Robust Execution-time Coordination in DEC-POMDPs Under Model Uncertainty

Jun-young Kwak, Rong Yang, Zhengyu Yin, Matthew E. Taylor*, Milind Tambe
University of Southern California, Los Angeles, CA, 90089
*Lafayette College, Easton, PA 18042
{junyounk,yangrong,zhengyuy,tambe}@usc.edu, *taylorm@lafayette.edu

ABSTRACT

Despite their worst-case NEXP-complete planning complexity, DEC-POMDPs remain a popular framework for multiagent teamwork. This paper introduces effective teamwork under model uncertainty (i.e., potentially inaccurate transition and observation functions) as a novel challenge for DEC-POMDPs and presents MODERN, the first execution-centric framework for DEC-POMDPs explicitly motivated by addressing such model uncertainty. MODERN's shift of coordination reasoning from planning-time to execution-time avoids the high cost of computing optimal plans whose promised quality may not be realized in practice. There are three key ideas in MODERN: (i) it maintains an exponentially smaller model of other agents' beliefs and actions than in previous work and then further reduces the computation-time and space expense of this model via bounded pruning; (ii) it reduces execution-time computation by exploiting BDI theories of teamwork, and limits communication to key trigger points; and (iii) it limits its decision-theoretic reasoning about communication to trigger points and uses a systematic markup to encourage extra communication at these points – thus reducing uncertainty among team members at trigger points. We empirically show that MODERN is substantially faster than existing DEC-POMDP execution-centric methods while achieving significantly higher reward.

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence

General Terms

Algorithms

Keywords

Distributed POMDPs, Model Uncertainty, Teamwork

1. INTRODUCTION

Despite their NEXP-complete policy generation complexity [1], *Distributed Partially Observable Markov Decision Problems* (DEC-POMDPs) have become a popular paradigm for multiagent teamwork [2, 7, 8, 10]. DEC-POMDPs are able to quantitatively express observational and action uncertainty, and yet optimally plan communications and domain actions.

The Sixth Annual Workshop on Multiagent Sequential Decision-Making in Uncertain Domains (*MSDM-2011*), held in conjunction with *AAMAS-2011* on May 3, 2011 in Taipei, Taiwan.

This paper focuses on teamwork under *model uncertainty* (i.e., potentially inaccurate transition and observation functions) in DEC-POMDPs. In many domains, we only have an approximate model of agent observation or transition functions. To address this challenge we rely on execution-centric frameworks [9, 13, 14], which simplify planning in DEC-POMDPs (e.g., by assuming cost-free communication at plan-time), and shift coordination reasoning to execution time. Execution-centric frameworks appear better-suited to address model uncertainty as they (i) lead to provably exponential improvement in worst-case complexity [8, 9]; (ii) avoid paying a high planning cost for a "high-quality" DEC-POMDP policy that cannot be realized in practice; and (iii) allow for coordination reasoning at execution-time to mitigate model uncertainty.

Unfortunately, past work in execution-centric approaches [9, 13, 14] also assumes a correct world model, and the presence of model uncertainty exposes three key weaknesses in that work. First, they maintain the entire set of the team's joint belief states for execution-time reasoning, a costly undertaking that is not well-justified given model uncertainty. Second, they reason at execution-time about the right action and communication before each decision step, leading to inefficient computation. Third, their detailed and expensive reasoning about communication and action is based on the assumption of an accurate model — again, given model uncertainty, such precise computation is wasteful due to its inaccuracy.

This paper provides two sets of contributions. The first is a new execution-centric framework for DEC-POMDPs called MODERN (MOdel uncertainty in Dec-pomdp Execution-time ReasoNing). MODERN's key insight is that given model uncertainty, it is wasteful to maintain a very detailed model of other agents or the team's belief states (it will be inaccurate anyway) and reason in depth with such a detailed model — the inferences will also be inaccurate. Instead, model uncertainty drives MODERN to simplify modeling and reasoning of other agents and boost communication at key junctures instead.

MODERN is the first execution-centric framework for DEC-POMDPs explicitly motivated by model uncertainty. It is based on three key ideas. First, MODERN reasons with an exponentially smaller model of other agents' beliefs and actions than the entire set of joint beliefs as done in previous work [9, 13, 14]; then it further reduces the computation time and space expense of this model via bounded pruning. Second, MODERN reduces execution-time computation by: (i) engaging in decision-theoretic reasoning about communication only at *Trigger Points* — instead of every agent reasoning about communication at every step, only agents encountering trigger points perform such reasoning; and (ii) utilizing a pre-planned policy for actions that do not involve interactions, avoiding on-line planning at every step. Our approach has significant advantages in domains with interaction-sparseness. Third, MODERN

increases communication at trigger points by doing a markup of the expected utility gain under the assumption that communication has significant value in reducing uncertainty. We justify our design decisions in MODERN through a systematic empirical evaluation. As our evaluation shows, MODERN outperforms competing algorithms in terms of its run-time performance while finding higher quality solutions.

This paper's second set of contributions are in opening up model uncertainty as a new research direction for DEC-POMDPs and emphasizing the similarity of this problem to the *Belief-Desire-Intention* (BDI) model for teamwork [3, 6, 11]. In particular, BDI teamwork models also assume inaccurate mapping between real-world problems and domain models. As a result, they emphasize robustness via execution-time reasoning about coordination [11]. Given some of the successes of prior BDI research in teamwork, we leverage insights from BDI in designing MODERN.

2. PROBLEM STATEMENT

DEC-POMDPs have been used to tackle real-world multi-agent collaborative planning problems under transition and observation uncertainty, which are described by a tuple $\langle I, S, \{A_i\}, \{\Omega_i\}, T, R, O, \mathbf{b}^0 \rangle$. $I = \{1, ..., n\}$ is a finite set of agents, and $S = \{s_1, ..., s_k\}$ is a finite set of joint states. A_i is the finite set of actions of agent i, $A = \prod_{i \in I} A_i$ is the set of joint actions, where $\mathbf{a} = \langle a_1, ..., a_n \rangle$ is a particular joint action (one individual action per agent). Ω_i is the set of observations of agent i, $\Omega = \prod_{i \in I} \Omega_i$ is the set of joint observations, where $\mathbf{o} = \langle o_1, ..., o_n \rangle$ is a joint observation. $T : S \times A \times S \mapsto \mathbb{R}$ is the transition function, where $T(s'|s, \mathbf{a})$ is the transition probability from s to s' if joint action $\mathbf{a}$ is executed. $O : S \times A \times \Omega \mapsto \mathbb{R}$ is the observation function, where $O(\mathbf{o}|s', \mathbf{a})$ is the probability of receiving the joint observation $\mathbf{o}$ if the end state is s' after $\mathbf{a}$ is taken. $R(s, \mathbf{a}, s')$ is the reward that agents get by taking $\mathbf{a}$ from s and reaching s', and $\mathbf{b}^0$ is the initial joint belief state.

