A Multi-Agent System for Agent Coordination in Uncertain Environments

Lucian Vlad Lita, Jamieson Schulte, Sebastian Thrun
$\{llita, jschulte, thrun\}$@cs.cmu.edu
Carnegie Mellon University

ABSTRACT

We present a multi-agent architecture for coordinating large numbers of mobile agents (e.g. robots) cooperating in uncertain environments. Our approach addresses the problem of navigating large numbers of goal-driven agents through an environment whose state is unknown and has to be discovered during navigation. In our approach, each agent uses an approximate POMDP planner for generating contingency plans, and an efficient control brokering scheme for dispatching agents to goal locations. Extensive experimental results have been obtained in the context of natural disaster relief. Our experiments have been carried out in a realistic simulation of Honduras after Hurricane Mitch destroyed most of the country's infrastructure.

1. INTRODUCTION

This paper proposes a solution to the multi-agent Canadian Traveler Problem (CTP). The CTP is the problem of finding a shortest path to a goal location in a graph, where individual edges of the graph might or might not be traversable [1]. Whether or not an edge is traversable can only be found out by moving there. Hence, an optimal solution to a CTP is a contingency plan, which offers alternative routes if edges are not available. The CTP differs from other hard planning and optimization problems in that the state of the environment is only partially observable. It can be viewed as a partially observable Markov Decision Process (POMDP) with local and deterministic observations [7, 8, 2]. Finding an optimal contingency plan is known to be NP-hard.

The focus of the paper is a multi-agent CTP, which involves multiple agents attempting to reach multiple target locations. Finding an optimal solution is even harder, since the space of actions at each point in time is exponential in the number of agents. In practice optimal solutions can only be found for very small graphs (e.g., 10 edges) and a small number of agents (e.g., 3), whereas many practical multi-agent CTP problems possess hundreds of edges and agents.

The multi-agent CTP plays an important role in many practical problems. A classical example is packet routing in the Internet, where communication nodes might be up or down at random [6]. Clearly, packet routing has to be performed with extremely limited computation, making extensive planning (as proposed here) infeasible. More related to the work described here is the multi-robot exploration problem [5, 11, 12] where teams of robots jointly explore an unknown environment. Here, the relative speed of the moving agents (robots) is much slower than in the packet routing problem, offering the opportunity for planning and multi-agent coordination.

The problem that motivated the research described here is based on a disaster relief operation, which plays a central role in DARPA's Control of Agent-Based Systems program (known there as the MIATA Technology Integration Experiment). The problem is isomorphic to the two problems above: a large number of mobile agents (e.g., supply trucks) are tasked to bring emergency supplies to a number of population centers in a country devastated by a natural disaster. The road conditions are unknown. Hence, the optimal solution must offer backups from non-traversable roads. In particular, the natural disaster studied here takes place in October 1998 in Honduras, while and after Hurricane Mitch (Figure 1) destroyed up to 90% of the country's infrastructure in certain regions. The simulation used to validate our approach is based on high-fidelity data collected by the US Geological Survey in the aftermath of the disaster, in an attempt to improve relief operations for future disasters.

As noticed above, the multi-agent CTP has been studied extensively in the networking literature, with a strong emphasis on protocols and reactive policies that require little, if any, run-time computation or additional communication. The relative slowness of physical agents, such as relief trucks or robots, open the opportunity for on-line communication, planning, and coordination [9], which play central roles in the approach proposed here. Within AI and OR, the problem of acting under uncertainty has been studied in the POMDP literature, but virtually all existing approaches address the single-agent planning problem only. The standard solutions are exponentially hard in the number of agents and require exponential communication overhead, which renders them inapplicable to the multi-agent CTP.

