Computational Models for Multiagent Coordination Analysis: Extending Distributed POMDP Models

Hyuckchul Jung, Ranjit Nair, Milind Tambe, and Stacy Marsella

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

Abstract. Recently researchers in multiagent systems have begun to focus on formal POMDP (Partially Observable Markov Decision Process) models for analysis of multiagent coordination. However, prior work has mostly focused on analysis of communication, such as via the COM-MTDP (Communicative Markov Team Decision Problem) model. This paper provides two extensions to this prior work that goes beyond communication and analyzes other aspects of multiagent coordination. In particular, we first present a formal model called R-COM-MTDP that extends COM-MTDP to analyze team formation and reorganization algorithms. R-COM-MTDP enables a rigorous and systematic analysis of complexity-optimality tradeoffs in team (re)formation approaches in different domain types. It provides the worst-case complexity analysis of the team (re)formation under varying conditions, and illustrates under which conditions role decomposition can provide significant reductions in computational complexity. Next, we propose COM-MTDP as a formal framework to analyze DCSP (Distributed Constraint Satisfaction Problem) strategies for conflict resolution. Different DCSP strategies are mapped onto policies in the COM-MTDP model, and agents compare strategies by evaluating their mapped policies. Thus, the two COM-MTDP based methods could open the door to a range of novel analyses of multiagent team (re)formation, and facilitate automated selection of the most efficient strategy for a given situation.

1 Introduction

Research in multiagent teamwork and cooperative multiagent systems has led researchers to develop successful practical multiagent applications. As the systems have matured, new competitive evaluation techniques are becoming increasingly popular (e.g., Robocup[2], RoboCup Rescue[3], TAC[11], Planning[6]). However, systematic techniques for performance analysis of these systems are still lacking. Thus, it is difficult to quantitatively compare the coordination approaches or behaviors employed in these systems, and understand which approach will dominate in particular circumstances.

Recently developed approaches[8, 9, 15] based on Distributed POMDPs and MDPs are beginning to remedy this problem, by providing tools for analysis as well as synthesis of multiagent coordination algorithms such as STEAM[10]. For instance, Pynadath and Tambe introduced COM-MTDP[9] (COMmunicative Multiagent Team Decision Problem) for analysis of communication actions in teams. While this work is promising, it has mostly focused on analysis of communication actions in multiagent systems. We extend this prior work to analyze other types of coordination beyond communication. We present two key extensions. The first extension focuses on taking and changing of roles in a team. The second extension illustrates analysis of value selection strategies in DCSP (Distributed Constraint Satisfaction Problem)[13].

Based on COM-MTDP, we build R-COM-MTDP, a model for analysis of team formation and reformation. The point here is not just that we can build R-COM-MTDP, but that we establish a methodology via which further such analyses could be performed. Using R-COM-MTDP, we specifically provide the worst case complexity analysis of the team (re)formulation under varying communication and observability conditions, and illustrate under which conditions role decomposition can provide significant reductions in computational complexity.

In addition to the extension of COM-MTDP model for team (re)formation, we propose COM-MTDP as a formal framework to analyze different value selection strategies of DCSP (Distributed Constraint Satisfaction Problems). DCSP value selection strategies are mapped onto policies in COM-MTDP, and compared by evaluating the policies. DCSP is a key method for conflict resolution in multiagent systems, and this COM-MTDP based approach will enable agents to predict the performance of their strategies and select the best one for fast conflict resolution convergence.

2 COM-MTDP: Communicative Markov Team Decision Problem

Given a team of agents, α, COM-MTDP model[9] 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 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 controls the effect of agents' actions in a dynamic environment: $P(s, \mathbf{a}, s') = Pr(S^{t+1} = s|S^t = s', A^t_\alpha = \mathbf{a})$ where S^t denotes the world state at time t. $\Omega_\alpha = \prod_{i\in\alpha}$ is a set of observations where Ω_i is the set of observation of an agent i. Observation function, $O_\alpha(s, \mathbf{a}, \omega) = Pr(\Omega^t_\alpha = \omega|S^t = s, A^{t-1}_\alpha = \mathbf{a})$, specifies the probability distribution of joint observations of the agent team α, and can be classified as follows:

- **Collective Partial Observability**: no assumption on the observations.
- **Collective Observability**: unique world state for the combined observation of the team $\alpha : \forall\omega \in \Omega_\alpha, \exists s \in S$ such that $\forall s' \neq s, Pr(\Omega^t_\alpha = \omega|S^t = s') = 0$.
- **Individual Observability**: unique world state for each individual agent's observation: $\forall\omega \in \Omega_i, \exists s \in S$ such that $\forall s' \neq s, Pr(\Omega^t_i = \omega|S^t = s') = 0$.

