Multiagent Teamwork: Hybrid Approaches

P. Paruchuri, E. Bowring, R. Nair, J.P. Pearce, N. Schurr, M. Tambe, P. Varakantham
University of Southern California, Los Angeles, CA 90089, USA
http://teamcore.usc.edu

ABSTRACT

Today within the multiagent community, we see at least four competing methods to building multiagent systems: belief-desire-intention (BDI), distributed constraint optimization (DCOP), distributed POMDPs, and auctions or game-theoretic methods. While there is exciting progress within each approach, there is a lack of cross-cutting research. This article highlights the various hybrid techniques for multiagent teamwork developed by the teamcore group. In particular, for the past decade, the TEAMCORE research group has focused on building agent teams in complex, dynamic domains. While our early work was inspired by BDI, we will present an overview of recent research that uses DCOPs and distributed POMDPs in building agent teams. While DCOP and distributed POMDP algorithms provide promising results, hybrid approaches allow us to use the complementary strengths of different techniques to create algorithms that perform better than either of their component algorithms alone. For example, in the BDI-POMDP hybrid approach, BDI team plans are exploited to improve POMDP tractability, and POMDPs improve BDI team plan performance.

1. INTRODUCTION

The long-term goal of our research is to facilitate building heterogeneous teams composed of software agents, robots, people etc operating in dynamic and real-time domains. Such teamwork is important in several applications like virtual environments for training [25, 21], RoboCup robot soccer [24], office work [20], disaster rescue applications [19] etc. Today within the multiagent community, we see at least four competing methods to building multiagent teams acting in complex and dynamic environments. First, Distributed Constraint Optimization (DCOP) methods exploit locality of interaction in seeking a local or global optimum [10, 1, 8, 18]. Second, distributed Partially Observable Markov Decision Problems (POMDPs) focus on team coordination in the presence of uncertainty in actions and observations in real-world domains [20, 13, 2]. Third, game-theoretic and auction based techniques focus on coordination among self-interested agents using market-oriented mechanisms [7] which may also be applied in team settings. Fourth, BDI approaches, inspired by logic and psychology, are symbolic approaches which arguably enable better human understanding of the methodology employed.

While there has been excellent progress in each of the four methods outlined above, there is an unfortunate lack of hybrid models that enable interactions among the four approaches, allowing them to overcome each other's weaknesses. For example, current BDI team approaches lack tools for quantitative performance analysis under uncertainty. Distributed POMDPs on the other hand are well-suited for such analysis but the complexity of finding optimal policies in such models is highly intractable. Fortunately, with a BDI-POMDP hybrid approach, BDI team plans are exploited to improve POMDP tractability, and POMDPs improve BDI team plan performance. Similarly, a hybrid DCOP-POMDP approach combines the DCOP strength of reasoning about local interactions among agents with a POMDP's ability to reason about uncertainty. We outline several such interactions in this article.

2. TEAMWORK APPLICATIONS

Our early work focused on teams of pilots-agents flying simulated helicopters for mission rehearsal simulations [23] and teams for RoboCup Soccer simulations [24]. While this early work focused on small-scale homogeneous agent teams in simulated environments, our recent work addresses larger-scale heterogeneous teams. We describe here two recent application domains and our continuing work in these domains.

Personal assistant agents: Individual software agents embedded within an organization can represent each human user in the organization and act on their behalf. Such agentified organizations may be highly beneficial in domains like disaster rescue, where teams composed of agent-assisted response vehicles, robots and people may enable more rapid crisis response. Personal assistant teams are also useful in office environments like the "Electric Elves", an agent system deployed at USC that ran continuously for nine months [20]. The team of 15-20 agents aided in daily tasks like rescheduling meetings, selecting presenters for research meetings and ordering meals. Section 3.1 describes the hybrid approach adopted in these proxies. Partly building on this experience, work has begun on a more comprehensive joint project with SRI International called CALO.