We propose a scalable multi-agent architecture that approaches the above mentioned set of intractable problems in a effi-

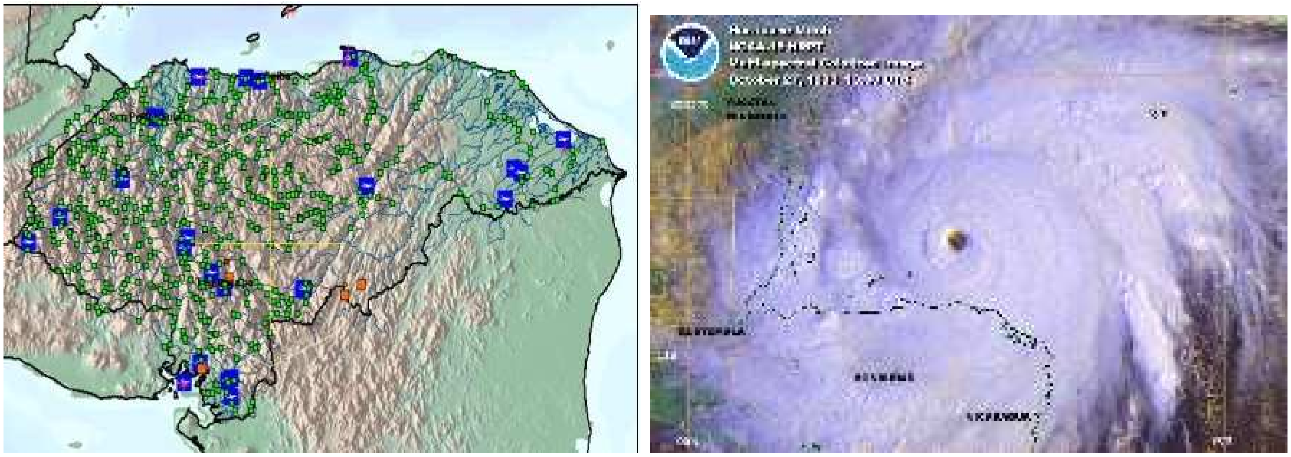

Figure 1: Multi-Agent Honduras Simulation and Hurricane Mitch

cient, real-time manner. The architecture supports a large number of mobile, goal-driven information agents that strive to maximize their reward for reaching goals. These agents are coordinated at a higher level by dispatcher agents whose purpose is to maximize the total reward cumulated over time.

2. PROBLEM DESCRIPTION

In our setting the agent's position l is certain but the state w of the unseen environment is unknown. Thus the agent state, which we call *belief state,* is represented as a pair:

$$(l, P(w|o)) \tag{1}$$

where $P(w|o)$ is the distribution representing the agent's belief of the state of the world given its observations. The belief space is the set of all possible belief states an agent could have given its observations. Our goal is to efficiently generate a good plan by searching the space of possible actions and the resulting belief states. This setting is commonly used in the POMDP literature [4].

When an action a_i is taken the belief state is updated according to success or failure. On success the agent traverses the edge and arrives at a new location l'. On failure the agent remains in the original location l. In either case the *observed history* is updated with the new observation o_i:

$$(l, P(w|o_{1..i-1})) \rightarrow \begin{cases} (l', P(w|o_{1..i})) & \text{on success} \\ (l, P(w|o_{1..i})) & \text{on failure} \end{cases} \tag{2}$$

The scenario consists of a natural disaster relief mission where immediate aid has the most impact and where the value of subsequent actions decreases exponentially with time. The agent attempts to maximize its reward in an uncertain environment. The reward $R(t)$ is discounted exponentially over time :

$$R(t) = e^{-\alpha t} \tag{3}$$

where α is the decay parameter. The agent receives reward only at the goal state. Therefor during planning, the agent should take into account the conditions under which it receives reward as well as the fact that reward decreases exponentially.

Finding the optimal solution involves an exhaustive search to determine the effect of every action in every belief state. We examine the space of actions, but limit the belief state exploration. We estimate the effects of actions in a trade-off of computation time for accuracy.

The problem becomes even harder in the multi-agent regime since the optimal solution is the best pairing of agents and goals. In a deterministic setting, for a large number of agents, the dispatching complexity is exponential in the number of agents. However, since the dispatchers have to analyze combinations of plans and joint probabilities of failure, the problem becomes even harder.

We have devised a planner that is very efficient under resource limitations (i.e. belief space exploration limitation), and we have incorporated it into our goal-driven information agents. In order to overcome the intractability of multi-agent coordination, we propose a dispatcher that makes use of plan information precomputed by information agents in parallel.

3. INFORMATION AGENTS

In this paper we consider information sharing, goal driven agents that operate according to optimal plans limited by their computational power and response (action) time. Bounded by these resources, each agent generates a set of contingency plans based on its own observed history as well as observations shared by its peers.

