R-COM-MTDP: Comparing and Forming Plans for Team Formation in Multiagent Systems

Ranjit Nair, Milind Tambe, Stacy Marsella, David V. Pynadath
Computer Science Department and Information Sciences Institute
University of Southern California
Los Angeles CA 90089
{nair,tambe}@usc.edu, {marsella,pynadath}@isi.edu

Abstract

Team formation, i.e., allocating agents to roles within a team or subteams of a team, and the reorganization of a team upon team member failure or arrival of new tasks are critical aspects of teamwork. Despite significant progress, research in multiagent team formation and reorganization has failed to provide a rigorous analysis of the computational complexities of the approaches proposed or their degree of optimality. This shortcoming has hindered quantitative comparisons of approaches or their complexity-optimality tradeoffs, e.g., is the team reorganization approach in practical teamwork models such as STEAM optimal in most cases or only as an exception? To alleviate these difficulties, this paper presents *R-COM-MTDP*, a formal model based on decentralized communicating POMDPs, where agents explicitly take on and change roles to (re)form teams. R-COM-MTDP significantly extends an earlier COM-MTDP model, by analyzing how agents' roles, local states and reward decompositions gradually reduce the complexity of its policy generation from NEXP-complete to PSPACE-complete to P-complete. We also encode key role reorganization approaches (e.g., STEAM) as R-COM-MTDP policies, and compare them with a locally optimal policy derivable in R-COM-MTDP, thus, theoretically and empirically illustrating the complexity-optimality tradeoffs.

Introduction

Team formation, i.e., allocating agents to roles within a (sub)team (Tidhar *et al* 1996) is a critical pre-requisite for multiagent teamwork, i.e. before agents can act together as team, we need to decide on how best to form these team. Teams formed must often reform (reorganize role allocation) upon team member failure or arrival of new tasks(Grosz & Kraus 1996; Tambe 1997). Unfortunately, there are three key shortcomings in the current work on team formation and reformation. First, each team formation/reformation episode is considered independent of future reformations. Yet, in many dynamic domains, teams continually reorganize. Thus, planning ahead in team formation can minimize costly reorganization given future tasks or failures. We call this *Team Formation for Reformation*, i.e. forming teams taking into account future eventualities like change in tasks and failures that will necessitate a reformation. For instance, distributed sensor agents may form subteams and continually reorganize to track mobile targets(Modi *et al.* 2001); planning ahead may minimize reorganization and tracking disruption. Such planning may also benefit dynamic disaster rescue teams(Kitano *et al* 1999).

A second key shortcoming is the lack of analysis of complexity-optimality tradeoffs in team (re)formation algorithms. Certainly, the computational complexity of the overall process of optimal team formation and reformation, as suggested above, is unknown. Furthermore, both the optimality and complexity of current team reformation approaches remains uninvestigated, e.g., is the team reformation algorithm in STEAM(Tambe 1997), a key practical teamwork model, always optimal or only as an exception? Even if suboptimal, understanding the computational advantages of these algorithms could potentially justify their use. Finally, while *roles* are central to teams(Tidhar *et al* 1996), we still lack analysis of the impact of different role types on teamwork computational complexity.

This paper presents *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 Partially Observable Markov Decision Processes (POMDP), 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 -executing policies in multiagent teams. We show that while generation of optimal policies in R-COM-MTDPs is NEXP-complete, different observability and communication conditions, and role definitions, significantly reduce such complexity. We also encode key role reorganization approaches (e.g., STEAM) as R-COM-MTDP policies; we compare them with a locally optimal policy derivable in R-COM-MTDP, thus, analytically and empirically quantifying their optimality losses for efficiency gains. Thus, policies derived directly using the R-COM-MTDP model are calculated by planning for all possible reformations that may be

required in the future. However, it is possible to analyze reactive approaches like STEAM that don't perform "look-ahead" within this model.

Previous Work

While there are related multiagent models based on MDPs (Bernstein *et al* 2000; Boutilier 1996; Peshkin *et al.* 2000; Pynadath & Tambe 2002; Xuan *et al* 2001) 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. The most closely related is COM-MTDP(Pynadath & Tambe 2002), which, we discuss first before discussing the others.

Given a team of selfless agents, α, a COM-MTDP (Pynadath & Tambe 2002) is a tuple, $\langle S, A_\alpha, \Sigma_\alpha, P, \Omega_\alpha, O_\alpha, B_\alpha, R_\alpha \rangle$. 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. S^t is a random variable for the world state at time t which takes values from S. The same notation is used for A^t_α, Ω^t_α, etc. $P(s_b, \mathbf{a}, s_e) = Pr(S^{t+1}{=}s_e | S^t{=}s_b, A^t_\alpha{=}\mathbf{a})$ 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 function, $O_\alpha(s, \mathbf{a}, \boldsymbol{\omega}) = Pr(\Omega^t_\alpha{=}\omega | S^t{=}s, A^{t-1}_\alpha{=}\mathbf{a})$, specifies a probability distribution over the agents' observations jointly 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 determine 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 *et al* 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 *s*tate-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 state estimator functions could be as simple appending new observation or communication to previous history.

