Comparing Market and Token-Based Coordination

Yang Xu
School of Info Sciences
University of Pittsburgh
Pittsburgh, PA 15260, USA
yxu@sis.pitt.edu

Paul Scerri and **Katia Sycara**
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{pscerri, katia}@cs.cmu.edu

Michael Lewis
School of Info Sciences
University of Pittsburgh
Pittsburgh, PA 15260, USA
ml@sis.pitt.edu

Abstract

Many coordination algorithms claim to be *general*, implying that they can be used to coordinate agents in a variety of domains. However, little work has been done to quantitatively compare distinctly different approaches to coordination across a range of domains, in part because of the amount of effort required to implement the approaches for different domains. In this paper, we present a detailed comparison of two published coordination algorithms, performed in an abstract coordination simulation environment that allows extensive, quantitative experimentation. The abstract environment preserves critical coordination issues but abstracts away domain level details allowing a high degree of parameterization and large volume of experiments. The simulator is used to compare two distinct approaches to coordination, token-based coordination and market based coordination. The results largely show the generality of different approaches, but show that performance and performance tradeoffs varies greatly across domains.

Introduction

Autonomous coordination is a complex process because several distributed algorithms are required to interact to produce agile, cohesive and efficient coordinated behavior. If effective coordination can be achieved, it is applicable to a diverse range of domains from commerce, to disaster response and to the military. Because of the importance of autonomous coordination, many approaches have been developed, including approaches based on markets (N. Kalra B. Dias & Stentz. 2005; Gerkey & Mataric. 2003), tokens (Y. Xu & Lewis. 2005) and swarms (Cicirello & Smith. 2001). Typically, each of these approaches is designed to work in a specific domain, but the authors, usually with good reason, claim they will work in a wide range of domains. However, such claims are rarely quantitatively verified and, more importantly, competing approaches are rarely systematically compared. Thus, when a developer needs to select an approach to use in a particular domain, they are confronted with many claims but little concrete data with which to make a decision.

The need to compare competing algorithms is well understood, as is the difficulty of doing so. Major initiatives such as RoboCup (Yanco. 2001), Urban Search and Rescue (USAR) (Jacoff & Evans. 2001) and the Trading Agent competition (E. David M. He & Jennings. 2005), have been created partly for the purpose of comparing alternative approaches. However, with important exceptions e.g., (Kaminka. 2000), these initiatives have not led to scientifically valid comparisons to date. This is partly because the target problem does not have enough flexibility to allow testing across a range of settings and partly because performing scientifically valid comparisons is simply too resource consuming within such environments. In more limited scenarios, and typically on more abstracted problems, many researchers have compared specific algorithms that might be part of an approach to coordination. Sometimes, the results are less than conclusive. For example, Modi (Modi & Veloso. 2005) and Mailler (Mailler & Lesser. 2004) have written papers in recent AAMAS conferences showing that their respective algorithms outperform each other on subtly different problems and with different metrics used to measure performance. The consequence of the relative lack of algorithmic comparison within the multi-agent community is that whole competing sub-fields of multiagent systems have never been carefully compared with one another.

In this paper, we present an initial attempt at systematically and scientifically comparing distinct approaches to coordination. To make such a comparison both feasible and interesting, we have developed an abstract simulation environment that is sufficiently rich to capture a variety of real world concerns, but sufficiently abstract to be highly configurable and very fast. Such an environment provides enough realism to verify that an algorithm can deal with a range of issues that "real" coordination presents, but is sufficiently abstract for statistically significant numbers of experiments to be performed in a reasonable amount of time. We used the coordination simulator to investigate the relative strengths of three distinct approaches to coordination: auction-based coordination (Dias & Stentz. 2003); token-based coordination (Y. Xu & Lewis. 2005); and a hybrid of the two. The first two approaches to coordination were chosen because they had sufficiently similar capabilities, were published within the agents community and claims had been made about the generality of each approach. The hybrid algorithm was developed not to be superior to the other two, but to investigate an hypothesis about the observed relative strengths of

the other algorithms.

Unfortunately, the overlap in coordination tasks that can be performed by both tokens and auctions is limited to task and resource allocation, hence the focus of the comparison is on those capabilities. In the experiments, other tasks required for coordination, such as initiating joint tasks and sharing key information are always performed by the token algorithm. Based on an analysis of previous literature (Y. Xu & Lewis. 2005; N. Kalra B. Dias & Stentz. 2005), several hypotheses can be formed about the relative performance of the algorithms. Auctions are focused on maximizing overall utility taking into account the *bids* of all team members (N. Kalra B. Dias & Stentz. 2005). Token-algorithms are focused on scalability, hence they minimize communication, sometimes at the expense of overall utility. Thus, the clearest hypothesis is that auctions will communicate more than token algorithms, but result in better allocations of tasks and resources. More subtly, the performance advantage of an auction should be most pronounced when small changes in allocations lead to big differences in performance, i.e., typically highly constrained cases, while the token algorithms should maximize their communication advantage when the probabilistic models they rely on are most advantageous, i.e., weakly constrained cases. The empirical results support these hypotheses.