From the set of constructed plans the agent selects the best plan and then acts according to it, sharing its discoveries with interested agents within the system. We consider action cost to be dependent only on the base state, regardless of the outcome state. Finally, before each action the agent needs to reevaluate the soundness of its plans and replan if necessary.

```
BASTAR(goal, set of contingency plans, knowledge)
agent's task is to reach the goal; the set of contingency plans is initially empty. One contingency plan will eventually be
selected by the cummulative FDR measure; the agent also possesses an acquired knowledge based on observed history

• INITIALIZE a priority queue with initial belief state
• empty set of contingency plans ← initial belief state node
• UNTIL the queue is empty or the expansion limit is reached, DO
    1. belief state ← state with highest estimated FDR value in queue
    2. if belief state = goal
        – skip to the next iteration
    3. FOR ALL possible actions from belief state
        – FOR action's BOTH success and failure
            ∗ if action does not contradict the belief state's history or the agent's knowledge
                · new belief state ← belief state, action, observation (successs or failure)
                · new belief state → queue
                · new node in the set of contingency plans ← new belief state
• SELECT AND MARK best plan in the set of contingency plans based on the cumulative FDR measure
```

Table 1: The BA* algorithm

3.1 Planning with BA*

Traditionally, planning problems in deterministic spaces involve finding a path from an agent to its goal. A commonly used algorithm is A* which uses an optimistic (and thus *admissible*) measure of goal distance to efficiently find the optimal path [10]. A* makes use of a heuristic which provides an optimistic estimate of the remaining cost to the goal. This does not apply to uncertain environments where a path to the goal might be shorter but might also have a low probability of being traversable. Under uncertainty, agents have to plan for contingencies and cannot ignore the uncertainty inherent in the world. Thus, agents must plan in the space of all possible plans, accounting for contingencies [3].

We propose **Belief A* (BA*)**, an algorithm based on A* which efficiently handles planning under uncertainty. BA* searches in the space of all possible plans and produces the best contingency plan over the belief space found under time restrictions. For our planning problem, the algorithm generates a contingency plan represented by a tree and not a by a path as in the deterministic case. For practical purposes the BA* implementation uses memoization to cut down on the planning complexity in a similar fashion to value iteration.

BA* uses a heuristic that maximizes the return based on the expected future discounted reward (FDR) and on the probability of an agent passing through a particular belief state given the plan so far. An information agent associates an *observed* history h to each belief state it plans for. Since this is the planning stage and the agent has not actually *observed* anything, this history represents the sets of observations made by the agent in its path to the belief state. The reward $R(g|h)$ the agent would receive upon reaching the goal state g is:

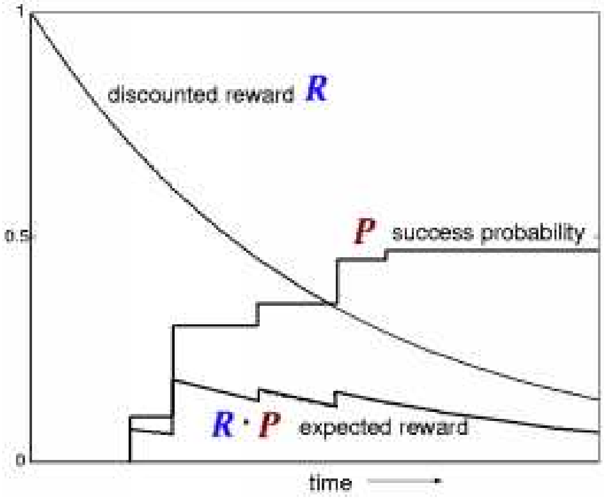

Figure 2: Expected Reward

$$R(g|h) = e^{-\alpha a(g)} \tag{4}$$

where $a(g)$ is the time it took the agent to reach the goal.

For planning, the agent needs to know the utility of each intermediate belief state encountered during the search. For any belief state s we can describe the expected reward $R^*(s)$ under an optimal plan:

$$R^*(s) = \sum_{w \in W} P(w|h) R(s,w) \tag{5}$$

where W is the set of all possible state configurations of the environment, $P(w|h)$ is the distribution of world states given the observed history, and $R(s,w)$ is the reward gained by applying the optimal plan from the state s to the goal (shown in Figure 2). Unfortunately, computing $R(s,w)$ is intractable since it is exponential in the number of belief states.