The COM-MTDP reward function represents the team's joint utility over states and actions, $R_\alpha : S \times \Sigma_\alpha \times A_\alpha \rightarrow \mathbb{R}$, and is the sum of two rewards: a domain-action-level reward, $R_{\alpha A} : S \times A_\alpha \rightarrow \mathbb{R}$, and a communication-level reward, $R_{\alpha\Sigma} : S \times \Sigma_\alpha \rightarrow \mathbb{R}$. COM-MTDP (and likewise R-COM-MTDP) domains can be classified based on the allowed communication and its reward as i) **General Communication:** no assumptions on Σ_α nor $R_{\alpha\Sigma}$, ii) **No Communication:** $\Sigma_\alpha = \emptyset$, and iii) **Free Communication:** $\forall \sigma \in \Sigma_\alpha$, $R_{\alpha\Sigma}(\sigma) = 0$. $R_{\alpha\Sigma}$ is just the immediate cost of the communication act and does not include the future reward resulting from this communication.

The COM-MTDP model subsumes the other previous multiagent coordination models. For instance, the *decentralized partially observable Multiagent decision process* (DEC-POMDP) (Bernstein *et al* 2000) model focuses on generating decentralized policies in *collectively partially observable* domains with *no communication*; *Multiagent Markov Decision Processes*(MMDPs) (Boutilier 1996) focuses on *individually observable* domains with *no communication*; while the Xuan-Lesser model (Xuan *et al* 2001) focuses only on a subset of collectively observable environments. The POIPSG model (Peshkin *et al.* 2000) is similar to DEC-POMDP) (Bernstein *et al* 2000) in that it considers *collectively partially observable* domains with *no communication*.

The R-COM-MTDP Model

Roles reduce the complexity of action selection and also enable better modeling of real systems, where each agent's role restricts its domain-level actions. Hence, we build on the existing multiagent coordination models, especially COM-MTDP, to include roles. Another key multiagent concept that is missing in current models is "local state". We, thus, define a R-COM-MTDP as an extended tuple, $\langle S, A_\alpha, \Sigma_\alpha, P, \Omega_\alpha, O_\alpha, B_\alpha, R_\alpha, \mathcal{RL} \rangle$.

Extension for Explicit Roles

$\mathcal{RL} = \{r_1, \ldots, r_s\}$ is a set of all roles that α can undertake. Each instance of role r_j requires some agent $i \in \alpha$ to fulfill it. 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 $\Upsilon_i = \{\upsilon_{ir_j}\}$ contains the role-taking actions for agent i. $\upsilon_{ir_j} \in \Upsilon_i$ means that agent i takes on the role $r_j \in \mathcal{RL}$. An agent's role can be uniquely determined from its belief state and policy.

Role-Execution Actions: $\Phi_\alpha = \prod_{i \in \alpha} \Phi_i$ is a set of combined execution actions, where $\Phi_i = \bigcup_{\forall r_j \in \mathcal{RL}} \Phi_{ir_j}$. Φ_{ir_j} is the set of agent i's actions for executing role $r_j \in \mathcal{RL}$, thus restricting the actions that an agent can perform in a role.

The distinction between role-taking and -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_j, agent i can perform only those role-execution actions, $\phi \in \Phi_{ir_j}$, which may not contain all of its available actions in Φ_i. Another agent ℓ may have a different set of available actions, $\Phi_{\ell r_j}$, allowing us to model the different methods by which agents i and ℓ may fill role r_j. These different methods can produce varied effects on the world state (as modeled by the transition probabilities, P) and the team's utility. Thus, the policies must ensure that agents for each role have the capabilities that benefit the team the most.

As in the models seen in the previous section (Xuan *et al* 2001; Pynadath & Tambe 2002), each decision epoch consists of two stages, a communication stage and an action stage. However, the action stage in each successive epoch, alternates between role-taking and role-execution actions. 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. This allows us to easily reason about role taking actions and role execution actions in isolation as will be shown in section 6. 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: $R_{\alpha A}(s, \mathbf{a}) = R_{\alpha\Upsilon}(s, \mathbf{a}) + R_{\alpha\Phi}(s, \mathbf{a})$. By definition, $R_{\alpha\Upsilon}(s, \phi) = 0$ for all $\phi \in \Phi_\alpha$, and $R_{\alpha\Phi}(s, \upsilon) = 0$ 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 $\pi_{\alpha\Upsilon}$, $\pi_{\alpha\Phi}$ and $\pi_{\alpha\Sigma}$ that will maximize the total reward over a finite horizon T.

Extension for Explicit Local States: S_i

In considering distinct roles within a team, we consider that only a distinct part of the state space S is 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 of the world state that affect the observation of agent i. We denote this subset of features as its *local state*, $S_i = \Xi_{i1} \times \Xi_{i2} \times \cdots \times \Xi_{im_i}$. The local state of an agent is dynamic and could change with change in the agent's role, world state, etc. By definition, the observation that agent i at time t receives is independent of any features not covered by S_i^t: $\Pr(\Omega_i^t = \omega | S^t = \langle \xi_1, \xi_2, \ldots, \xi_n \rangle, A_\alpha^{t-1} = \mathbf{a}, \Omega_{\alpha-i}^t = \boldsymbol{\omega}_{\alpha-i}) = \Pr(\Omega_i^t = \omega | S_i^t = \langle \xi_{i1}, \ldots, \xi_{im_i} \rangle, A_\alpha^{t-1} = \mathbf{a}, \Omega_{\alpha-i}^t = \boldsymbol{\omega}_{\alpha-i})$, where $\Omega_{\alpha-i}^t = \prod_{j \in \alpha \setminus \{i\}} \Omega_j^t$.