Initial experiments suggested that auctions find superior allocations because they compare many options, while tokens use little communication by quickly focusing on the agents most likely able to perform tasks or having most use for resources. If these are the correct reasons for the relative algorithm strengths, then a hybrid algorithm that uses tokens to solicit auction bids from those agents most likely to submit winning bids then uses an auction to select from between the small number of bids should perform well. However, this hybrid algorithm should only perform well under restricted circumstances. If the problem is so tightly constrained that the auctions need to see many bids to make good allocations then using tokens to solicit bids only adds overhead. Conversely, if the coordination is so underconstrained that the tokens can reliably and accurately target the best agents, then the auction only adds unnecessary overhead. We implemented the hybrid algorithm and compared its performance to the other algorithms.

Problem

In the following, we formally describe the coordination problem that the algorithms must contend with.

Agents, $A = \{a_1, \ldots, a_k\}$, are cooperating on a joint goal G. Information, $I = \{i_1, \ldots, i_n\}$, are discrete pieces of information that are either $true$ or $false$ at a particular time. G is broken into discrete sub-tasks $\alpha_1, \ldots, \alpha_n$, typically performed by individuals. A subtask, α_i is applicable when the predicate $Applicable(I_{\alpha_i}), I_{\alpha_i} \subseteq I$ is $true$, where $Applicable(I_{\alpha_i}) \equiv \bigwedge_{i \in I_{\alpha_i}} i$. The applicability of a task must be determined by the team and the team must ensure that only one instance of an applicable task is being executed. We refer this process as plan instantiation and deconfliction.

Agents must perform the individual tasks α, when they applicable, for the team to receive reward. The reward received by the team when an agent performs a task is a function of the agent and task, as well the resources the agent has. Specifically:

$$Reward(a, \alpha, Holds(a)) \rightarrow \mathcal{R}$$

The function $Assigned(a, \alpha) = 1$ if agent a is assigned to task α, otherwise it is equal to 0. Only one agent may be assigned a task at any time, i.e., $\sum_{a \in A} Assigned(a, \alpha) \leq 1$.

Agents always require sharable resources to perform tasks. These resources, $R = \{r_1, \ldots, r_m\}$, are discrete and non-consumable. Agent a has exclusive access to resources $Holds(a) \subseteq R$. Only one agent may hold a resource at any point in time, i.e., $\forall a, b \in A, a \neq b, Holds(a) \cap Holds(b) = \emptyset$.

We specifically distinguish between *necessary* and *useful* resources. We define $IR_i \subseteq R$ as a set of substitutable resources. Necessary resources IR_i^* are those where if $Holds(a)$
$\cap IR_i^* = \emptyset$ then $Reward(a, \alpha, Holds(a)) = 0$. Useful resources IR_i^+ are those where if $Reward(a, \alpha, Holds(a)) > Reward(a, \alpha, Holds'(a))$ then $Holds(a) \cap IR_i^+ \neq \emptyset$ and $Holds(a) \cap IR_i^+ = \emptyset$. In this paper, we consider only necessary resources.

The coordination problem is to maximize the reward to the team, while minimizing the *costs of coordination*. The overall reward is simply:

$$\sum_{i=0}^{n} \sum_{a \in A} Assigned(a, \alpha_i) Reward(a, \alpha_i, Holds(a))$$

The costs of coordination can be very general and in some cases difficult to define. Here we are specifically concerned with only the volume of communication. The coordination simulator that we are using, implements this abstract coordination problem. More details about this simulator are in Section 4.

Algorithms

In the following, we describe the three coordination approaches that are compared and point to key literature describing the expectations for those algorithms. Notice that we only focus on the problems of task and resource allocation because other issues are not addressed in a comparable way by the respective algorithms.

Auction-Based Coordination

The first of the algorithms we compared used a market-based approach to task and resource allocation. Our implementation of this approach was based on TraderBots (Dias & Stentz. 2003) with adaptations where are necessary to make a comparison possible. In our market-based approach, one agent acts as auctioneer and both tasks and resources are treated as merchandise. Agents bid for either single items or combinatorial sets of items in order to maximize their own

utilities. The auctioneer maximizes its utility by "selling" their "merchandise". In this approach, Sandholm's winner determination algorithm (Sandholm. 2002) is used to determine the allocation for tasks and resources by the auctioneer. Because of the centralized position of the auctioneer, it develops a complete knowledge of how agents will use a task or resource if allocated. Thus, the auctioneer can perform assignments that maximize the team utility. Notice that several constraints also apply to this approach. To be fair to all the bidders, the auction should last for a fixed period of time. Where early determination is infeasible; Agents are allowed to bid for resources after tasks have been allocated. Moreover, to prevent deadlock in resource allocation, agents are only allowed to bid for resources for their *first* pending task.

