Exploring auction mechanisms for role assignment in teams of autonomous robots

Vanessa Frias-Martinez[1], Elizabeth Sklar[1], and Simon Parsons[2]

[1] Department of Computer Science
Columbia University
1214 Amsterdam Avenue, New York, NY 10027, USA
vf2001,sklar@cs.columbia.edu

[2] Department of Computer and Information Science
Brooklyn College, City University of New York
2900 Bedford Avenue, Brooklyn, NY 11210, USA
parsons@sci.brooklyn.cuny.edu

Abstract. We are exploring the use of auction mechanisms to assign roles within a team of agents operating in a dynamic environment. Depending on the degree of collaboration between the agents and the specific auction policies employed, we can obtain varying combinations of role assignments that can affect both the speed and the quality of task execution. In order to examine this extremely large set of combinations, we have developed a theoretical framework and an environment in which to experiment and evaluate the various options in policies and levels of collaboration. This paper describes our framework and experimental environment. We present results from examining a set of representative policies within our test domain — a high-level simulation of the RoboCup four-legged league soccer environment.

1 Introduction

Multi agent research has recently made significant progress in constructing teams of agents that act autonomously in the pursuit of common goals [12, 15]. In a multi agent team, each agent can function independently or can communicate and collaborate with its teammates. When collaborating, the notion of *role assignment* is used as a means of distributing tasks amongst team members by associating certain tasks with particular roles. The assignment of roles can be determined *a priori* or can change dynamically during the course of team operation.

Collaboration enables a team of agents to work together to address problems of greater complexity than those addressed by agents operating independently. In general, using multiple robots is often suggested to have several advantages over using a single robot [4, 7]. For example, [11] describes how a group of robots can perform a set of tasks better than a single robot. Furthermore, a team of robots can localize themselves better when they share information about their environment [7]. But collaboration in a team of robots may also add undesirable delays through the communication of information between the agents.

We are exploring — within dynamic, multi-robot environments — the use of auction mechanisms to assign roles to agents dynamically and the effect of different approaches to collaboration within the team. In order to evaluate this set, we have developed a theoretical framework and a simulation environment. The theoretical framework helps us to identify the space of possibilities, and the simulation environment helps us to evaluate the various degrees of collaboration.

This paper begins by highlighting some background material on auctions and the use of auction mechanisms in multi agent systems. Then we describe our theoretical framework. Next we detail our experimental environment — a high-level simulation of the RoboCup Four-Legged Soccer League. We then present results of simulation experiments evaluating both collaborative and non-collaborative models of information sharing as well as various auction policies. Finally, we close with a brief discussion and directions for future work.

2 Auctions

Following Friedman [9], we can consider an *auction* to be a mechanism that regulates how commodities are exchanged by agents operating in a multi agent environment. An *auction mechanism* defines how the exchange takes place. It does this by laying down rules about what the traders can do — what *messages* they can exchange in an interaction — and rules for how the allocation of commodities is made given the actions of the traders. Auctions have been used in different environments for resource allocation, such as electronic institutions [6], distributed planning of routes [13] or assignment of roles to a set of robots to complete a common task [10].

3 Theoretical framework

In our auction, there are two types of agents: the *auctioneer* and the trader — a *player* in the RoboCup soccer game. The player makes an *offer* and the auctioneer's job is to coordinate the offers from all the players and perform role assignment. There are five main components to our model.

First, we define $\mathcal{R}$ to be the set of possible roles: $\mathcal{R} = \{PA, OS, DS\}$, where PA is a primary attacker, OS is an offensive supporter, and DS is a defensive supporter. Note that the goalie is not considered a role to be assigned in this manner, since it cannot change during the course of the game.

Next, we define $\mathcal{P}$ to be a set of player attributes: $\mathcal{P} = \{d_{ball}, d_{goals}, d_{mates}, d_{opps}\}$ where d_{ball} contains the distance from the player (who is making the offer) to the ball; d_{goals} contains the distance from the player to each goal; d_{mates} contains the distance from the player to each of its teammates; and d_{opps} contains the distance from the player to each player on the opposing team.

Third, we define F to be a set of functions which define the method for sharing perception information between agents. This information could be shared with teammates, the auctioneer, or both. Fourth, we define $\mathcal{M}$ to be a *matching function*, the method used by the auctioneer for clearing the auction, i.e., matching the offers with roles. In other words, the matching function captures the coordination strategy. Finally, we define an *auction*, $\mathcal{A}$, to be: $\mathcal{A} = \langle P, R, M, f \rangle$ where $P \subseteq \mathcal{P}$ and $P \neq \emptyset$; $R \subseteq \mathcal{R}$ and $R \neq \emptyset$; $M \subseteq \mathcal{M}$ and $M \neq \emptyset$; and $f \in F$.

