24th International Symposium on Automation & Robotics in Construction (ISARC 2007)
Construction Automation Group, I.I.T. Madras

COOPERATIVE TASK PLANNING OF MULTI-ROBOT SYSTEMS

B.B.Biswal
Professor
Department of Mechanical Engineering
National Institute of Technology, Rourkela,
Orissa, 769008
bbbiswal@nitrkl.ac.in

B.B.Choudhury
Ph.D. Student
Department of Mechanical Engineering
National Institute of Technology, Rourkela,
Orissa, 769008
bbcnit@rediffmail.com

ABSTRACT

Multiple cooperating robots hold the promise of improved performance and increased fault tolerance for large-scale problems. A robot team can accomplish a given task more quickly than a single agent by executing them concurrently. A team can also make effective use of specialists designed for a single purpose rather than requiring that a single robot be a generalist. Multi-robot coordination, however, is a complex problem. An empirical study is described in the present paper that sought general guidelines for task allocation strategies. Different task allocation strategies are identified, and demonstrated in the multi-robot environment. A simulation study of the methodology is carried out in a simulated grid world. The results show that there is no single strategy that produces best performance in all cases, and that the best task allocation strategy changes as a function of the noise in the system. This result is significant, and shows the need for further investigation of task allocation strategies.

KEYWORDS

Multi-Robot, Task Allocation, Strategies, Algorithms

1. INTRODUCTION

There is no general theory of task allocation in uncertain multi-robot domains. In this paper, an attempt is made to empirically derive some guidelines for selecting task allocation strategies for multi-robot systems. The explored strategies are individualistic in that they do not involve explicit cooperation and negotiation among the robots. However, they are a part of a large class approaches that produce coherent and efficient cooperative behavior.

Given the empirical nature of this work and the scope of the problem addressed, these guidelines are necessarily incomplete, though they provide useful insight. The demonstration that the choice of task allocation strategy is far from trivial and also empirically show that no optimal task allocation strategy exists for all domains, and that it can be very difficult to identify the optimal task allocation strategy even for a particular task. These results are derived through the use of a framework developed for understanding the task allocation problem, which illustrates a common approach to decomposing the problem. The approach presented in this paper can be advantageously used in real-world problems.

2. DYNAMIC TASK ASSIGNMENT

In the context of multi-robot coordination, dynamic task allocation can be viewed as the selection of appropriate actions [1] for each robot at each point in time so as to achieve the completion of the global task by the team as a whole. From a global perspective, in multi-robot coordination, action selection is based on the mapping from the combined robot state space to the combined robot action space. For homogeneous robots, it is the mapping

$$S^{|R|} \rightarrow A^{|R|}$$

where S is the state space of a robot, $|R|$ is the number of robots, and A is the set of actions available to a robot [2].In practice, even with a small number of robots, this is an extremely high-dimensional mapping, a key motivation for decomposing and distributing control.

Based on the approach introduced in [3], the task allocation problem is decomposed into the following three steps:

1. each robot bids on a task based on its perceived fitness to perform the task;
2. an auctioning mechanism decides which robot gets the task;
3. the winning robot's controller performs one or more actions to execute the task.

The above decomposition is to construct a general formulation for the multi-robot coordination problem. In this formulation, a bidding function determines each robot's ability to perform a task based on that robot's state. Next, the task allocation mechanism determines which robot should perform a particular task based on the bids. Finally, the robot controllers determine appropriate actions for each robot, based on the robot's current task engagement. This partitioning, illustrated in Fig. 1, serves two purposes: it reduces the dimensionality of the coordination problem, and it reduces the amount of inter-robot communication required. We now have the mapping

$$B^{|R||T|} \rightarrow T^{|R|}$$

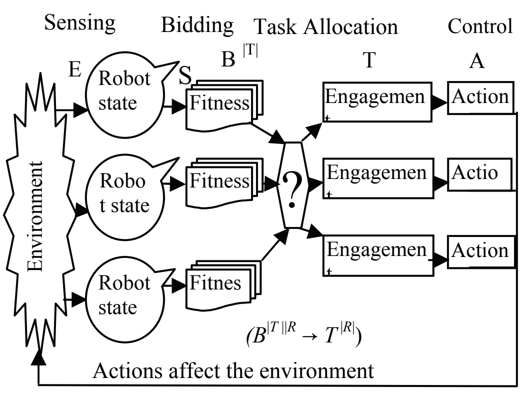

Figure 1 Reducing the Dimensionality of Multi-Robot Coordination

Instead of mapping namely from all robots' bids *B* for all tasks *T* to a task assignment for each robot, this overall mapping is called the Task Allocation Strategy for the system as a whole. The overall mapping is treated here as a global, centralized process (as depicted in Fig. 1), but distributed auctioning mechanisms [3,4], blackboard algorithms [5] , and cross-inhibition of behaviors [6] are some validated methods for distributing the task allocation function. In this methodology, the focus is on what the task allocation function should be, rather than on how it should be distributed. The above framework is a general way that dynamic task allocation for multi-robot systems can be formulated.