Algorithm 1: AgentAuction

```
(1) ApplicableTasks=[],   Bids=[],   OwnTasks=[],
    Holds=[], AuctionList=[];
(2) while true
(3)     foreach (α in a, α ∉ ApplicableTasks)
(4)         if (Applicable(α)
(5)             ApplicableTasks.append(α);
(6)             SendToAuctioneer(α);
(7)     Update(AuctionList, BidList);
(8)     msg → recvMsg;
(9)     if (msg is NewTaskAuction(α))
(10)        BidTask(α);
(11)    else if (msg is NewResourceAuction(r))
(12)        OpenResources.add(r);
(13)    else if (msg is TaskAllocated(α))
(14)        OwnTasks.append(α);
(15)    else if (msg is ResourceAllocated(r))
(16)        Holds(append(r));
(17)        CheckExecution(OwnTasks.getfirst(),
            Holds)
(18)    BidResources(OwnTasks.getfirst());
(19)    if (OwnTasks.getfist() is complete)
(20)        OwnTask.removeFirst();
(21)        SendToAuctioneer(CheckUnneeded(Holds));
```

Agents using auction based coordination will act at each step in the following way (see Algorithm 1). The agent first checks whether new tasks have become applicable. If so, the agent will submit the tasks for auction (line 3-6). The agent will then update its AuctionList and BidList (line 7). Next, agents will be required to receive messages. For each message, agent process in one of the four ways. If a message notifies any new task, the agent will consider a bid for that task and any other open task auctions (line 9-10). The value of a bid is calculated as

$$Reward(a, \alpha) = a.cap(\alpha) - dist(a.location, r.location)$$

Thus, the agent bids proportionally to its capability to perform the task but inversely proportionally to the time it will take to perform the task. If the message is to inform of an open resource auction, this resource will be added to the agent's OpenResources list (line 11-12). If the message is to notify the agent that is allocated to a task, this task will be added to OwnTasks (line 13-14). If it is allocated a resource, the agent checks whether any task is now executable (line 15-17). After processing all the messages, the agent will try to bid for required resources to perform the first task pending to be performed in OwnTasks (line 18). Notice that some resources are interchangeable, so the agent can bid for any of those resources. For example, for a fire fighting a bucket of water is interchangeable with a fire extinguisher. The agent will send bids for all combinations of OpenResources that will allow it to perform its first pending task. Finally, if any task has been completed, the resources will be released to the auctioneer for allocation to other agents (line 19-21).

The auctioneer allocates tasks and resources as described in Algorithm 2. The auctioneer processes all incoming messages (lines 3), records bids (lines 11-12) and open new auctions as required (lines 5-10). Then it makes a list for all auctions to be closed (lines 13). The auctioneer will determine an allocation for all the items in the list and they will be allocated (lines 14). Finally, bids for closed auctions are removed from lists (lines 15).

Algorithm 2: Auctioneer Algorithm

```
(1) while (true)
(2)     Auctions=[], Bids=[], ClosedTasks=[], Close-
        dResource=[];
(3)     Msgs → getMsgs();
(4)     foreach (m in Msgs)
(5)         if (m is Resource(r))
(6)             Broadcast(new Auction(r));
(7)             Auctions.append(r);
(8)         else if (m is Task(α))
(9)             Broadcast(new Auction(α));
(10)            Auctions.append(α);
(11)        else if (m is Bid)
(12)            Bids.append(m);
(13)    ClosingAuction ← toClose(Auctions);
(14)    DetermineWinner(ClosingAuction);
(15)    RemoveBids(ClosingAuction);
```

Token-Based Coordination

Token-based algorithms are a relatively new approach to coordination, designed for coordination of many agents(P. Scerri & Tambe. 2005; Guralnik. 2003; Y. Xu & Lewis. 2005). Specifically, here we use the approach as described in (Y. Xu & Lewis. 2005). Tokens, encapsulating both information and control, are the basis for all coordination. Control information, included with the token, allows actors to locally decide what to do with the token. For example, a *task token* contains control information allowing an actor to decide whether to perform the task or pass it off for another actor. An intelligent routing algorithm (Y. Xu & Lewis. 2005) is built in the token-based approach to help agents build local decision theoretic models to determine when and where to pass tokens. By utilizing the relevance between tokens, i.e, tokens representing resources useful for a particular task should be passed to the same agent as the token representing that task was, intelligent routing algorithm is able to efficiently deploy tokens to make higher utility with less communication. In this paper, tasks are allocated by the LA-DCOP token algorithm (P. Scerri & Tambe. 2005) where different with the basic

task allocation algorithm in (Y. Xu & Lewis. 2005), agent is allow to reject previous accepted task but accept another task that it can get more reward.

