Decomposing Centralized Multiagent Policies

Ping Xuan and Victor Lesser

CMPSCI Technical Report 2007-52
October 30, 2007

Computer Science Department
140 Governors Drive
University of Massachusetts
Amherst, MA 01003-9264

Keywords: Multiagent system, communication, planning

Decomposing Centralized Multiagent Policies

Ping Xuan
Department of Math and Computer Science
Clark University
950 Main Street
Worcester, MA 01610
pxuan@clarku.edu

Victor Lesser
Department of Computer Science
University of Massachusetts at Amherst
Amherst, MA 01003
lesser@cs.umass.edu

Abstract

Using or extending Markov decision processes (MDPs) or partially observable Markov decision processes (POMDPs) to model multiagent decision problems has become an important trend. Generally speaking, there are two types of models: centralized ones and decentralized. The centralized ones focus on finding the best *joint action* given any *global state*, while the decentralied ones try to find out that for *each agent*, what is the best *local action* given all the *partial information* available in that agent. Although decentralized models better capture the nature of decentralization in multiagent systems, they are much harder to solve compared to centralized models. In this paper, we show that, by studying the communication needs of the centralized models, we can establish a connection between the two models, and the solutions to centralized models (i.e. *centralized policies*) can be used to derive solutions to decentralized models (i.e. *decentralized policies*) – a process we call *plan decomposition*. We show that the amount of communication needed could be greatly reduced during the decomposition, and there are techniques that could be applied to produce a set of decentralized policies based on the same centralized policy. While this method does not solve decentralized models optimally, it does offer a great deal of flexibility and allows us to tradeoff the quality of the policies with the amount of communication needed, and gives us better insights about the need and timing for effective coordination in multiagent planning.

1 Introduction

The problem of formulating plans is a key problem in multiagent collaboration. Proper coordination among decentralized agents means that the agents' plans must be compatible with each other. Several authors view distributed AI as a distributed goal search problem, including Lesser [17], Durfee and Montgomery [9], etc. The planning problem can be visualized by a distributed goal search tree, as illustrated in the work of Lesser [17] and Jennings[15]. The global search space is then divided into several local search spaces, with each agent working on its own local goals. The difficulty with this approach is the sub-goal interaction problem: different agents' plans may be interrelated due to the relationships among the planned tasks. Naturally, most of the research work on multiagent planning focus on how to produce a globally consistent plan that resolve the interrelationships. However, the issue of how to dynamically adapt such a global plan across distributed agents has not been adequately addressed. For example, when agent actions have non-deterministic outcomes and each agent only observe parts of the system state, the agents must implement mechanisms to make sure that they choose the correct subsequent actions as specified in the plan. In this paper we extend our earlier results [26] on the transformation of centralized multiagent plans into decentralized policies. We formalize this *plan decomposition* problem and propose a methodology for solving it. Furthermore, we show that solving the plan decomposition problem will help us solve the decentralized planning problem in return.

To date, numerous multiagent planning approaches have been proposed. While their approaches differ widely, in decision theoretic perspective they can be categorized in two classes: one that uses a *centralized view*, and one that uses a *decentralized view*. Typically, the centralized view is the view taken by the designer of the system, which incorporates the observations of each agent. The behavior of the agents are often considered collectively, in terms of joint intentions [18] and joint actions. There, a plan, which specifies the problem solving strategy of the agents, is often described through a mapping from the global system states (as observed by the collection of the agents) to the joint actions of the agents, which is in the form of a centralized policy (CP). The decision theoretic model for solving CPs is proposed by Boutilier [5]. In contrast, the decentralized view is often the view of a situated agent in the system, which deals with a local observation of the global system state (i.e., local state) and makes local decisions. There, a plan – the decentralized policy (DP) – is often a mapping from local beliefs to local actions. A decision theoretic model for solving DPs was initially proposed by Xuan *et al* [27], where the decentralized multiagent MDP is defined and the formal definition of a DP is given. In this framework communication among the agents are also explicitly modeled via communication actions and communication decisions. Since this work, similar or generalized frameworks (include extensions for POMDPs) for decentralized multiagent decision processes include the DEC-MDP/POMDP framework by Bernstein *et al*[2], the Communicative Multiagent Team Decision Problem (COM-MTDP) by Pynadath and Tambe [23], and the DEC-POMDP with communication (Dec-POMDP-Com) by Goldman and Zilberstein [12] have been developed. In these decentralized views, each agent has its own local decision policy based on the information available in each agent (directly observed or communicated), and decides only the local actions (including communication actions) for this agent, unlike the centralized view where one CP specifies the joint actions of the agents.

DPs are the preferred, readily implementable form of plans for decentralized agents, but unfortunately it is computationally infeasible to obtain optimal DPs when there is no restrictions on their structure [4, 2] (the complexity is NEXP, while solving MDP/POMDP - the key part in obtaining the optimal CP in a centralized multiagent MDP - is P or PSPACE complexity), and furthermore it is very hard to to generate approximate solutions in general [24], although recent work shows that certain subclass(es) of the decentralized multiagent MDP may have exact or approximated solutions, or heuristics [1, 6, 20, 3]. In contrast, CPs are computationally easier to solve and there are well-known algorithms for obtaining approximate solutions. However, a CP is not directly implementable in decentralized agents because it does not specify the interaction of the agents when they face uncertainty caused by no or limited communication. Thus, we need a way of translating CPs into DPs, and which we will call the plan decomposition problem. In decision theoretic terms, this problem amounts to the following: suppose that (1) at any time, the status of the overall system can be captured by a global state, and (2) we have a plan that specifies the joint action the agents should take for every possible global state during the plan execution (this is called a centralized policy (CP in short), which is really a mapping from global states to joint actions), but (3) each agent only sees its local view – which contains partial information about the global state, how to let each agent know what action it should perform according to the centralized policy? Once we know the answer to this, we now have a decentralized policy (DP in short): each agent is now capable of locally deciding what action (include communication action and regular local action) to take given the information available to this agent (include agent direct observations and the information obtained through communications.) A decentralized policy is simply a mapping from local information to local action or communication.

Typically, one CP may lead to many possible DPs that can implement it in decentralized agents, but they differ in the amount of communication used to synchronize the agents. Also, by modifying the CP we may be able to make tradeoffs between the plan quality and the amount of communication in the resulting DPs. The latter issue of optimizing information exchange is also present when solving decentralized multiagent MDP/POMDPs, for example in the work of Goldman and Zilberstein [12] and the work by Shen *et al*[25], where some approximate algorithms are proposed. Thus, our approach may indeed provide insights for the same problem from a different viewpoint, although in our approach the resulting DPs may not be optimal.

This plan decomposition problem is related to some earlier work on multiagent planning that deals with the management of decentralized plans. For example, one can envision that in some systems there exists a manager agent which has (or is able to obtain) complete knowledge about the system and thus it is able to generate a CP that specifies joint actions for all agents. For example, in the work on contract nets [7], in some sense, the manager agent can be viewed as having the centralized plan established after the bidding process is finished: only the selected bidder has the task in its schedule (strictly speaking, the bidders are autonomous rather than being controlled by the manager agent - it is really the binding nature of the contracts that makes the system appear cooperative or controlled). In this case the plan decomposition is trivial because the contract net protocol also specifies who needs to communicate and what is to be communicated regarding the tasks – in case the selected bidder fails, the manager agent will begin another round of contracting. Fujita and Lesser [10] have generalized this centralized task distribution problem to complex tasks with uncertain outcomes and timing constraints, in which one agent is responsible for assigning subtasks and their execution order to a set of (slave) agents. Due to the uncertain outcomes (such as failures) of agent tasks, the agents may find the subplans undesirable and thus the master agent would often need to revise the central plan. In their work they study the issue of communication (as the transmission of rescheduling requests based on the slave agents' assessments on how the subplan's progress has deviated from the intended plan) and the rescheduling efficiency. Specifically, there exists a phase transition phenomena - frequent rescheduling was ineffective for both very easy and very difficult problems, but effective in the transition region (mid-level) of difficulty. Obviously, the plan decomposition problem we are addressing here is different from the centralized task distribution since there is no such master agent, and communication is among the peer agents. The assumption here is that each agent has the CP pre-calculated or stored, but they need to find out their own part of actions (subplan) from the CP. Because of the uncertainty associated with the tasks, they need to communicate with each other to be aware of their current situations in order to coordinate effectively with others as implied in the centralized plan.

An obvious solution is to simply let the agents synchronize themselves at each step (when the global state changes), thus piecing together the actual global state via communication. Once this is done, each agent just looks up the policy, find the joint action specified there, and choose the component that corresponds to the agent's local action. But this would obviously require a lot of communication. A natural question is, can we use less communication? And when is communication necessary? Also, if communication carries a cost, how should we choose the communication strategy so that the cost of communication is justified? In this paper, we provide the answers to these questions: that we can reduce a great deal of the communication, that communication is necessary when there is ambiguity in the system regarding which action an agent should pick, and that there are tradeoffs strategies that can help us balance the communication cost with the expected gain. We will show the methodology for solving this plan decomposition problem and discuss how the understanding of this problem can help us in the design of multiagent plans.

An important feature of our plan decomposition methodology is that, for a given CP, there are a spectrum of possible DPs. This can be achieved by varying the combination of transformation mechanisms to be applied during the process. In turn, by looking at the effectiveness of the various mechanisms, we can better understand how decomposable the problem is, and this give us better intuition about how to choose the right kind of coordinate strategies for different problems.