Thus, observations made by an agent depend on only i) its local state, ii) the last combined domain-level action, and iii) the observations of some (possibly none) of the other agents' observations . This definition of local state allows us to identify a new class of observation functions to closely model (possibly as an approximation) key types of complex multiagent systems such as RoboCupRescue(Kitano *et al* 1999), where each agent knows its local state perfectly at all times.

Local Observability: Each individual's observation uniquely determines its local state: $\forall \omega \in \Omega_i$, $\exists s \in S_i$ such that $\forall s' \neq s$, $\Pr(\Omega_i^t = \omega | S_i^t = s') = 0$.

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 agents receives an observation, ω_i^0 which is part of a joint observation $\boldsymbol{\omega}^0$ drawn from probability distribution $O_\alpha(S^0, null, \boldsymbol{\omega}^0)$ (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 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, $\pi_{i\Upsilon}$. By the central assumption of teamwork, all of the agents receive the same joint reward, $R_\alpha^0 = R_\alpha(S^0, \Sigma_\alpha^0, A_\alpha^0)$. 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 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 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 enables a critically needed systematic investigation of the complexity for generating optimal policies under different communication and observability conditions. (Given space limits, detailed proofs for all our theorems are at: *http://www.isi.edu/teamcore/uai*).

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

Proof: Reduction from COM-MTDP to R-COM-MTDP is easy. Given that R-COM-MTDP is a generalization

of COM-MTDP, we set its role taking action to null. The difficult direction is the reduction from R-COM-MTDP to COM-MTDP, where the key idea is that we generate a new COM-MTDP where its state space has all the features of the original R-COM-MTDP state space $S = \Xi_1 \times \cdots \times \Xi_n$, and an additional feature $\Xi_{\text{phase}} = \{\text{taking}, \text{executing}\}$. 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 P with an alternating behavior for this new feature: $P'(\langle \xi_{1b}, \ldots, \xi_{nb}, \text{taking} \rangle, \upsilon, \langle \xi_{1e}, \ldots, \xi_{ne}, \text{executing} \rangle) = P(\langle \xi_{1b}, \ldots, \xi_{nb} \rangle, \upsilon, \langle \xi_{1e}, \ldots, \xi_{ne} \rangle)$ where υ is a role-taking action in the R-COM-MTDP(similarly from executing to taking).□

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). New results in Table 1 come from analyzing the key addition in R-COM-MTDP, that of local states and local observability.

Theorem 2 *Given a collectively observable R-COM-MTDP, $\langle S, A_\alpha, \Sigma_\alpha, P, \Omega_\alpha, O_\alpha, B_\alpha, R_\alpha, \mathcal{RL} \rangle$, there is an equivalent locally observable R-COM-MTDP, $\langle S', A_\alpha, \Sigma_\alpha, P', \Omega_\alpha, O'_\alpha, B_\alpha, R'_\alpha, \mathcal{RL} \rangle$, such that any policies, $\pi_{\alpha\Upsilon}$, $\pi_{\alpha\Phi}$, and $\pi_{\alpha\Sigma}$, will achieve identical expected rewards under both R-COM-MTDPs.*

Proof: We can define a R-COM-MTDP such that each agent's local state is exactly the observation it receives according to the original collectively observable R-COM-MTDP. The world state in the resulting R-COM-MTDP is exactly the combined observations of the agents. The resulting R-COM-MTDP is locally observable and equivalent to the original collectively observable R-COM-MTDP□

Thus, while collective observability is a team's global property, we can still generate from it a locally observable R-COM-MTDP. A locally observable R-COM-MTDP is not collectively observable however. By definition, a locally observable R-COM-MTDP is collectively partially observable (the most general observability class). Since under no communication, the complexity of both collectively observable R-COM-MTDP and collectively partially observable R-COM-MTDP is NEXP-complete, Theorem 2 implies that the complexity of locally observable R-COM-MTDP under no communication is also NEXP-complete. This explains the NEXP-complete entries for local observability in Table 1. Finally, we can also show the following result:

Theorem 3 *The decision problem of determining if there exist policies, $\pi_{\alpha\Sigma}$ and $\pi_{\alpha A}$, for a given R-COM-MTDP with* free communication *and* local observability*, that yield a total reward at least K over finite horizon T is PSPACE-complete.*

Table 1: Computational Complexity

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

Role Decomposition

While roles are seen to be central in designing multi-agent systems, some designers exploit roles further by decomposition of the multiagent coordination problem into smaller subproblems, isolating the specific factors relevant to each of the separate roles(Marsella *et al* 2001; Yoshikawa 1978). The qualitative intuition behind this *role decomposition* is that this separation simplifies the overall problem facing the agent team, even if it only approximates optimal behavior.