Specifically, in the token-based approach, each agent executes Algorithm 3. As with the auction-based approach, agents first check whether new tasks have become applicable. If so, the agent will embed the task to a token and add it into its token list, Tokens, to be processed (line 3-5). Next, the agent will receive all the tokens passed from other agents (line 6). It then processes all the tokens in the Tokens. If a token represents a task, the agent will accept the task if its capability to perform that task is higher than token's threshold (P. Scerri & Tambe. 2005) (lines 8-11), otherwise, the agent will choose a neighbor to pass that token to (line 13). If the token is a resource token, and the agent's need for that resource to perform his waiting tasks is higher than token's current threshold (Y. Xu & Lewis. 2005), this resource will be held otherwise it is passed to a neighbor (lines 14-21). Note that when a token is sent, the token will be removed from that agent's list. Finally, the agent will check whether any task which is pending can now be executed (lines 22) and release any resources from completed tasks (lines 23-28).

Algorithm 3: AgentToken

```
(1) ApplicableTasks=[], OwnTasks=[], Holds=[], Tokens=[]; while (true)
(2)     foreach (α in a, α ∉ ApplicableTasks)
(3)         if (Applicable(α))
(4)             ApplicableTasks.append(α);
(5)             Tokens.append(CreateTokens(α));
(6)     Tokens.append(recvTokens());
(7)     foreach (t ∈ Tokens)
(8)         if (t is TaskToken(α))
(9)             if (GetCap(α) >t.threshold)
(10)                if (α ∉ OwnTasks)
(11)                    OwnTasks.append(α);
(12)            else
(13)                SendToNeighbour(t);
(14)        else if (t is ResourceToken(r))
(15)            t.threshold+= δ;
(16)            if (GetNeed(r)>t.threshold)
(17)                if (r ∉ Holds)
(18)                    Holds.append(r);
(19)            else
(20)                t.threshold -=δ;
(21)                SendToNeighbour(token);
(22)    CheckExecution(OwnTasks, Holds);
(23)    foreach (α ∈ OwnTasks)
(24)        if (α is complete)
(25)            OwnTask.remove(α);
(26)            foreach (r ∈ ChkUnneed(OwnTask, Holds))
(27)                Hold.remove(r);
(28)                SendToNeighbour(CreateToken(r));
```

Hybrid Approach: Token-Based Auctions

The two algorithms described above take very different approaches and are based on distinctly different principles. The auction algorithm gathers lots of information, i.e., bids, and then makes an intelligent decision about how to allocate tasks and resources. On the other hand, the token algorithm makes informed estimates of about what good allocations will be like and attempts to directly target only those agents involved in an allocation of that quality or better. Intuitively, these principles can be combined into a hybrid algorithm that has the key advantages of both basic algorithms. Notice that the intention here is not to design a new algorithm but instead fuse two principles to see whether it performs best in the cases where neither of the two basic algorithms are particularly suited.

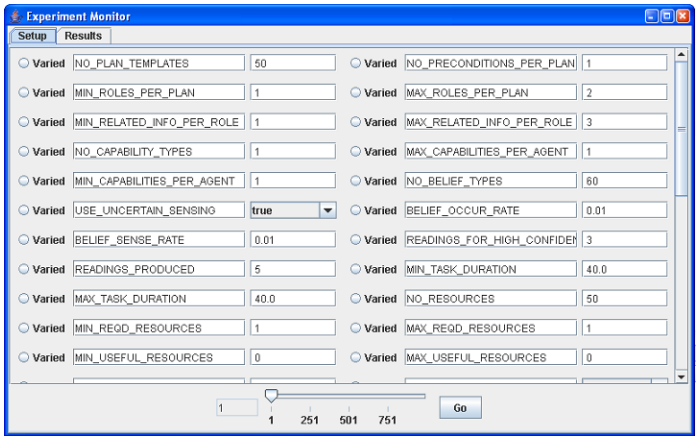

Figure 1: CoordSim allows us to test coordination algorithms by varying many parameters

The hybrid algorithm works in the following way. The auctioneer algorithm runs exactly as before, except that instead of broadcasting announcements for auctions an *auction token* is created. Each auction token is allowed to exist from the starting of the auction to the end of the auction being closed. The auctioneer has a probabilistic model of the team state, just as all agents do in the token-based approach. The auction token is then intelligently routed to the agents most likely to be able to submit the best bids. The token stops moving after the auction it presents is closing or has visited a fixed number of teammates. Note that although the intelligent routing algorithm should work to route tokens for higher bids, it should not work better than the token-based approach. The reason is that intelligent routing algorithm cannot make use of the relevance (Y. Xu & Lewis. 2005) between tasks and resources which have been encapsulated into auction tokens. When an agent is receiving an auction token, it cannot infer any knowledge about the sender whether it cannot make use of that task or resource. The auctioneer determines the winner of the auction and allocates tasks and resources the same as in the basic auction case.