$B_\alpha = \prod_{i \in \alpha} B_i$ is the set of combined belief states, where B_i circumscribes the set of possible belief states for an agent i. Agent i makes decisions of which action to take and which message to communicate based on its belief state $b_i^t \in B_i$ derived by its observations and communicated messages through time t. With communication, belief state is divided into *pre-communication* belief state and *post-communication* belief state denoted by $b^t_{\bullet\Sigma}$ and $b^t_{\Sigma\bullet}$, respectively. π_{iA} is a domain-level *policy* defined as a mapping from belief states to actions, π_{iA} : $B_i \rightarrow A$.

COM-MTDP reward function represents a team's joint preference over states and the cost of actions and communication, R : $S \times \Sigma_\alpha \times A \rightarrow \Re$. Here, R is the sum of two rewards: (i) domain-action-level-reward, $R_{\alpha A}$:$S \times A_\alpha \rightarrow \Re$, and (ii) communication-level-reward,$R_{\alpha\Sigma}$:$S \times \Sigma_\alpha \rightarrow \Re$. COM-MTDP domains can be classified based on communication availability and its cost: (i) **General communication**: no assumption on Σ_α or $R_{\alpha\Sigma}$, (ii) **No communication**: $\Sigma_\alpha = 0$, and (iii) **Free communication**: $\forall \sigma \in \Sigma_\alpha, R_{\alpha\Sigma}(\sigma) = 0$.

3 R-COM-MTDP Model

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

$\mathcal{RL} = \{r_1, \ldots, r_s\}$ 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 divided into 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. 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. The domain-action-level-reward of R-COM-MTDP is further separated into reward for role-taking and role-execution actions. Here, we view the role taking reward as the cost for taking up different roles in different teams: e.g., if a satellite agent changes its role to join a new sub-team tracking a new star, there can be a delay in tracking. However, change of roles may potentially provide significant future rewards.

Furthermore, 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 for an agent team α, $\pi_{\alpha\Upsilon}$, $\pi_{\alpha\Phi}$ and $\pi_{\alpha\Sigma}$ that will maximize the total reward over a finite horizon T.

Extension for Explicit Local States (S_i): we often find that only a distinct part of the state space S is relevant for each individual agent that performs distinct roles within a team. Representing the world state as orthogonal features (i.e., $S = \Xi_1 \times \Xi_2 \times \cdots \times \Xi_n$), we can identify the subset of features of the world state that affect the observation of agent i. For each agent i, this subset of features is referred as its *local state*, $S_i = \Xi_{i1} \times \Xi_{i2} \times \cdots \times \Xi_{im_i}$. The local state is dynamic and could vary with the change of agents' roles, world states, etc. By definition, the observation that agent i receives at time t is independent of any features not covered by S_i^t: $\Pr(\Omega_i^t = \omega | S^t = \langle \xi_1, \xi_2, \ldots, \xi_n \rangle, A_\alpha^{t-1} = \mathbf{a}, \Omega_{\alpha\setminus\{i\}}^t = \omega_{\alpha\setminus\{i\}}) = \Pr(\Omega_i^t = \omega | S_i^t = \langle \xi_{i1}, \ldots, \xi_{im_i} \rangle, A_\alpha^{t-1} = \mathbf{a}, \Omega_{\alpha\setminus\{i\}}^t = \omega_{\alpha\setminus\{i\}})$, where $\Omega_{\alpha-i}^t = \prod_{j\in\alpha\setminus\{i\}} \Omega_j^t$. Here, another class of observation function is defined:

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

4 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. Refer to [5] for detailed theorem proofs.

Theorem 1. *R-COM-MTDP is reducible to an equivalent COM-MTDP and vice versa.*

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[9]. Table 1 shows the computational complexity for various classes of R-COM-MTDP domains, where the results for individual, collective, and collective partial observability follow from COM-MTDPs. New results in Table 1 come from analyzing the key addition in R-COM-MTDP, that of local states and local observability.