Hence, there is a need to *estimate* the utility of intermediate states in reaching the goal. Rather than explicitly attempting to compute the expected reward for intermediate belief states, the agents use a heuristic to guide the search in belief space [3]. The BA* algorithm uses this heuristic in order to direct the search toward states with higher expected reward. The estimated utility is measured by the expected FDR.

In a deterministic setting the value of any state s is be bounded by the minimal cost of any path that must travel through s to reach the goal. This value is used by A* as a heuristic. To account for uncertainty, the BA* algorithm introduces an optimistic estimate $P(s|h) \geq P(g|h)$ of the probability of reaching the goal along a path that travels through state s. Thus we can define a heuristic guaranteed to be optimistic in its estimate $\widehat{R}(s|h)$ of the expected reward of a sub-plan that passes through state s:

$$\widehat{R}(s|h) = P(s|h) \cdot e^{-\alpha(a(s)+b(s))} \tag{6}$$

where $a(s)$ is time the agent would take to reach belief state s following a trajectory in h, and $b(s)$ is a lower bound on the time the agent will take to reach the goal from s.

The Belief A* planning scheme reflects a preference for shorter paths through higher probability belief states. Response time and computational resources dictate how much BA* expands belief states during planning. Table 1 describes an information agent planner using the BA* algorithm with an estimated FDR heuristic.

3.2 Plan Evaluation

Each information agent evaluates the quality of a potential plan and computes the expected reward of reaching the goal R_{plan} over the set of potential future histories $|H|$ leading to the goal by following *plan*:

$$R_{plan} = \sum_{h \in |H|} R(g|h) \cdot P(g|h) \tag{7}$$

In order to choose the best available plan, the information agent maximizes the expected future discounted reward R_{max} over the spectrum of generated plans:

$$R_{max} = argmax_{plan_i} \{R_{plan_i}\} \tag{8}$$

The advantage of the BA* algorithm is that it tends to generate relevant belief states towards reaching the goal. Thus, based on the expected reward, the information agent efficiently chooses a plan that is very likely to produce a generous reward.

3.3 Acting

The architecture allows agents to move through the environment and provides appropriate responses when the traversability of an edge is discovered. These external observations place the agent in a non-deterministic partially observable world which it has to traverse towards the goal. The reaction time and the computation power limit the quality of the produced plan. However, the quality of the plan also affects which actions are being performed. Thus, the quality of the solution found by the agent is related to its processing power and dependent on action cost - the time it takes to perform an action.

Upon discovery of new information, agents broadcast the finding, and interested peers can choose to remember it and process it if they consider it useful. Agents could also choose to ignore externally discovered information. In assessing the overall solution quality, we also considered probability-blind agents which act upon action cost alone and disregard the probability of reaching a state. They act close to optimal when state transition probabilities are high - i.e. when natural disasters affect the world lightly. In reality natural disasters do affect the environment and agents perform better when choosing actions with using cost and success probability in the interest of increasing expected reward values.

3.4 Replanning

Individual agents have the opportunity to replan as soon as the current belief state changes. Belief states change as a result of actions and new knowledge acquisition and plans become obsolete if goals are fulfilled by other agents. Replanning could be done as differential process, updating and enhancing the current plan. However, if the information inflow is considerable, the agent could choose to discard the current plan. In this case, the information agent would construct and evaluate a new set of plans.

4. DISPATCHER AGENTS

A multi-agent architecture that assumes independence among agents is inefficient and does not maximize the overall reward. Under independence, agents select their own goals based on some utility measure. If a goal is faster to attain or has a higher probability to be reached, many agents

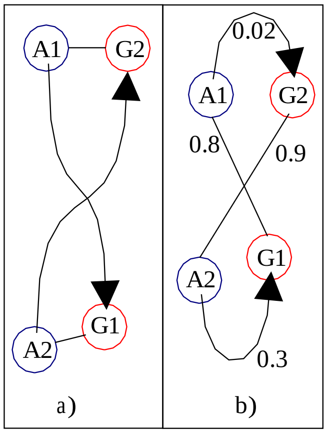

Figure 3: a) Sequential pairing of agents to goals (A1,G1); (A2,G2) without accounting for distances. b)Euclidean pairing of agents to goals (A1,G2); (A2,G1) without accounting for probabilities.