Our work is systematically exploring the space of all possible auctions $\mathcal{P} \times \mathcal{R} \times \mathcal{M} \times F$. $\mathcal{B}$ denotes the set of possible types of offers in a particular auction, $A \in \mathcal{A}$: $\mathcal{B} = \{\mathbf{r}, \mathbf{w}\}$ where: $\mathbf{r} \subseteq R$ is a set of roles for which the player bids; $\mathbf{w}$ is a set of real-valued weights, one weight corresponding to each of the roles in $\mathbf{r}$ (a weight of 0 means that the player is not interested in making an offer for the corresponding role); and $f(p), p \subseteq P$, is the mechanism by which perceptual data is used to determine $\mathbf{r}$ and $\mathbf{w}$.

To date, we have defined two different types of auctions within this framework — a *simple* auction [5] and a *combinatorial* auction [3]. We can define a simple auction $b_t \in \mathcal{B}$ as: $b_t = \{r, w\}$, where the role r and w are singletons(unique offer). And a *combinatorial auction*, is defined as: $b_t = \{(r_0, r_1, r_2), (w_0, w_1, w_2)\}$ where r_i and w_j are singletons. Using different combinations of weights allows the agent to bid for different combinations of roles, and this makes the auction combinatorial [1].

4 SimRob: our Simulated Approach to a RoboCup Game

We are using RePast[14] to implement our environment. RePast allows us to build a simulation as a state machine in which all the changes to the state machine occur through a schedule. In order to model a RoboCup soccer game in RePast, we need to define the agents, the environment and the state machine that RePast will execute at each scheduled *tick*, i.e., simulated time step.

4.1 Agent parameters

The RoboCup Four-Legged League environment has four Sony AIBO robots per team and a bright orange ball. Each one of the robotic agents is associated with an array containing the values that define their perception and localization:

$$(x, y, \phi, d_{ball}, d_{goals}, d_{opps}, d_{mates}, b_{ball}, b_{goals}, b_{opps}, b_{mates}) \tag{1}$$

where (x, y) are the 2D coordinates of the robot on the field[3]; ϕ is the orientation of the robot[4]; d_{ball} is the distance from the robot to the ball, d_{goals} is the distance from the robot to each goal, d_{opps} is an array containing the distance from the robot to each opponent, and d_{mates} is an array containing the distance from the robot to each teammate. The boolean values in the second half of equation (1) indicate if the ball has been detected by the player (b_{ball}), if each goal has been detected by the player (b_{goals}), if each opponent has been detected nearby (b_{opps}) and if each teammate has been detected nearby (b_{mates}).

4.2 Simulation skeleton

We use RePast in order to simulate the development of a game with the agents. At the beginning of the simulation, we define four agents (per team) and a ball in the field.

[3] The field itself is broken down into the same discretized grid that we use for localization on the AIBOs.

[4] The 360° of orientation are divided into eight 45° sections, numbered 0 through 7.

Each of the agents is defined as explained above, by means of an array as in equation (1). The simulation run in RePast can be divided into the following steps:

(1a) Generation of the agent parameters In this first step, we obtain the parameters of each of the agents in the field. The localization of the robot is expressed with the coordinates (x, y) in a 2D field. We also obtain the distances to the ball d_{ball}, to the goal d_{goals} and to the opponents d_{opps}.

(1b) Amount of information shared by the agents The information shared by the agents is: $mingoal$, a boolean variable that is true when the agent is the one closest to the goal. This variable can be defined when the agents share the variable d_{goals} among them. $maxopp$ is a boolean variable that is true when the agent is farthest away from the opponents in the field. This variable can be defined when the agents share d_{opps}. And $maxball$ is a boolean variable that is true when the agent is farthest away from the ball. This value can be defined when the variable d_{ball} is shared among the agents.

(2a) Defining a bidding policy for the agents For each simulation tick of the game play, the agent's bid will be the role associated by the policy being tested to the set of perceptions gathered by the agent at that simulation tick.

(2b) Defining an auction policy for the auctioneer The auction is responsible for distributing the roles between the agents on the field. The auctioneer will go through the different roles in the bid until one of the roles in the array is assigned to the agent, meaning that the bid is won.

(3) Game Play Once the agent-roles are defined, we have to actually simulate the joint task to be developed by the agents. As stated before, our aim is that of simulating a soccer game. The game model is very simple. Each role has a state graph that will output a certain behavior depending on the perceptions gathered by the agent:

- PA BEHAVIOR: If the agent sees the goal and the ball, then it kicks the ball, otherwise it turns to look for the ball without losing track of the goal.
- OS BEHAVIOR: If the ball is seen, the agent kicks it.
- DS BEHAVIOR: If the ball is seen, the agent follows it in order to prevent an agent from the opposing team scoring.