This section presents a *quantitative* evaluation of the computational savings gained by role decomposition, as well as of the sub-optimality introduced by its approximation. This fits in with our idea of analyzing the extremes to understand boundary conditions. For role decomposition, the following three constraints must hold. First, the dynamics of the local state must depend on only the current local state and the agent's domain-level action: $\Pr(S_i^{t+1} | S^t = \langle \xi_1, \ldots, \xi_n \rangle, A_\alpha^t = \Pi_{j \in \alpha} a_j) = \Pr(S_i^{t+1} | S_i^t = \langle \xi_{i1}, \ldots, \xi_{im_i} \rangle, A_i^t = a_i)$. Second, agent's observations are independent and governed by the following observation functions, $O_i(s, a, \omega) = Pr(\Omega_i^t = \omega | S_i^{t-1} = s, A_i^{t-1} = a)$ which implies that the observations of agent i at time t are unaffected by the observations and actions of other agents. Finally, we also structure the reward function so that the agents' actions earn independent rewards: $R_\alpha(s, \prod_{i \in \alpha} a_i) = \sum_{i \in \alpha} R_i(s_i, a_i)$, where R_i is the local reward function for agent i and s_i is its local state. We now examine the computational savings given role decomposition.

Theorem 4 *The decision problem of determining if there exist policies, $\pi_{\alpha\Upsilon}$, $\pi_{\alpha\Phi}$ and $\pi_{\alpha\Sigma}$, for a R-COM-MTDP with role decomposition, that yield a total reward at least K over some finite horizon T is PSPACE-complete.*

Proof: Given role decomposition, the value of the optimal policy for the whole team is identical to the sum of the values of the locally optimal policies for the separate agents. Thus, the problem is equivalent to solving $|\alpha|$ single-agent POMDPs, a problem known to be PSPACE-Complete.□

Theorem 5 *The decision problem of determining whether there exist policies, $\pi_{\alpha\Upsilon}$, $\pi_{\alpha\Phi}$ and $\pi_{\alpha\Sigma}$, for a R-COM-MTDP with role decomposition in a* locally observable *domain, that yield a total reward at least K over some finite horizon T is P-complete.*

Proof: Under local observability, we can reduce the overall R-COM-MTDP to $|\alpha|$ separate single-agent *MDPs*.□

Table 2 demonstrates that role decomposition can significantly lower computational complexity and together with Table 1, it allows us to compare the relative

Table 2: Computational Complexity after Role Decomposition

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

value of communication and role decomposition in simplifying the decision problem facing an agent team. Examining the bottom two rows of Table 1, we see that, under collective observability, having the agents communicate all of their observations all of the time reduces the problem from NEXP to P. Examining the difference between Tables 1 and 2, we see that role decomposition, in contrast, reduces the problem to only PSPACE under collective observability (top row, Table 2). However, under *local* observability, full communication reduces the problem from NEXP to PSPACE, while role decomposition produces a decision problem that is only P.

Most real-world domains are not exactly decomposable, so role decomposition often gains its computational savings at the cost of an approximate model of the domain. Similarly, full communication may sacrifice optimality when communication incurs nonzero costs. We can quantify the relative optimality of these two methods.

Theorem 6 *Given an expected reward loss from role decomposition of K, and a maximum communication cost C per epoch such that $K > T \cdot C$, where T is the finite horizon, the optimal policy using full communication strictly dominates the optimal policy with role decomposition.*

New Role Replacement Algorithms

R-COM-MTDP analysis of the complexity of forming the optimal policy for an entire team, including team formation and reformation, illustrates its difficulty. Hence, implemented systems often restrict the problem to "Role Replacement". For example, STEAM(Tambe 1997) assumes an initial team formation performed by a human, and focuses on reformation via role replacement, where a failed agent must be replaced by another. Similarly, the SharedPlans theory focuses on unreconciled actions(Grosz & Kraus 1996), where an agent or a subteam considers substituting for an unfilled (or failed) role. However, the complexity of the decision problem faced in such role substitution, or the optimality of the approaches proposed, has previously not been analyzed.

R-COM-MTDP can analyze the complexity and optimality of current approaches to the "Role Replacement" problem. As an illustration, we focus on the STEAM policy for replacing a failed agent within a R-COM-MTDP. Consider a simple scenario where a task is being executed by a team of agents jointly, and that a single agent of this team fails in its role and needs to be replaced. An agent A_R observes (or is informed of) this failure and needs to decide whether it should replace the failed agent or not. In STEAM, the policy A_R will follow is this: an agent in role R will replace a failed agent in role F only if the following inequality holds:

$$Criticality\,(F) - Criticality\,(R) > 0 \quad (1)$$
$$Criticality\,(x) = 1 \;\; if\; x\; is\; critical; = 0 \;\; otherwise$$

In other words, replacement occurs if role F is considered critical and role R is not critical. A role is critical if its non-achievement results in the task remaining unfulfilled. Criticality is determined in $O(|\alpha|)$ by processing role-dependency conditions supplied by a human. While this STEAM approach is tractable, it is difficult to evaluate its optimality or quantitatively compare it with other approaches. To address this problem, R-COM-MTDP can be used to come up with an optimal joint role-taking policy. This policy, which we refer to as, the "globally optimal" policy, allows any number of agents to perform *any* role taking action at any point in time (even before the actual failure). The time complexity for finding the globally optimal joint policy by searching this space is thus: $O\left(\left(|\Upsilon_\alpha|^{\frac{|\Omega_\alpha|^T-1}{|\Omega_\alpha|-1}}\right)^{|\alpha|} \cdot (|S| \cdot |\Omega_\alpha|)^T\right)$, i.e. doubly exponential in the finite horizon and the number of agents. The intractability of the "globally optimal" policy search motivated us to look for policy that is "locally optimal" in the sense that we consider only one agent's decision about another agent's failure and the decision is further restricted to either replace or not at the moment the agent learns of the failure.