2.1 The Auction Algorithm

The auction algorithm is an intuitive method for solving the classical assignment problems. It outperforms substantially its main competitors for important types of problems, both in theory and practice, and is also naturally well suited for parallel computation. In the process, the user submits jobs to the auctioneer to start the process. An auctioneer is responsible for submitting and monitoring jobs on the user's behalf. The auctioneer creates an auction and sets additional parameters of the auction such as job length, the quantity of auction rounds, the reserve price and the policy to be used. The auctioneer informs the robots (Robot-1, Robot-2 and Robot-3) that an auction is about to start. Then, the auctioneer creates a call for proposals, sets its initial price, and broadcasts calls to all the robots (Robot-1, Robot-2 and Robot-3). Robots formulate bids for selling a service to the user to execute the job. The robots evaluate the proposal; they decide not to bid because the price offered is below what they are willing to charge for the service. This makes the auctioneer to increase the price and send a new call for proposal with this increase in the price. Meanwhile, the auctioneer keeps updating the information about the auction. In the second round, Robots are decided to bid.

The auctioneer clears the auction according to the policy specified beforehand. Once the auction clears, it informs the outcome to the user and the robots. The flowchart for the process is presented in Figure 2.

The algorithm described here can be utilized in task allocation in multi-robot applications, and is particularly suitable for parallel computation. This approach attempts to find the best assignment between tasks and robots, while maximizing the total benefit. It iterates between robots and in each

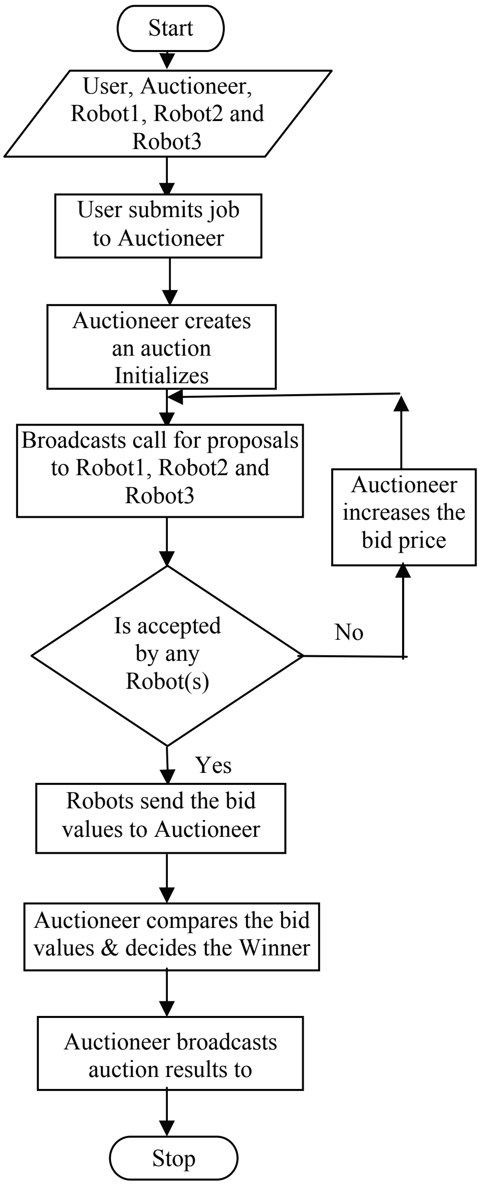

Figure 2 Flowchart of the Auction for Task Allocation

iterations tries to assign a task to a robot who offers the most. In consecutive iterations, other robots may bid for other tasks and if more than one bids for the same task, it will increase the cost of task until finally just one task-robot pair match takes place, i.e., iterative improvement. The iteration terminates when all robots are pleased with their match, otherwise an unhappy robot will bid higher for another task and this process will continue. Although auction algorithm may have some similarities to the free market approach, there is a little difference. One difference is that in the free market approach, agents can cooperate in order to gain a maximum profit for all of them, however in the auction algorithm every robot is considered rival. Another dissimilarity is that the auction algorithm uses an exclusive mathematical model for all the applications, while the free market approach does not. In addition, the free market technique is based on the collection of heterogeneous agents, while in the auction algorithm the robot set is homogeneous.