We denote the joint observation history at time step t with $\mathbf{h}^t = \{\mathbf{o}^1, \mathbf{o}^2, \ldots, \mathbf{o}^t\}$ and the set of $\mathbf{h}^t$ with H^t. $H = \bigcup_t H^t$ is the set of all possible joint observation histories at all time steps. A joint policy $\pi : H \mapsto A$ is a mapping from joint observation history to joint action. Let $h_i^t = \{o_i^1, o_i^2, \ldots, o_i^t\}$ be the individual observation history of agent i at time step t. The set of all possible h_i^t is denoted by H_i^t. Let $H_i = \bigcup_t H_i^t$ be the set of all possible individual observation histories at all time steps for agent i. The individual policy for agent i, $\pi_i : H_i \mapsto A_i$, is a mapping from agent i's individual observation history to its individual action. We use h_{-i}^t to denote an observation history of all agents except i and H_{-i}^t to denote the set of all possible h_{-i}^t. Similarly we use π_{-i} to represent the policy for all agents except i.

Here, we assume the presence of model uncertainty, which is modeled with a Dirichlet distribution [4]. A separate Dirichlet distribution for the observation and transition function is used for each joint state, action, and observation. An L-dimensional Dirichlet distribution is a multinomial distribution parameterized by positive hyper-parameters $\boldsymbol{\beta} = \langle \beta_1, \ldots, \beta_L \rangle$ that represents the degree of model uncertainty. The probability density function is

$$f(x_1, ..., x_L; \boldsymbol{\beta}) = \frac{\prod_{i=1}^{L} x_i^{\beta_i - 1}}{B(\boldsymbol{\beta})}, B(\boldsymbol{\beta}) = \frac{\prod_{i=1}^{L} \Gamma(\beta_i)}{\Gamma(\sum_{i=1}^{L} \beta_i)},$$

and $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ is the standard gamma function. The maximum likelihood point can be easily computed: $x_i^* = \frac{\beta_i}{\sum_{j=1}^{L} \beta_j}$, for $i = 1, ..., L$. Let $\mathbf{T}_{s,\mathbf{a}}$ be the vector of transition probabilities from s to other states when $\mathbf{a}$ is taken and $\mathbf{O}_{s',\mathbf{a}}$ be the vector of observation probabilities when $\mathbf{a}$ is taken and s' is reached. Then $\mathbf{T}_{s,\mathbf{a}} \sim Dir(\boldsymbol{\beta})$ and $\mathbf{O}_{s',\mathbf{a}} \sim Dir(\boldsymbol{\beta}')$, where $\boldsymbol{\beta}$ and $\boldsymbol{\beta}'$ are two different hyper-parameters.

We assume that the planner is not provided the precise amount of model uncertainty (i.e., the precise amount of uncertainty over transition or observation uncertainty). Our goal is effective teamwork, i.e., achieving high reward in practice, at execution time.

3. RELATED WORK

3.1 DEC-POMDPs

Related work includes DEC-POMDP planning that specifically focuses on optimal communication [2, 8]. In addition to its lack of emphasis on execution-time reasoning, this research has not tackled the challenge of model uncertainty. Furthermore, given general communication costs, the policy generation problem remains NEXP-complete. Although execution-centric approaches [9, 12, 13, 14] lead to a provably exponential improvement in worst-case complexity over optimal DEC-POMDP planners, they have also assumed model correctness.

Xuan and Lesser [14] studied the trade-offs between centralized and decentralized policies in terms of communication requirements, as well as provided a method to transform centralized policies to distributed ones. This research differs from our own given its focus on distributed MDPs rather than DEC-POMDPs (and thus does not face the challenges of modeling other agents' belief distributions), and its assumption of model correctness.

ACE-PJB-COMM (APC), one of the first execution-centric system [9] assumes free communication when planning and collapses the multi-agent problem to a single-agent POMDP. At execution time, agents execute the plan in a decentralized fashion, communicating to avoid miscoordination. If the expected utility gain from communication is greater than communication cost, an agent will communicate its history to the rest of the team. Similarly, MAOP-COMM (MAOP) [13] communicates whenever it detects a *history inconsistency* that might cause miscoordination. APC and MAOP respectively use *GrowTree* and *JointHistoryPool*, the set of possible belief nodes to reason about the team's belief space when an agent does not know its teammate's actions or observations.

Our work differs from these works as we assume model uncertainty, leading to several major differences. First, MODERN maintains an exponentially smaller set of beliefs than *GrowTree* and *JointHistoryPool* mentioned above. It further reduces the computation time and space via bounded pruning, which allows it to significantly improve the scalability compared to APC with particle filtering and MAOP with belief merging. Second, MODERN provides selective reasoning about communication, reducing the computation burden, while APC and MAOP reason about action and communication at every time step in an online-manner. Third, assuming model uncertainty, MODERN boosts communication, while APC and MAOP assume model correctness, and thus they rely on precise local computation over joint beliefs. Fourth, MODERN explicitly considers two-way communication to synchronize beliefs, while APC only considers one-way broadcasts. Although APC pays less in communication cost than MODERN, relying on reasoning over uncertain belief states causes more conservative action selection, leading to lower reward. An additional difference with MAOP is that it does not consider communication cost, while MODERN reasons about communication based on cost-utility analysis, which allows us to show significantly superior results as communication cost increases.

3.2 BDI Teamwork

While BDI is unable to quantitatively reason about costs and uncertainties, nevertheless, there have been some successes in prior BDI work [3, 6, 11]. Here, we draw upon three key ideas from BDI teamwork in MODERN. First, BDI approaches focus on execution and thus emphasize the execution-time teamwork reasoning rather than plan-time reasoning, as a pre-planned team coordination may lead to dramatic failures when unanticipated events occur. BDI teamwork frameworks simplify planning by focusing on team-oriented programs that abstract away from "low-level coordination," instead shifting coordination reasoning (i.e., communication) to execution time. This in essence is similar to the execution-centric DEC-POMDP framework. Second, agents differentiate between individual actions and actions that require interaction with others; in many teamwork domains agents are seen to act individually for significant portions of the task performance and then occasionally perform a tightly coupled action. Third, agents only reason about communication when the plan requires interactions with others; agents do not reason about communication at every step. For example, key teamwork execution systems have been based on joint commitments [6]. A joint commitment between two agents to a joint goal P leads to two types of communication:

- In order to form a joint commitment, an agent requests others to commit to its goal, P. We refer to this as "asking," and here an agent's action changes based on response from the other agent.
- Once jointly committed to P, if an agent privately comes to believe that P is achieved, unachievable, or irrelevant, it communicates this to its teammates. We refer to this as "telling." The other agent's action changes due to the communication.