The information barrier due to the decentralized nature of a multiagent system is a problem that is central to multiagent research. In many studies, for example in [19] (Markov games), a stochastic model of other agents are introduced. In such models, the current states can be represented via a probabilistic distribution of possible global states. However, it should be noted that such approach effectively assumes that the other agents can be regarded as not adaptive in nature - their behavior is regarded as a random process independent of other agents' strategies. In reality, both collaborative agents and adversarial agents would be highly adaptive, thus such models are approximate at best. Another attempt to directly model other agents in the presence of the information barrier is to reason about what are the beliefs that other agents have about itself. This would lead to rather complex recursive modelings [11] because the other agents' beliefs about this agent in turn depend on this agent's belief about them. In comparison, our approach does not attempt to directly model other agents' belief, instead, it relies on the common ground among the agents: in our case, the common set of information that can be established in an "imaginary" monitoring agent. The introduction of the "imaginary" agent helps the understanding of how the beliefs in the agents are affected by communication, which is eavesdropped by the imaginary monitoring agent. This way, we do not have recursive modeling nor maintain probabilistic models for other agents. It should be noted that it is still possible to associate our common belief state with a probabilistic distribution, but such information is not necessary for the computations discussed in this paper.

We need to note that the term communication we are using here does not mean the transmission of domain-level information such as the necessary input/output data, as in the work of Shen et al [25]. Rather, we are concerned about the transmission of meta-level information - the kind of information that concerns the plan itself, such as the outcome status of the agent actions (success, failure, utility values, agent state, etc.) In other words, communication is part of the situation assessment decision/action: to gather the information needed to relate the local observations to the actual global system state. The communication decision is thus the assessment of whether such information exchange is necessary. Also, we are not concerned about the actual details of the message protocols but rather focus on whether the information exchange has occurred or not. For simplicity, we focus on a special kind of communication – where the agents synchronize there local findings and converge at the consensus: the actual global state. In our approach we define a structure called the *common belief states* in each agent to represent the common ground among the agents' different views about which state is the current global state. In particular, it allows us to reason about which global states may be possible at a particular time point in problem solving which is key to our approach for transforming a CP to DP. The local observations of the agent (i.e. the private information before communication) can be seen as the additional information that enables the agent to further narrow down the candidate states and form its *local belief states*. Common belief state is identical across the agents, while local beliefs may vary from agent to agent, because each agent observes a different part of the global state. We show that a communication strategy can be defined via this common belief state, and based on the communication strategy, an agent can update its local belief states after the completion of local actions or after communication. Also, once the communication strategy is known to all agents, even no communication implies some information: basically, the states that would have triggered communication can be ruled out. This is the basic technique for reducing communication.

To study the necessity of communication, we note that in episodic (i.e. one step) decision making, information may have value only if the information would cause the agent to alter its decision and choose a different action. Likewise, if the agent would choose the same action regardless whether or not it receives the information, communication is not necessary at this step. Thus, given the consensus common belief state and local outcomes, if all the possible global state lead to the same joint action, then there is no need to communicate. Otherwise, there are some global states that require different joint actions, therefore it is important for the agents to use communication to make sure they do not mix them up and thus deviate from the given CP. Thus, the purpose of communication can be viewed as a way to avoid ambiguity.

A greedy approach thus can be employed to reduce the communication: to communicate only when there is ambiguity. We will show that there are several transformations to be used in this greedy approach. When the communication costs is really high, we can also show that if we are allowed to slightly deviate from the original CP, there could be other transformations (called *nonconforming transformations* that could significantly further reduce the amount of communication. This way, a tradeoff between the amount of communication and the deviation from the original CP could be sought.

In the following, we first formally layout the decomposition problem, followed by the transformation methodology, including the the choice of communication policies and nonconforming transformations. We then discuss some experimental results and summarize the major results of the paper.

2 CP vs. DP: a Decision-Theoretic Perspective

To study the plan decomposition problem we need to introduce the multiagent MDP framework to formalize this problem. Let us start with some examples of multiagent cooperation problems. First, let us look at a grid world problem. Suppose two agents, X and Y, try to meet each other in a 4×4 grid. (Our framework applies to n-agent systems, with only notational differences. For clarity we use examples with only 2 agents.) They start from some starting position, as illustrated in Figure 1. In each step, an agent can choose a direction to move, or stay where it is. However, a move may have nondeterministic outcomes: there is probability q (called the success rate) that it moves to the neighbor cell in the direction of the move, $(1 - q)/4$ chance resulting in any of the other neighbor cells, and the rest of the time it does not move (i.e., gets stuck in the current cell). The goal is for the agents to meet each other (to be in a same grid cell) before a certain time (the deadline) to receive a global reward.

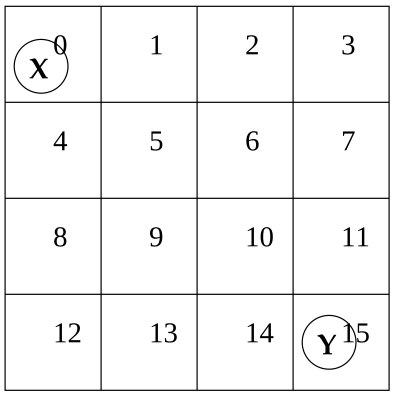

Figure 1: A Grid World Meeting Problem

The assumption here is that both agents know their own positions in the grid, but they do not observe the current position of the other agent unless they communicate. Furthermore, communication may incur a cost, and the agents would like to minimize the amount of communication in addition to maximizing the expected utility. As such, the agents would need to decide whether to communicate or not in addition to the decision of which action to take.

Let us use the notation (s_x, s_y) to represent the agent positions, for example $(0, 15)$ is the starting position for the two agents. Also, we use $(a_x|a_y)$ to represent a joint action, for example $(\rightarrow \mid \uparrow)$ means for X to move to the right and for Y to move up in the grid. Here we use $\leftarrow$, $\rightarrow$, $\uparrow$, $\downarrow$, and $\odot$ to represent the agent action of moving to the left, moving to the right, moving up, moving down, and staying where it is. First, we introduce the centralized policy used in the decentralization process: if the position of the agents is (s_x, s_y), we first compute the goal state $(\hat{s})$ for both agents by choosing the grid position closest to the straight-line mid-point between s_x and s_y (in case of multiple choices, just choose the rightmost grid position). For example, position 6 is the goal position for the starting state (0,15). Then, we choose each agent's local action that maximally reduces the straight-line distance between the agent's current position and the goal position (in case of multiple choices, choose the vertical move). For example, moving to the right $\rightarrow$ is the action that reduces the straight-line distance between 0 and 6 most. Similarly, $\uparrow$ is the action if the agent is at 15 and the goal position is 6. The following table shows a partial list of the CP:

(s_x,s_y)	joint action
(0,15)	$(\rightarrow \mid \uparrow)$
(1,11)	$(\downarrow \mid \uparrow)$
(5,7)	$(\rightarrow \mid \leftarrow)$
(0,14)	$(\downarrow \mid \uparrow)$

Thus, if there is no uncertainty, i.e., the *success rate* is 1, the intended path for the agents to meet is the following:

$$(0, 15) \longrightarrow (1, 11) \longrightarrow (5, 7) \longrightarrow (6, 6).$$

For decentralized policies, the same path would be guaranteed if the success rate is 1, and no communication needed due to no uncertainty - even though none of the agents see each other's position. However, when the success rate is less then 1, the agents may deviate from this intended path and therefore may need to communicate - share each other's position - in order to follow the exact same centralized policy. The goal position can be viewed as a mutual commitment between the agents, and the change of goal positions can be viewed as establishing new commitments (and decommitting the previous ones), as discussed in [27, 12]. Specifically, if the success rate is less than 1, then when the first step is finished (with the joint action $(\rightarrow \mid \uparrow)$), there might be 9 possible resulting global states (s_x could be 0, 1, or 4, and s_y could be 15, 14, or 11) according to the centralized policy above. Thus, for some episodes the next global state may be (0,15) again - thus the next joint action is $(\rightarrow \mid \uparrow)$ according to the CP, and for some other episodes the next global state may be (0,14), with the next joint action to be $(\downarrow \mid \uparrow)$ according to the CP. Thus, without knowing Y's actual position, even if X knows its next position is 0, it still cannot decide if it should take $\rightarrow$ or $\downarrow$ action. This means that the centralized policy is not directly implementable in decentralized agents, as there is the danger of deviating from the original policy if the wrong action is chosen. However, if X chooses to communicate when it notices that its position is 0 after its action, and thus let both agents know each other's current position (i.e., synchronize), the agents can then continue to follow the CP. Interestingly, if this strategy is known to both agents, then if Y does not see communication from X, Y can infer that X does not have outcome 0. Thus, no communication actually carries information (if the other agent's communication strategy is known)!

Another example of multiagent cooperation often deals with planning and scheduling of hierarchical task networks. Figure 2 shows a planning problem involving two agents selecting and sequencing their tasks in order to achieve a certain joint goal expressed in the TAEMS framework [8]. The tasks are organized in a hierarchical structure according to how the utilities of low-level tasks translate into the utilities of high-level tasks, and there are interrelationships among the tasks and also other constraints such as deadlines. The goal is for the agents to plan their actions so as to maximize the expected total utility.