Distributed sensor nets: This domain consists of multiple stationary sensors, each controlled by an independent agent, and targets moving through their sensing range (see Figure 1) [4] [9]. Each sensor is equipped with a doppler radar with multiple sectors. An agent may activate one sector at a time or switch the sensor off. The sensor agents must act as a team to cooperatively track the targets. In particular, in order for a target to be tracked accurately, multiple agents must concurrently turn on overlapping sectors. There may not be enough sensors to track all possible targets so agents have to sacrifice tracking some lower priority targets in order to ensure that they globally optimize tracking performance. Additionally, sensor readings may be noisy, and the situation may be dynamic with targets moving through the sensing range. Our early work utilized a standard DCOP approach to address the resource allocation problem — allocating sensors to targets — that arises in this domain. While DCOP addresses the locality of agent interactions in this domain, it is unable to address the sensor uncertainty by itself. A hybrid approach that combines DCOPs with POMDPs called ND-POMDPs (refer 3.4), promises to address this shortcoming.

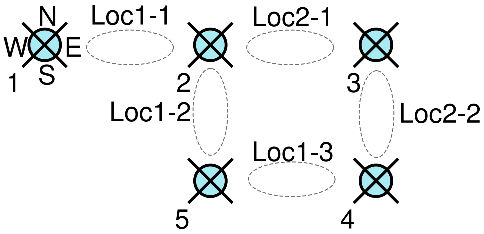


Figure 1: We show five sensors, with overlapping areas between the sensors. Loci-j refers to the jth step in target i's trajectory

Disaster Response Simulation: We have constructed a research prototype, called DEFACTO (Demonstrating Effective Flexible Agent Coordination of Teams through Omnipresence) to explore human-multiagent interaction (see Figure 2). Though DEFACTO can be used for deployed applications, it is initially being used as a modeling and simulation tool to improve on current disaster response training methods [21]. DEFACTO represents each team member (simulated fire-engines and human) with a proxy, which handles both the coordination and communication for the human-multiagent team. Experiments have been conducted that have humans interact with the teams of agents varying the adjustable autonomy strategies (see 3.1) and the size of the team. This prototype has been demonstrated to the Los Angeles Fire Department with positive and helpful feedback [22].

Building selected: Yes
Building Name: 1861
Fire Brigade Assigned: NO
Total area: 43627
Fieryness: 0
Type: Refuge
HasBeenExplored: true
AgentsInside: 7
Height: 40000

Figure 2: Disaster Response Simulator

3. CASE STUDIES IN HYBRID MODELS

This section provides an overview of four specific projects which employed hybrid approaches and the benefits gained.

3.1 Human-agent task allocation: BDI-POMDP hybrids

Adjustable autonomy refers to agents in a human-agent team dynamically varying their own autonomy in order to allow decisions to be made by the best teammate, be it human or agent. The main issue that adjustable autonomy addresses is whether and when agents should make a decision autonomously or transfer decision making control to other entities. Previous research framed the problem in terms of two choices: either transfer control or take autonomous action. With only these two options an agent is forced to either take a risky decision or risk incurring the cost of miscoordination (as a result of waiting for human response). To reduce this risk, we introduced the notion of a *transfer-of-control strategy*, which is a pre-planned sequence of transfer-of-control and deadline delaying actions. Thus, the key adjustable autonomy problem in agent-team settings is to select the right transfer-of-control strategy, i.e. the one that provides the benefit of high-quality decisions without risking significant costs in interrupting the user or miscoordination with the team. Furthermore, an agent must select the right strategy despite significant uncertainty about whether the user will respond to a request for input and whether the agent itself can make a correct decision.

Our hybrids in this area apply decision theoretic techniques (MDP or POMDP) to the team problem of strategy selection, whereas the rest of the team coordination is handled by BDI inspired methods. Though MDPs provide for sequential decision making in the presence of transitional uncertainty [20], they cannot handle observational uncertainty. In order to address this issue, we use POMDPs to model the adjustable autonomy problem. We have developed efficient exact algorithms for POMDPs, deployed in service of adjustable autonomy, by exploiting the notions of progress in the environment [26].

The usefulness of the hybrids is seen in that we are not using optimization methods to solve the whole team coordination problem. The team is actually executing a team-oriented program (TOP), i.e. abstract symbolic specifications of sequences of team activities, and the communication among the team members is controlled by BDI coordination [23, 20]. It is while executing a single task or a role in service of executing this TOP, that MDPs or POMDPs get employed to find optimal transfer-of-control strategies. Thus, instead of the complex reasoning about all of team coordination, we restrict it to specific team tasks.