would pursue it. While only one agent would attain the goal first, many others would have wasted time and resources and would have had to turn their attention towards other goals. Another problem is that agents could enter an oscillation mode if they are between two sets of goals that are serviced by other agents in close proximity. An oscillating agent might be indeterminately attempting to reach goals but never actually succeeding. A solution that involves full communication between agents must assume that the agents construct very frequently plans for all the goals taking into account their peers' plans as well. This solution cannot be considered under the real-time constraint.

We introduce *dispatcher agents,* a dynamic, fast and scalable mechanism that coordinates the assignment of agents to goals. Dispatcher agents attempt to maximize the total reward attained by the information agents R_{total}, using only an estimate of individual agent reward expectations $\hat{R}_{total}$:

$$R_{total} = \sum_{g \in G} e^{-\alpha t_g} \tag{9}$$

This formulation represents the goal-centric view held by the dispatcher, which attempts to minimize the times t_g required by the information agents to reach the full set of goals G.

With large numbers of agents, optimal dispatching is intractable if we are assessing the expected reward of all the combinations of agents and goals. We propose a fast response time, greedy dispatcher that allows agents to communicate. The dispatcher avoids the redundant computation of expected reward. Instead, the dispatcher uses an expected FDR measure computed in parallel by information agents. Based on the precomputed FDR the dispatcher assigns agents to goals.

The fastest dispatcher agent algorithm performs sequential, queue-based allocation of agents to goals. If there is no favored ordering on the agent and goal queues, the dispatcher agent could have a very low performance (Figure 3 a). A greedy algorithm based on Euclidean distance performs better, but still does not account for the probabilistic nature of the environment. It also does not account for the exponential decrease in reward over time (Figure 3 b).

We propose a fast, greedy expected FDR dispatcher agent. The dispatching algorithm overcomes the need to analyze the quality of individual assignments of agents to goals. The dispatcher instructs each information agent to find a plan for each goal. It then uses the information agents' precomputed FDR and plan information in order to assign appropriate agent-goal pairs in a greedy fashion.

5. A MULTI-AGENT ARCHITECTURE

Different types of information agents and dispatcher agents coexist in a scalable architecture (Figure 4). Their goal and planning engine are encapsulated such that communication is minimal and still conveys the desired information. The information agents perform the initial planning and communicate with their dispatcher agents. Upon receiving their goal assignments, the mobile agents attempt to reach their goals. Planning and acting are the information agent's responsibility and discoveries are shared with their peers. Communication is agent type free and the architecture does not coerce agents to plan or acquire any shared knowledge.

One requirement is to minimize on-line communication so that the architecture remains scalable. However, the architecture must also allow enough communication so that agents benefit from knowledge sharing. The greedy expected FDR dispatcher allows these restrictions to be practical. The FDR dispatcher absorbs planning information using little communication while benefiting from the parallelism and coordination inherent in the multi-agent architecture.

6. EVALUATION

Experiments show that our multi-agent architecture supports a large number of agents that plan and act *efficiently*, in *real-time* in realistic uncertain environments.

6.1 The Simulator

The MapleSim simulator was used to prepare data for agent experiments, and can be used as a test environment in which planning agents work in a realistic setting. The simulator and the agent architecture presented here were developed for the MIATA technology integration experiment group within the DARPA CoABS initiative. The simulation provides a means for distributed control of many heterogeneous agents in the Honduras disaster relief scenario using real data. In the scenario, the network of roads, cities, airports, and communications is critically damaged as hurricane Mitch passes through, and foreign aid groups are tasked with providing supplies and rebuilding the infrastructure quickly. What makes the problem particularly difficult is that the extent of the damage is largely unknown in advance, and must be discovered in the process of providing relief.