Theorem 2. *A collectively observable R-COM-MTDP is reducible to an equivalent locally observable R-COM-MTDP.*

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

	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.

Table 1. Computational Complexity

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. We can also show:

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.*

Role Decomposition: while roles are seen to be central in designing multiagent 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[14]. The qualitative intuition behind this *role decomposition* is that this separation simplifies the overall problem facing the agent team.

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.*

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.*

	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.

Table 2. Computational Complexity after Role Decomposition

Table 2 demonstrates that role decomposition can significantly lower computational complexity and together with Table 1, it allows us to compare the relative value of communication and role decomposition in simplifying the decision problem. 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.

5 DCSP Strategy Analysis with COM-MTDP

Distributed Constraint Satisfaction Problem (DCSP) techniques have been applied to various application problems such as distributed sensor network [4]. In this section, we introduce DCSP and a COM-MTDP-based model to analyze the performance of different DCSP value selection strategies.

5.1 Distributed Constraint Satisfaction Problem (DCSP)

A Constraint Satisfaction Problem (CSP) is commonly defined by a set of n variables, $X = \{x_1, ..., x_n\}$, each element associated with value domains D_1, ..., D_n respectively, and a set of k constraints, $\Gamma = \{C_1, ..., C_k\}$. A solution in CSP is the value assignment for the variables which satisfies all the constraints in Γ. A distributed CSP is a CSP in which variables and constraints are distributed among multiple agents[13]. Formally, there is a set of m agents, $Ag = \{A_1, ..., A_m\}$. Each variable (x_i) belongs to an agent A_j. There are two types of constraints based on whether variables in the constraint belong to a single agent or not:

- For a constraint $C_r \in \Gamma$, if all the variables in C_r belong to a single agent $A_j \in Ag$, it is called a *local constraint*.
- For a constraint $C_r \in \Gamma$, if variables in C_r belong to different agents in Ag, it is called an *external constraint*.

Figure 1-a illustrates an example of DCSP: each agent A_i (denoted by a big circle) has a local constraint LC_i and there is an external constraint C_{ij} between

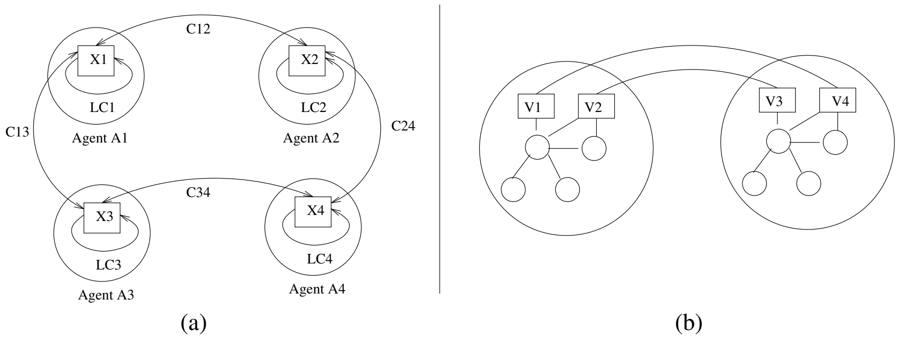


Fig. 1. Model of agents in DCSP

A_i and A_j. As illustrated in Figure 1-b, each agent can have multiple variables. There is no limitation on the number of local/external constraints for each agent. Solving a DCSP requires that agents not only satisfy their local constraints, but also communicate with other agents to satisfy external constraints.

Asynchronous Weak Commitment Search Algorithm (AWC): AWC is a sound and complete algorithm which shows the best performance among the published DSCP algorithms[12, 13]. In the AWC approach, agents asynchronously assign values to their variables from available domains, and communicating the values to neighboring agents. Each variable has a non-negative integer priority that changes dynamically during search. A variable is consistent if its value does not violate any constraints with higher priority variables. A solution is a value assignment in which every variable is consistent.

For simplification, suppose that each agent has exactly one variable and constraints between variables are binary. When the value of a variable is not consistent with the values of neighboring agents' variables, there can be two cases: (i) *good* case where there exists a consistent value in the variable's domain; (ii) *nogood* case that lacks a consistent value. In the nogood case, an agent increases its priority to *max+1*, where *max* is the highest priority of neighboring agents' variables. This priority increase makes previously higher agents select new values to satisfy the constraint with the new higher agent.