Finally, if a goal is scored, the robots are sent back to their initial positions and the ball randomly changes location. Then, the three step (parameter generation, auction execution and game play) simulation is run again.

5 Experiments

This section describes our experimental work to date. We have started to explore the range of possible auctions and their effect on the coordination of a team, as measured by their performance in simulated games. We have experimented with four very simple types of coordination and describe policies that we have used for experimentation, chosen somewhat *ad hoc*. In current work, we are learning policies [8].

Table 1. Example non-collaborative simple(S) and combinatorial (C) auctions

Ball seen	Opponent seen	Mate seen	Role(S)	Role(C)
0	0	0	DS	[OS,.7,DS,.2,PA,.1]
0	0	1	DS	[OS,.7,DS,.2,PA,.1]
0	1	0	DS	[OS,.7,DS,.2,PA,.1]
0	1	1	DS	[OS,.7,DS,.2,PA,.1]
1	0	0	DS	[OS,.7,DS,.2,PA,.1]
1	0	1	DS	[OS,.7,DS,.2,PA,.1]
1	1	0	OS	[OS,.7,DS,.2,PA,.1]
1	1	1	OS	[OS,.7,DS,.2,PA,.1]

5.1 Non-collaborative simple auction

This approach defines a team of agents that don't share any perception data. Hence, each one relies on the information that it gathers independently of the others. The offers made by the agents follow the policy in Table 1 column Role(S). This shows that we have defined the agent to offer to be OS when both ball and opponent are seen. In any other case, our agent will offer to be DS. We have chosen a simple matching policy that just associates a fixed role to each of the possible sets of perceptions.

5.2 Non-collaborative combinatorial auction

In this case there is still no sharing of perception, but the bid now contains a vector defining the agent's role preferences For our experiments, we have defined two different bidding policies. The *offensive* policy, defined in Table 1, column Role(C), represents a team with an attacking approach, always looking for the goal and aiming to score. The other policy is more defensive. The offensive policy assigns the array of roles [DS,.7,OS,.2,PA,.1] to each of the agents. The matching is the same as before.

5.3 Collaborative simple auction

In this case, the agents share all the perception data. Hence, when defining the bids, we can also share the three variables related to the minimum and maximum distances to the ball, opponents and goal. The table defining the bidding policy is huge. In Table 2, column Role(S), we show a few lines to give the sense of it, but it is deliberately similar to the policy for the non-collaborative auction to give a reasonable comparison. When no elements are seen by any of the agents, the agent bids for the role DS. When everything is seen and the distances are minimum, the agents bid to be OS. The matching policy is also the same as for the non-collaborative examples.

5.4 Collaborative combinatorial auction

Here the bidding Table 2, column Role(C), is similar to the previous one, but contains a vector of bids and weights instead of only one role, and this vector is like that for the

Table 2. Collaborative simple(S) and combinatorial(C) auctions

Ball seen	Opp seen	Mate seen	MinGoal	MaxOpp	MaxBall	Role(S)	Role(C)
0	0	0	0	0	0	[DS]	[OS,.7,DS,.2,PA,.1]
0	0	0	0	0	1	[DS]	[OS,.7,DS,.2,PA,.1]
...	.	.	.	.	.	.	.
1	1	1	1	1	1	[OS]	[OS,.7,DS,.2,PA,.1]

Table 3. Results

Number of goals scored after 2000 ticks.

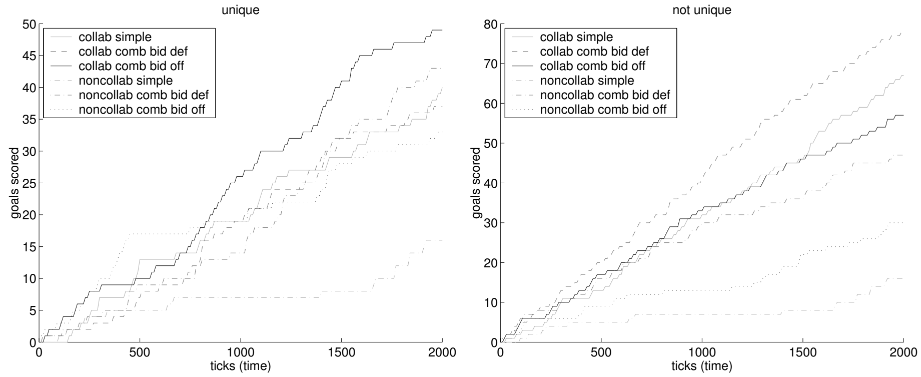
	unique		not unique	
	offensive bid	defensive bid	offensive bid	defensive bid
noncollab simple	16	–	16	–
noncollab comb	33	43	30	47
collab simple	40	–	67	–
collab comb	49	37	78	67

non-collaborative combinatorial auction. Again we ran experiments with an attacking bidding policy and a defensive bidding policy, and the matching table is the one used in the previous examples.