6.2 Environment

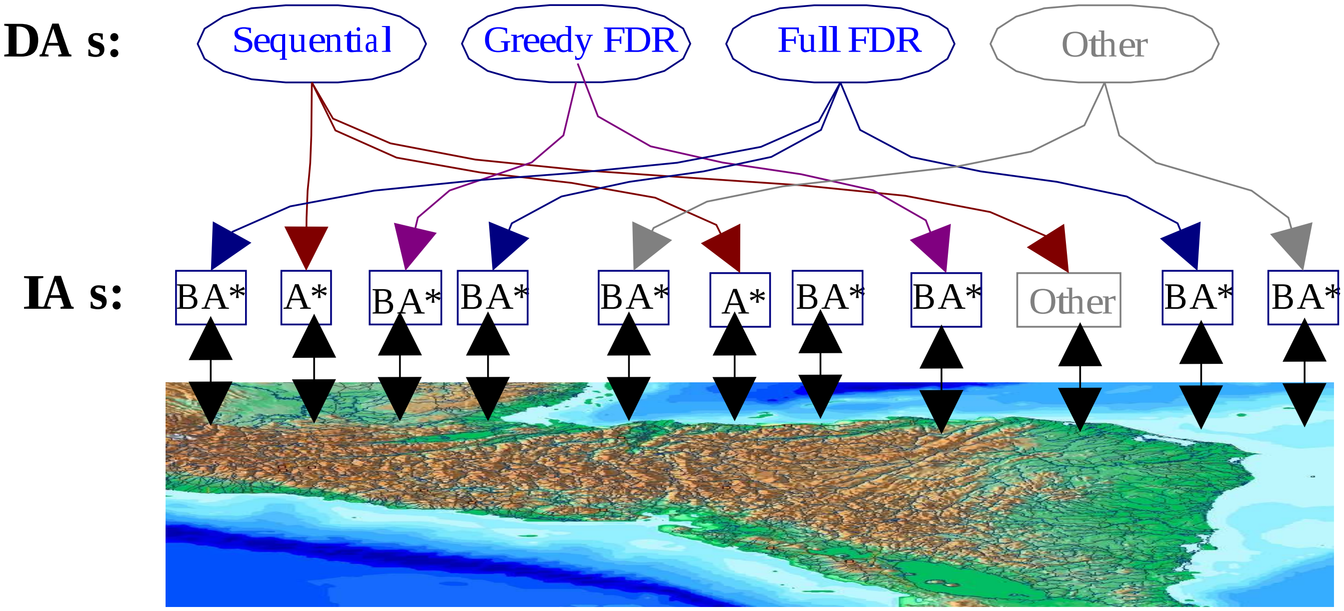

Figure 4: A scalable, extensible multi agent architecture. Dispatcher Agents (DAs) coordinate Information Agents (IAs)

The experiments were performed in a large environment simulating Honduras following a natural disaster. We have implemented a synchronous multi-agent architecture that keeps track of the environment responses, time and rewards, and information exchange. Agent actions trigger environment responses which let the information agents know the traversability of edges. These responses also entail the subsequent belief state of the information agent. The architecture synchronizes the information agents' actions based on cost (time). The environment also provides the structure and state transition mappings that information agents use for planning purposes. Information exchange is also supported and it is initiated by each individual information agent upon new discovery, or by dispatcher agents for agent-goal matching.

We analyzed the performance of the system using different disaster scenarios which entail different state transition probabilities. We varied the world instance as a sampling of random variables representing more than 300 road segments and having the distributions dictated by each specific scenario. For these world instances we varied the number of agents and goals as well as their physical locations. We considered different information agent types that could communicate and we also considered different dispatchers in each case. The simulations were performed on up to 100 information agents and we found that a single dispatcher managed these information agents well.

6.3 Information Agent

Our information agent experiments involved simulating incremental numbers of agents, in many different scenarios. In most situations the BA* based algorithm took advantage of state transition probability information in order to weigh the impact of low cost, but high risk actions. Using realistic scenarios, we compared information agents using a BA* heuristic versus information agents using a deterministic, shortest path approach.*Performance improvement* is the ratio of *naive* agents' average reward to that of the agents under evaluation. The *naive* agents use simple shortest-path planning, do not communicate, and use a random dispatcher. The plots in Figure 5 present results on two realistic scenarios and do not represent the best case scenario for the BA* information agent.

