Team Formation for Reformation

Ranjit Nair and **Milind Tambe** and **Stacy Marsella**

Computer Science Department and Information Sciences Institute
University of Southern California
Los Angeles CA 90089
{nair,tambe}@usc.edu, marsella@isi.edu

Introduction

The utility of the multi-agent team approach for coordination of distributed agents has been demonstrated in a number of large-scale systems for sensing and acting like sensor networks for real-time tracking of moving targets (Modi *et al.* 2001) and disaster rescue simulation domains, such as RoboCup Rescue Simulation Domain (Kitano *et al.* 1999; Tadokoro *et al.* 2000) These domains contain tasks that can be performed only by collaborative actions of the agents. Incomplete or incorrect knowledge owing to constrained sensing and uncertainty of the environment further motivate the need for these agents to explicitly work in teams. A key precursor to teamwork is team formation, the problem of how best to organize the agents into collaborating teams that perform the tasks that arise. For instance, in the disaster rescue simulation domain, injured civilians in a burning building may require teaming of two ambulances and three nearby fire-brigades to extinguish the fire and quickly rescue the civilians. If there are several such fires and injured civilians, the teams must be carefully formed to optimize performance.

Our work in team formation focuses on dynamic, real-time environments, such as sensor networks (Modi *et al.* 2001) and RoboCup Rescue Simulation Domain (Kitano *et al.* 1999; Tadokoro *et al.* 2000). In such domains teams must be formed rapidly so tasks are performed within given deadlines, and teams must be reformed in response to the dynamic appearance or disappearance of tasks. The problems with the current team formation work for such dynamic real-time domains are two-fold: i) most team formation algorithms (Tidhar, Rao, & Sonenberg 1996; Hunsberger & Grosz 2000; Fatima & Wooldridge 2001; Horling, Benyo, & Lesser 2001; Modi *et al.* 2001) are static. In order to adapt to the changing environment the static algorithm would have to be run repeatedly, ii) Team formation has largely relied on experimental work, without any theoretical analysis of key properties of team formation algorithms, such as their worst-case complexity. This is especially important because of the real-time nature of the domains.

In this paper we take initial steps to attack both these problems. As the tasks change and members of the team fail, the current team needs to evolve to handle the changes. In both the sensor network domain (Modi *et al.* 2001) and RoboCup Rescue (Kitano *et al.* 1999; Tadokoro *et al.* 2000), each re-organization of the team requires time (e.g., fire-brigades may need to drive to a new location) and is hence expensive because of the need for quick response. Clearly, the current configuration of agents is relevant to how quickly and well they can be re-organized in the future. Each reorganization of the teams should be such that the resulting team is effective at performing the existing tasks but also flexible enough to adapt to new scenarios quickly. We refer to this reorganization of the team as ”Team Formation for Reformation”. In order to solve the “Team Formation for Reformation” problem, we present *R-COM-MTDPs* (**R**oles and **Com**munication in a **M**arkov **T**eam **D**ecision **P**rocess), a formal model based on communicating decentralized POMDPs, to address the above shortcomings. R-COM-MTDP significantly extends an earlier model called COM-MTDP (Pynadath & Tambe 2002), by making important additions of roles and agents' local states, to more closely model current complex multiagent teams. Thus, R-COM-MTDP provides decentralized optimal policies to take up and change roles in a team (planning ahead to minimize reorganization costs), and to execute such roles.

R-COM-MTDPs provide a general tool to analyze role-taking and role-executing policies in multiagent teams. We show that while generation of optimal policies in R-COM-MTDPs is NEXP-complete, different communication and observability conditions significantly reduce such complexity. In this paper, we use the disaster rescue domain to motivate the “Team Formation for Reformation” problem. We present real world scenarios where such an approach would be useful and use the RoboCup Rescue Simulation Environment (Kitano *et al.* 1999; Tadokoro *et al.* 2000) to explain the working of our model.