2.2. Task Allocation Strategies

The dynamic task allocation problem, i.e., the mapping from bids to tasks, can be performed in numerous ways. The focus is limited here to Markovian systems, where the task allocation mapping for a given robot is based on the mapping between that robot's current task assignments and every other robot's current bid on each task, to the given robot's new task assignment, as shown in Fig. 3. Given each robot's bid on each task and each robot's current task engagement, each robot's new task assignment need to be determined. The effects of two key aspects of distributed control, commitment and coordination, on performance are explored.

Given the large space of possibilities, only the extreme cases of each: no commitment and full commitment, and no coordination and full coordination are considered. The combination of these extremes results in four task allocation strategies as shown in Figure 4.

Along the commitment axis, a fully committed strategy meant a robot would complete its assigned task before considering any new engagements, while a fully opportunistic strategy allowed a robot to drop an ongoing engagement at any time in favor of a new one.

Along the coordination axis, the uncoordinated (individualistic) strategy meant each robot performed based on its local information, while a coordinated strategy simply implemented mutual exclusion, so only one robot could be assigned to a task, and no redundancies were allowed. It is noted that this notion of coordination is simple, and not intended to represent explicit cooperation and coordination strategies instantly closes (i.e., the fixed time-cost was 0). Three new tasks appear every twelve time-steps at random positions on the grid. The tasks are structured so that one robot is sufficient for completion of an individual task assignment. Thus, mutual exclusion is the simplest yet effective form of coordination.

Current engagement	Bids	A	B	C	D	New engagement
A	R1	6	4	2	5	?
--	R2	4	1	0	3	?
C	R3	7	2	3	2	?

Figure 3 An Example Task Allocation Scenario

Commitment ⬇	Coordination➡	
	Individual	Mutually Exclusive
Commitment	Strategy.1	Strategy.2
Opportunity	Strategy.3	Strategy.4

Figure 4 the Four Task Allocation Strategies

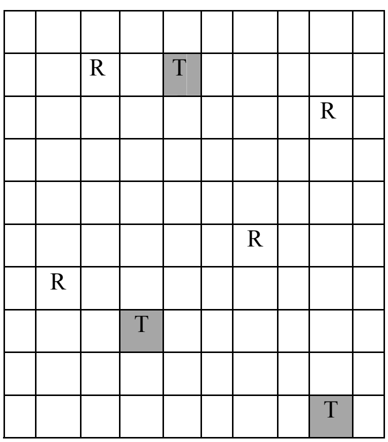

Figure 5 An Example 10 X 10 Grid World with Four Robots and Three Tasks

As an example, the fully committed mutually exclusive strategy is as follows:

1. If a robot is currently engaged in a task, and its bid on that task is greater than zero, remove the row and column of the bid from the table, and set the robot's new assignment to its current one.

2. Find the highest bid in the remaining table. Assign the corresponding robot to the corresponding task. Remove the row and column of the bid from the table.

3. Repeat from step 2 until there are no more bids. In case of individualistic (uncoordinated) strategies, the same algorithm is run on a separate table for each robot. In the opportunistic (uncommitted) case, step 1 above is skipped.

3. GRID WORLD EXPERIMENTAL FRAMEWORK

A simplified version of the above described multi-robot task in a grid world is illustrated in Fig. 5.. As the base case of the grid world implementation, a 10×10 grid inhabited by 10 robots is considered. Robots bid on tasks depending on their capability(expressed by a number) to those tasks. The bid was set to 20 − d, where d is the Manhattan distance to the task. In each time-step, any robot assigned to a particular task selects that task. When a robot selects a task, that task .

Strategy:	I,O	I,C	M,O	M,C
Results:	980	1045	435	722

Figure 6 Results from Base Case Grid World

In order to explore the parameter space of the task, we focused on *commitment* and coordination. In the context of emergency handling, commitment means that robots stay focused on a single task, until the task is over. The opposite, *opportunism,* means that robots can switch tasks, if for example another task is found with greater intensity or priority. In the experiments, coordination is linked to communication, namely the ability of robots to communicate about who should service which tasks, as opposed to *individualism,* where robots have no awareness of each other. Communication is used to prevent multiple robots from trying to accomplish the same task; robots inhibit others from engaging in the same task. The goal is to reduce interference among robots, and to prevent loss of coverage in some areas because all the robots rush to perform task in another area. Deciding the level of commitment and collaboration are key aspects of the multi-robot task allocation problem. Four experiments were designed resulting from the combinations in varying the two parameters, coordination and commitment. The results of the grid world simulation are presented in Fig.6.On one axis we test commitment versus opportunism, and on the other we test individualism versus mutual exclusion.