We expect that the hybrid approach should reduce communication over the basic auction, by targeting only those agents likely to make good bids and reduce computation by limiting the number of bids the auctioneer must deal with.

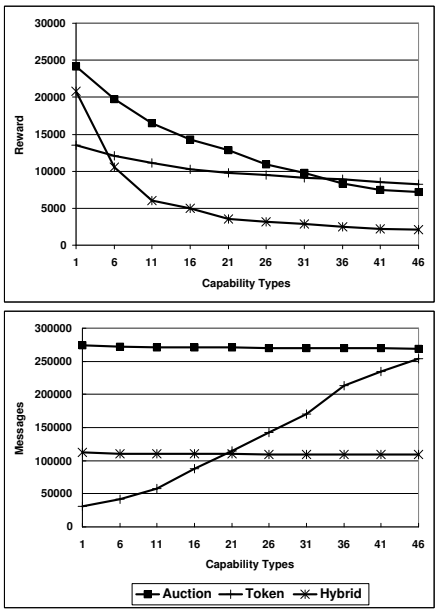

Figure 2: The average reward for heterogeneous teams dramatically decrease in auction and hybrid approaches. Token-based approach maintains constant reward but requires more messages

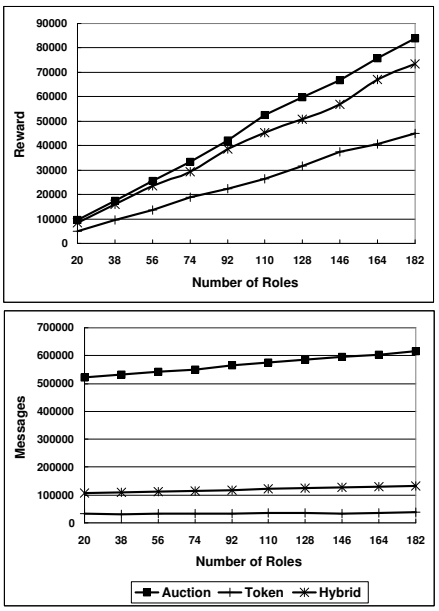

Figure 3: Reward and Messages increase dramatically with the number of tasks.

Experiments

In this section, we show how the comparisons were performed. The three approaches were implemented in an abstract simulator called CoordSim. This simulator is capable of simulating the major aspects of coordination including sensor fusion, plan management, information sharing, task assignment and resource allocation. CoordSim abstracts away the environment, instead just simulating its effects on the team. Uncertain sensor readings are received randomly by one or more agents in the team at a parameterizable rate. Agents cannot ”know” anything they do not sense or is not communicated to them from a teammate.

In the experiments, we use a consistent algorithm for sensor fusion and information sharing, specifically the algorithms described in (B. Yu & Lewis.. 2006; Y. Xu & Scerri. 2004). Physical resources required for tasks are simulated, only allowing one agent to access them at any time. There is no cost for transferring resources and resources cannot be consumed or lost. We simulate the spatial layout of tasks, distributing them randomly in an 500×500 environment. In these experiments all agents move at equal speed. Time is designed and all agents are allowed to ”think” and ”act” at each step, although the effects of their ”actions” are abstractly simulated. Communication is implemented via object passing, making it very fast. Reward is simulated as being received by the team when the agent is allocated the task, its simulated location is at the task location and it has exclusive access to required resources. Reward is received while the agent is simulating to take the task, which takes one time step.

CoordSim allows a large number of parameters to be varied and statistics to be recorded. Figure 1 shows the interface for setting up experiments and viewing results. If not otherwise stated, the experiments are configured as follows. There are 100 agents to perform 50 tasks with 50 resources. Each task requires only one resources which could be interchangeable with four others. In the default setup, there is only one type of capability required and all agents have none-zero value for this capability, i.e., all agents are at least somewhat capable of all tasks. Auctions are held open for 40 time steps and the task tokens, resource tokens are allowed to move unless accepted. The initial threshold on a task token is 100, meaning that the task will not be accepted by an agents until it can get a reward more than 100 by performing this task. We measured two key statistics required to support or refute our hypothesis about the algorithms. ”Reward” is the sum of reward received by each agent. ”Messages” is the number of times agents communicated, either between themselves or with the auctioneer. The ”messages” count indicates messages sent to perform sensor fusion, plan initiation and information sharing. Simulation runs for 2000 time steps. The experiment results below are based on 100 runs.