DCSP Value Selection Strategies in AWC Framework: While AWC relies on the *min-conflict* value selection strategy[7] that minimizes conflicts with other agents, new novel value selection strategies were introduced based on *local cooperativeness* (*local cooperativeness* measures how many compatible values are available in neighboring agents' domains)[1]:

- S_{low}: Each agent selects a new value from its consistent values maximizing the sum of compatible values with its *lower* priority neighbor agents.
- S_{high}: Each agent selects a new value from its consistent values maximizing the sum of compatible values with its *higher* priority neighbor agents.

- S_{all}: Each agent selects a new value from its consistent values maximizing the sum of compatible values with *all* neighbor agents.

Note that a value is consistent if the value is compatible with the values of higher priority agents. In the *nogood* case, since an agent increases its priority, every value in its domain is consistent. The three strategies above and the original *min-conflict* strategy can be applied to the *good* and the *nogood* case. Therefore, there are 16 strategy combinations such as $S_{low} - S_{high}$ (S_{low} is applied in the *good* case and S_{high} is applied to the *nogood* case). Since we will consider only strategy combinations, henceforth, they are referred as strategies for short. While agents' strategies can be heterogeneous, for simplification, we assume that every agent applies the same strategy. Performance evaluation with heterogeneous strategies will be considered in our future work.

It was shown that the value selection strategies described above have a great impact on conflict resolution convergence in solving DCSPs[1]. However, there was no universal best strategy for different problem domains. To gain maximum efficiency, it would be essential to predict the right strategy to use in a given domain. In the next section, we propose COM-MTDP as a formal framework for such strategy performance analysis. The DCSP strategies defined in [1] are used as exemplar strategies for the analysis. However, note that the approach described in the next section can be applied to other types of DCSP strategies.

5.2 Mapping DCSP onto COM-MTDP

In this section, we provide an initial mapping of DCSP onto COM-MTDP. While communication, observation, observation function, and belief state are key parts of COM-MTDP, they are not as relevant here and will not be discussed in this initial mapping. For instance, in AWC, each agent always communicates its changed value to neighboring agents without reasoning for communication. Future work will take them into account for strategy analysis: e.g., an agent may consider communication cost before communicating its local solution.

In a general mapping, the first question to address is the issue of state representation. One typical representation could be a vector of values of all variables in a given DCSP. However, this representation leads to a huge problem space. To avoid combinatorial space explosion in problem space, abstract state representation is used in the initial mapping. In particular, a COM-MTDP state s is represented by the combination of agents' local states (s_i) that specify the status of constraint violation. A status indicates whether an agent is in the *good* case or in the *nogood* case.

To further reduce the problem space with a large number of agents, we use a small scale model that represents a local interaction. In a 2D grid configuration (each agent is externally constrained with four neighbors except for the ones on the grid boundary), the small scale model consists of five agents (A_1, A_2, A_3, A_4, A_5) with a middle agent (A_3) surrounded by the other four agents. Thus, a state in the COM-MTDP model is the tuple of the five agents' local states.

In the mapping, S_{low}, S_{high}, S_{all}, and $min - conflict$ are the actions for agents in the COM-MTDP model. A DCSP strategy such as $S_{low} - S_{high}$ is akin to a policy in COM-MTDP: S_{low} in the *good* state and S_{high} in the *nogood* state. The performance of DCSP strategies are compared by evaluating their mapped policies. Here, the evaluation of a policy is done by computing the value of an initial state of the COM-MTDP model under the policy. Note that we do not attempt to find an optimal policy but try to evaluate given policies.

In building a COM-MTDP model, the state transition can be derived by combining the local state changes of individual agents. The local state change for an agent is governed by the agent's local state, its neighboring agents' states, and the policies (strategies) selected by agents. Now, the state transition probability is defined as a product of the probabilities of local state changes:

- $P(s, a, s') = \Pi_{i=1}^{5} Pr(s'_i | s, a)$ where $s' = \langle s'_1, s'_2, s'_3, s'_4, s'_5 \rangle$ denotes the next state after a state $s = \langle s_1, s_2, s_3, s_4, s_5 \rangle$ with an action a.