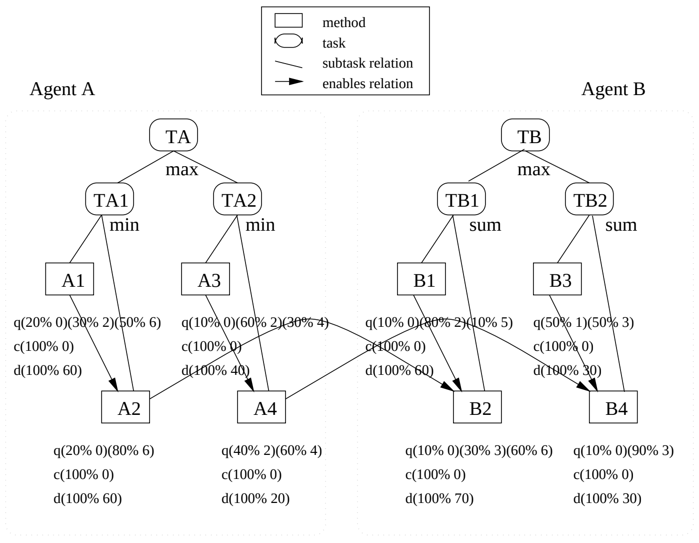

Figure 2: A Multiagent Task Planning Problem

Typically, heuristic schedulers are used to find solution plans as solving it optimally is usually computationally intractable. Some schedulers specify the solution in terms of joint actions sequences, which are very much like centralized policies; while others may specify local schedules for each agent. But a schedule alone does not fully specify what has to be done in each agent. There must also be coordination mechanisms that specify how schedules in different agents needs to be collated to handle things like synchronization points or mutexes, and there must also be mechanisms to handle uncertainty. For example, rescheduling may be needed if the failure of one task renders the schedule obsolete, such as in the example illustrated in Figure 2: when agent A's original plan is to perform action A1 and then A2, but task A1 fails to enable task A2. In this case, A has to reschedule, but agent B would also need to be aware of the change in order to change its schedule. Communication is thus a necessary part of coordination.

Just like the previous problem, here an agent's action may have nondeterministic outcomes, and such outcomes are not automatically observed by other agents. This introduces uncertainty in one agent's view of the system, which in turn affects the decision-making of the agents, and as a result affects the coordination between the agents. Communication may resolve the uncertainty, but in many cases it is not possible to communicate all the time. Furthermore, communication may carry a cost. Sometimes, too much communication regarding not needed information may become a distraction to the agent planning process. Thus, during agent planning, whether or not it is necessary to obtain the information via communication becomes a decision the agents have to make.

In each of the two examples, there is a difference between what the agent observes (i.e., the local state) and the actual global state of the system. Most of the research work on multiagent planning have been focusing on how to generate a multiagent plan that describes how the global system state should evolve over time. Such a plan would specify the action of each agent given the current system state. However, when the global system state is not automatically observed by the agents, the plan cannot be implemented: an agent does not know exactly the current global state, and therefore may have difficulty in deciding which action to perform. If the wrong action is chosen, the agents would deviate from the plan. Through communication, the global system state may be discovered by the agents, and this would lead to the correct choices of actions. However, we will show that the agents do not need to communicate all the time to behave in a coordinated way. In fact, the proper choices of action may be achieved via far less communication.

In decision-theoretic terms, the centralized view of the system can be described by a multiagent Markov decision process model, as proposed in [5]. In the simplest form, this is a standard Markov decision process [22] that consists of a set of global states S, a set of joint actions A (each joint action specifies one action for each agent), a transition probability matrix $Pr(s'|s,a)$ – the probability that the system moves from state s to state s' after joint action a, and a reward function $r(s)$ that specifies the global utility received when the system is in state s. In this framework, a CP is precisely a policy for a Markov decision process. Figure 3 shows how problem solving is carried out under such a policy (in a two-agent system with agents X and Y): at each stage the joint action is chosen according to the CP and is performed, and system moves into one of the possible next states depending on the outcome of the joint action. The expected utility of the CP can be calculated using a standard policy evaluation algorithm for Markov decision processes.

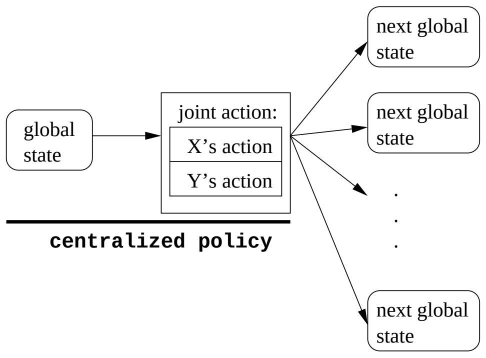

Figure 3: Centralized View and Policy

In the above we assumed that the collective results of observations of all the agents (i.e. the sum of all agents' local findings) identifies the true global system state. In many systems, however, the real system state is not directly observed. Rather, the sum of the agents' observation is a partial observation of the global state. In those cases, a more general model, a partially-observable MDP (POMDP), may be used instead of a standard (fully observable) MDP to describe the decision problem, and the CP is then a mapping from a partial observation to a joint action. The model and methods proposed in this paper applies to both MDP and POMDP models, but for clarity we will adopt the MDP model and hence will not distinguish the difference between the result of collective observation and the global state in the following discussion.

With the decentralized view, an agent cannot see other agents' local states and local actions and their

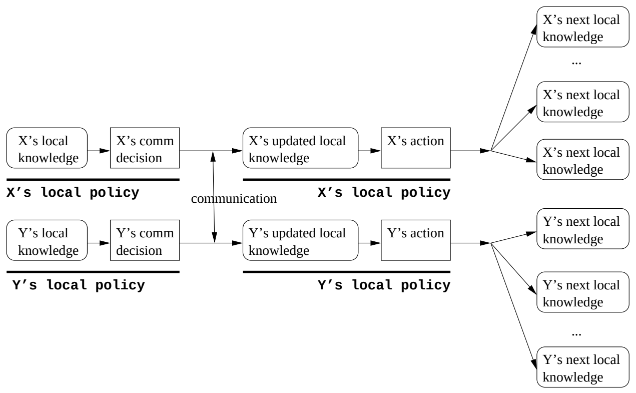

Figure 4: Decentralized View and Policy

outcomes, and has to decide the next local action on its own. Thus, each agent has only a partial, decentralized observation of the system's global state (which, as mentioned above, corresponds to the collective findings of all the agents), and different agents have different partial views/findings. Of course, this does not necessarily mean that the agents' information is limited to its local observations. Rather, an important ability of decentralized agents is their ability to communicate. We view communication as a way of expanding an agent's partial view by exchanging local information not observed by other agents. For simplicity, we define communication as the action of agents updating each other with local state information and thus collectively discovering the current global state (in other words, the agents synchronize themselves so that they all observe the current global state). Clearly, a DP has to deal with communication explicitly, in particular when communication incurs a cost, or when continuous, unlimited, communication is not feasible. In [27], we proposed a decentralized multiagent decision process framework that provides a basis for decision-theoretic study of decentralized policies. The system is modeled by each agent having an individual state space, its own local action set, and local state transition probability measure, but uses a global reward function to connect the effects of the actions in the agents. Thus, it is not a standard Markov decision process. The DP under this framework explicitly includes communication decisions. Specifically, the DP should define not only what local action to take based on the current local knowledge of the agent, but also whether communication is needed after its local action is completed. Typically, such decisions are not made solely based on current local state information but may need to include some of the agent's history information, e.g. some past states and/or past communications. In general, we are dealing with history-dependent policies when it comes to DP, whereas typically we need only deal with history-independent, or Markovian, policies, when studying CPs. A DP thus contains two mappings: from local knowledges to communication decisions (before choosing next action), and from local knowledges to local actions.

Figure 4 shows the computation model under this decentralized view: at any stage t, each agent first observes the outcome of the previous local action (as specified by the DP). Then, it decides (according to the DP) whether there is need for communication when the action finishes (and the local observation is made). Next, the agents perform the communications (if any). When the communication sub-stage finishes, the local knowledge of each agent may be augmented by the communication messages. Each agent then chooses the local action as specified by the DP and executes it. After the action finishes, each agent moves into the next stage with the new local knowledge, which depends on the outcome the action. This two sub-stage process is a natural illustration of a common problem solving procedure: first gather the information needed, and then choose an action. The problem of deciding communication decisions thus relates to the classical *value of information* problem in AI planning: in this case, the cost of information is in the form cost of communication, and the value of information is reflected through the potential gain of the overall utility via a better informed action decision.

Optimally solving a decentralized multiagent decision problem is extremely difficult: as Bernstein *et al* pointed out, solving a decentralized Markov decision process (a decision process without communication) is of NEXP-time complexity[4]. Pynadath and Tambe [23] have recently categorized the complexity for several

$$\mathrm{CP} \longrightarrow \begin{pmatrix} \mathrm{DP}^X \\ \mathrm{DP}^Y \end{pmatrix}$$

Figure 5: Transforming a centralized Policy to decentralized policies: one local policy for each agent

classes of multiagent MDP problems and the optimal solution for general cases are typically NEXP-time. As a result, heuristic approaches and approximation methods for developing DPs are extremely important. However, due to the infancy of research into this question, only a number of simple heuristic approaches have been studied so far [27, 6, 25, 20]. These approaches are often quite domain-specific and therefore cannot be easily extended to derive general solution methodologies. Goldman et al [12] also proved that it is hard to even get approximation solutions to DEC-MDPs that are close to optimal (say within a small ϵ). On the other hand, CPs are easier to solve and there are systematic methods for obtaining them. For example, solving a POMDP with polynomial number of stages is of PSPACE complexity class [21], a lower class on the complexity scale.

The transformation from CP to DP can thus provide a way of systematically generating DPs in addition to making CPs implementable. For a two-agent system containing agents X and Y, this transformation is illustrated in Figure 5. Note that the decentralized policy consists of a separate policy for each agent (what we call *local policies*) - in our case, two local policies, one for agent X (DP^X) and one for agent Y (DP^Y). Together they form a complete DP derived from the CP.

Note that the two substage model and the multiagent MDP framework described here [27] explicitly separates a DP into a action policy and a communication policy, whereas in a typical decentralized POMDP model (such as [2]) there is no such separation - a communication action is treated as a special kind of action (since it involves multiple agents rather than just acts locally). Although theoretically equivalent, the separation model has the advantage of clearly distinguishes local problem solving versus coordinated actions, and also allows DPs to be specified in a more natural form - as a combination of local plans and coordination mechanisms - the form often adopted in heuristic solutions.