4. DESIGN DECISIONS

During planning, MODERN has a standard single-agent POMDP planner [5] plan a policy for the team of agents by assuming zero-cost communication. Then, at execution-time, agents model other agents' beliefs and actions, reason about when to communicate with teammates, reason about what action to take if not communicating, etc.

MODERN's design is driven by the model uncertainty, leading to three major novel ideas. MODERN (i) maintains a very small set of beliefs to reason about other agents because a complete set of joint beliefs (which may be erroneous under model uncertainty) is not well-justified, (ii) selectively reasons about communication to reduce execution-time computational burden because precise reasoning at every step does not provide additional benefits, but rather exacerbates the computational problems, and (iii) simplifies its decision-theoretic reasoning and marks up the expected utility gain to overcome model uncertainty by boosting communication at trigger points.

Thus, *it is precisely due to model uncertainty* that MODERN simplifies modeling and reasoning about other agents by maintaining a bounded approximate model (compared to previous work [9, 13] which maintains a very detailed model). Instead, MODERN boosts communication at trigger points. As shown in our experimental results, it is precisely due to this aggressive reliance on communication rather than detailed reasoning that MODERN outperforms its competitors who are much more reliant on their models of other agents. We describe these ideas in the following.

4.1 Modeling Other Agents

In contrast with the complete tree of joint beliefs in [9, 13], MODERN maintains an approximate and exponentially smaller set of beliefs to model other agents via (i) *Individual estimate of joint Beliefs (IB)* and (ii) *Bounded Pruning*.

IB is a concept used in MODERN to decide whether or not communication would be beneficial and to choose a joint action when not communicating. IB can be conceptualized as a subset of team beliefs that depends on an agent's local history, leading to an exponential reduction in belief space compared to *GrowTree* mentioned earlier. In particular, for two agents, *GrowTree* has $(|o_1||o_2|)^t$ number of nodes for the team while IB maintains only $|o_1|^t$ or $|o_2|^t$ number of nodes, where $|o_i|$ is the number of agent i's local observations, and t is a time step. IB^t describe the set of nodes of the possible belief trees of depth t. Each node θ in IB^t has a tuple consisting of $\langle \mathbf{b}(\theta), \mathbf{h}(\theta), \mathbf{a}(\theta), p(\theta) \rangle$, where $\mathbf{b}(\theta)$ is the joint belief given that $\mathbf{h}(\theta)$, $\mathbf{h}(\theta)$ is the joint observation history, $\mathbf{a}(\theta)$ is the joint action obtained from a given policy tree, and $p(\theta)$ is the likelihood of observing $\mathbf{h}(\theta)$.

For example, consider a 2×3 grid world with two agents (see Figure 1). The agents can wait (W) or move in 4 directions. The team's goal is to perform a joint task (P_j, e.g., defusing a bomb) or individual task (P_i, e.g., charging the battery) at pre-specified locations to achieve a high reward. There are three observations per agent: *location-of-joint-task* (o), *not-location-of-any-task* ($\bar{o}$), and *location-of-individual-task* ($\hat{o}$). We will describe the domain in detail later in Section 6. An agent's IB in this domain can be represented as a tree shown in Figure 2. The initial distribution of possible joint beliefs is composed of a single leaf at belief $\mathbf{b}^0$, the starting belief of the team, with probability 1 and an empty observation history. Suppose that the team chooses to execute the action $\langle \mathit{Move}_{\mathit{east}}, \mathit{Move}_{\mathit{west}} \rangle$. The IB of each agent only maintains nodes that are consistent with its local observation history. Thus, IB for agent 1 has 3 nodes in total, $\langle o, o \rangle$, $\langle o, \bar{o} \rangle$, $\langle o, \hat{o} \rangle$ assuming that agent 1's local observation is o, while a joint team belief such as *GrowTree* maintains a full tree consisting of 9 nodes at time horizon 1. This shows how our individual model can be much smaller than the joint team belief. However, the number of possible beliefs in IB grows rapidly, particularly when agents choose not to communicate for long time periods.

Hence, we propose a new pruning algorithm that provides further savings. In particular, it keeps a fixed number of *most likely* beliefs per time step in IB. Our pruning method first expands beliefs using the Bayes update rule and then selects the most likely belief at each time step until the selected number of beliefs reaches a pre-defined upper-limit. This reduced belief set is used to detect trigger points and reason about communication in MODERN. Note that MODERN uses a *sync* action in communication (discussed below) that is useful to ensure that all agents create an identical belief. This provides a way to ascertain the team's joint status and avoid miscoordination.

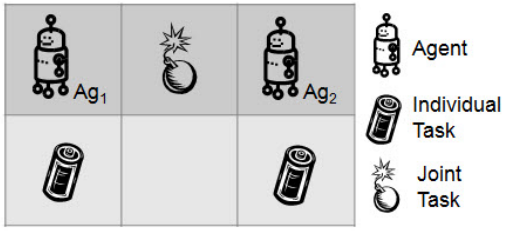


Figure 1: Illustrative Domain

4.2 When to Reason: Trigger Points

The policy provided to each agent from MODERN's planning maps the agent team's joint observation to joint actions of the team. Unlike APC or MAOP, MODERN does not require agents to reason from scratch about what action or communication to execute at

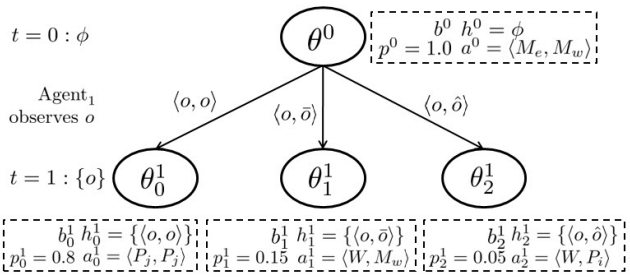


Figure 2: Example of an individual belief maintained by agent$_1$

every time step. Instead, agents follow the provided policy, mapping their own observation in the policy to their own action, except at trigger points inspired by joint commitments. Note that trigger points include any situation involving ambiguity in mapping an agent's observation to its action in the joint policy. The key idea is that in sparse interaction domains, agents will not have to reason about coordination at every time step and only infrequently encounter trigger points, thus significantly reducing the burden of execution-time reasoning. First, we define trigger points inspired by BDI teamwork:

Definition Time step t is a trigger point for agent i if either of the following conditions are satisfied.