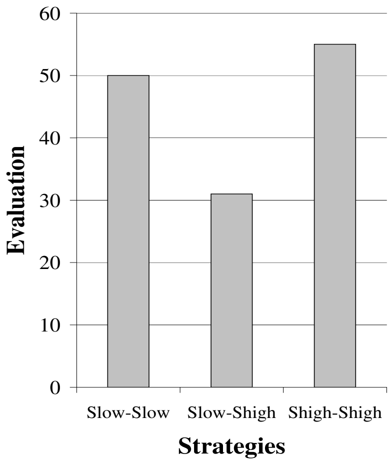
Note that the individual state transition probabilities are derived from the simulation of the whole system in a given problem setting, not from a simple experiment with only five agents.

Performance analysis is based on the fact that the best performing strategy has less chance of forcing neighboring agents into the *nogood* case than other strategies. Here, rewards (costs) are given to a state in proportion to the number of agents in the *nogood* cases. Therefore, as a policy (DCSP strategy) performs worse, its evaluation value increases. In comparing two strategies, a strategy with a smaller evaluation value is better than that with a larger evaluation value.

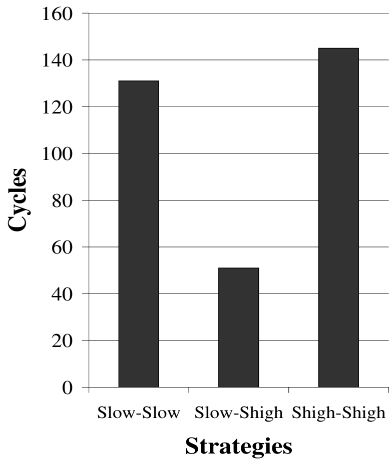
Analytical Results of DCSP Strategy Performance: Figure 2 shows the analytical results of three different strategies ($S_{low} - S_{low}$, $S_{low} - S_{high}$, and $S_{high} - S_{high}$) and the experimental results in two different problem settings (shown in [1]). In the analytical results (Figure 2-a & c), the vertical axis plots the evaluation value of each strategy computed from the COM-MTDP model. In the experimental results (Figure 2-b & d), the vertical axis plots the number of cycles until a solution in DCSP is found. Note that the lower evaluation values and cycles indicate the better performance in the analysis and the experiments respectively.

Here, the analytical results in Figure 2 match to the real experimental results in two different problem settings. That is, the ordering of strategies in analytical results is same with the empirical performance ordering. For instance, in the problem setting 1 (Figure 2-a & b), the experimental results (Figure 2-b) show that $S_{low} - S_{high}$ is the best with the least number of cycles, and $S_{low} - S_{low}$ and $S_{high} - S_{high}$ performs worse. The analytical results (Figure 2-a) clearly show that $S_{low} - S_{high}$ has the least evaluation value, and $S_{low} - S_{low}$ and $S_{high} - S_{high}$ have larger values. Note that, as a strategy performs worse, the evaluation value increases in the COM-MTDP model.

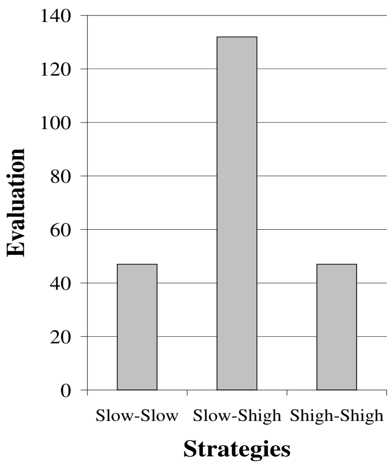
Our initial results illustrate that the strategy analysis with the COM-MTDP model match the actual experimental performance comparisons. Thus, the model



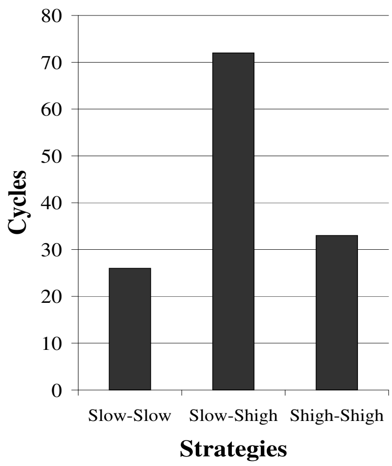
(a) Analytical results from setting 1



(b) Experimental results from setting 1



(c) Analytical results from setting 2



(d) Experimental results from setting 2

Fig. 2. DCSP strategy analysis with COM-MTDP

could potentially form a basis for predicting strategy performance in a given situation, and enable agents to flexibly adapt their strategies to changing circumstances.