Domain and Motivation

The RoboCup-Rescue Simulation Domain (Kitano *et al.* 1999; Tadokoro *et al.* 2000), provides an environment where large-scale earthquakes can be simulated and heterogeneous agents can collaborate in the task of disaster mitigation. Currently, the environment is a simulation of an earthquake in the Nagata ward in Kobe, Japan. As a result of the quake many buildings collapse, civilians get trapped, roads get damaged and gas leaks cause fires which spread to neighboring buildings. There are different kinds of agents that participate in the task of disaster mitigation viz. fire brigades, ambulances, police forces, fire stations, ambulance centers and police stations. In addition to having a large number of heterogeneous agents, the state of the environment is rapidly changing – buried civilians die, fires spread to neighboring buildings, buildings get burnt down, rescue agents run out of stamina, etc. There is uncertainty in the system on account of incorrect information or information not reaching agents. In such a hostile, uncertain and dynamically changing environment, a purely reactive method of team formation that relies on only current information is not likely to perform as well as a method that takes into account future tasks and reformations that are required. Reformations take time and hence we would like to form teams keeping in mind the future reformations that may be necessary.

While teams can be formed without look-ahead, the following scenarios illustrate the need for "Team Formation for Reformation".

1. A factory B catches fire at night. Since it is known that the factory is empty no casualties are likely. Without looking ahead at the possible outcomes of this fire, one would not give too much importance to this fire and might assign just one or two fire brigades to it. However, if by looking ahead, there is a high probability that the fire would spread to a nearby hospital, then more fire brigades and ambulances could be assigned to the factory and the surrounding area to reduce the response time. Moving fire brigades and ambulances to this area might leave other areas where new tasks could arise empty. Thus other ambulances and fire brigades could be moved to strategic locations within these areas.
2. There are two neighborhoods, one with small wooden houses close together and the other with houses of more fire resistant material. Both these neighborhoods have a fire in each of them with the fire in the wooden neighborhood being smaller at this time. Without looking ahead to how these fires might spread, more fire brigades may be assigned to the larger fire. But the fire in the wooden neighborhood might soon get larger and may require more fire brigades. Since we are strapped for resources, the response time to get more fire brigades from the first neighborhood to the second would be long and possibly critical.
3. There is an unexplored region of the world from which no reports of any incident have come in. This could be because nothing untoward has happened in that region or more likely, considering that a major earthquake has just taken place, that there has been a communication breakdown in that area. By considering both possibilities, it might be best if police agents take on the role of exploration to discover new tasks and ambulances and fire brigades ready themselves to perform the new tasks that may be discovered.

Each of these scenarios demonstrate that looking ahead at what events may arise in the future is critical to knowing what teams will need to be formed. The time to form these future teams from the current teams could be greatly reduced if the current teams were formed keeping this future reformation in mind.

From COM-MTDP to R-COM-MTDP

In the following sub-sections we present R-COM-MTDP, a formal model based on COM-MTDP (Pynadath & Tambe 2002), that can address the shortcomings of existing team formation approaches, and its application to RoboCup Rescue.

COM-MTDP

Given a team of selfless agents, α, a COM-MTDP (Pynadath & Tambe 2002) is a tuple, $\langle S, A_\alpha, \Sigma_\alpha, P, \Omega_\alpha, O_\alpha, B_\alpha, R\rangle$. S is a set of world states. $A_\alpha = \prod_{i\in\alpha} A_i$ is a set of combined domain-level actions, where A_i is the set of actions for agent i. $\Sigma_\alpha = \prod_{i\in\alpha}\Sigma_i$ is a set of combined messages, where Σ_i is the set of messages for agent i. $P(s_b, \mathbf{a}, s_e) = Pr(S^{t+1}{=}s_e|S^t{=}s_b, A^t_\alpha{=}\mathbf{a})$ governs the domain-level action's effects. $\Omega_\alpha = \prod_{i\in\alpha}\Omega_i$ is a set of combined observations, where Ω_i is the set of observations for agent i. Observation functions, $O_\alpha = \prod_{i\in\alpha} O_i$, specify a probability distribution over an agent's observations: $O_i(s, \mathbf{a}, \omega) = Pr(\Omega^t_i{=}\omega|S^t{=}s, A^{t-1}_\alpha{=}\mathbf{a})$ and may be classified as:

Collective Partial Observability: No assumptions are made about the observability of the world state.

Collective Observability: Team's combined observations uniquely determines world state: $\forall\boldsymbol{\omega}\in\Omega_\alpha$, $\exists s\in S$ such that $\forall s'\neq s$, $\Pr({\Omega_\alpha}^t = \boldsymbol{\omega}|S^t = s') = 0$.