Asking Let h_i^t be the actual individual observation history of agent i. Time step t is an Asking trigger point for agent i if there exist two different $h_{-i}^t, \tilde{h}_{-i}^t \in H_{-i}^t$ such that $\pi_i(\mathbf{h}^t) \neq \pi_i(\tilde{\mathbf{h}}^t)$, where $\mathbf{h}^t = h_i^t \otimes h_{-i}^t$ and $\tilde{\mathbf{h}}^t = h_i^t \otimes \tilde{h}_{-i}^t$.

Telling Time step t is a Telling trigger point for agent i if there exists at least one $h_{-i}^t \in H_{-i}^t$, and two different $h_i^t, \tilde{h}_i^t \in H_i^t$, such that $\pi_{-i}(\mathbf{h}^t) \neq \pi_{-i}(\tilde{\mathbf{h}}^t)$, where $\mathbf{h}^t = h_i^t \otimes h_{-i}^t$ and $\tilde{\mathbf{h}}^t = \tilde{h}_i^t \otimes h_{-i}^t$.

Let us consider the example discussed in Section 4.1. As shown in Figure 2, there is no ambiguity when selecting a joint action at t=0. Thus, no trigger point is detected and the agents select a joint action, $\langle \mathit{Move}_{\mathit{east}}, \mathit{Move}_{\mathit{west}} \rangle$, from the given policy. At t=1, agent 1 detects a trigger point by the *asking* condition because agent 1's action choice is affected by agent 2's local observation: either P_j if agent 2 observes o or W if agent 2 observes $\bar{o}$ or $\hat{o}$. Thus, agents start reasoning about communication.

4.3 Reasoning at Trigger Points

MODERN's reasoning about communication is governed by the following formula: $f(\kappa, t) \cdot (U_{\mathrm{C}}(i) - U_{\mathrm{NC}}(i)) > \sigma$, where κ is a markup rate, t is a time step, $U_{\mathrm{C}}(i)$ is the expected utility of agent i if agents were to communicate, $U_{\mathrm{NC}}(i)$ is the expected utility of agent i when it does not communicate, and σ is a given communication cost. The two novelties in MODERN's reasoning are how it computes $U_{\mathrm{C}}(i)$ and $U_{\mathrm{NC}}(i)$, and how it uses the markup function $f(\kappa, t)$. Both of these are motivated by model uncertainty.

In MODERN, agents reason about whether or not communication would be beneficial. If they do communicate, all agents synchronize by sending all local observation histories to others. Thus, all the agents reach a specific belief node, θ, and can choose a joint action for the team. Otherwise, if no agent chooses to communicate, each agent chooses the best locally optimal action based on estimated most likely actions of the other agents. Computation of $U_{\mathrm{C}}(i) - U_{\mathrm{NC}}(i)$ is performed as following:

$$U_{\mathrm{C}}(i) = \sum_{\theta \in \mathrm{IB}_i} p(\theta) \cdot V\left(\mathbf{b}(\theta), \mathbf{a}(\theta)\right),$$

$$U_{\mathrm{NC}}(i) = \max_{a_i \in A_i} U_{\mathrm{IB}_i}(\langle a_i, a_{-i}^* \rangle),$$

Table 1: Markup values: experimental observations

σ	Sparse-interaction	Highly-coupled
Low	κ=1.0	κ=1.0
Medium	1.0< κ <2.0 (*)	κ=1.0
High	1.0< κ <1.5 (*)	1.0< κ <1.5 (*)

$$a_{-i}^* = a_{-i}(\theta^*) \text{ s.t. } \theta^* \in \Theta,$$

$$U_{\mathrm{IB}_i}(\mathbf{a}) = \sum_{\theta \in \mathrm{IB}_i} p(\theta) \cdot V(\mathbf{b}(\theta), \mathbf{a}),$$

where $V(\mathbf{b}, \mathbf{a})$ is the expected utility when an action $\mathbf{a}$ is taken at belief state $\mathbf{b}$. a_{-i}^* is agent i's estimate of the most likely action of all other agents. This is greedily selected using the most likely sequence, Θ, at every time step, where agent i optimistically assumes that all other agents obtain the most likely observations. $U_{\mathrm{C}}(i)$ is calculated by considering two-way synchronization, which emphasizes the benefits from communication. $U_{\mathrm{NC}}(i)$ is computed based on the individual evaluation of heuristically estimated actions of other agents.

The markup function, $f(\kappa, t)$, helps agents to reduce uncertainty among team members by marking up the expected utility gain from communication rather than perform precise local computation over erroneous models. In this work, we use an exponential markup rate, $f(\kappa, t) = \kappa^t$. Because uncertainty among team members increases as time passes, the markup rate should increase according to the time step. An optimal markup function, $f(\kappa, t)$, can lead to substantial improvements. While we do not have a theoretical method of computing κ, our extensive experimental results illustrate some potential guidelines for setting κ. In Table 1, (*) indicates where κ significantly impacts the overall performance. In particular, when communication cost (σ) is low, marking up appears to have little impact on total expected reward, hence we suggest κ=1.0 (row 2, columns 2 and 3). In a sparse-interaction domain, when σ is medium, κ significantly impacts the overall performance, thus 1.0< κ <2.0 appears to be beneficial (row 3, column 2). When σ is high, overestimating κ can degrade performance, thus κ between 1.0 and 1.5 is suggested (row 4, columns 2 and 3). We empirically justify this heuristic in Section 6.

5. THE MODERN ALGORITHM

The MODERN algorithm first takes a provided single joint policy from the offline planning as means for the underlying decision making. Each node in IB is expanded using possible observations and joint actions from the given policy, and then MODERN detects trigger points based on the belief tree. Once the agents detect a trigger point, they reason about whether or not communication would be beneficial, as discussed in the previous section. If agents do not detect trigger points, this implies there is little chance of miscoordination, and they take individual actions as per the given policy.

MODERN is presented in detail in Algorithm 1. The joint policy π is provided as an input to MODERN. On line 1, the initial distribution of possible beliefs, IB^0, is composed of a single node at belief $\mathbf{b}^0$ (the starting belief of the team), which has probability 1, an empty observation history, and a joint action, $\mathbf{a}^0$, which is described by the root of π. In line 6, IB is updated by the standard Bayes update rule. Then, MODERN decides whether or not a trigger point exists on the current time step (line 7). If a trigger point is detected, MODERN reasons whether the expected utility gain caused by communication justifies communication cost (lines 9–19). Specifically, when a trigger point is detected, agent i communicates when the expected utility gain by communication is higher than a given communication cost. Otherwise, the agent

simply selects and executes its action from π. When agents decide to communicate, they share their local histories (i.e., synchronize their histories). Agents can then know the actual joint observation histories and execute the joint action given by the policy (line 17). Otherwise, they execute the estimated best joint action (line 19). If the trigger point is not detected, agents take their individual actions from π (line 21).