3 Transforming a CP into DPs

Now we discuss how to transform a given CP into DPs. As mentioned before, a CP is often described in terms of a policy for a standard Markov decision process (MDP). The policy consists of a mapping from the set of states (global system states) in the MDP to the action (joint agent actions) set, telling the joint action $a(s)$ to be performed at each global state s. We use $a_X(s)$ and $a_Y(s)$ to represent X's and Y's local action part in the joint action, respectively. Also, we use $\mathrm{next}(s)$ to describe the set of possible next states for state s, i.e. $\{s'|Pr(s'|s,a(s)) > 0\}$. Note that in practice, the mapping only needs to cover the *reachable* states. This can reduce the size of a CP considerably.

Although throughout this paper we are using a 2-agent system to describe our approach, it should be noted that the methodology applies the same way to N-agent systems. The definition of synchronization remains the same: all agents would use communication to reach a common consensus: the discovered global state.

To describe a DP, we introduce three concepts: (1) *common belief state*, (2) *local belief set*, and (3) *local history set*. Roughly speaking, an agent's local policy can be described through a mapping between local belief sets to local actions, and a mapping between local history sets to communication decisions. The common belief state, which describes the consensus of the agents, is the key to the computation of local belief sets and local history sets. We shall now define these structures and also show that how to update them during the course of problem solving.

Formally, we can use the notation S^k to represent the possible global states that the system may reach at time k according to the CP, i.e.

$$S^0 = \{s_0\}$$

$$S^{k+1} = \bigcup_{s \in S^k} \text{next}(s).$$

Here s_0 is the starting global state. Clearly, at time k, each agent knows that the current global state s_k must be in S^k, but often it does not know which one. However, it can use local information to rule out some states in S^i that are deemed incompatible with local observation and thus identify a subset of S^k that contains global states that are truly possible. This subset is the local belief set of the agent. An agent can use its current local belief plus newly observed information to compute the local belief at next stage (i.e. time $k+1$). By representing local belief this way, we can show how the DP is represented, and how the computation is done.

For clarity, we assume that each agent has complete knowledge of the CP and also knows the dynamics of the state space. A trivial transformation is to let the agents communicate at all stages, therefore maintaining the observation of the global state all the time in all agents, i.e., establishing a centralized view in each agent. However, our interest is in how to minimize communication. Intuitively, an agent does not need to communicate if it knows precisely what is the next local action it needs to perform. If it knows that the actual global state is one of several possible states *and* all the possible states indicate the same *local* action for this agent (not necessarily the same joint action), then the agent may choose not to communicate. In this case, uncertainty regarding the exact global state will not affect its choice of actions.

As such, each agents' problem solving episode can be characterized as a series of non-communicating intervals. At the beginning of each interval, agents are synchronized and know the global state. The agents then remain silent in the next several stages until one of them (or both) has insufficient knowledge to decide its local decision. At that time, they communicate (initiated by an agent with insufficient information). After communication, they are again synchronized with the global state. We assume that communication is without error or loss.

When synchronized, the agent knows the current global state. Otherwise, the agent does not know the exact state but knows the set of possible states: the *local belief set*, or LB in short. Obviously, each agent's local belief may be different, so we use LB^X, LB^Y for each agent's LB set. At time k, the LB sets are $LB^{X,k}$ and $LB^{Y,k}$, respectively, but for clarity we omit the k in the notations whenever appropriate. Clearly, $LB^{X,k+1}$ is a subset of next($LB^{X,k}$), but the exact nature of $LB^{X,k+1}$ would depend not only on the local action outcome but also on the information exchange between the agents. Assuming that the global system can be determined by combining all local observations in all agents (as in a DEC-MDP), the intersection of the local beliefs in different agents uniquely determines the current global state, i.e. $s^k = LB^{X,k} \cap LB^{Y,k}$. Of course, for DEC-POMDPs, the joint observations would only reveal the current belief state, not the actual state.

Since agents have different LB sets, LB sets cannot be used as the consensus of the agents. Consider an imaginary "monitoring" agent MA who reads all the communication in the system but does not observe the outcome of any local actions. MA knows everything else the agents knows, such as the state space, initial state, the CP, the communication strategy, etc. Suppose the monitoring agent MA would discover the global state when the agents synchronized (i.e., the synchronization messages contain such information). But when there is no message, the monitoring agent's LB set ($LB^{MA,k}$) would be "bigger" than both LB^X and LB^Y because the monitoring agent does not have the local outcome information to rule out some impossible states (similar observations are noted, for example, in the work on diagnosis by Kaminka and Tambe [16]). Since both agents – just like the monitoring agent – read all communication messages, they can both "pretend" as the monitoring agent and independently calculate the LB^{MA} set and use it as the consensus of the agents. In the following discussion, we use the term *common belief state* (or simply *belief state*, B in short) to represent LB^{MA}. Clearly, B is calculated based on common knowledge only, and as such the evolution of B^k (the B set at time k) can be computed at both agents.

As a simple example, let us consider two agents, X and Y, and initially they are perfectly synchronized and both are at the global starting state ($S0$ in Figure 6.) Thus X, Y, as well as MA (the monitoring agent) all have this knowledge. Figure 6 shows the initial LB sets for X and Y, as well the LB set for MA (i.e. the common belief set B) in the rectangles nearing each agent. Suppose that the centralized policy specifies that the joint action is $(x|y)$. Also, we assume that *both* actions may have two possible outcomes, and each

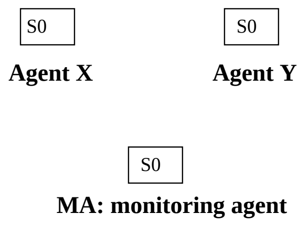

Figure 6: Initial Local Beliefs and Common Belief Set

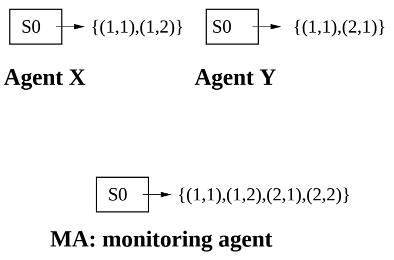

Figure 7: The possible next states, after action $(x|y)$ finishes, but before communication

is labeled as 1 and 2 (for convenience, we will use expression like x=1 to mean x has outcome 1. Thus, the four possible next global states are (1,1), (1,2), (2,1), (2,2) – represented by the joint outcomes of $(x|y)$. Now suppose in this episode, we have the outcomes x=1 and y=1. To agent X, before the communication stage begins, the possible current global states are in the set {(1,1),(1,2)} - because X does not know Y's local outcome, and similarly, Y thinks the possible states set is {(1,1),(2,1)}.

But for the monitoring agent, without hearing any communication between X and Y, and unable to read the agents' local outcomes, can only assume that – the common belief state – consists of all 4 possible states: {(1,1), (1,2), (2,1), (2,2)}. This set, let's call it N, is simply the next states from B^k: $N = \cup_s \text{next}(s)$ for all $s \in B^k$. N is important to us because the calculate of the B set in the next stage (i.e. the computation from B^k to B^{k+1}) depends on it: B^{k+1} (and hence the local belief states at time $k+1$ must be a subset of N. Figure 7 shows how the possible global states set changes in agent X, Y. and MA.

Note that if the monitoring agent knows the communication strategy of the agents (for example, X would communicate if and only if x=1), then it is possible to exclude certain states in the N set when deciding the B^{k+1} in the current episode. This is the key idea which we will explore shortly.

Also note that it is not difficult for both X and Y to internally (and independently) compute the B and N = next(B) sets. As such, both B and N are common knowledge to the agents. Clearly, N is calculated without the need for knowing the *actual* outcome in X or Y: N is decided by the model alone and it shall be the same even if actions x or y can produce more than one outcome. Figure 8 shows the content of N: the states are listed in a matrix where each row corresponds to the same outcome for action x and each column corresponds to the same outcome for action y.

At this point, agent X cannot distinguish the two possible global states in the set {(1,1),(1,2)} based on the local observation x=1. Similarly, if the outcome is x=2 instead of x=1, agent X won't be able to distinguish the two possible global states in the set {(2,1),(2,2)}. That is because that the states in the set share the same state trajectory up to this point, and thus the local information (including past observations and past communications) are the same for both states. We will call sets like {(1,1),(1,2)} and {(2,1),(2,2)} *local history sets* for X, or LH sets, because the states in each LH set share the same local history. In this example, given that the outcome of actions (x=1,y=1), the current LH set for X (LH^X) is {(1,1),(1,2)} and LH^Y is {(1,1),(2,1)}. While LH sets include all the possible states at time $k+1$, it is not equal to the LB sets, because during the communicate stage, the agents may be able exclude certain states in the LH sets based on the communication messages (or the lack of them). The, the LB sets in the agents should be a

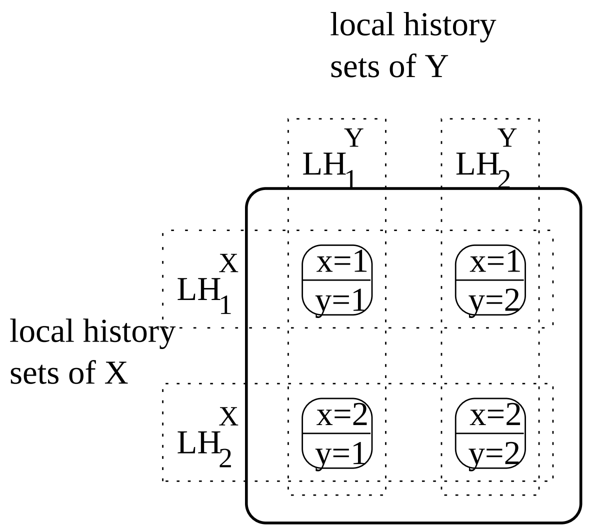

Figure 8: The N set, partitioned into LH sets

subset of the LH sets.