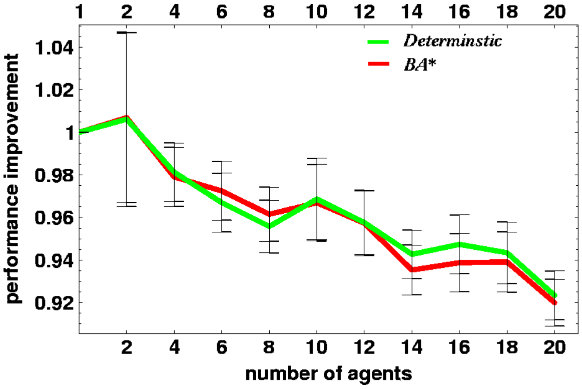

Figure 6: Not a favorable scenario for the BA* Information Agents. They do however behave similar to the Deterministic Agents

However, since the BA* heuristic exploits probability information in expected reward calculations, we evaluated a hypothetical scenario in a nearly-deterministic world where roads are up or down with high certainty. We found that in this case, the BA* approach also exhibits a similar behavior (Figure 6).

We limited planning time through the BA* belief state ex-

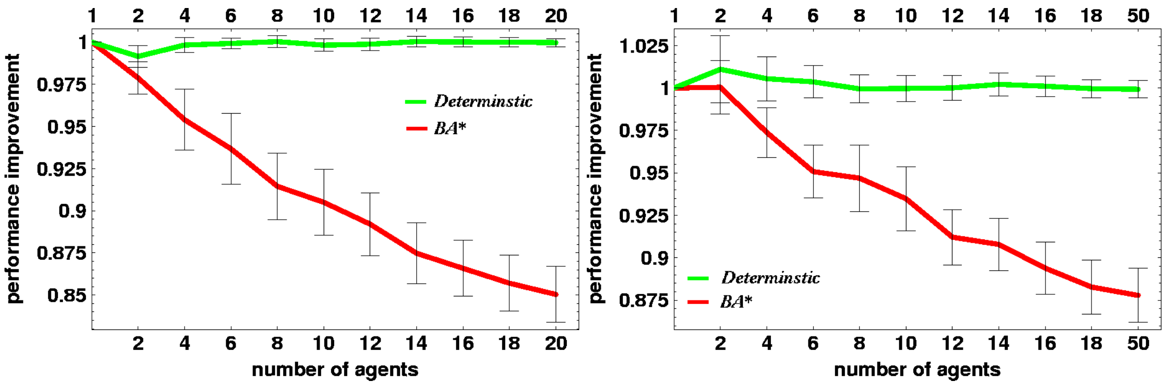

Figure 5: Two realistic scenarios showing that BA* Information Agents perform better than Deterministic Agents

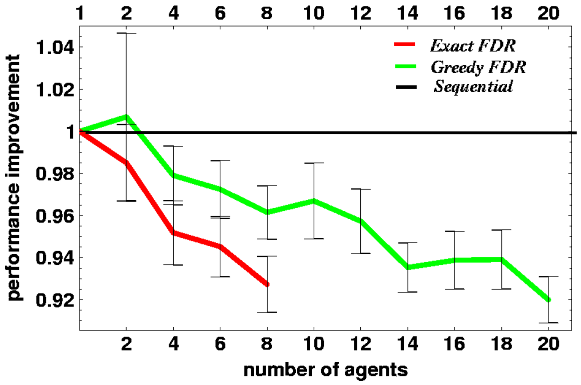

Figure 7: Greedy Dispatchers Agents outperform Sequential Dispatchers. The exact (exhaustive) FDR is also shown for comparison purposes

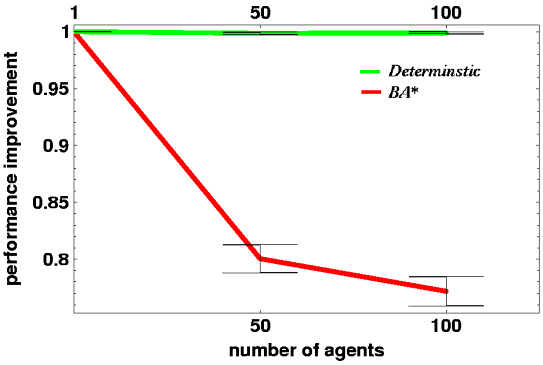

Figure 8: Large scale multi-agent experiment showing the performance increase of BA* Information Agents over Deterministic Information Agents

pansion limit so that a tradeoff exists between response time and accuracy. The information agents develop high-reward plans in our large scale Honduras environment in under 5 seconds on a 500MHz PC.