As an example, again consider the 2×3 grid domain from Figure 1. MODERN first initializes IB with a single node that has the initial belief, $\mathbf{b}^0$, with likelihood $p^0 = 1.0$ (line 1). The initial joint action $\mathbf{a}^0 = \langle M_e, M_w \rangle$ is obtained from the provided policy (line 2). Then the two agents execute their local actions respectively: agent 1 moves east and agent 2 moves west to approach the task location. As illustrated in Figure 2, agent 1 observes o, i.e., it is at the joint task location. It expands IB_1 considering all possible joint observations that are consistent with o: $\langle o, o \rangle$, $\langle o, \bar{o} \rangle$, and $\langle o, \hat{o} \rangle$. At the same time, agent 2 also updates IB_2 based on its local observation (line 6). At $t = 1$, agent 1 checks whether the current situation meets any of the trigger point conditions (line 7). According to IB_1, we see that agent 1's local action is affected by agent 2's observation, and thus a trigger point is detected by the *asking* condition. Because a trigger point is detected at $t = 1$, agent 1 reasons whether communication would be beneficial to improve team performance. First, it computes $U_C(1) = 0.8V(\mathbf{b}_0^1, \mathbf{a}_0^1) + 0.15V(\mathbf{b}_1^1, \mathbf{a}_1^1) + 0.05V(\mathbf{b}_2^1, \mathbf{a}_2^1)$ (line 9). Then, agent 1 considers possible individual actions given by the policy: $\{P_j, W\}$ (line 10), and estimates the most likely action of agent 2 assuming agent 2 keeps obtaining the most likely observation (line 11). Let us assume agent 2's most likely action is P_j (i.e., performing a joint task). Then, $U_{\mathrm{NC}}(1) = \max\{0.8V(\mathbf{b}_0^1, \langle P_j, P_j \rangle) + 0.15V(\mathbf{b}_1^1, \langle P_j, P_j \rangle) + 0.05V(\mathbf{b}_2^1, \langle P_j, P_j \rangle), 0.8V(\mathbf{b}_0^1, \langle W, P_j \rangle) + 0.15V(\mathbf{b}_1^1, \langle W, P_j \rangle) + 0.05V(\mathbf{b}_2^1, \langle W, P_j \rangle)\}$ (line 13). Agent 1 then checks whether the utility gain, $f(\kappa, 1) \cdot (U_C(1) - U_{\mathrm{NC}}(1))$, is greater than the communication cost σ (line 14). If so, it communicates (line 15), otherwise it selects the estimated best action based on $U_{\mathrm{NC}}(1)$ (line 19). If no trigger point is detected, agents execute their local actions from the given policy tree (line 21).

Algorithm 1 MODERN(JOINTPOLICY π, AGENTINDEX i)

1: $\{$Initialize individual estimate IB_i^0; $\tau \leftarrow false\}$
2: $\{\mathbf{a}^0 \leftarrow \pi(\mathrm{IB}_i^0)$; Execute a local action $a_i^0\}$
3: **for** $t = 1, \ldots, T-1$ **do**
4: $\quad o_i^t \leftarrow$ Get the observation from the environment
5: $\quad h_i^t \leftarrow$ Update agent i's own local history with o_i^t
6: $\quad \mathrm{IB}_i^t \leftarrow$ EXPAND($\mathrm{IB}_i^{t-1}, o_i^t$)
7: $\quad \tau \leftarrow$ DETECTTRIGGERPOINT(π, IB_i^t)
8: $\quad$ **if** $\tau = true$ **then**
9: $\quad\quad U_{\mathrm{C}} \leftarrow \sum_{\theta \in \mathrm{IB}_i^t} p(\theta) \cdot V(\mathbf{b}(\theta), \mathbf{a}(\theta))$
10: $\quad\quad A_i^t \leftarrow \{\pi_i(\mathbf{h}^t(\theta)) | \theta \in \mathrm{IB}_i^t\}$
11: $\quad\quad a_{-i}^* = a_{-i}(\theta^*)$
12: $\quad\quad a_i^* \leftarrow \arg\max_{a_i \in A_i^t} \sum_{\theta \in \mathrm{IB}_i^t} p(\theta) V\left(\mathbf{b}(\theta), \langle a_i, a_{-i}^* \rangle\right)$
13: $\quad\quad U_{\mathrm{NC}} \leftarrow \sum_{\theta \in \mathrm{IB}_i^t} p(\theta) \cdot V(\mathbf{b}(\theta), \langle a_i^*, a_{-i}^* \rangle)$
14: $\quad\quad$ **if** $f(\kappa, t) \cdot (U_{\mathrm{C}} - U_{\mathrm{NC}}) > \sigma$ **then**
15: $\quad\quad\quad$ Sync h_i^t with other agents
16: $\quad\quad\quad \mathrm{IB}_i^t \leftarrow$ Update via communicated joint history $\mathbf{h}^t$
17: $\quad\quad\quad \{a_i^t \leftarrow \pi_i(\mathrm{IB}_i^t)$; $\tau \leftarrow false\}$
18: $\quad\quad$ **else**
19: $\quad\quad\quad a_i^t \leftarrow a_i^*$
20: $\quad$ **else**
21: $\quad\quad a_i^t \leftarrow \pi_i(\mathrm{IB}_i^t)$
22: $\quad$ Execute the action a_i^t

6. EMPIRICAL VALIDATION

We evaluate the performance of MODERN on several domains and compare it with two previous techniques: APC [9] and MAOP [13]. Since APC and MAOP have only limited abilities to scale up to large domains, we first show results in small domains. Then, we scale up MODERN and show results for larger domains. The *planning* time for all algorithms is identical and thus we only measure the average execution-reasoning time per agent.

Noise in transition matrix and observation matrix follow a Dirichlet distribution (which is not known by the planner or the agents). The level of model error is represented by a parameter α ($\sum_i^L \beta_i$) in Dirichlet distribution: error *increases* as α *decreases*. We evaluate the performance of MODERN under four different amounts of error by varying α from 10 to 10000. The experiments were run on Intel Core2 Quadcore 2.4GHz CPU with 3GB main memory. All techniques were evaluated for 600 independent trials throughout this section. We report the average rewards.