3.2 Multiagent task allocation: BDI and Distributed POMDPs

We next shift our focus from single agent to distributed POMDPs. We now describe a more complex hybrid BDI-POMDP approach [12], where BDI team plans are exploited to improve POMDP tractability and POMDP analysis betters BDI team plan performance through improved role allocation, i.e. which agents to assign to the different roles in the team.

This hybrid approach (see Figure 3) combines the strengths of BDI plans and RMTDP (role-based multiagent team decision problem), an extension of MTDP that enables quantitative evaluation of role allocations. This interaction enables RMTDPs to improve the performance of BDI-based teams. We have also identified four ways in which BDI team plans make it easier to build RMTDPs and to efficiently search

RMTDP policies. First, we use the pre-conditions and post-conditions in the BDI plans to mathematically define the domain for an RMTDP. Second, the BDI plans provide partial policies to RMTDPs, restricting the policy search. Next, the BDI plan hierarchy helps decompose the RMTDP policy search, thus improving its efficiency. In particular, we use the plan hierarchy to come up with an admissible heuristic called MAXEXP that allows us to do a branch-and-bound search in the role allocation policy space. Finally, the belief representation in BDI team plans is exploited to enable faster RMTDP policy evaluation.

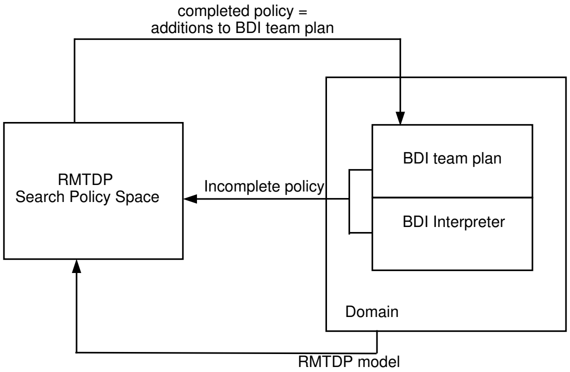
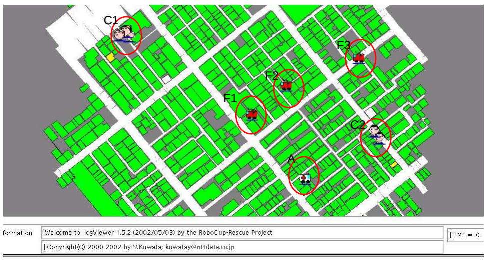


Figure 3: Integration of BDI and POMDP

We demonstrate the advantages of this hybrid approach via a scenario (see Figure 4) from the RoboCupRescue disaster simulation environment [24]. Here, five fire engines at three different fire stations (two each at stations 1 & 3 and the last at station 2) and five ambulances stationed at the ambulance center must collaborate to put out two fires (in top left and bottom right corners of the map) and to save the surviving civilians. The first goal is to determine which fire engines to assign to each fire. Once the fire engines have gathered information about the number of civilians at each fire, this is transmitted to the ambulances. The next goal is to allocate the ambulances to fires to rescue the civilians trapped there.



Welcome to logViewer 1.5.2 (2002/05/03) by the RoboCup-Rescue Project

Copyright(C) 2000-2002 by Y.Kuwata; kuwatay@nttdata.co.jp

Figure 4: RoboCupRescue Scenario: C1 and C2 denote the two fire locations, F1, F2 and F3 denote fire stations 1, 2 and 3 respectively and A denotes the ambulance center.

We compare the performance of the various allocations found via the role allocation policy search against the performance of human subjects (human1, human2, human3) and RescueISI (the third place team in RoboCupRescue 2001). This comparison was done via multiple runs in the RoboCupRescue simulation environment. We used two different settings for the distribution from which civilian locations were drawn: uniform and skewed. The metric for comparison was the number of civilian casualties and the amount of building damage. The three human subjects were familiar with the RoboCupRescue domain and were given time to study the setup and to provide their allocations. As can be seen in Figure 5(a), the RMTDP allocation did better than the other five allocations in terms of a lower number of civilians dead. Using the skewed distribution, the difference between the allocations was greater (see Figure 5(b)). The RMTDP allocation does much better than the humans in terms of the number of civilians dead.