Individual Observability: Each individual's observation uniquely determines the world state: $\forall\omega\in\Omega_i$, $\exists s\in S$ such that $\forall s'\neq s$, $\Pr(\Omega^t_i = \omega|S^t = s') = 0$.

Agent i chooses its actions and communication based on its belief state, $b^t_i \in B_i$, derived from the observations and communication it has received through time t. $B_\alpha = \prod_{i\in\alpha} B_i$ is the set of possible combined belief states. Like the Xuan-Lesser model(Xuan, Lesser, & Zilberstein 2001), each decision epoch t consists of two phases. In the first phase, each agent i updates its belief state on receiving its observation, $\omega^t_i\in\Omega_i$, and chooses a message to send to its teammates. In the second phase, it updates its beliefs based on communication received, Σ^t_α, and then chooses its action. The agents use separate state-estimator functions to update their belief states: initial belief state, $b^0_i = SE^0_i()$; pre-communication belief state, $b^t_{i\bullet\Sigma} = SE_{i\bullet\Sigma}(b^{t-1}_{i\Sigma\bullet}, \omega^t_i)$; and post-communication belief state, $b^t_{i\Sigma\bullet} = SE_{i\Sigma\bullet}(b^t_{i\bullet\Sigma}, \Sigma^t_\alpha)$.

The COM-MTDP reward function represents the team's joint utility (shared by all members) over states and actions, $R : S\times\Sigma_\alpha\times A_\alpha\to\mathbb{R}$, and is the sum of two rewards: a domain-action-level reward, $R_A{:}S{\times}A_\alpha{\to}\mathbb{R}$, and a communication-level reward, $R_\Sigma : S\times\Sigma_\alpha\to\mathbb{R}$. COM-MTDP (and likewise R-COM-MTDP) domains can be classified based on the allowed communication and its reward:

General Communication: no assumptions on Σ_α nor R_Σ.

No Communication: $\Sigma_\alpha = \emptyset$.

Free Communication: $\forall\sigma\in\Sigma_\alpha$, $R_\Sigma(\sigma) = 0$.

Analyzing the extreme cases, like free communication (and others in this paper) helps to understand the computational impact of the extremes. In addition, we can approximate some real-world domains with such assumptions.

R-COM-MTDP Extensions to COM-MTDP Model

We define a R-COM-MTDP as an extended tuple, $\langle S, A_\alpha, \Sigma_\alpha, P, \Omega_\alpha, O_\alpha, B_\alpha, R, \mathcal{PL}\rangle$. The key extension over the COM-MTDP is the addition of subplans, $\mathcal{PL}$, and the individual roles associated with those plans.

Extension for Explicit Sub-Plans $\mathcal{PL}$ is a set of all possible sub-plans that α can perform. We express a sub-plan $p_k \in \mathcal{PL}$ as a tuple of roles $\langle r_1, \ldots, r_s\rangle$. r_{jk} represents a *role instance* of role r_j for a plan p_k and requires some agent $i \in \alpha$ to fulfill it. Roles enable better modeling of real systems, where each agent's role restricts its domain-level actions (Wooldridge, Jennings, & Kinny 1999). Agents' domain-level actions are now distinguished between two types:

Role-Taking actions: $\Upsilon_\alpha = \prod_{i\in\alpha} \Upsilon_i$ is a set of combined role taking actions, where $\Upsilon_i = \{\upsilon_{ir_{jk}}\}$ contains the role-taking actions for agent i. $\upsilon_{ir_{jk}} \in \Upsilon_i$ means that agent i takes on the role r_j as part of plan p_k. An agent's role can be uniquely determined from its belief state.

Role-Execution Actions: $\Phi_{ir_{jk}}$ is the set of agent i's actions for executing role r_j for plan p_k (Wooldridge, Jennings, & Kinny 1999). $\Phi_i = \bigcup_{\forall r_{jk}} \Phi_{ir_{jk}}$. This defines the set of combined execution actions $\Phi_\alpha = \prod_{i\in\alpha} \Phi_i$.