6.1 Domain Descriptions

Small Grid Domains: 1×5 and 2×3 domains were used for evaluation. In both domains, there are two agents trying to perform one joint task. In the 1×5 grid domain, each agent has three actions: *move east*, *move west* and *perform joint task*. Each agent can obtain two observations: whether or not it is at a joint location. In this domain, we have 50 joint states, 9 joint actions and 4 joint observations. In the 2×3 grid, each agent has two additional actions: *move north* and *move south*. There are 72 joint states, 25 joint actions and 4 joint observations in this configuration. In both configurations, each movement action incurs a small penalty of -0.2. The joint task requires that both agents perform the task together at the joint location. If the joint task is successfully performed, a reward of +10 is obtained. If only one agent performs the joint task or the joint task action is performed at the wrong location (i.e., miscoordination), a penalty of -5 or -2 is given to the team, respectively.

Large Grid Domain: In Sections 6.3–6.5, we consider a scale-up of the 2×3 grid domain since we can quickly solve the small grid domains. As shown in Figure 1, two individual tasks are added to the grid, which require only one agent to perform. In this new domain, each agent has one additional observation regarding an individual task location and two additional actions: *perform individual task* and *wait*. The number of joint states is 288, the number of joint actions is 49, and the number of joint observations is 9. If any agent performs the individual task action at the correct location, the team receives a reward of +5. If both agents try to perform the same individual task, the team reward will be unchanged (+5). On the other hand, if an agent attempts to perform the individual task at any incorrect location, a penalty of -1 will be assessed. If an agent chooses the action *wait*, there will be no penalty or reward.

Dec-Tiger Domain: The fourth domain used for our evaluation is the Dec-Tiger domain [7]. It has 2 joint states, 4 joint observations, and 9 joint actions. In this domain, miscoordination happens when both agents open the door with the tiger or when each agent opens a different door. We included this domain because APC was built specifically for this domain. While MODERN focuses on domains where interactions among agents are sparse, the tiger domain has highly coupled actions among agents.

6.2 Comparison: Solution Quality

We compared the average rewards achieved by all algorithms for three different communication costs in two small grid domains and the tiger domain. The communication costs are selected proportional to the expected value of the policies: $5\%, 20\%$, and 50%. The time horizon was set to 3 in this set of experiments.

Table 2: Comparison MODERN (MD) with APC and MAOP: Average Performance in Small Domains

		1×5 Grid				2×3 Grid				Dec-Tiger			
σ	α	MD(κ_1)	MD(κ_2)	APC	MAOP	MD(κ_1)	MD(κ_2)	APC	MAOP	MD(κ_1)	MD(κ_2)	APC	MAOP
5%	10	5.36	5.38	-1.20	1.52	5.28	5.30	-2.25	-0.36	11.45	11.36	12.56	-3.09
	50	5.24	5.11	-1.20	1.49	5.28	5.33	-2.04	-0.68	10.95	10.94	11.92	-3.44
	100	5.16	5.20	-1.20	1.47	5.02	5.03	-1.85	-0.63	11.18	11.23	12.33	-3.37
	10000	4.46	4.38	-1.20	1.13	4.62	4.61	-1.80	-0.78	10.92	10.96	11.92	-3.51
20%	10	4.70	4.65	-1.20	0.38	4.62	4.68	-1.20	-1.47	8.35	8.41	10.70	-5.69
	50	4.58	4.71	-1.20	0.28	4.62	4.66	-1.20	-1.72	7.59	7.56	10.64	-6.13
	100	4.50	4.46	-1.20	0.28	4.36	4.40	-1.20	-1.68	8.09	8.11	10.55	-6.04
	10000	3.80	3.71	-1.20	-0.12	3.96	3.91	-1.20	-1.86	7.77	7.73	10.31	-6.21
50%	10	3.38	3.39	-1.20	-1.90	3.30	3.29	-1.20	-3.69	0.24	0.17	-6.0	-11.78
	50	3.26	3.25	-1.20	-2.15	3.30	3.34	-1.20	-3.80	-0.81	-0.80	-6.0	-12.40
	100	3.18	3.16	-1.20	-2.12	3.04	3.06	-1.20	-3.79	-1.42	-1.39	-6.0	-12.27
	10000	2.48	2.52	-1.20	-2.61	2.64	2.62	-1.20	-4.01	-1.18	-1.26	-6.0	-12.51

Table 3: Average Number of Messages in Small Domains (α=10, κ=1.0)

	1×5 Grid			2×3 Grid			Dec-Tiger		
σ	MODERN	APC	MAOP	MODERN	APC	MAOP	MODERN	APC	MAOP
5%	1.95	0.05	1.72	1.99	0.53	1.68	3.22	1.86	0.87
20%	1.87	0.05	1.72	1.90	0.03	1.68	2.44	1.86	0.87
50%	1.80	0.03	1.72	1.81	0.02	1.68	2.38	0.0	0.87

In Table 2, σ in column 1 displays the different communication cost and α in column 2 represents the level of model error. Columns 3–6 display the average reward achieved by each algorithm in the 1×5 grid domain. Columns 7–10 show the results in the 2×3 grid domain. Columns 11–14 are for the multi-agent tiger domain. For the markup function in MODERN (MD in Table 2), κ_1=1.0 and κ_2=1.25 were used. We performed experiments with a belief bound of 10 nodes per time-step for our algorithm.

Table 2 shows that MODERN (columns 3–4, and 7–8) significantly outperformed APC (columns 5 and 9) and MAOP (columns 6 and 10) in the grid domains that have sparse interactions. MODERN received statistically significant improvements (via t-tests, p<0.01), relative to other algorithms.

In the highly-coupled tiger domain, APC (column 13) had slightly higher reward than MODERN (columns 11–12) when communication cost was low (5%, rows 3–6) or medium (20%, rows 7–10), but the difference was only about 10% in reward. However, when communication cost was high (50%, rows 11–14), MODERN outperformed APC. In particular, even at this high communication cost, MODERN selectively utilized communication to successfully perform a joint task, and thus it achieved higher reward. MAOP (column 14) showed the worst results regardless of α and σ. We discuss more in detail below.

Table 2 also shows that as communication cost increases, the reward obtained by all three algorithms decreases since agents have to pay more to avoid miscoordination or face higher chance of miscoordination with less communication, both leading to lower solution quality. For instance, when model uncertainty was high (i.e., α=10), the average reward decreased as communication cost increased from 5% (row 3) to 50% (row 11). The average reward of MAOP is directly affected by communication cost since it does not consider this communication cost during reasoning. As shown in Table 3, MAOP triggered the same number of communications regardless of communication cost.