5.5 Results

Teams using each of the types of coordination described above (including separate offensive and defensive techniques in the combinatorial auction) were run in simulation against the same, simple, opponent in order to evaluate the effectiveness of the collaboration policy. The opposing team moved randomly around the field, but was not intended as serious opposition, rather it was intended as a baseline against which all mechanisms could be judged equally. For each coordination mechanism, we ran two sets of experiments. In one, the "unique" experiments, we made the auctioneer assign unique roles to agents. In the "not unique" experiments, the auctioneer was allowed to assign duplicate roles. The average number of goals scored for each of the different kinds of collaboration are given in Table 3, and plots of the goals scored over time for a sample game are given in Figures 1.

In the non unique approach, collaborative teams score almost double the number of goals of the non-collaborative teams. In the unique role approach, differences in the score of the games between the collaborative and non-collaborative approaches for both simple and combinational auctions are not so marked. This is due to the fact that our matching policy is very demanding and since we do not allow repeated roles, the auctioneer often ends up distributing roles randomly. In order to prove this last assertion, we defined a parameter called *success ratio* in the simulation. The success ratio is as-



(a) unique matching policy

(b) non-unique matching policy

Fig. 1. Goals scored over the course of a game

sociated with the acceptance of the bids made by an agent. The higher the ratio, the more times its bid has been accepted. In the not uniqueness experiments, we obtained very low ratios, meaning that the agents almost never won a bid, and so, the roles were distributed randomly.

6 Conclusions and Future work

This paper has described our preliminary work in exploring the use of auction mechanisms to coordinate players on a RoboCup team. While this work is only just beginning, we believe that the results demonstrate the potential of the approach to capture a wide range of types of coordination, and to be able to demonstrate their effectiveness through simulation. In addition, this approach makes it simple to explore more complex, and potentially more flexible, kinds of role allocation than have been previously used in the legged-league, for example [2, 16].

Our longterm work is to build on this foundation and explore a wide range of possible auctions through simulation and on real (physical) robots. We are currently using learning techniques to automatically explore the space of auctions. We further intend to implement the most effective bidding and matching policies developed on our real Legged-League team.

7 Acknowlegements

This work was made possible by funding from NSF #REC-02-19347 and NSF #IIS 0329037.

References

1. C. Boutilier and H. H. Hoos. Bidding languages for combinatorial auctions. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 1211–1217, San Francisco, CA, 2001. Morgan Kaufmann.
2. D. Cohen, Y. Hua, and P. Vernaza. The University of Pennsylvania Robocup 2003 Legged Soccer Team. In *Proceedings of the RoboCup Symposium*, 2003.
3. S. de Vries and R. Vohra. Combinatorial auctions: A survey. *INFORMS Journal of Computing*, (to appear).
4. G. Dudek, M. Jenkin, E. Emilios, and D.Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4), 1996.
5. R. Engelbrecht-Wiggans. Auctions and bidding models: A survey. *Management Science*, 26:119–142, 1980.
6. M. Esteva and J. Padget. Auctions without auctioneers: distributed auction protocols. In *Agent-mediated Electronic Commerce II, LNAI 1788*, pages 20–28. Springer-Verlag, 2000.
7. D. Fox, W. Burgard, H. Kruppa, and S. Thrun. Collaborative multi-robot localization. In *Proceedings of the 23rd German Conference on Artificial Intelligence*. Springer-Verlag, 1999.
8. V. Frias-Martinez and E. Sklar. A team-based co-evolutionary approach to multi agent learning. In *Proceedings of the 2004 AAMAS Workshop on Learning and Evolution in Agent Based Systems*, 2004.
9. D. Friedman. The double auction institution: A survey. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories and Evidence*, Santa Fe Institute Studies in the Sciences of Complexity, chapter 1, pages 3–25. Perseus Publishing, Cambridge, MA, 1993.
10. B. Gerkey and M. Mataric. Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 2000.
11. D. Guzzoni, A. Cheyer, L. Juli, and K. Konolige. Many robots make short work. *AI Magazine*, 18(1):55–64, 1997.
12. G. A. Kaminka, D. V. Pynadath, and M. Tambe. Monitoring deployed agent teams. In Jörg P. Müller, Elisabeth Andre, Sandip Sen, and Claude Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 308–315. ACM Press, 2001.
13. T. L. Lenox, T. R. Payne, S. Hahn, M. Lewis, and K. Sycara. Agent-based aiding for individual and team planning tasks. In *Proceedings of IEA 2000/HFES 2000 Congress*, 2000.
14. Repast. `http://repast.sourceforge.net`.
15. M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.
16. M. Veloso and S. Lenser. CMPAck-02:CMU's Legged Robot Soccer Team. In *Proceedings of the RoboCup Symposium*, 2002.