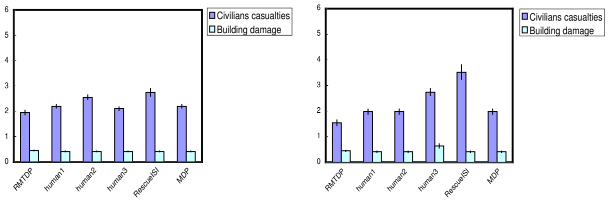


Figure 5: Comparison of performance in RoboCupRescue, a: (left) uniform, and b: (right) skewed.

3.3 Multiagent task allocation: Graphical games and DCOPs

Distributed constraint optimization problems (DCOPs) are a promising framework for modeling team optimization. In a DCOP, each agent assigns a value to one or more variables. Constraints that exist between subsets of these variables generate a cost or reward to the agent team, depending on the values chosen for the variables. The agents must coordinate their choices of variable values to maximize team reward. Figure 6 shows a DCOP with three agents each in control of one variable, with constraints between variables 1 and 2 and variables 2 and 3 both generating rewards for the team; the optimal solution is assigning 0 for all variables.

While *complete* DCOP algorithms, such as ADOPT (Asynchronous Distributed OPTimization) [11] reach a globally optimal solution, *incomplete* DCOP algorithms compute a local optimum. A more precise classification of incomplete algorithms is useful to understand the tradeoff between runtime and solution quality or the likelihood of finding the global optimal. We provide a hybrid solution concept, called k-optimality [17, 6], that draws from both graphical games and constraint reasoning to categorize incomplete DCOP algorithms and the local optima they reach. A k-optimal DCOP solution is an assignment of values to variables such that no subset S of k or fewer agents can improve its local utility, defined as the sum of the rewards on all constraints on agents in S; a k-optimal algorithm is an algorithm guaranteed to converge to a k-optimal solution. Under some assumptions, algorithms with higher k-optimality provide higher expected solution quality, and may require fewer restarts to reach a global optimum.

In experiments, while lower k algorithms converged to a stable solution more rapidly, higher k algorithms achieved a higher solution quality on average. Figure 7 [6] shows

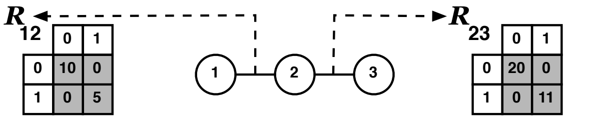


Figure 6: An example DCOP with three agents

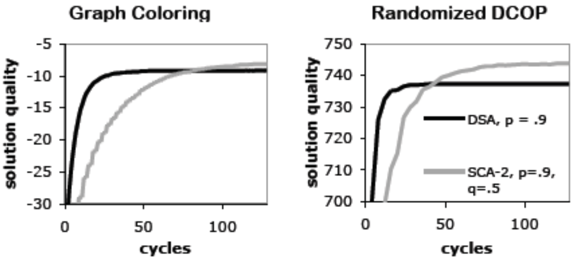


Figure 7: SCA-2 vs. DSA in two DCOP domains

the performance of DSA, an existing 1-optimal algorithm, against SCA-2, a new 2-optimal algorithm. The comparison was done over many examples, both in a three-coloring domain and a domain with constraint costs chosen from a uniform random distribution.

K-optimality also provides a novel tool to enumerate sets of multiple solutions with desirable properties. A set of k-optima is guaranteed to have a certain level of diversity (any two solutions must be separated by a Hamming distance of at least $k+1$) as well as relative quality (any solution X is of higher quality than any solution $\tilde{X}$ within a Hamming distance of k). Upper bounds on the number of possible k-optimal solutions to a DCOP can be obtained by leveraging results from coding theory [17]. In many domains, agent teams must generate multiple possible joint actions, either to execute in series or to provide a choice to a human operator. Each joint action generated may consume a resource, such as fuel (for vehicles), supplies (for troops) or time (for a human who must choose among the generated options). These bounds allow a human operator to choose a value of k in order to guarantee a particular level of diversity in the solution set, as well as to ensure that resources are not exhausted before all k- optimal solutions are found.