Another trend in Table 2 is that the solution quality generally increases with a lower α. For example, when communication cost was low (5%), as model uncertainty increased (i.e., α decreased from 10000 (row 6) to 10 (row 3)), the average reward increased. Inclusion of model uncertainty into communication reasoning allows for better use of communication to reduce miscoordination. Thus, when model uncertainty increases, execution-centric algorithms get better results. Specifically, when model uncertainty is high, the true transition and observation probabilities in the world have larger differences from the values in the given model; more model uncertainty means more models with higher probabilities of success or failure. When the true probabilities are lower than given model values, agents recover in models with high failure probability using communication and thus avoid huge penalties. When the true values are higher, agents exploit gains in models with high success probability by successfully performing joint actions leading to a higher solution quality. This results in higher average reward.

For instance, consider a 1×5 grid world with two agents. An agent continues to move as long as it believes that it is not at the target location and performs the joint action when it believes that it is at the target location. There are the following four possibilities: (1) both agents believe that they are at the target location, are at the target location, and they thus successfully perform the joint action and achieve a reward of +9.2; (2) both agents do not believe that they are at the target location, and they thus do not attempt the joint action and achieve a reward of -1.2; (3) both agents believe that they are at the target location but one or both of them are not at the target location, and they thus fail to perform the joint action and achieve a reward of -2.8; and (4) exactly one agent believes that it is at the target location, and it thus attempts the joint action alone and achieve a reward of -6.0. In this case, more model uncertainty results in more models with higher probabilities of success or failure (i.e., not case #2). MODERN, by using a better strategy for reasoning about communication at execution time, improves the probability of achieving success (case #1) and exploits gains from them. Additionally, communication decreases the likelihood that the agents fall into cases #3 and #4, which have large negative rewards, and instead fall into case #2, which has a smaller negative reward by exchanging the observation history at the right moment. On the other hand, when model uncertainty is low, it will have lower occurrences of cases #1, #3, or #4 than higher model uncertainty. Thus, the benefits obtained by either higher success probabilities (case #1) or by

avoiding miscoordination by execution-time communication (case #3 or #4) decrease. As a result, MODERN can lead to a higher average team reward with higher model uncertainty.

By exploiting better use of communication, MODERN is more robust to model uncertainty than other methods. In Table 2, MODERN's gain with more model uncertainty is higher than gains of MAOP and APC in most cases. For instance, when communication cost was small (σ=5%) in the 1×5 grid domain, MODERN gained 0.9–1.0 reward with model uncertainty (columns 3–4) while MAOP only gained 0.39 (column 6). APC showed no reward gain with model uncertainty as shown in the table (column 5).

In these small domains, the average reward in MODERN was similar regardless of the markup rate (columns 3–4, 7–8, and 11–12). *Indeed, without carefully tuning κ (e.g., κ=1.0), MODERN's rewards were still statistically significantly higher than others.*

The results in Table 2 show that even though MODERN uses exponentially smaller models, it achieves significantly higher reward, and Table 3 reveals why. Rows show different communication cost and columns display the number of messages for each algorithm in different domains. For example, in the 1×5 grid domain, MODERN (column 2) used more communications than other algorithms (columns 3 and 4). Comparing MODERN and APC, MODERN communicates significantly more messages, paying the high communication cost but achieving higher returns. However, APC very rarely communicates, relying on its full *GrowTree* to perform a coordinated joint action. MODERN communicates more because it anticipates higher expected reward after performing a *sync*; with more conservative communications APC fails to anticipate them. As a result, agents in APC are fully coordinated, but are only able to perform the least harmful joint action (e.g., move). On the other hand, agents in MODERN can realize that they are at the joint location via communication and then achieve higher reward by performing the joint task.

6.3 Comparison: Runtime

Here, we compare the average (execution) runtime per agent of the algorithms. MODERN used 10 belief nodes for the bounded pruning (for small domains, this limit was never reached). Communication cost was 5% of the expected utility. The maximum runtime per trial was set to 1,800 seconds. As shown in Table 4, all algorithms showed similar results in the tiger domain. In small 1×5 and 2×3 grid domains, MODERN and APC took similar amounts of time. The runtime of MAOP was 1.39–1.89 times that of MODERN's runtime in both domains, where this difference was statistically significant (via t-tests, p<0.01). We also tested the algorithms in a scaled-up 2×3 grid domain with longer time horizon (T=5). MAOP was not able to finish running within the time limit. APC uses a particle filtering technique to improve speed, but even with only one particle, APC took about more than half hour to finish a trial, exceeding the time limit, whereas MODERN took less than 125 seconds.

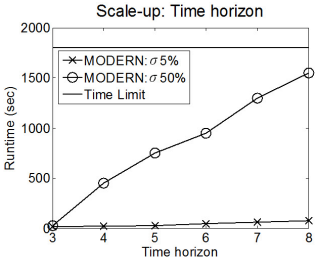
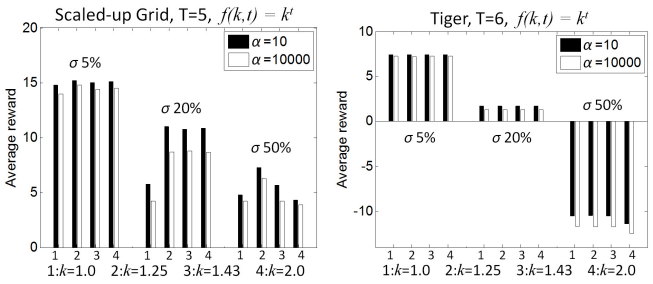


Figure 3: Scale-up: T

We then ran experiments in the larger grid domain with increased time horizons. Figure 3 shows the runtime on the y-axis and the time horizon on the x-axis. We tested the algorithm under two different communication costs: 5% (low) and 50% (high). MAOP and APC (with 1 particle) could not solve the problem within the given time limit for even the shortest time horizon — while MODERN took significantly less time than other algorithms. As the time horizon increased, MODERN obtained higher rewards (from 9.8 to 15.7), since there was more time for agents to recover from any failed actions. With σ=50%, MODERN took more time than with σ=5%, although still scaling linearly with time horizon.



(a) Scaled-up Grid domain (b) Dec-Tiger domain

Figure 4: Benefit of the markup function: three communication costs are tested per domain.

Table 4: Runtime Comparison in Different Domains (sec)

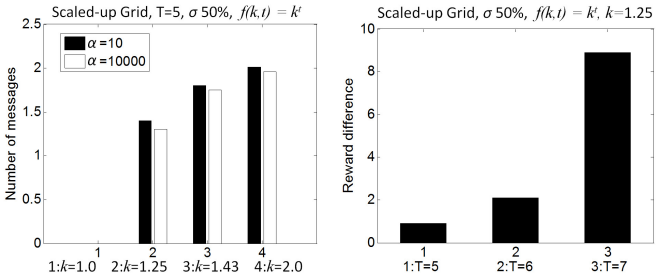
	MODERN	APC	MAOP
Dec-Tiger (T=3)	0.0012	0.0018	0.0014
1×5 Grid (T=3)	12.02	11.68	22.76
2×3 Grid (T=3)	20.02	21.54	27.77
Scaled-up 2×3 Grid (T=5)	124.14	-	-

6.4 Evaluating the Impact of Markup

We now show how the markup function, $f(\kappa, t)$, impacts the performance with respect to the markup rate, κ. This allows us to provide guidelines on selecting an appropriate κ. To that end, we answer the following three questions, which explain important properties of $f(\kappa, t)$.