For both X and Y, the local history set can be viewed as the set of states that are the next states of the previous local belief set (which summarizes all previous observations) that are compatible with the most recent local outcome. Formally, this means that, for each possible current state s, after the joint action $a = a(s)$ (according to the CP), X observes the action outcome o_X, and let outcome(s, o_X, X) be the subset of next(s) such that X's local observation is o_X, then

Equation 1:

$$LH^{X,k+1} = \bigcup_{s \in LB^{X,k}} \text{outcome}(s, o_X, X) \tag{1}$$

Similarly we can define LH^Y. However, for the monitoring agent, because it does not observe any local outcome, it is not able to know the actual LB set in either agent and hence do not know the actual LH set. However, since the set B includes all possible LB sets in each of the agents, the monitoring agent can use N = next(B) and partition N into possible LH^X and LH^Y sets.

In this example, the previous LB^X set is {S0} and the next states are simply N. But only (1,1) and (1,2) has outcome x=1. In other words, an LH set can be viewed as a projection of the N set according to the local observation history. Basically, when the local action finishes (but before any communication begins), an agent may narrow down its belief about the set of possible global states by using its action outcome: it can rule out states in N that do not match the current and previous outcome history. This narrowed-down set is the local history set. It is easy to see that the LH sets can also be computed from one stage to the next according to the CP, according to Equation 1, but replacing the LB set with B, and enumerate all possible o_X outcomes. The important property about the LH sets is that the states in the same LH set all have the exact same local outcome histories: they are indistinguishable locally unless external information is received via communication. Thus, the set of possible LH sets partitions the N set based on the local history: each possible LH set uniquely defines a branch of the state trajectory for the local problem solving. For our example, Figure 8 shows the partition of N into two possible LH sets for X: {(1,1),(1,2)} is LH_1^X, where x=1, and {(2,1),(2,2)} is LH_2^X, where x=2 – each set corresponds to one row in the state table in the figure. Similarly the LH^Y sets are the columns: LH_1^Y is {(1,1),(2,1)} (when y=1) and LH_2^Y is {(1,2),(2,2)} (when y=2).

Now that the set N is partitioned and the current LH sets are known, the agents need to decide whether

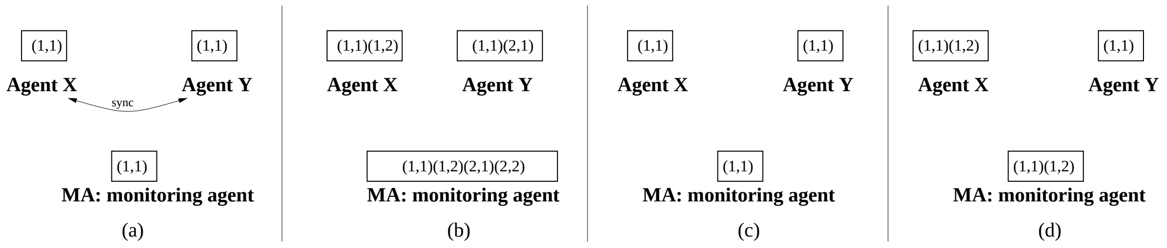

Figure 9: The local belief sets for different communication strategies

to communicate or not. Clearly, since states in LH sets are locally indistinguishable, LH sets are the smallest units that a communication strategy can be acted upon - it is impossible for agent X to devise a strategy that would behave differently for state (1,1) or (1,2), because at the moment it cannot tell the difference between these two states. This is the key point in reasoning about communication strategies. In our example, this means that the communication strategy for X is simply two boolean questions: (1) whether to communicate or not if current LH set is LH_1^X, and (2) whether to communicate or not if current LH set is LH_2^X. Similarly, Y's communication strategy is the same questions to LH_1^Y and LH_2^Y. In other words, the communication decision can be viewed as a mapping from each local history sets to a yes/no choice. Thus, based on the communication strategy, the agents can look up the communication decision based on its current LH set. For example, if the strategy for X says to communicate if its LH set is LH_1^X, then X would initiate the synchronization process.

Now let us look at the possible communication strategies. If either agent's communication strategy says yes to the current LH set, then the results will be simple: the agents will share their local observation and synchronize to the global state. The LB sets for X, Y, and the common belief state become the same set containing the discovered global state, and the problem solving will continue by the agents performing their actions according to the joint action indicated in the CP. Part (a) of Figure 9 shows the new LB sets for X, Y, and MA in our example.

The more interesting case is when neither agent communicates. Thus, the monitoring agent MA hears nothing. The question now is, how do the LB sets of X and Y and the common belief set (remember that this is the same as the LB set for MA) change? Are they simply their current LH sets and the common belief set is simply the set N? Not always! This actually depends on the communication strategy.

Let us use our example above to illustrate this. Again, (1,1) is the actual global state, and therefore LH_1^X and LH_1^Y are the current LH sets for the agents.

- If X's policy is to say *no* to both LH_1^X and LH_2^X, and Y's policy is to say *no* to both LH_1^Y and LH_2^Y, then the agents cannot infer any information by noticing that no communication has occurred (in fact, the MA expects no communication, based on the strategies). Thus, the LB set for X would be the same of LH_1^X, and the LB set for Y is LH_1^Y, and the common belief state, or the LB set for MA, becomes the set N. This is illustrated in part (b) of Figure 9.

- Now suppose X's policy is LH_1^X:*no* and LH_2^X:*yes*, and Y's policy is LH_1^Y:*no* and LH_2^Y:*yes*. This time, X can infer that since Y does not initiate communication, Y must be having LH_1^Y. Thus, the actual states must be $LH_1^X \cap LH_1^Y$ = (1,1). Similarly, Y can reach the same conclusion. Even MA can infer the same thing: because X does not communicate, X is in LH_1^X, and similarly Y must be in LH_1^Y, and therefore the current global state must be (1,1). This is illustrated in part (c) of Figure 9. We have the same belief states as in part (a), but this is done without any communication!

- Now suppose X's policy is LH_1^X:*no* and LH_2^X:*yes*, and Y's policy is LH_1^Y:*no* and LH_2^Y:*no*. Now, agent X cannot infer anything new from Y's silence: Y could be having LH_1^Y or LH_2^Y. But Y can infer something from X's silence: so X must have LH_1^X. Thus, Y knows that the actual state is $LH_1^X \cap LH_1^Y$ = (1,1). This is quite interesting, since the symmetry is broken: Y now knows the actual state but X does not. It is also easy to see that the local belief state of MA is the same as X's LB set, because MA can infer X is having LH_1^X through X's silence, but cannot infer anything from Y's silence. This case is illustrated in part (d) of Figure 9.

- The other case – where X's policy is LH_1^X:*no* and LH_2^X:*no*, and Y's policy is LH_1^Y:*no* and LH_2^Y:*yes* – is symmetrical to the above case: X's LB becomes (1,1), and Y and MA have the same LB as LH_1^Y.

The basic idea is that when no communication is received, an agent can rule out some LH sets (i.e. the ones that would communicate according to the strategies) in the other agent, therefore helping us narrow down the set B in the next stage. This type of gaining information without communication is a direct result of each agent knowing each other's communication policy.

A more intricate point here is that since both X and Y can mimic the kind of reasoning in MA, they also know how much information the other agent has induced of their state as a result of no communication. For example, in the case of part (c) of Figure 9, agent X would know that agent Y has discovered the actual global state (but X is not sure whether that state is (1,1) or (1,2): X would think that in some episodes it could be (1,1), and in others, (1,2)), and agent Y would know that agent X has not (i.e. Y knows X cannot differeniate between (1,1) and (1,2)). And similarly, Y knows that X knows Y has discovered the state... and so on. A recursive model of what can be inferred about the other agent can still be constructed this way. But those inferences has no effects on establishing a consensus about possible global states. In our model, the distinction between LB and B allows us to specify the difference between an agent's private belief and the consensus among the community: it is possible that some agents know better than others, but without communication, the other agents cannot have enough information in their reasoning in order to narrow down the choices. Thus, the consensus must be based on the common knowledge that enables everyone to perform the reasoning. Our model is not recursive, because our agents are not trying to model the other agent's local belief, but instead focus on establishing the consensus information base about the global state (the common belief state). To this end, the monitoring agent's view is sufficient in order to decide what information is available to all agents.

Now we are ready to formally define the sets we introduced earlier. This definition is inductive:

Definition 1: B^k – the common belief state at time k, and $LB^{X,k}$ and $LB^{Y,k}$ – the local belief state of agent X and Y, as well as the local history (LH) sets as the following:

1. $B^0 = S^0$.
2. Let $N^{k+1} = \text{next}(B^k)$.
3. $LH^{X,k+1}$ and $LH^{Y,k+1}$ are partitions of N^{k+1} such that any two states are in the same LH set if and only if they share the same outcome histories: there exists a sequence of states such that both states can be reached through this sequence with the same local actions, local observations, and synchronization messages. Alternatively, since $LB^{X,k}$ summarizes the local history up to time k, for each possible $LB^{X,k}$ in B, if the local outcome is o_X, then its $LH^{X,k+1}$ can be defined by Equation 1.
4. $B^{k+1} = \{s^{k+1}\}$ – the actual global state to be communicated through synchronization – if s^{k+1} belongs to either one of $LH^{X,k+1}$ sets or one of $LH^{Y,k+1}$ sets that maps to a *yes* synchronization decision; otherwise,
$$B = N - \bigcup LH_{i'}^X - \bigcup LH_{j'}^Y$$
for all i' such that $f(LH_{i'}^{X,k+1}) = yes$ and all j' such that $g(LH_{j'}^{Y,k+1}) = yes$. Here the functions f and g are the communication policy for agent X and Y.
5. $LB^{X,0} = LB^{Y,0} = S^0$; and for $k \geq 0$, $LB^{X,k+1} = B^{k+1} \cap LH^{X,k+1}$ and $LB^{Y,k+1} = B^{k+1} \cap LH^{Y,k+1}$, where $LH^{X,k+1}$ and $LH^{Y,k+1}$ refers to the current LH set for X and Y, i.e. the LH set that matches the actual outcome histories in this episode.