Assume K_F is the earliest time when any agent can replace the failed agent after learning about the failure. We consider the decision at time $K_F = t_0$ to perform role-taking action υ_F, which will replace the failed agent, by an agent $A_R = i$ which knows of the failure and has a pre-communication belief state $b_{i\bullet_\Sigma} = \beta$. In order to quantify the difference between performing the role-replacement action υ_F and doing nothing, we define the following term:

$$\Delta^T(t_0, i, \beta) \equiv E\left[\sum_{t=0}^{T-t_0} R_\alpha^{t_0+t} \,\middle|\, \Upsilon_i^{t_0} = \upsilon_F, K_F = t_0, A_R = i, b_{i\bullet_\Sigma}^{t_0} = \beta\right]$$
$$- E\left[\sum_{t=0}^{T-t_0} R_\alpha^{t_0+t} \,\middle|\, \Upsilon_i^{t_0} = \text{null}, K_F = t_0, A_R = i, b_{i\bullet_\Sigma}^{t_0} = \beta\right]$$

We assume that, for all times other than K_F, the agents follow some role-taking policy, π_Υ. Thus, Δ^T measures the difference in expected reward that hinges on agent i's specific decision to perform or not perform υ_F at time t_0. It is locally optimal for agent i to perform the role replacing action, υ_F, at time t_0, if and only if $\Delta^T \geq 0$.

The R-COM-MTDP model can be used to derive an operational expression of $\Delta^T \geq 0$. For simplicity, we define notational shorthand for various sequences and combinations of values. We define a partial sequence of random variables, $X^{<t}$, to be the sequence of random variables for all times before t: $X^0, X^1, \ldots, X^{t-1}$. We make similar definitions for the other relational operators. The expression, $(S)^T$, denotes the cross product

over states of the world, $\prod_{t=0}^{T} S$, as distinguished from the time-indexed random variable, S^T, which denotes the value of the state at time T. The notation, $s^{\geq t_0}[t]$, specifies the element in slot t within the vector $s^{\geq t_0}$.

In inequality 2, the function Λ is a shorthand to compactly represent a particular subsequence of world and agent belief states occurring, conditioned on the current situation, as follows: $\Pr(\Lambda(\langle t_i, t_f \rangle, s, \boldsymbol{\beta}_{\bullet\Sigma}, t_0, i, \beta)) \equiv \Pr(S^{\geq t_i, \leq t_f} = s, \mathbf{b}_{\bullet\Sigma}{}^{\geq t_i, \leq t_f} = \boldsymbol{\beta}_{\bullet\Sigma} \,|\, K_F = t_0, A_R = i, b^{t_0}_{i\bullet\Sigma} = \beta)$. The function, $\beta_{\Sigma\bullet}$, maps a pre-communication belief state into the post-communication belief state that arises from a communication policy: $\beta_{\Sigma\bullet}(\boldsymbol{\beta}_{\bullet\Sigma}, \boldsymbol{\pi}_\Sigma) \equiv \boldsymbol{SE}_{\Sigma\bullet}(\boldsymbol{\beta}_{\bullet\Sigma}, \boldsymbol{\pi}_\Sigma(\boldsymbol{\beta}_{\bullet\Sigma}))$

Theorem 7 *If we assume that, upon failure F, no action other than υ_F by agent i is possible to replace the failed agent, then condition $\Delta^T(t_0,i,\beta) \geq 0$ holds if and only if:*