3.4 Multiagent task allocation: DCOPs and Distributed POMDPs

In many real-world multiagent applications, e.g. distributed sensor nets, a network of agents is formed based on each agent's interactions with a small number of neighbors. While distributed POMDPs capture the real-world uncertainty in multiagent domains, they fail to exploit the locality of interaction. Hence to exploit locality of interaction, we introduce the networked distributed POMDP (ND-POMDP) model [14] which is a hybrid of distributed POMDP and DCOP. The ND-POMDP model assumes transition and observation independence, with the reward function expressed as the sum of rewards for interacting agents. For instance in sensor nets, the reward is sum of the rewards of the interacting sensor agents. ND-POMDP can be mapped to a *n-ary* DCOP, where the agents are variables, domain of variables is set of agent's policies, and constraint (or interaction) graph is derived from the reward function of the ND-POMDP.

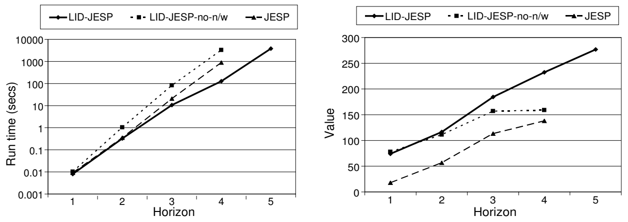


Figure 8: (a) Run time (secs), (b) Value

We developed a locally optimal policy generation algorithm called LID-JESP (locally interacting distributed joint equilibrium search for policies), based on the DCOP algorithm, DBA [27] and a distributed POMDP solver, JESP [13]. We present some initial results for the sensor net scenario in Figure 1, where there are five sensors (each capable of scanning in N, S, E, and W directions) trying to track two moving targets. Two neighboring sensors are needed to successfully track a target in the sector, with the sensors receiving false positive and false negative observations. We compared LID-JESP with Nair*et al.*'s the centralized JESP algorithm [13] (which does not consider the *interaction graph*) and LID-JESP-no-nw (LID-JESP with fully connected *interaction graph*). Figure 8(a) indicates run time comparisons on a logarithmic scale, with run-time in seconds on y-axis and time horizon, T on the x-axis, while Figure 8(b) shows the value comparisons, with value indicated on y-axis and time horizon, T on x-axis. The values obtained for LID-JESP, JESP and LID-JESP-no-nw are quite similar, although LID-JESP and LID-JESP-no-nw often converged on a higher local optimum than JESP. In comparing the run times, LID-JESP outperforms LID-JESP-no-nw and JESP which highlights the advantage of exploiting network structure to reduce the complexity of distributed POMDPs.

4. TEAMWORK: TOWARDS THE FUTURE

This article emphasized hybrid representations for scalability, and expressiveness in our current research on agent teams. It illustrated the interactions between distributed POMDP, DCOP, BDI and game theoretic representations. While this research focused on teams where team members are fully dedicated to their common goal (i.e. team members do not have additional explicitly represented selfish constraints), our recent research has begun focusing on such additional constraints. These constraints arise as we push teamwork into domains where there may be individual resources or privacy considerations. We describe four issues in current research addressing these problems:

- Formalization of resource-constrained teamwork using distributed MDPs: While previous distributed POMDP frameworks focused on agents with a joint reward function, we have also introduced the EMTDP framework to model agent teams where agents have additional individual resource constraints [15].

- Multiply-constrained optimization: While previous work in DCOP optimized a single global function, in multiply-constrained DCOP the goal is to also satisfy agents' individual constraints. We developed a unified algorithm that tailors its performance to the structure of the network and whether the constraint is to be kept private [3].

- Privacy in DCOP: While a key motivation for using DCOPs has been privacy, the effectiveness of DCOP algorithms in achieving this goal has not been investigated quantitatively across multiple metrics. We developed a framework [5] that allowed us to identify several key properties that lay hidden under the assumption that distribution automatically provides privacy.
- Security in POMDP teams: While the above work assumes that the agents act in environments where there is no adversary present, we started investigating the issue of teamwork in hostile environments. The technique we developed is called policy randomization [16] where the policies are developed by solving the multi-criterion problem that maximizes the policy randomness while maintaining reward constraints.