Note that the first 4 items above only involve information available at the monitoring agent. Only the last item involves information locally at the agents, which leads to a narrowed-down LB set in the agents: if we know N (and thus B) and the local history, we can easily identify the actual LH set in each agent. In other words, LH^X and LH^Y can be determined based on N and local information at runtime, so they do not need to be stored.

Therefore, the above discussion shows how the LB sets and the B set evolve during the course of problem solving. Basically, from the set B we can calculate N, and partition it into possible LH sets (this partition process can be iterative, because we can use $LH^{X,k}$ sets to determine $LH^{X,k+1}$ sets – by expanding the latest action outcomes, e.g. Equation 1 – therefore it is straightforward given the CP), and use the communication policy to update the set B. Once we have the new B, we can easily calculate the new LB sets by incorporating local information. And this calculation (from a given B to possible B's in the next stage) can even be performed offline by a process that simulates all possible combinations of outcomes in different episodes. Thus, it is possible to speed up the calculation of B by caching the results from offline processing. Note that for if we could establish an offline model of how the B set may evolve during the course of the problem solving, then all we need at runtime is the actual local observations, which can be used in conjunction with the B set to compute the LH sets and also the LB sets.

Now, given the partition of N into $\cup_i LH_i^X$ and $\cup_j LH_j^Y$, if the current state is $s = LH_i^X \cap LH_j^Y$, then s will be synchronized if and only if at least one of LH_i^X and LH_j^Y maps to *yes* (to synchronize). The new set B will contain all such state s: $\{s = LH_i^X \cap LH_j^Y \mid$ both LH_i^X and LH_j^Y maps to no $\}$. This set includes all states that will not be synchronized, i.e. the LB set of the monitoring agents when no message is exchanged.

The following describes one stage of problem solving in each agent when dynamically transforming the CP into a DP, given the initial LB sets and B:

1. From LB^X and LB^Y, choose the local action a_X and a_Y according to the CP.

2. From B, calculate $N = \cup\text{next}(s)$ for all $s \in B$.

3. From N and all possible outcomes of a_X and a_Y, calculate all LH_i^X and LH_j^Y.

4. Use the mapping from LH sets to *yes/no* to calculate B' (the updated common belief set) and the set $K = N - B'$.

5. Let's assume that a_X has outcome i and a_Y has outcome j (which means the current state is $s = LH_i^X \cap LH_j^Y$. If there is communication, i.e., at least one of LH_i^X and LH_j^Y maps to *yes*, then at the next stage, the common belief state = the local belief of X = the local belief of $Y = \{s\}$ (synchronized.)

6. Else, at the next stage, the common belief state = B', the local belief of X is updated to $LH_i^X \cap B'$, and the local belief of Y is $LH_j^Y \cap B'$.

For the step 1 to be valid, all states in LB^X must have the same a_X according to the CP, otherwise there is ambiguity toward the choice of local actions. Likewise, all states in LB^Y must have the same a_Y. All calculations in the steps are well defined, with the only exception of the choice of the mappings in step 4. As such, the choice of the mappings must ensure that any possible LB sets resulting from step 6 must have the same local action. This no-ambiguity rule translates to a constraint on the choice of communication policy: if LH_i^X is not communicated, all states in $LH_i^X \cap B'$ must have the same local action for X. A similar constraint applies to LH_j^Y.

A matrix representation can be used if the system consists of two agents X and Y: since $N = \cup_i LH_i^X = \cup_j LH_j^Y$, we can write N as the matrix M: a $|LH^X| \times |LH^Y|$ matrix whose elements are the states $s_{i,j} = LH_i^X \cap LH_j^Y$. (For n-agent systems, a multi-dimensional matrix – a tensor – may be used in the same spirit.) Also let M^A be the matrix of the *joint actions* $a_{i,j} = a(s_{i,j})$ according to the CP. Figure 10 illustrates both M (on the left) and M^A (on the right). Then, choosing to communicate on LH_i^X (or LH_j^Y) can be symbolized by crossing out all matrix elements on row i (or column j). After crossing out all communicating LH_i^X's and

$$\begin{array}{c|cccc} & LH_1^Y & LH_2^Y & \dots & LH_n^Y \\ \hline LH_1^X & s_{1,1} & s_{1,2} & \dots & s_{1,n} \\ LH_2^X & s_{2,1} & s_{2,2} & \dots & s_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ LH_m^X & s_{m,1} & s_{m,2} & \dots & s_{m,n} \end{array} \qquad \begin{array}{c|cccc} & LH_1^Y & LH_2^Y & \dots & LH_n^Y \\ \hline LH_1^X & a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ LH_2^X & a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ LH_m^X & a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{array}$$

Figure 10: Local history set matrix and the actions

$$\begin{array}{c|ccc} & LH_1^Y & LH_2^Y & \mathbf{LH_3^Y} \\ \hline LH_1^X & (a|b) & (a|b) & (\mathbf{a}|\mathbf{b'}) \\ LH_2^X & (a|b) & (a|b) & (\mathbf{a}|\mathbf{b'}) \\ \mathbf{LH_3^X} & (\mathbf{a'}|\mathbf{b}) & (\mathbf{a'}|\mathbf{b}) & (\mathbf{a''}|\mathbf{b''}) \end{array}$$

Figure 11: The default communication strategy

LH_j^Y's, the remaining matrix is B'. Thus, the constraint is to make sure that for each row (and column) of the remaining matrix, there is no ambiguity about the next local action for X (or for Y), i.e., the $X(Y)$ component of all $a_{i,j}$ for each row (column) should be the same for the remaining M^A matrix.

4 Communication Policies

Now let us consider the choice of the communication policies, i.e., the mappings of LH sets. Let Com^X be the mapping from possible LH^X sets to $\{yes/no\}$ and Com^Y be the mapping from possible LH^Y sets to $\{yes/no\}$. They define the communication decision of each agent when entering the next stage.

If every LH_i^X or LH_j^Y has unambiguous local actions, the constraint is automatically satisfied, and no communication is necessary. Otherwise, we should choose to communicate on some LH_i^X and/or LH_j^Y sets so that the remaining matrix meets the no-ambiguity constraint. Clearly, this problem belongs to the family of constraint satisfaction problems, which are often NP-hard and solved by approximation methods. In this paper we discuss only two simple approximation solutions - the *default* strategy and the *hill-climbing* strategy.

The default strategy simply maps all ambiguous LH_i^X and LH_j^Y sets to *yes* - let $\text{Com}^X(LH_i^X) = yes$ if and only if LH_i^X has ambiguous actions for X, and $\text{Com}^Y(LH_j^Y) = yes$ if and only if LH_j^Y has ambiguous actions for Y:

The Default Strategy:
If X's actions of LH_i^X are not unique (i.e. $\big|\{a_X(s)|s \in LH_i^X\}\big| > 1$), then we have $\text{Com}^X(LH_i^X) =$ *yes.* Similarly, $\text{Com}^Y(LH_j^Y) = yes$ if Y's actions of LH_j^Y are not unique. Otherwise, $\text{Com}^X(LH_i^X)$ and $\text{Com}^Y(LH_j^Y) = no$.

Figure 11 shows an example of the default strategy. The matrix elements shown are the joint actions for the states (the M^A matrix.) A joint action has the form $(a|b)$ where a is X's local action and b is Y's local action. A boldface row and/or column in the matrix means that the corresponding LH set is mapped to *yes.* It is easy to see that the remaining matrix (B') satisfies the no-ambiguity constraint.

The hill-climbing strategy further reduces communication by performing a heuristic search process that greedily minimizes the total number of *yes* decisions, so that the B' set contains as many states as possible. It is based on heuristic search methods: it evaluates a matrix by a heuristic value — the sum of the number of ambiguous rows and the number of ambiguous columns. Given a matrix, it selects the row or column that would best reduce the heuristic value (hence the hill-climbing), then applies the best selection (i.e., crosses out the selected column/row) and produces a new matrix, and applies the same method on the new matrix.

$$\begin{array}{c c}
 & \begin{array}{ccc} LH_1^Y & LH_2^Y & LH_3^Y \end{array} \\
\begin{array}{c} LH_1^X \\ LH_2^X \\ \mathbf{LH_3^X} \end{array} & \left(\begin{array}{ccc} (a|b) & (a|b) & (a|b') \\ (a|b) & (a|b) & (a|b') \\ (\mathbf{a}'|\mathbf{b}) & (\mathbf{a}'|\mathbf{b}) & (\mathbf{a}''|\mathbf{b}'') \end{array} \right)
\end{array}$$

Figure 12: Example: the Hill Climbing Strategy

When the heuristic value is zero (the no-ambiguity constraint satisfied), the search stops and the current matrix is the result:

The Hill-Climbing Strategy:

1. Let $\mathcal{M} = M^A$.

2. For each non-selected column or row r of $\mathcal{M}$, calculate the heuristic value of $\mathcal{M}'(r)$, which is $\mathcal{M}$ with r selected (crossed out.)