4. THE BLACKBOARD ALGORITHM

To ensure reasonable scalability and robustness, communication among the robots is done through a "blackboard". To simulate experiments with inter-robot communication, each robot sends its relevant state information to the blackboard, and the blackboard information is read by all the robots. In the case of no communication, the blackboard just contains information from one robot (itself). The information on the blackboard is the current engagement of each robot. Intuitively, if all robots have the same blackboard information available and execute the same algorithm, they should all come to the same conclusion as to which robot should pursue which task.

Table 1 Quantitative Results

Results	Individ.		Mut. Exclusion	
Commitment	2063	1	2325	2
	2016	2	1919	1
	1786	2	2008	1
Opportunism	1087	0	2061	2
	928	0	1406	1
	1917	0	1078	0
			1322	0

To facilitate validation of the experiments, all parameters are held constant, except the way the information on the blackboard is handled. The algorithm for deciding which robot should do what is as follows:

Step 1: All robots engaged in a task cannot have their engagement set to “none”

Step 2: In case of commitment, all entries in the blackboard for robots already pursuing a task is set top zero, along with all entries for task already being pursued. In case of opportunism, this step is skipped.

Step 3: The highest non-zero score in the table is found, and the robot corresponding to this entry is assigned to the task corresponding to this entry.

Then all entries of the robot and the tasks are set to zero. Step 3 is repeated with the new table, until no new assignments are made. Table 1 presents the quantitative where the larger numbers are total sum of on-line for tasks in each trial. The smaller numbers are tasks left at the end of the trial. The algorithm has the effect that in the case of commitment robots keep themselves engaged in performing a task until it is complete, while in the opportunism, robots keep switching engagements.

5. DISCUSSION

The grid world results are interesting if they actually represent real world system behavior. The fact that the best performing task allocation strategy changes as we vary noise parameters in the grid world implies that it can be very difficult to decide *apriori* which task allocation strategy should be used in a given task for any real world implementation. The experiments clearly show that the opportunistic strategy worked significantly better than the commitment-based strategy. This might be because the time to reach a task was significantly larger than the time to complete a task, once a robot was there. This choice of parameters favors opportunism over commitment since the former effectively uses the presence of robots near emergencies by harnessing them immediately. In other regions of the parameter space of the emergency handling task (e.g., where the ratio of time-to-reach-task to time-to-complete-task is small) opportunism might not be as effective. The present study excluded the case where several robots would be required to do a task in a cooperative fashion, a regime in which performance might improve with commitment.

The four task allocation strategies we examined are *extreme*, in that they take into consideration only the complete presence or absence of commitment and coordination in the given context. Arguably, the best strategy for any particular task would most likely be a carefully balanced compromise. However, as stated previously, the goal of this work was not to attempt to find the best strategy (which is necessarily task- and parameter-specific), but rather to gain some insight into task allocation in general. The four strategies we explored provide a reasonable span of strategy space and provide leading insights for further study. In practice, the robot capability ratings can be obtained from the databases. Therefore, one can automatically select appropriate candidate for a given task by using the proposed matching procedure and databases.

6. CONCLUSION

The paper describes an empirical study that sought general guidelines for task allocation strategies in systems of multiple cooperating robots. Four distinct task allocation strategies are identified that aim at studying tradeoffs between commitment and coordination. The data from the simulations show that there is no single strategy that produces best performance in all cases, and that the best task allocation strategy changes as a function of the noise in the system. This result is significant, and shows the need for further investigation of task allocation strategies. The described work is a small step toward the larger goal of principled analysis and synthesis of multi-robot coordination strategies for complex and uncertain domains, such as space exploration. The entire exercise has relevance to real world distributed robotics terrestrial and space applications.

7. REFERENCES

[1] Maes, P. (1994) Modeling adaptive autonomous agents. Artificial Life, I, 1(2),135-162.

[2] Mataric, M.J., Sukhatme, G.S., et al. (2003) Multirobot Task Allocation in uncertain Environments. Autonomous Robots, Vol. 14, 255-263.

[3] Gerkey, B. and Mataric, M.J. (2001) Principled communication for dynamic multi-robot task allocation, Experimental Robotics VII, LNCIS 271, D. Rus and S. Singh (Eds.), Springer-Verlag: Berlin, 353-362.

[4] Dias, M.B. and Stentz, A.T. (2000) A free market architecture for distributed control of a multirobot system, Proceedings, 6th International Conference on Intelligent Autonomous Systems (IAS-6), 115— 122.

[5] Corkill, D.D. (1991) Blackboard systems. AI Expert, 6(9), 40-47.

[6] Werger, B. and Mataric, M. (2000) Broadcast of local eligibility for multi-target observation. In Proceedings, 5th International Symposium on Distributed Autonomous Robotic Systems (DARS), Knoxville, TN, Oct. 4-6, 347-356.