Heterogeneous Team

In the first experiment, we examined team performance by varying team composition and the capabilities required to perform tasks. For example, in an emergency response experiment some agents might only be able to fight fires while others could only provide medical treatment. As capabilities grew more varied fewer agents were available to perform particular tasks. In this experiment, we varied the number of capabilities from 1 to 46 where in the most heterogeneous

condition, only two agents on average are capable to performing a task.

The experimental results in Figure 2 show that for heterogeneous teams, auction and hybrid approaches earn less reward as the team becomes more heterogeneous because there are fewer agents able to compete for the more specialized tasks. The advantages of teamwide maximization of utility by the auctioneer decrease as there are progressively fewer feasible alternative bids. In contrast, reward for the token-based approach remain almost flat with increasing specialization. We propose two reasons. One is that token-based approach greedily finds reasonable solution rather than searching for the optimal. As the other reason, by passing a higher number of tokens around the network and making use the relevance between them, intelligent routing algorithm gets better knowledge to route tokens. This is manifested that although the average distance to route a token increases with heterogeneity as reflected in an increase in messages around the team, token-based approach maintains the same level of reward.

Time Critical Tasks

In the second experiment, we investigated team performance when many tasks needed to be performed within a short period of time. To increase their reward, teams were required to perform tasks and allocate resources as rapidly as possible. In this study we varied the number of tasks the teams were required to finish from 20 to 182. After 2000 time steps, the accumulated reward and message count were recorded as shown in Figure 3.

All three approaches performed more tasks in order to get higher reward. As expected, the auction approach attained higher reward than the hybrid or token-based approaches. Considering both reward and messages, however, the hybrid approach performs well by almost matching the reward obtained by the auction at just a quarter of the communication cost. The reason the hybrid approach achieves such good performance with so little communication overhead is that the intelligent routing algorithm limits communication to a small number of agents while high bidders must always be informed in auctions.

Competitive Resources

The third experiment used 200 tasks each requiring an average of four resources with no interchange possibilities. As available resources are increased from 4 to 40, competition for them declines and they become less likely to be a bottleneck.

The experiment was stopped after 1000 time steps. Figure 4 shows that the reward for the auction based approach increased rapidly with increases in resources. Both the token-based and hybrid approaches remained flat with token-based approach earning the highest reward at all levels of scarcity while the hybrid approach yielded very limited reward.

We hypothesized that because resource contention in this experiment was high the centralized control of the auction and hybrid approaches would often force agents to either bid for all four resources together or miss the task while the distributed token-based approach weakened this constraint.

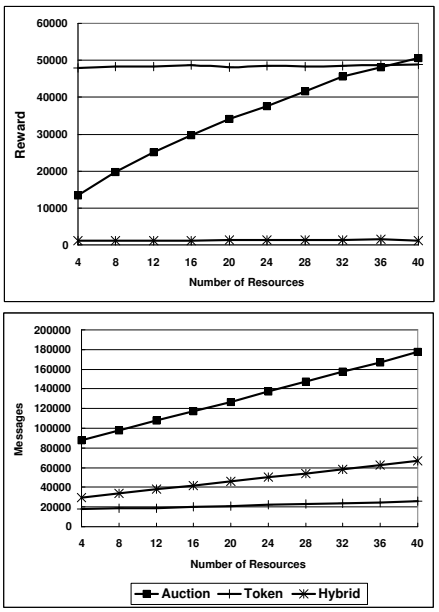

Figure 4: Reward increase with available resources in auction approach and are very low in hybrid approach, but with token-based approach are uniformly high

If our hypothesis were true, auction and hybrid approaches would get more reward if we either increased the simulation length or reduced the length of the auction to weaken the constraint. Figure 5 shows the effect of shortening auction length from 40 to 20 steps (a) and increasing session length from 2000 and 4000 steps (b). The token-based approach continues to produce its constant level of reward while the hybrid approach obtain slightly better rewards. The auction-based approach, however, improves with increasing resources exceeding the token-based approach at most levels in the two alternative experiments.

Interchangeable Resources

In the fourth experiment, there were 100 tasks each requiring three resources. The number of interchangeable resources were varied from 1 to 5. Experiments were stopped at 1000 steps. Results are shown in Figure 6.