6.4 Dispatcher Agents

In order to test the performance of our proposed dispatcher, we set up an experiment where each dispatcher type coordinates information agents in different world scenarios. The sequential dispatcher matches the agents and goals as they are appear in queue. The greedy expected FDR dispatcher requires the information agents to plan for every goal, and selects a corresponding set of best plans. The greedy dispatcher would then proceed and couple goals and agents.

In order to assess the quality of the greedy expected FDR dispatching scheme, we constructed an "exact" expected FDR dispatcher, which fully searches the space of agent-goal assignments to maximize the expected reward. Due to the combinatorial nature of the search, this exact dispatcher can only handle very few agents, but offers an insight into the quality of the greedy expected FDR scheme. Its procedure is similar to the greedy expected FDR dispatcher, but it searches all combinations, rather than performing hill-climbing. The results shown in Figure 7 allow agents to communicate. We found that dispatching overhead is insignificant compared to the amount of planning information agents are required to perform.

6.5 Large scale experiment

In a large scale experiment, we averaged the performance of 100 information agents over a set of instances of a scenario with multiple random sampling the physical locations of goals and agents. BA* based agents have a 20% performance improvement in a realistic setting (Figure 8).

7. DISCUSSION

Through a multi-agent architecture we address the problem of coordinating large numbers of mobile agents cooperating in uncertain environments. Large numbers of goal-driven agents act in an environment with an initially unknown state. Each information agent uses an approximate POMDP planner for generating contingency plans. The information agents are coordinated by dispatcher agents and assigned to goal locations that yield a good overall reward.

Experiments show that our multi-agent architecture when applied to large number of coordinated information agents is both practical and efficient. Information agents are able to devise efficient plans based on the realistic measure of expected FDR. Dispatcher agents use the plan information precomputed in parallel by information agents in order to map agents to goals. Significant improvement is seen in simulated realistic disaster scenarios, while simulations of environments under normal conditions yield results close to those of deterministic planners.

The design of a practical large scale multi-agent architecture poses many difficult problems some of which we plan to address in future work. The current approach does not consider the sequence of actions information agents choose. If two agents have similar plans, then the probability that one will fail is closely correlated with the probability that the other will fail as well. Future work will focus on improving dispatcher agents by reducing the joint probability of failure in information agents.

Further future work will address an ever-changing environment. The setting entails that agent observations do not preserve their information content over time. State transition probabilities in this case would not be fixed, and therefor agents would possess ephemeral knowledge. Re-dispatching and additive dispatching are two more issues we are considering for future research.

8. REFERENCES

[1] A. Bar-Noy and B. Schieber. The canadian traveler problem. In *Proceedings of the 2nd ACM-SIAM Symposium on Discrete Algorithms*, pages 261–270, 1991.

[2] D. M. Blei and L. P. Kaelbling. Shortest paths in a dynamic uncertain domain. In *IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments*, Menlo Park, CA, 1999.

[3] B. Bonet and H. Geffner. Planning with incomplete information as heuristic search in belief space. *Proc. 5th Int. Conf. on AI Planning and Scheduling (AIPS)*, April 2000.

[4] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of AI Research (JAIR)*, 11:1–94, 1999.

[5] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, 2000. IEEE.

[6] A. Itai and H. Shachnai. Adaptive source routing in high-speed networks. *Journal of Algorithms*, 20:218–243, 1996.

[7] L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[8] M. Littman, T. Cassandra, S. Hanks, and L. Kaelbling, editors. *AAAI Fall Symposium, Workshop on Planning with Partially Observable Markov Decision Processes*, Orlando, FL, October 1998. AAAI Press.

[9] M. J. Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, January 1997.

[10] N. J. Nilsson. *Principles of Artificial Intelligence*. Springer Publisher, Berlin, New York, 1982.

[11] R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, TX, 2000. AAAI.

[12] B. Yamauchi, P. Langley, A. Schultz, J. Grefenstette, and W. Adams. Magellan: An integrated adaptive architecture for mobile robots. Technical Report 98-2, Institute for the Study of Learning and Expertise (ISLE), Palo Alto, CA, May 1998.