3. Choose the r such that $\mathcal{M}'(r)$ produces the smallest heuristic value h.

4. If $h = 0$, $M' = \mathcal{M}'(r)$; otherwise, let $\mathcal{M} = \mathcal{M}'(r)$, and go back to step 2.

5. If row i (or column j) of M' is selected, $\text{Com}^X(LH_i^X)$ (or $\text{Com}^Y(LH_j^Y)$) is *yes*, otherwise *no*.

Figure 12 shows an example of the hill-climbing strategy, using the same matrix in Figure 11. Instead of selecting both LH_3^X and LH_3^Y, here only LH_3^X is selected (alternatively we can also just select LH_3^Y, depending on the implementation of the hill-climbing algorithm). In this hill-climbing strategy, if the outcome is $(s_{2,3})$ – meaning LH_2^X for agent X and LH_3^Y for agent Y, there will not be a need for communication since agent Y can rule out from its local belief state $s_{3,3}$ (when it does not receive any communication message) since X would have communicated if it had the outcome LH_3^X. This avoids the communication that would have occurred in the default strategy.

Once we have the communication strategies, the DP is defined: the communication decision is specified by the communication strategy, which consists of mappings from LH sets to *yes/no.* Action decisions, which are mappings from LB sets to local actions, are specified via the CP: the X's (Y's) action is simply the X (Y) component of the joint actions for all states in LB^X (LB^Y). The no-ambiguity rule ensures that the local action is same for all states in the LB set.

From the above discussion, we can see that the most complex part of the algorithm involves some operations on the matrix M, which contains $|N|$ possible states in the next stage. In each stage, the transformation is a polynomial-time operation. In the worst case (when no states are ruled out in any stage), the number of states in each stage equals the size of the reachable states in that stage according to the CP. Thus, the worse case needs polynomial time with regard to the size of reachable states in the CP. Typically, since the global states could be revealed after communication – and even when there is no communication, many states could be ruled out from N – the algorithm does not have to deal with many states. Also, this algorithm is dynamic – it only needs to be evaluated in reaction to online information (i.e., it picks the new set B from the set of possible new B sets), therefore it is not necessary to explore the complete CP and hence the computation cost could be generally quite low.

Note that in general, the optimal communication decisions in a decentralized MDP is history-dependent, i.e., the communication decision should be based on the agent's local history information. In this work, the local history information is summarized in the N set (more specifically, the current LH set), and therefore we are able to define the communication decision as simply a mapping from current LH sets to yes/no answers. This, in some sense, is similar to the technique of translating a history-dependent non-Markov decision process into a history-independent MDP by incorporating history information in the state description.

5 Evaluating a DP

Based on different communication policies, there are different DPs. The key criteria for DPs are the expected utility (EU) and the expect total amount of communication (AoC). (If an explicit communication cost is given, we may combine the two criteria into one overall utility measure.) Evidently, the EU of a derived DP is the same as the EU of the CP, which can be evaluated by the policy evaluation algorithm of Markov decision processes. Moreover, once a CP is given, we can calculate the value of each state, $v(s)$, which is the expected future reward when the system is in state s, and $p(s)$, the probability of the reaching state s.

The calculation of AoC is also straightforward. For simplicity let's assume that each synchronization counts as 1. Since the imaginary monitoring agent sees the same amount of communication as the real agents do, we can calculate the AoC observed by the monitoring agent. Let $c(B)$ be the expected future amount of communication when the agents' current common belief is B (i.e., the monitoring agent's LB). Then, $c(B')$ is the expected future amount of communication in the next stage if there is no communication. Thus, the chance of having communication in the next stage is $1 - p(B')/p(B)$. Thus,

Equation 2:

$$c(B) = \frac{p(B')}{p(B)}c(B') + (1 - \frac{p(B')}{p(B)}) + \sum_{s \in K} \frac{p(s)}{p(B)} c(\{s\}) \tag{2}$$

where $p(Q) = \sum_{s \in Q} p(s)$. We can rewrite the above as

Equation 3:

$$\begin{aligned} p(B)c(B) &= p(B')c(B') + (p(B) - p(B')) + \sum_{s \in K} p(\{s\})c(\{s\}) \\ &= (p(B) - p(B')) + \sum_{Q \in \text{next}(B)} p(Q)c(Q) \end{aligned} \tag{3}$$

where next(B) is the set of all possible common belief state at the next stage.

Equation 3 shows the relationship between the AoC of belief state B and the AoC of the belief states in the next stage. For a terminal belief state (i.e., where every state $s \in B$ is a terminal state), $c(B) = 0$, because agents would realize that the global state is a terminal state (i.e., no more local actions) without communication.

We can thus use dynamic programming to calculate the $p(B)c(B)$ in all stages and all the way up to the starting common belief state, $p(B_0)c(B_0)$, which is the AoC of the whole DP (note that the starting probability $p(B_0) = 1$.)

6 Non-Conforming DPs

So far, we discussed how to choose communication policies according to a given CP if the current local belief states are LB^X and LB^Y and the common belief state is B. Once the communication policies are decided, we can calculate all possible resulting local belief states and common belief states in the next stage. Thus, if we start from the beginning state and iterate over the next stages, we would obtain a DP. By applying different communication policies we would get different DPs. Thus, a CP may be transformed to many DPs.

DPs derived this way would adhere to the exact CP in the eyes of an outsider observer who sees the global state and the joint actions of the agents. We call these DPs *conforming* DPs. The set of reachable global states in any conforming DP is exactly the same set of reachable global states defined by the CP. In other words, conforming DPs share the same global state space as the CP.

Conforming DPs offer no loss of EU compared to the CP while decreasing the AoC. However, the decrease of AoC may be limited since we must choose communication policies based on the joint actions specified by

$$\begin{array}{c} \\ LH_1^X \\ LH_2^X \\ LH_3^X \end{array}\begin{array}{c} \begin{array}{ccc} LH_1^Y & LH_2^Y & LH_3^Y \end{array} \\ \begin{pmatrix} (x|y) & (x|y) & (x|y') \\ (x|y) & (x|y) & (x|y') \\ (x'|y) & (x'|y) & (x''|y'') \end{pmatrix} \end{array} \longrightarrow \begin{array}{c} \\ LH_1^X \\ LH_2^X \\ LH_3^X \end{array}\begin{array}{c} \begin{array}{ccc} LH_1^Y & LH_2^Y & LH_3^Y \end{array} \\ \begin{pmatrix} (x|y) & (x|y) & (x|y') \\ (x|y) & (x|y) & (x|y') \\ (\lambda|\lambda) & (\lambda|\lambda) & (\lambda|\lambda) \end{pmatrix} \end{array}$$

Figure 13: Marking some global states terminal

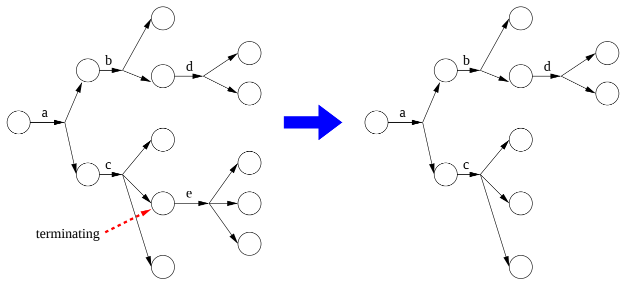

Figure 14: Terminating a State

the CP. In contrast, there are also possible *non-conforming* DPs - if we can also modify the joint actions, then we would have more choices of communication policies. Obviously, by modifying the joint actions we are in fact modifying the CP. This would result in a different state space and thus different EU values. However, we also have the potential of further reducing AoC to the extent not possible by conforming DPs. This offers a tradeoff between EU and AoC when selecting DPs, which the conforming DPs lack. This is why it is interesting to study non-conforming DPs: it offers a wider selection of DPs and offers tradeoff choices.

In order to derive non-conforming DPs from a given CP, we first examine how to derive new CPs from a given one, i.e., how to change to joint actions of states in the set N given current common belief state B. We are mostly interested in domain-independent techniques, which modifies the given CP based on its structure, not on domain knowledge. Specifically, since a policy can be viewed as a directed tree with each node representing a state, and the outgoing edges representing possible transitions to the next states, as shown in the left half of Figure 14. Note that the label of the outgoing edge represents the action to be taken at that state, and there are multiple states from a single action due to non-deterministic action outcomes.

Evidently, there are many many possibilities to change the joint actions of the states in the N set. In this paper we will only examine a few choices based on common intuition. One domain-independent technique is *terminating*: to mark one or more non-terminal states in the original policy to be terminal states in the new policy, i.e., the joint action would become $(\lambda|\lambda)$ where λ means no action. Figure 13 shows an example of the changed joint actions in the matrix representation. The graph view of the policy due to the operation is illustrated in Figure 14. The resulting policy would terminate when reaching the marked states. Clearly, the new policy may not receive any additional reward beyond the marked states, therefore the expected utility of the new policy is different from the original policy's. This technique is well-suited for typical planning problems, where the agents make sequential decisions regarding the execution of tasks, and the problem solving may terminate at any stage and the utility is calculated at that time. Thus, the new terminal states in the new CP is valid.

We identify two simple strategies for applying the terminating operation here:

Terminating 1 (T1): For a given belief state B, one strategy is to mark all of its next states (i.e. any state in N) terminal. In this case all next belief states are terminal, and as a result no more communication is needed. This corresponds to marking all states in local history matrix M terminal and therefore the next joint action (in the form of $(a_X|a_Y)$) for each state in M is $(\lambda|\lambda)$, where λ means no action. In this case M automatically satisfies the no-ambiguity condition, thus K (which is defined as $N - B'$ in section 3: the removed columns and rows in M) is empty (by default).