The distinction between role-taking and role-execution actions ($A_\alpha = \Upsilon_\alpha \cup \Phi_\alpha$) enables us to separate their costs. We can then compare costs of different role-taking policies analytically and empirically as shown in future sections. Within this model, we can represent the specialized behaviors associated with each role, and also any possible differences among the agents' capabilities for these roles. While filling a particular role, r_{jk}, agent i can perform only those role-execution actions, $\phi \in \Phi_{ir_{jk}}$, which may not contain all of its available actions in Φ_i. Another agent ℓ may have a different set of available actions, $\Phi_{\ell r_{jk}}$, allowing us to model the different methods by which agents i and ℓ may fill role r_{jk}. These different methods can produce varied effects on the world state (as modeled by the transition probabilities, P) and the team's utility (as modeled by the reward function, R_Φ). Thus, the policies must ensure that agents for each role have the capabilities that benefit the team the most.

In R-COM-MTDPs (as in COM-MTDPs), each decision epoch consists of two stages, a communication stage and an action stage. In each successive epoch, the agents alternate between role-taking and role-execution epochs. Thus, the agents are in the role-taking epoch if the time index is divisible by 2, and are in the role execution epoch otherwise. Although, this sequencing of role-taking and role-execution epochs restricts different agents from running role-taking and role-execution actions in the same epoch, it is conceptually simple and synchronization is automatically enforced. As with COM-MTDP, the total reward is a sum of communication and action rewards, but the action reward is further separated into role-taking action vs. role-execution action: $R_A(s, \mathbf{a}) = R_\Upsilon(s, \mathbf{a}) + R_\Phi(s, \mathbf{a})$. By definition, $R_\Upsilon(s, \phi) = 0$ for all $\phi \in \Phi_\alpha$, and $R_\Phi(s, \upsilon) = 0$ for all $\upsilon \in \Upsilon_\alpha$. We view the role taking reward as the cost (negative reward) for taking up different roles in different teams. Such costs may represent preparation or training or traveling time for new members, e.g., if a sensor agent changes its role to join a new sub-team tracking a new target, there is a few seconds delay in tracking. However, change of roles may potentially provide significant future rewards.

We can define a role-taking policy, $\pi_{i\Upsilon} : B_i \rightarrow \Upsilon_i$ for each agent's role-taking action, a role-execution policy, $\pi_{i\Phi} : B_i \rightarrow \Phi_i$ for each agent's role-execution action, and a communication policy $\pi_{i\Sigma} : B_i \rightarrow \Sigma_i$ for each agent's communication action. The goal is to come up with joint policies π_Υ, π_Φ and π_Σ that will maximize the total reward.

Extension for Explicit Local States: S_i In considering distinct roles within a team, it is useful to consider distinct subspaces of S relevant for each individual agent. If we consider the world state to be made up of orthogonal features (i.e., $S = \Xi_1 \times \Xi_2 \times \cdots \times \Xi_n$), then we can identify the subset of features that agent i may observe. We denote this subset as its *local state*, $S_i = \Xi_{k_{i1}} \times \Xi_{k_{i2}} \times \cdots \times \Xi_{k_{im_i}}$. By definition, the observation that agent i receives is independent of any features not covered by S_i: $\Pr(\Omega_i^t = \omega | S^t = \langle \xi_1, \xi_2, \ldots, \xi_n\rangle, A_\alpha^{t-1} = \mathbf{a}) = \Pr(\Omega_i^t = \omega | S_i^t = \langle \xi_{k_{i1}}, \ldots, \xi_{k_{im_i}}\rangle, A_\alpha^{t-1} = \mathbf{a})$.

Application in RoboCup Rescue

The notation described above can be applied easily to the RoboCup Rescue domain as follows:

1. α consists of three types of agents: ambulances, police forces, fire brigades.
2. Injured civilians, buildings on fire and blocked roads can be grouped together to form tasks. We specify sub-plans for each such task type. These plans consist of roles that can be fulfilled by agents whose capabilities match those of the role.
3. A sub-plan, $p \in \mathcal{PL}$ comprises of a variable number of roles that need to be fulfilled in order to accomplish a task. For example, the task of rescuing a civilian from a burning building can be accomplished by a plan where fire-brigades first extinguish the fire, then ambulances free the buried civilian and one ambulance takes the civilian to a hospital. Each task can have multiple plans which represent multiple ways of achieving the task.
4. Each agent receives observations about the objects within its visible range. But there may be parts of the world that are not observable because there are no agents there. Thus, RoboCup Rescue is a collectively partially observable domain. Therefore each agent, needs to maintain a belief state of what it believes the true world state is.
5. The benefit function can be chosen to consider the capabilities of the agents to perform particular roles, e.g., police agents may be more adept at performing the "search" role than ambulances and fire-brigades. This would be reflected in a higher value for choosing a police agent to take on the "search" role than an ambulance or a fire-brigade. In addition, the reward function takes into consideration the number of civilians rescued, the number of fires put out and the health of agents.