$$
\begin{aligned}
&\sum_{s^{\leq t_0} \in (S)^{t_0}} \sum_{\boldsymbol{\beta}^{\leq t_0}_{\bullet\Sigma} \in \boldsymbol{B}^{t_0}} \Pr(\Lambda(\langle 0, t_0 \rangle, s^{\leq t_0}, \boldsymbol{\beta}^{\leq t_0}_{\bullet\Sigma}, t_0, i, \beta) | \Upsilon^{t_0}_i = \upsilon_F) \\
&\cdot \Bigg(\sum_{s^{>t_0} \in (S)^{T-t_0}} \sum_{\boldsymbol{\beta}^{>t_0}_{\bullet\Sigma} \in \boldsymbol{B}^{T-t_0}} \Pr\left(\Lambda(\langle t_0+1, T \rangle, s^{>t_0}, \boldsymbol{\beta}^{>t_0}_{\bullet\Sigma}, t_0, i, \beta) \right. \\
&\qquad \left. \Big| \Upsilon^{t_0}_i = \upsilon_F, \Lambda(\langle 0, t_0 \rangle, s^{\leq t_0}, \boldsymbol{\beta}^{\leq t_0}_{\bullet\Sigma}, t_0, i, \beta) \right) \\
&\qquad \cdot \sum_{t=t_0+1}^{T} R_\alpha\left(s^{>t_0}[t], \boldsymbol{\pi}_\Sigma\left(\boldsymbol{\beta}_{\bullet\Sigma}{}^{>t_0}[t]\right), \boldsymbol{\pi}_A\left(\beta_{\Sigma\bullet}\left(\boldsymbol{\beta}_{\bullet\Sigma}{}^{>t_0}[t], \boldsymbol{\pi}_\Sigma\right)\right)\right) \\
&\qquad - \Pr\left(\Lambda(\langle t_0+1, T \rangle, s^{>t_0}, \boldsymbol{\beta}^{>t_0}_{\bullet\Sigma}, t_0, i, \beta) \right. \\
&\qquad \left. \Big| \Upsilon^{t_0}_i = \mathit{null}, \Lambda(\langle 0, t_0 \rangle, s^{\leq t_0}, \boldsymbol{\beta}^{\leq t_0}_{\bullet\Sigma}, t_0, i, \beta) \right) \\
&\qquad \cdot \sum_{t=t_0+1}^{T} R_\alpha\left(s^{>t_0}[t], \boldsymbol{\pi}_\Sigma\left(\boldsymbol{\beta}_{\bullet\Sigma}{}^{>t_0}[t]\right), \boldsymbol{\pi}_A\left(\beta_{\Sigma\bullet}\left(\boldsymbol{\beta}_{\bullet\Sigma}{}^{>t_0}[t], \boldsymbol{\pi}_\Sigma\right)\right)\right) \Bigg) \\
&\geq - \sum_{s \in F} \sum_{\boldsymbol{\beta} \in \boldsymbol{B}} \Pr\left(\Lambda(\langle t_0, t_0 \rangle, s, \boldsymbol{\beta}, t_0, i, \beta)\right) R_{\alpha\Upsilon}(s, \upsilon_F) \qquad (2)
\end{aligned}
$$

Proof: We can rewrite the expression for $\Delta^T(t_0, i, \beta)$ as an explicit summation over the possible state and belief state sequences:

$$
\begin{aligned}
\Delta^T(t_0, i, \beta) = &\sum_{s^{\leq T} \in (S)^T} \sum_{\boldsymbol{\beta}_{\bullet\Sigma}{}^{\leq T} \in (\mathbf{B})^T} \\
&\Bigg(\sum_{\boldsymbol{\beta}_{\Sigma\bullet}{}^{\leq T} \in (\mathbf{B})^T} \Pr\Big(S^{\leq T} = s^{\leq T}, \mathbf{b}_{\bullet\boldsymbol{\Sigma}}{}^{\leq T} = \boldsymbol{\beta}_{\bullet\Sigma}{}^{\leq T}, \mathbf{b}_{\boldsymbol{\Sigma}\bullet}{}^{\leq T} = \boldsymbol{\beta}_{\Sigma\bullet}{}^{\leq T} \\
&\qquad | \Upsilon^{t_0}_i = \upsilon_F, K_F = t_0, A_R = i, b^{t_0}_{i\bullet\Sigma} = \beta\Big) \\
&\qquad \cdot \sum_{t=0}^{T} R_\alpha(s^{\leq T}[t], \boldsymbol{\pi}_A(\boldsymbol{\beta}_{\Sigma\bullet}{}^{\leq T}[t]), \boldsymbol{\pi}_\Sigma(\boldsymbol{\beta}_{\bullet\Sigma}{}^{\leq T}[t])) \\
&- \sum_{\boldsymbol{\beta}_{\Sigma\bullet}{}^{\leq T} \in (\mathbf{B})^T} \Pr\Big(S^{\leq T} = s^{\leq T}, \mathbf{b}_{\bullet\boldsymbol{\Sigma}}{}^{\leq T} = \boldsymbol{\beta}_{\bullet\Sigma}{}^{\leq T}, \mathbf{b}_{\boldsymbol{\Sigma}\bullet}{}^{\leq T} = \boldsymbol{\beta}_{\Sigma\bullet}{}^{\leq T} \\
&\qquad | \Upsilon^{t_0}_i = \text{null}, K_F = t_0, A_R = i, b^{t_0}_{i\bullet\Sigma} = \beta\Big) \\
&\qquad \cdot \sum_{t=0}^{T} R_\alpha(s^{\leq T}[t], \boldsymbol{\pi}_A(\boldsymbol{\beta}_{\Sigma\bullet}{}^{\leq T}[t]), \boldsymbol{\pi}_\Sigma(\boldsymbol{\beta}_{\bullet\Sigma}{}^{\leq T}[t])) \Bigg)
\end{aligned}
$$

We then separate $\boldsymbol{\pi}_A$ into $\boldsymbol{\pi}_\Upsilon$ and $\boldsymbol{\pi}_\Phi$, the policies for role-taking and role-execution. We define $\boldsymbol{\pi}_{\Upsilon\upsilon}$ and $\boldsymbol{\pi}_{\Upsilon null}$ as two policies that differ only in the action for belief state β for agent i. The reward earned up to t_0 will be identical. By isolating the cost of role replacement and then substituting the resulting expression into $\Delta^T \geq 0$, we produce exactly the inequality from the statement of the theorem. □