5. REFERENCES

[1] S. Ali, S.Koenig, and M.Tambe. Preprocessing techniques for accelerating the dcop algorithm adopt. In *AAMAS*, 2005.

[2] D. Bernstein, S.Zilberstein, and N.Immerman. The complexity of decentralized control of markov decision processes. In *UAI*, 2000.

[3] E. Bowring, M. Tambe, and M. Yokoo. Distributed multi-criteria coordination in multi-agent systems. In *Workshop on DALT*, 2005.

[4] V. Lesser, C.Ortiz, and M.Tambe. *Distributed sensor nets: A multiagent perspective*. Kluwer academic publishers, 2003.

[5] R. Maheswaran, J. Pearce, P. Varakantham, E. Bowring, and M. Tambe. Valuation of possible states: A unifying quantitative framework for evaluating privacy in collaboration. In *AAMAS*, 2005.

[6] R. Maheswaran, J. P. Pearce, and M. Tambe. Distributed algorithms for DCOP: A graphical-game-based approach. In *PDCS*, 2004.

[7] R. Maheswaran and T.Basar. Coalition formation in proportionality fair divisible auctions. In *AAMAS*, 2003.

[8] R. Mailler. Comparing two approaches to dynamic, distributed constraint satisfaction. In *AAMAS*, 2005.

[9] P. Modi, H.Jung, M.Tambe, W.Shen, and S.Kulkarni. A dynamic distributed constraint satisfaction approach to resource allocation. In *CP*, 2001.

[10] P. Modi, W. Shen, M. Tambe, and M. Yokoo. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *AIJ*, 161:149–180, 2005.

[11] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1-2):149–180, 2005.

[12] R. Nair and M.Tambe. Hybrid bdi-pomdp framework for multiagent teaming. *JAIR*, 23:367–413, 2005.

[13] R. Nair, M.Tambe, M.Yokoo, D.Pynadath, and S.Marsella. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *IJCAI*, 2003.

[14] R. Nair, P.Varakantham, M.Yokoo, and M.Tambe. Networked distributed pomdps: A synergy of distributed constraint optimization and pomdps. In *IJCAI*, 2005.

[15] P. Paruchuri, M.Tambe, F.Ordonez, and S.Kraus. Towards a formalization of teamwork with resource constraints. In *AAMAS*, 2004.

[16] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. Security in multiagent systems by policy randomization. In *AAMAS*, 2006.

[17] J. P. Pearce, R. T. Maheswaran, and M. Tambe. Solution sets for DCOPs and graphical games. In *AAMAS*, 2006.

[18] P. Scerri, A.Farinelli, S.Okamoto, and M.Tambe. Allocating tasks in extreme teams. In *AAMAS*, 2005.

[19] P. Scerri, L.Johnson, D.Pynadath, P.Rosenbloom, M.Si, N.Schurr, and M.Tambe. A prototype infrastructure for distributed robot, agent, person teams. In *AAMAS*, 2003.

[20] P. Scerri, D. Pynadath, and M. Tambe. Towards adjustable autonomy for the real-world. *JAIR*, 17:171–228, 2002.

[21] N. Schurr, J. Marecki, P. Scerri, J. Lewis, and M. Tambe. The defacto system: Training tool for incident commanders. In *IAAI*, 2005.

[22] N. Schurr, P. Patil, F. Pighin, and M. Tambe. Using multiagent teams to improve the training of incident commanders. In *Industry Track of AAMAS*, 2006.

[23] M. Tambe. Towards flexible teamwork. *JAIR*, 7:83–124, 1997.

[24] M. Tambe, G.Kaminka, S.Marsella, I.Muslea, and T.Raines. Two fielded teams and two experts: A robocup response challenge from the trenches. In *IJCAI*, 1999.

[25] M. Tambe, W. Johnson, R. Jones, F. Koss, J. Laird, P. Rosenbloom, and K. Schwamb. Intelligent agents for interactive simulation environments. *AI Magazine*, page 16(1), 1995.

[26] P. Varakantham, R. Maheswaran, and M. Tambe. Exploiting belief bounds: Practical pomdps for personal assistant agents. In *AAMAS*, 2005.

[27] M. Yokoo and K.Hirayama. Distributed breakout algorithm for solving distributed constraint satisfaction problems. In *ICMAS*, 1996.