The R-COM-MTDP model works as follows: Initially, the global world state is S^0, where each agent $i \in \alpha$ has local state S_i^0 and belief state $b_i^0 = SE_i^0()$ and no role. Each agent i receives an observation, ω_i^0, according to probability distribution $O_i(S_i^0, \mathit{null}, \omega_i^0)$ (there are no actions yet) and updates its belief state, $b_{i\bullet\Sigma}^0 = SE_{i\bullet\Sigma}(b_i^0, \omega_i^0)$ to incorporate this new evidence. Each agent then decides on what to broadcast based on its communication policy, $\pi_{i\Sigma}$, and updates its belief state according to $b_{i\Sigma\bullet}^0 = SE_{i\Sigma\bullet}(b_{i\bullet\Sigma}^0, \Sigma_\alpha^0)$. Each agent, based on its belief state then executes the role-taking action according to its role-taking policy, $\pi_{i\Upsilon}$. Thus, some police agents may decide on performing the "search role", while others may decide to "clear roads", fire-brigades decide on which fires "to put out". By the central assumption of teamwork, all of the agents receive the same joint reward, $R^0 = R(S^0, \Sigma_\alpha^0, A_\alpha^0)$. The world then moves into a new state, S^1, according to the distribution, $P(S^0, A_\alpha^0)$. Each agent then receives the next observation about its new local state based on its position and its visual range and updates its belief state using $b_{i\bullet\Sigma}^1 = SE_{i\bullet\Sigma}(b_{i\Sigma\bullet}^0, \omega_i^1)$. This is followed by another communication action resulting in the belief state, $b_{i\Sigma\bullet}^1 = SE_{i\Sigma\bullet}(b_{i\bullet\Sigma}^1, \Sigma_\alpha^1)$. The agent then decides on a role-execution action based on its policy $\pi_{i\Phi}$. It then receives new observations about its local state and the cycle of observation, communication, role-taking action, observation, communication and role-execution action continues.

Complexity of R-COM-MTDPs

R-COM-MTDP supports a range of complexity analysis for generating optimal policies under different communication and observability conditions.

Theorem 1 *We can reduce a COM-MTDP to an equivalent R-COM-MTDP.*

Proof: Given a COM-MTDP, $\langle S, A_\alpha, \Sigma_\alpha, P, \Omega_\alpha, O_\alpha, B_\alpha, R\rangle$, we can generate an equivalent R-COM-MTDP, $\langle S, A'_\alpha, \Sigma_\alpha, P', \Omega_\alpha, O_\alpha, B_\alpha, R'\rangle$. Within the R-COM-MTDP actions, A'_α, we define $\Upsilon_\alpha = \{\mathit{null}\}$ and $\Phi_\alpha = A_\alpha$. In other words, all of the original COM-MTDP actions become role-execution actions in the R-COM-MTDP, where we add a single role-taking action that has no effect (i.e., $P'(s, \mathit{null}, s) = 1$). The new reward function borrows the same role-execution and communication-level components: $R'_\Phi(s, \mathbf{a}) = R_A(s, \mathbf{a})$ and $R'_\Sigma(s, \sigma)$. We also add the new role-taking component: $R'_\Upsilon(s, \mathit{null}) = 0$. Thus, the only role-taking policy possible for this R-COM-MTDP is $\pi'_{i\Upsilon}(b) = \mathit{null}$, and any role-execution and communication policies (π'_Φ and π'_Σ, respectively) will have an identical expected reward as the identical domain-level and communication policies (π_A and π_Σ, respectively) in the original COM-MTDP. □

Theorem 2 *We can reduce a R-COM-MTDP to an equivalent COM-MTDP.*[1]