6 Conclusion

In this paper, first we presented a formal model called R-COM-MTDP for modeling team formation and reorganization approaches by extending an existing formal model COM-MTDP that specializes in communication action analysis. R-COM-MTDP enables a rigorous and systematic analysis of complexity-optimality tradeoffs in team formation and reorganization approaches for different domain types. 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.

In addition to the extension of COM-MTDP for team (re)formation, we proposed COM-MTDP as a formal framework for DCSP value selection strategy analysis. Different strategies are mapped onto COM-MTDP policies, and their performance can be compared by evaluating the policies. Thus, R-COM-MTDP could open the door to a range of novel analyses of multiagent coordination. The two extensions to COM-MTDP provided in this paper could open the door to a range of novel analyses of multiagent coordination.

7 Acknowledgement

The research in this paper was supported by National Science Foundation grant #0208580 and NASA Jet Propulsion Laboratory subcontract "Continual Coherent Team Planning". We thank David Pynadath for valuable discussion and feedback.

References

1. Jung, H., Tambe, M., Kulkarni, S.: Argumentation as Distributed Constraint Satisfaction: Applications and Results, *Proceedings of International Conference on Autonomous Agents* (2001)
2. Kitano, H., Tambe, M., Stone. P., Veloso, M., Coradeschi, S., Osawa, E., Matsubara, H., Noda, I., Asada, M.: The Robocup Synthetic Agent Challenge 97, *Proceedings of International Joint Conference on Artificial Intelligence* (1997)
3. Kitano, H., Tadokoro, S., Noda, I., Matsubara, H., Takahashi, T., Shinjoh, A., Shimada, S.: Robocup Rescue: Search and rescue for large scale disasters as a domain for multi-agent research, *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* (1999)
4. Modi, P., Jung, H., Tambe, M., Shen, W., Kulkarni, S.: A Dynamic Distributed Constraint Satisfaction Approach to Resource Allocation, *Proceedings of International Conference on Principles and Practice of Constraint Programming* (2001)

5. Nair, R., Tambe, M., Marsella, S.: Team formation for reformation in multiagent domains like RoboCupRescue, *Proceedings of the International Symposium on RoboCup* (2002)
6. McDermott, D.: The 1998 AI Planning Systems Competition, *AI Magazine*, 21(2) (2000)
7. Minton, S., Johnston, M. D., Philips, A., Laird, P.: Solving Large-scale Constraint Satisfaction and Scheduling Problems Using a Heuristic Repair Method, *Proceedings of National Conference on Artificial Intelligence* (1990)
8. Peshkin, L., Meuleau, N., Kim, K., Kaelbling, L.: Learning to cooperate via policy search, *Proceedings of International Conference on Uncertainty in Artificial Intelligence* (2000)
9. Pynadath, D., Tambe, M.: Multiagent teamwork: analyzing the optimality complexity of key theories and models, *Proceedings of International Conference on Autonomous Agents and Multi Agent Systems* (2002)
10. Tambe, M.: Towards flexible teamwork, *Journal of Artificial Intelligence Research* 7:83-124 (1997)
11. Wellman, M., Greenwald, A. Stone, P., Wurman, P.: The 2001 Trading Agent Competition, *Proceedings of Innovative Applications of Artificial Intelligence Conference* (2002)
12. Yokoo, M.: Asynchronous Weak-commitment Search for Solving Distributed Constraint Satisfaction Problems, *Proceedings of International Conference on Principles and Practice of Constraint Programming* (1995)
13. Yokoo, M., Hirayama, K.: Distributed constraint satisfaction algorithm for complex local problems, *Proceedings of International Conference on Multi Agent Systems* (1998)
14. Yoshikawa T.: Decomposition of dynamic team decision problems, *Proceedings of IEEE Transactions on Automatic Control* (1978)
15. Xuan, P., Lesser, V., Zilberstein, S.: Communication decisions in multiagent cooperation, *Proceedings of Autonomous Agents* (2001)