Thus, Theorem 7 states that performing the role-replacement action υ_F is preferred when its total expected benefit exceeds its expected cost. More precisely, the outer summations on the left-hand side of the inequality iterate over all possible past histories of world and belief states, producing a probability distribution over the possible states the team can be in at time t_0. For each such state, the expression inside the parentheses computes the difference in domain-level reward, over all possible future sequences of world and belief states, between performing and not performing υ_F. The right-hand side of the inequality is a summation of the cost of performing the role replacement action υ_F over possible current states and belief states. Note that to calculate the inequality in Theorem 7, an agent would, in the worst case, need to sum rewards over all possible sequences of states and observations. This leads to a time complexity of $O((|S| \cdot |\Omega_\alpha|)^T)$. Thus, finding the locally optimal policy represents an $O\left(\left(|\Upsilon_\alpha|^{\frac{|\Omega_\alpha|^T - 1}{|\Omega_\alpha| - 1}}\right)^{|\alpha|}\right)$ speed-up over finding the globally optimal role replacement policy.

Contrast inequality 2 with inequality 1 (STEAM's role replacement policy): Criticality(F) is really an approximation of the total future reward for doing the role replacement, while criticality(R) is an approximation of the total future reward for not doing the role replacement. Thus, Theorem 7 provides some justification for STEAM's role replacement policy. Of course, inequality 1 is only an approximation; but it is at least compu-

tationally much cheaper ($O(|\alpha|)$). To further quantify this optimality-efficiency tradeoff, we rely on empirical evaluations (see next section).

Experimental Results

To demonstrate the utility of R-COM-MTDPs and specifically results of Theorem 7 (locally optimal policy), we now consider an example domain involving helicopter agents. These agents must decide whether to do a role replacement when a failure occurs. We compare the performance of the policies introduced in the previous section, across a space of distinct domains obtained by varying the cost of replacement and the probability of a failure.

Consider a scenario where 2 transport helicopters and 2 scouting helicopters have a joint goal to get from a point X to point Y along a path of 4 way-points. When a scouting helicopter moves along the path the traversed portion becomes safe for other helicopters to travel on. A helicopter may either move from its current position to the next point on the path with a single move action or may remain where it is. When a scout fails (e.g., it crashes) it can be replaced by a transport by incurring a *Role replacement cost* for expending additional man-power. Once a transport becomes a scout it cannot transform back. (We assume that there is an ordering that determines which transport will perform a role replacement.) Further, we assume that a helicopter can fail at any unscouted point x between X and Y based on some known (uniform) probability distribution. To ensure a focus on role replacement, we assume that the policies for role execution and communication are the same for all approaches. A helicopter's role execution policy while assigned to a scout role is that it will always go forward until it reaches Y, while the transport role execution policy is to wait at X until any one scout reaches Y. The reward is higher if more helicopters reach the destination safely and if they reach early rather than late.

We compared the performance of the two policies from the previous section and two straw-men policies, *Always Replace* and *Never Replace* to the globally optimal policy. In the *Locally Optimal* policy, transports use inequality 2 (from Theorem 7) to determine whether to replace a failed scout. In the *Never Replace* policy, no replacement is ever done. In the *Always Replace* policy, every failed scout is replaced; while in the *STEAM* policy, the transports use inequality 1 to determine whether to replace a failed scout. In STEAM, failure o f the last remaining scout would be seen as *critical* and all other roles as non-critical. We varied role replacement cost of a single replacement($R_{i\Upsilon}$) from 0 to 20 in steps of 1 on the Y-axis and the probability of failure of an agent given its position from 0.0 to 0.18 in steps of 0.02 on the X-axis, and plotted sub-optimality, i.e., the difference between the value of the globally optimal policy and the policy under consideration ($V_{opt} - V$) for these 210 configurations on the Z-axis of Fig. 1. The *Always* policy does much worse than the other policies for the configurations considered and so we omit it from this figure.

The uppermost, middle and lowest surfaces correspond to $V_{opt} - V_{Never}$, $V_{opt} - V_{STEAM}$ and $V_{opt} - V_{Locally\ Optimal}$, respectively. The closer the surface to the XY-plane the better is the performance. As seen in Fig. 1, in all cases that we consider, *Locally Optimal* does better than STEAM and *Never*. While STEAM does better than *Never* and *Always* in most cases, interestingly, it does worse than *Never* when replacement cost and probability of failure increases.

Fig. 2(a) shows a 2D-slice of Fig. 1 for probability of failure = 0.18. Fig. 2(a) indicates that the difference between STEAM(solid line) and *Never*(dotted line) reduces with increasing replacement cost. Eventually, *Never* starts overtaking STEAM — STEAM always chooses role substitution without considering the replacement cost (inequality 1). The difference between STEAM and the globally optimal policy(and locally optimal policy) initially decreases and then increases with increase in replacement cost, giving rise to a U-shape.