Proof: Given a R-COM-MTDP, $\langle S, A_\alpha, \Sigma_\alpha, P, \Omega_\alpha, O_\alpha, B_\alpha, R, \mathcal{PL}\rangle$, we can generate an equivalent COM-MTDP, $\langle S', A_\alpha, \Sigma_\alpha, P', \Omega_\alpha, O_\alpha, B_\alpha, R'\rangle$. The COM-MTDP state space, S', includes all of the features, Ξ_i, in the original R-COM-MTDP state space, $S = \Xi_1 \times \cdots \times \Xi_n$, as well as an additional feature, $\Xi_{\text{phase}} = \{\text{taking, executing}\}$. This new feature indicates whether the current state corresponds to a role-taking or -executing stage of the R-COM-MTDP. The new transition probability function, P', augments the original function with an alternating behavior for this new feature: $P'(\langle \xi_{1b}, \ldots, \xi_{nb}, \text{taking}\rangle, \upsilon, \langle \xi_{1e}, \ldots, \xi_{ne}, \text{executing}\rangle) = P(\langle \xi_{1b}, \ldots, \xi_{nb}\rangle, \upsilon, \langle \xi_{1e}, \ldots, \xi_{ne}\rangle)$ and $P'(\langle \xi_{1b}, \ldots, \xi_{nb}, \text{executing}\rangle, \phi, \langle \xi_{1e}, \ldots, \xi_{ne}, \text{taking}\rangle) = P(\langle \xi_{1b}, \ldots, \xi_{nb}\rangle, \phi, \langle \xi_{1e}, \ldots, \xi_{ne}\rangle)$. Within the COM-MTDP, we restrict the actions that agents can take in each stage by assigning illegal actions an excessively negative reward (denoted $-r_{max}$): $\forall \upsilon \in \Upsilon_\alpha$, $R'_A(\langle \xi_{1b}, \ldots, \xi_{nb}, \text{executing}\rangle, \upsilon) = -r_{max}$ and $\forall \phi \in \Phi_\alpha$, $R'_A(\langle \xi_{1b}, \ldots, \xi_{nb}, \text{taking}\rangle, \phi) = -r_{max}$. Thus, for a COM-MTDP domain-level policy, π'_A, we can extract role-taking and -executing policies, π_Υ and π_Φ, respectively, that generate identical behavior in the R-COM-MTDP when used in conjunction with identical communication-level policies, $\pi_\Sigma = \pi'_\Sigma$. □

Thus, the problem of finding optimal policies for R-COM-MTDPs has the same complexity as the problem of finding optimal policies for COM-MTDPs. Table 1 shows the computational complexity results for various classes of R-COM-MTDP domains, where the results for individual, collective, and collective partial observability follow from COM-MTDPs (Pynadath & Tambe 2002) (See *http://www.isi.edu/teamcore/COM-MTDP/* for proofs of the COM-MTDP results) . In the individual observability and collective observability under free communication cases, each agent knows exactly what the global state is. The P-Complete result is from a reduction from and to MDPs. The collectively partial observable case with free communication can be treated as a single agent POMDP, where the actions correspond to the joint actions of the R-COM-MTDP. The reduction from and to a single agent POMDP gives the PSPACE-Complete result. In the general case, by a reduction from and to decentralized POMDPs, the worst-case computational complexity of finding the optimal policy is NEXP-Complete.

The table 1 shows us that the task of finding the optimal policy is extremely hard, in general. However, by increasing the amount of communication we can drastically improve the worst-case computational complexity (by paying the price of the additional communication). As can be seen from the table, when communication changes from no communication to free communication, in collectively partially observable domains like RoboCup Rescue, the computational complexity changes from NEXP-Complete to PSPACE-Complete. In collectively observable domains, like the sensor domain, the computational savings are even greater, from NEXP-Complete to P-Complete. This emphasizes the importance of communication in reducing the worst case complexity. Table 1 suggests that if we were designers of a multiagent system, we would increase the observability of the system so as to reduce the computational

[1] The proof of this theorem was contributed by Dr. David Pynadath

	Ind. Obs.	Coll. Obs.	Coll. Part. Obs.
No Comm.	P-Comp.	NEXP-Comp.	NEXP-Comp.
Gen. Comm.	P-Comp.	NEXP-Comp.	NEXP-Comp.
Free Comm.	P-Comp.	P-Comp.	PSPACE-Comp.

Table 1: Computational complexity of R-COM-MTDPs.

complexity.