Interchangeable resources did not help the token-based approach, helped the auction-based approach very little but substantially increased reward for the hybrid approach. We contend that three required resources for each task is a high constraint for a centralized auction. The constraint have been weaken in auction based approach because this experiment lasts long enough for auctioneer to search bids and maximize the reward. In contrast, this constraint is higher in hybrid approach because within a limited number of moving, all resource auction tokens for a role are required to visit an agent who has this task pending. This is also a reason why hybrid approach gained so low reward in section 4.3. Moreover, interchangeable resources led to dramatic increases in the number of messages for auction based approach because

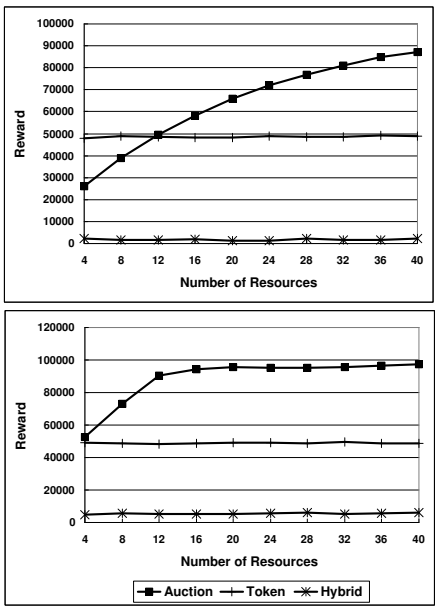

Figure 5: Auction approach gets more reward when auction last length is 20 (a) than as it is 40 or when experiments lasts from 2000 time points (b) than as it is 4000.

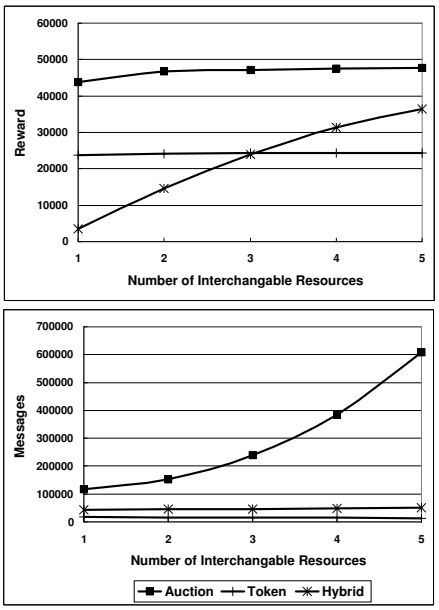

Figure 6: Reward for hybrid approach increases rapidly with more interchangeable resources

every agent could participate in every resource auction leading to the submission of a large number of multiple bids. For example, when interchangeable resources are 5, an agent should submit 5^3 resource bids.

Auction Length

In this experiment, we varied the length of auctions from 10 to 100 steps. In the hybrid approach, if the auction is open longer auction tokens can be passed to more agents and more agents have the opportunity to bid in the auction. In this experiment an auction lasting 100 steps, would provide every team member in the hybrid approach an opportunity to participate.

Figure 7 shows that for this experiment the auction approach obtained a uniformly high level of reward at all auction lengths. This reward, however, came at the cost of a large number of messages for short auctions. The hybrid approach, by contrast, had a uniformly low volume of messages and it approached a comparable level of reward with auction based approach very quickly as auction last long. This shows us that intelligent routing algorithm works as explained in section 4.2.

Handling Communication Failures

In this experiment, we investigated the performance of each approach working in an ad-hoc coordination domain where agents may randomly lose communication or fail. In this experiment, we varied the probability of communication failure from 0 to 9 percent.

Experimental results presented in Figure 8 show that uncertainty had greatest effect on the hybrid condition. As the bottom graph shows reward lose was greatest for the hybrid condition at all levels of failure. While the token-based approach had the poorest performance in this experiment, the rapid decline of reward with failure rate for the hybrid condition suggests that loss of auction tokens may have a disproportionate impact on system performance.

Conclusions

This paper presented a detailed, quantitative comparison of two distinct approaches to coordination. Our results showed that while both approaches might be used in a wide range of domains, their relative performance varied greatly. Moreover, there was a clear trade off, as expected, between quality of allocation and use of communication. The size of this trade off depended on the specific circumstances. Under some circumstances a hybrid of the two approaches appeared to provide a useful trade off by leveraging the strengths of both algorithms.

While this work represents an important first step towards quantitatively comparing distinct approaches to coordination, much work remains to be done. Critically in the comparison here, we used the simplest instantiations of the algorithms, ignoring the many performance enhancing techniques proposed in the literature. We intend to extend CoordSim to implement some of these extension. Just as importantly, while we considered many coordination issues, many others were ignored, e.g., individual failures, that may impact performance.