To understand the phenomenon such as the U-shape of STEAM's graph, Fig. 2(b) shows the number of role replacements for each of the policies, with increasing replacement cost for probability of failure=0.18. Policies, like STEAM, *Never*, and *Always* which don't consider replacement cost when deciding whether to do a replacement are not affected by the change in replacement cost, and hence the flat line for STEAM, *Never*, and *Always*. The locally optimal and globally optimal keep reducing the number of replacements made as the replacement cost is increased. Fig. 2(a) and Fig. 2(b) shows us that the sub-optimality of STEAM's policy is lowest when the number of replacements of the globally optimal policy and STEAM are closest. Initially STEAM was doing too few replacements and later too many replacements giving rise to the U-shape of STEAM's sub-optimality curve.

In this domain, the difference between values of the locally optimal policy and the globally optimal is very small as can be seen in Figs. 1 and 2(a). However, the locally optimal algorithm gave an order of magnitude speedup over the globally optimal, indicating that our "locally optimal" algorithm is a computationally advantageous alternative to the "globally optimal" strategy, and has optimality superior to current heuristic-based approaches. Thus, we have illustrated how we can design algorithms with different tradeoffs using R-COM-MTDP.

Summary

In this paper we present a formal model called *R-COM-MTDP* for *Team Formation for Reformation*. i.e., team formation keeping in mind future reformations that may be required. *R-COM-MTDP* which is based on decentralized communicating POMDPs, enables us to come up with policies for team (re)formation, communication and action and also enables a rigorous

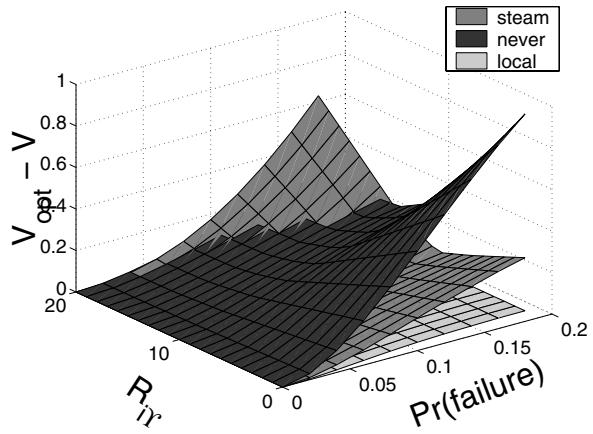
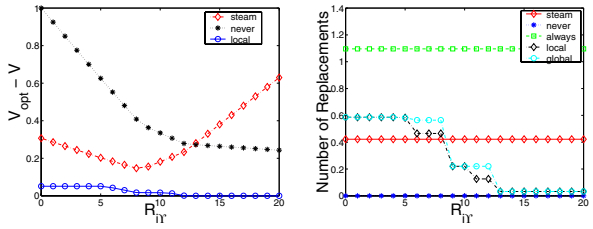


Figure 1: Sub-optimality of replacement policies:3D



(a) (b)

Figure 2: a: Sub-optimality of replacement policies, and b: Number of replacements vs. $R_{i\Upsilon}$ for Pr(failure)=0.18

analysis of complexity-optimality tradeoffs in coming up with these policies team formation and reorganization approaches. It provided: (i) worst-case complexity analysis of the team (re)formation under varying communication and observability conditions; (ii) illustrated under which conditions role decomposition can provide significant reductions in computational complexity; (iii) enabled theoretical and empirical analysis of specific role reorganization policies (e.g., showed where STEAM's role replacement does well). (These results have been rigorously proven, please see *http://www.isi.edu/teamcore/uai*). Thus, R-COM-MTDP could open the door to a range of novel analyses of multiagent coordination.

References

Bernstein, D. S.; Zilberstein, S.; and Immerman, N. 2000. The complexity of decentralized control of MDPs. In *UAI*.

Boutilier, C. 1996. Planning, learning & coordination in multiagent decision processes. In *TARK*.

Grosz, B., and Kraus, S. 1996. Collaborative plans for complex group action. *Artificial Intelligence* 86(2):269–357.

Kitano, H.; Tadokoro, S.; and Noda, I. 1999. Robocup-rescue: Search and rescue for large scale disasters as a domain for multiagent research. In *IEEE Conference SMC*.

Marsella, S.; Tambe, M.; and Adibi, J. 2001. Experiences acquired in the design of robocup teams: A comparison of two fielded teams. *JAAMAS* 4:115–129.

Modi, P. J.; Jung, H.; Tambe, M.; Shen, W.-M.; and Kulkarni, S. 2001. A dynamic distributed constraint satisfaction approach to resource allocation. In *Constraints Proc (CP)*.

Peshkin, L.; Meuleau, N.; Kim, K.-E.; and Kaelbling, L. 2000. Learning to cooperate via policy search. In *UAI*.

Pynadath, D., and Tambe, M. 2002. Multiagent teamwork: Analyzing the optimality complexity of key theories and models. In *AAMAS*.

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

Tidhar, G.; Rao, A.; and Sonenberg, E. 1996. Guided team selection. In *ICMAS*.

Xuan, P.; Lesser, V.; and Zilberstein, S. 2001. Communication decisions in multiagent cooperation. In *Agents*.

Yoshikawa, T. 1978. Decomposition of dynamic team decision problems. *Proceedings of the IEEE* AC-23(4):627–632.