Summary and Related Work

This work addresses two shortcomings of the current work in team formation for dynamic real-time domains: i) most algorithms are static in the sense that they don't anticipate for changes that will be required in the configuration of the teams, ii) complexity analysis of the problem is lacking. We addressed the first shortcoming by presenting *R-COM-MTDP*, a formal model based on decentralized communicating POMDPs, to determine the team configuration that takes into account how the team will have to be restructured in the future. *R-COM-MTDP* enables a rigorous analysis of complexity-optimality tradeoffs in team formation and reorganization approaches. The second shortcoming was addressed by presenting an analysis of the worst-case computational complexity of team formation and reformation under various types of communication. We intend to use the *R-COM-MTDP* model to compare the various team formation algorithms for RoboCup Rescue which were used by our agents in RoboCup-2001 and Robofesta 2001, where our agents finished in third place and second place respectively.

While there are related multiagent models based on MDPs, they have focused on coordination after team formation on a subset of domain types we consider, and they do *not* address team formation and reformation. For instance, the *decentralized partially observable Markov decision process* (DEC-POMDP) (Bernstein, Zilberstein, & Immerman 2000) model focuses on generating decentralized policies in *collectively partially observable* domains with *no communication*; while the Xuan-Lesser model (Xuan, Lesser, & Zilberstein 2001) focuses only on a subset of collectively observable environments. Finally, while (Modi *et al.* 2001) provide an initial complexity analysis of distributed sensor team formation, their analysis is limited to static environments (no reorganizations) — in fact, illustrating the need for R-COM-MTDP type analysis tools.

Acknowledgment

We would like to thank David Pynadath for his discussions on extending the COM-MTDP model to R-COM-MTDP, and the Intel Corporation for their generous gift that made this research possible.

References

Bernstein, D. S.; Zilberstein, S.; and Immerman, N. 2000. The complexity of decentralized control of MDPs. In *Proceeings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*.

Fatima, S. S., and Wooldridge, M. 2001. Adaptive task and resource allocation in multi-agent systems. In *Proceedings of the Fifth International Conference on Autonomous Agents*.

Horling, B.; Benyo, B.; and Lesser, V. 2001. Using self-diagnosis to adapt organizational structures. In *Proceedings of the Fifth International Conference on Autonomous Agents*.

Hunsberger, L., and Grosz, B. 2000. A combinatorial auction for collaborative planning. In *Proceedings of the Fourth International Conference on MultiAgent Systems*.

Kitano, H.; Tadokoro, S.; Noda, I.; Matsubara, H.; Takahashi, T.; Shinjoh, A.; and Shimada, S. 1999. Robocup-rescue: Search and rescue for large scale disasters as a domain for multi-agent research. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*.

Modi, P. J.; Jung, H.; Tambe, M.; Shen, W.-M.; and Kulkarni, S. 2001. A dynamic distributed constraint satisfaction approach to resource allocation. In *Proceedings of Seventh International Conference on Principles and Practice of Constraint Programming*.

Pynadath, D., and Tambe, M. 2002. Multiagent teamwork: Analyzing the optimality complexity of key theories and models. In *Proceedings of First International Joint Conference on Autonomous Agents and Multi-Agent Systems*.

Tadokoro, S.; Kitano, H.; Tomoichi, T.; Noda, I.; Matsubara, H.; Shinjoh, A.; Koto, T.; Takeuchi, I.; Takahashi, H.; Matsuno, F.; Hatayama, M.; Nobe, J.; and Shimada, S. 2000. The robocup-rescue: An international cooperative research project of robotics and ai for the disaster mitigation problem. In *Proceedings of SPIE 14th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls (AeroSense), Conference on Unmanned Ground Vehicle Technology II*.

Tidhar, G.; Rao, A. S.; and Sonenberg, E. 1996. Guided team selection. In *Proceedings of the Second International Conference on Multi-Agent Systems*.

Wooldridge, M.; Jennings, N.; and Kinny, D. 1999. A methodology for agent oriented analysis and design. In *Proceedings of the Third International Conference on Autonomous Agents*.

Xuan, P.; Lesser, V.; and Zilberstein, S. 2001. Communication decisions in multiagent cooperation. In *Proceedings of the Fifth International Conference on Autonomous Agents*
.