First, does the markup function help? In this set of experiments, time horizon was set to 5 in the grid domain and 6 in the tiger domain. $f(\kappa, t) = \kappa^t$ was used for the markup function. In Figure 4, κ=1.0 represents the result without the markup function. As shown in the figure, when communication cost was low (5%), the markup function did not impact the performance in both domains. Thus, even though we varied κ value (x-axis in Figure 4), the solution quality (y-axis) was roughly same. However, as communication cost increased ($\geq$ 20%), the grid domain started showing noticeable differences according to κ (Figure 4(a)). Specifically, there was up to two-fold improvement in expected reward with the markup function when communication cost was 20% in the grid domain, and the markup function improved the performance by about 50% when communication cost was 50%. The solution quality in the tiger domain was not affected by the markup function — the benefit by the mark up function in reward was less than 1% (Figure 4(b)). Thus, when communication cost is high, if the frequency of trigger points is low (i.e., a sparse-interaction domain as in the grid domain), there is significant benefit to marking up as compared to the results in a tightly coupled domain that has high frequency of trigger points.

Second, what markup value, κ, would be reasonable? As shown in Figure 4(a), when κ=1.25, MODERN performed best. This occurs because the markup function with different κ values results in different number of messages (y-axis in Figure 5(a)). In other words, when κ=1.25, it caused the right amount of communication for coordination. *Note that 1.0<κ<1.5 other than 1.25 does not de-*



(a) Number of messages (b) Scale-up: T

Figure 5: Effects by different markup functions

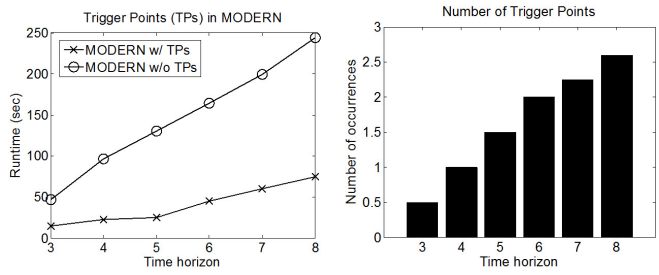
grade the overall performance, and even if κ is randomly selected between 1.0 and 1.5, it outperforms existing methods.

To justify the guideline in Table 1, we analyze the results shown in Figure 4. Figure 4 shows that marking up did not impact the overall performance when communication cost was low (5%), regardless of the domain properties. However, in the grid domain (Figure 4(a)), when communication cost increased ($\geq$ 20%), the markup function with 1.0<κ<1.5 improved the overall performance. κ can increase up to 2.0 when σ is medium (20%), but overestimating ($\kappa \geq$2.0) can degrade the performance when σ is not low ($\geq$ 20%). In the tiger domain (Figure 4(b)), when σ was medium (20%), the markup function did not impact the overall performance. When σ was high (50%) and 1.0<κ<1.5, the markup function did not impact the performance, but when $\kappa \geq$1.5, it caused slight degradation in total reward. These experimental results support our heuristic selection of κ in Table 1.

Lastly, we now ask how the markup function works as we scale up the time horizon. This set of experiments were tested at σ=50% with κ=1.25 and the exponential markup function. As shown in Figure 5(b), average reward difference with and without the markup function (y-axis) increased as time horizon was scaled up (x-axis). This is because when communication cost is high, there are more chances not to communicate for longer time steps as the time horizon increases, which increases uncertainty among team members.

6.5 Evaluating Trigger Points

Figure 6(a) shows the runtime of MODERN with and without selective reasoning: the x-axis is the time horizon and the y-axis is runtime in seconds. As shown in the result, MODERN can speedup runtime by over 300% using trigger points. In particular, the average number of trigger points for T=8 was about 2.6 (see Figure 6(b)). This means MODERN only reasons about communication for about 1/3 of the total time steps, which leads to roughly three-fold improvement in runtime.



(a) (b)

Figure 6: Runtime improvement by selective reasoning

7. CONCLUSION

This paper aims to open a new area of research for DEC-POMDPs: in many real-world domains, we will not have a perfect model of the world, and hence DEC-POMDPs must address such model uncertainty. To combat such model uncertainty, we presented a new framework called MODERN that simplifies DEC-POMDP planning (significantly reducing its complexity) and instead relies on agents' execution-time reasoning. There are three major new ideas in MODERN's execution time reasoning. MODERN: (i) avoids excessive reliance on a complete model by maintaining an approximate model of other agents by bounded pruning, resulting in exponentially smaller beliefs, (ii) reduces computational burden by exploiting BDI teamwork and sparse interactions between agents to limit reasoning about communication, and (iii) marks up the expected gain in utility to reduce uncertainty among team members by boosting communication. We justified our design decisions in MODERN through an empirical evaluation that considers several factors including communication costs and markup rates in different domains. We showed that MODERN can provide solutions much faster than existing algorithms while achieving significantly superior solution quality.

8. REFERENCES

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

[2] C. V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *AAMAS*, 2003.

[3] B. Grosz and S. Kraus. Collaborative plans for complex group actions. *Artificial lntelligence*, 86:269–358, 1996.

[4] R. Jaulmes, J. Pineau, and D. Precup. A formal framework for robot learning and control under model uncertainty. In *ICRA*, 2007.

[5] L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[6] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. In *AAAI*, 1990.

[7] R. Nair, M. Yokoo, M. Roth, and M. Tambe. Communication for improving policy computation in distributed POMDPs. In *AAMAS*, 2004.

[8] D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *JAIR*, 16:389–423, 2002.

[9] M. Roth, R. Simmons, and M. Veloso. Reasoning about joint beliefs for execution-time communication decisions. In *AAMAS*, 2005.

[10] S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. In *AAMAS*, 2008.

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

[12] S. A. Williamson, E. H. Gerding, and N. R. Jennings. Reward shaping for valuing communications during multi-agent coordination. In *AAMAS*, 2009.

[13] F. Wu, S. Zilberstein, and X. Chen. Multi-agent online planning with communication. In *ICAPS*, 2009.

[14] P. Xuan and V. Lesser. Multi-agent policies: from centralized ones to decentralized ones. In *AAMAS*, 2002.