Acknowledgements

This research has been sponsored in part by AFRL/MNK Grant F08630-03-1-0005 and AFOSR Grant F49620-01-1-

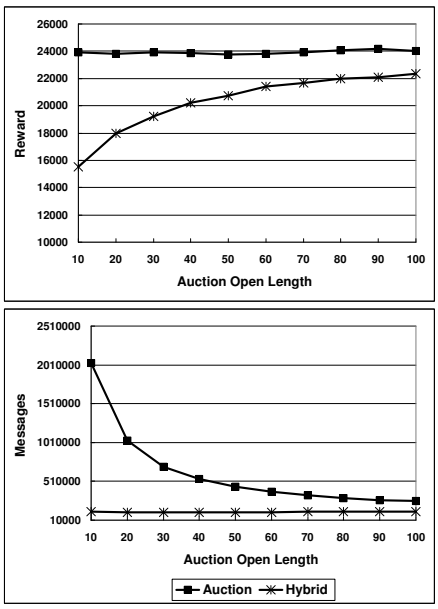

Figure 7: The hybrid approach can increase reward very rapidly with fewer messages as auction length becomes high

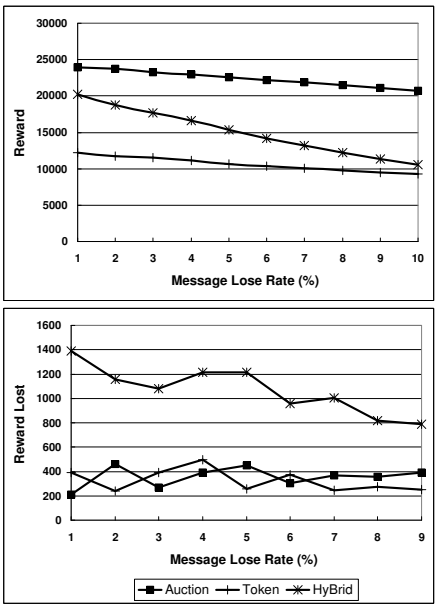

Figure 8: Hybrid loses more reward with communication failure

0542. We are grateful to Rob Zlot for invaluable help understanding the details of TraderBots algorithm.

References

B. Yu, P. Scerri, K. S. Y. X., and Lewis.., M. 2006. Scalable and reliable data delivery in mobile ad hoc sensor networks. In *Fifth Int. Conf. on Autonumous Agents and Multiagent Systems*.

Cicirello, V. A., and Smith., S. F. 2001. Wasp nests for self-configurable factories. In *The Fifth Int. Conf. on Autonomous Agents*.

Gerkey, B. P., and Mataric., M. J. 2003. Sold!: Auction methods for multirobot coordination. In *IEEE Trans. on Robotics and Automation, Special Issue on Multi-robot Systems*.

Dias, B., and Stentz., A. 2003. Traderbots: A market-based approach for resource, role, and task allocation in multirobot coordination. In *Technical report, CMU-RI - TR-03-19*.

E. David M. He, A. R., and Jennings., N. R. 2005. Designing and evaluating an adaptive trading agent for supply chain management applications. In *IJCAI-05 Workshop on Trading Agent Design and Analysis*.

Jacoff, E. M. A., and Evans., J. 2001. Experiences in deploying test arenas for autonomous mobile robots. In *2001 Performance Metrics for Intelligent Systems Workshop*.

Kaminka., G. 2000. The robocup-98 teamwork evaluation session: A preliminary report. In *RoboCup-99, LNAI*.

Mailler, R., and Lesser., V. 2004. Solving distributed constraint optimization problems using cooperative mediation. In *Third Int. Conf. on Autonumous Agents and Multiagent Systems*.

Modi, J., and Veloso., M. 2005. Bumping strategies for the multiagent agreement problem. In *Fourth Int. Conf. on Autonumous Agents and Multiagent Systems*.

N. Kalra B. Dias, R. Z., and Stentz., A. 2005. Market-based multirobot coordination: A survey and analysis. In *Technical report, CMU-RI-TR-05-13*.

P. Scerri, A. Farinelli, S. O., and Tambe., M. 2005. Allocating tasks in extreme teams. In *Fourth Int. Conf. on Autonumous Agents and Multiagent Systems*.

Sandholm., T. 2002. Algorithm for optimal winner determination in combinatorial auctions. In *Artificial Intelligence, 135*.

V. Guralnik T. Wagner and J. Phelps. 2003. A key-based coordination algorithm for dynamic readiness and repair service coordination. In *Second Int. Conf. on Autonomous Agents and Multiagent Systems*.

Y. Xu, P. Scerri, B. Y., S. O., and Lewis., M. 2005. An integrated token-based algorithm for scalable coordination. In *Fourth Int. Conf. on Autonumous Agents and Multiagent Systems*.

Y. Xu, S. K., Lewis., M., and P. Scerri. 2004. Information sharing among large scale teams. In *AAMAS'04 Workshop on Challenges in Coordination of Large Scale MultiAgent Systems*.

Yanco., H. 2001. Designing metrics for comparing the performance of robotic systems in robot competitions, measuring the performance and intelligence of systems. In *the 2001 PerMIS Workshop*.