Terminating 2 (T2): We notice that the above method may result in drastic changes — it eliminates all

$$\begin{array}{c} \\ LH_1^X \\ LH_2^X \\ LH_3^X \end{array}\begin{array}{ccc} LH_1^Y & LH_2^Y & LH_3^Y \\ \end{array}$$

$$\begin{array}{c|ccc} & LH_1^Y & LH_2^Y & LH_3^Y \\ \hline LH_1^X & (x|y) & (x|y) & (x|y') \\ LH_2^X & (x|y) & (x|y) & (x|y') \\ LH_3^X & (x'|y) & (x'|y) & (x''|y'') \end{array} \longrightarrow \begin{array}{c|ccc} & LH_1^Y & LH_2^Y & LH_3^Y \\ \hline LH_1^X & (x|y) & (x|y) & (x|y) \\ LH_2^X & (x|y) & (x|y) & (x|y) \\ LH_3^X & (x'|y) & (x'|y) & (x''|y) \end{array}$$

Figure 15: Merging: merge history set LH_3^Y into LH_2^Y

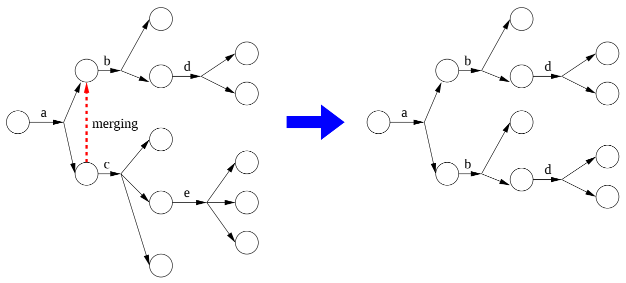

Figure 16: Merging a State

further communications, but also eliminates all further activities. As a result, the expected utility may suffer significantly. So an alternative method is to keep the remaining matrix $N - K$ intact (based on the conforming alternatives) and mark only a subset of the rest of the next states, i.e. the set K. Thus, the communication decision remains the same as the conforming alternatives, but some of the synchronized next belief states would be terminal. This may reduces communication in the further stages for those belief states according to the conforming strategies. And since it only marks a subset of the next states, the degradation of utility is more limited compared to T1.

Another domain-independent technique is *merging*. Illustrated in Figure 16, this technique marks one state to be merged: grafting the subtree beyond another state (target state) onto this state, replacing the original subtree of the merged state. Such a merging operation requires that the merged state is compatible to the target states, so that the subtree grafted correctly reflects the actual structure of the state space: the actions and states in the grafted subtree must exist in the state space, and correctly reflects the transition relationships. Figure 15 shows the joint actions of the matrix N before and after a merging operation.

Typically, this requirement means that the part of the state space beyond the merged state should be topologically identical to the part of the state space beyond the target state. Usually, this is satisfied between *sibling* states, meaning that the two states are both possible resulting states from the same state and action (see Figure 16). This requirement is met when dealing with sequential task planning problems, where the resulting sibling states represent the different possible outcomes of an action (in this case, the same as a task). There, two outcomes are often identical to each other except that they correspond to different utility values. The task interrelationships also remain the same for both outcomes. This means that, if a problem solving episode contains one outcome in its outcome sequence, there must also exist a possible episode that is the same episode except it replaces this outcome with the other one.

For example, an action produces two possible outcomes o_1 and o_2 (and ends in state s_1 and s_2), but the only difference between them is that o_1 produces utility 10 and o_2 produces utility 5, and otherwise they are exactly the same in the problem solving. In this case, the structures of the state spaces beyond s_1 and s_2 are the same.

Since utility value is typically a function of the outcomes, the calculation of the value of the merged state is straightforward: simply use the same process of calculating the utility value of the targeted state, but replace any occurrence of outcome o_2 with o_1 (assuming s_1 is merged to s_2). This technique is used in our calculation when dealing with merged states. The calculation of AoC is even simpler: it is simply the AoC of s_2 multiplied by the factor $p(s_1)/p(s_2)$ — to account for the fact that the probabilities of getting o_1 and o_2 are different.

For the merging operations, since communication decisions are based on local history sets (LH^X and LH^Y) rather than individual states, we focus on merging between *LH* sets. The goal is to merge communicating *LH* sets to non-communicating *LH* sets, so that the communication on the merged sets can be saved. Thus, the merging happens in the local history level rather than the state level: it replaces one local history with another one.

Merging: for a given belief state B, examine the *LH* sets (the rows and columns of M). For example, if a row is ambiguous (not counting elements already marked or merged), we try to find another row which is compatible to it but is unambiguous. If such a row exists, we can then merge each element in the original row to the same column element in the target row. The same can be applied to columns as well.

The third type of operations for generating non-conforming DPs we discuss here is *localizing*: when there is ambiguity in an LH set, we modify the local actions for some states in LH so that the states all have the identical local action. Using a matrix representation, suppose we have the matrix in Figure 17,

$$\begin{array}{c} \\ LH_1^X \\ LH_2^X \end{array} \begin{array}{c} \begin{array}{cc} LH_1^Y & LH_2^Y \end{array} \\ \begin{pmatrix} (x|y) & (x|y') \\ (x'|y) & (x''|y'') \end{pmatrix} \end{array} \longrightarrow \begin{array}{c} \\ LH_1^X \\ LH_2^X \end{array} \begin{array}{c} \begin{array}{cc} LH_1^Y & LH_2^Y \end{array} \\ \begin{pmatrix} (x|y) & (x|y') \\ (x'|y) & (x'|y') \end{pmatrix} \end{array}$$

Figure 17: Localizing LH_2^X by replacing x'' with x' and LH_2^Y by replacing y'' with y'

Clearly, there may be other alternatives such as replacing x' with x'' and replacing y' with y''. The resulting matrix would also be localized: the local action is uniquely decided given only local information. Thus, agents do not need to communicate at all. In our case, we use the following simple rule – to choose the most probable one – to decide which local action to keep and what to replace:

Localizing: For any LH set that is ambiguous, calculate the most *probable* action in this LH set (by summing up the probability of the states having the same local action), and replace all other local actions with the most probable one.

Of course, this policy deviates from the original centralized policy and therefore the system "wanders" into a state space unreachable in the original centralized policy (assuming the CP is defined only over reachable states, as it is usually the case). However, with a complete policy, the next states can be calculated and the next actions are also specified, provided that $(x'|y')$ is a legal joint action.

Note that there might be many other non-conforming techniques that could be used to explore new CPs, and also other strategies for applying the terminating and merging techniques discussed here. These are excellent areas for future developments, but not the focus of this paper. Next we shall see how our approach actually works in our experiments.

7 Experimental Results

To evaluate our approach, we implemented our CP to DP method and performed some experiments using the previous two examples mentioned in section 3.

7.1 The Grid World Problem

First we look at the grid world problem. Continuing from the description in section 2, we can see that when the success rate is less than 1, and in the first step (the common belief state is the singleton set {(0,15)}) the CP specifies the joint action $(\rightarrow \mid \uparrow)$), there might be 9 outcomes states, i.e., the N set contains 9 states. We can write down N and also the next joint actions, using the matrix representation, as the following:

$$\begin{pmatrix} (0,15) & (0,14) & (0,11) \\ (1,15) & (1,14) & (1,11) \\ (4,15) & (4,14) & (4,11) \end{pmatrix}, \text{ and } \begin{pmatrix} (\rightarrow \mid \uparrow) & (\downarrow \mid \uparrow) & (\rightarrow \mid \uparrow) \\ (\downarrow \mid \uparrow) & (\downarrow \mid \uparrow) & (\downarrow \mid \uparrow) \\ (\rightarrow \mid \uparrow) & (\downarrow \mid \uparrow) & (\rightarrow \mid \uparrow) \end{pmatrix}$$

We can see that the first and the third row of the action matrix have ambiguous local actions for X. This means that communication would be needed if the agents want to faithfully follow the centralized policy. Using the *Default* communication policy, the states in the first and third rows (total 6 states) would be communicated (in other words they form the K set), and the remaining 3 states in the second row is the $N - K$ set. The T set, (mentioned earlier for the terminal states), is empty.

Another choice would be to use the *localizing* non-conforming operation. First we compute the probability (assuming a success rate q of 0.92) for that each of these states:

$$\begin{pmatrix} 0.0036 & 0.0012 & 0.0552 \\ 0.0552 & 0.0184 & 0.8464 \\ 0.0012 & 0.0004 & 0.0184 \end{pmatrix}$$

Clearly, the states in the middle row have highest probability for their columns and the states in the last column have highest probability for the rows. So, their local actions will be chosen as the actions to be performed given their local history. Thus, using the localizing operation defined earlier, we will transform the next actions into:

$$\begin{pmatrix} (\rightarrow \mid \uparrow) & (\rightarrow \mid \uparrow) & (\rightarrow \mid \uparrow) \\ (\downarrow \mid \uparrow) & (\downarrow \mid \uparrow) & (\downarrow \mid \uparrow) \\ (\rightarrow \mid \uparrow) & (\rightarrow \mid \uparrow) & (\rightarrow \mid \uparrow) \end{pmatrix}$$

Now we can see that for each row, X's local action becomes unambiguous. The same is true for Y's local action (the columns). The two states that have a new joint actions are (0,14) and (4,14), and they both have joint action $(\rightarrow \mid \uparrow)$ now. As a result, each agent can now decide the next local action using local information only, therefore no communication is needed and the centralized policy is decomposed into local policies for this stage. Using our definitions, this means that the N set is the only next common belief state. In other words, the common belief state "grows" as the agents silently enter the next stage.

Similarly, we can apply the same localizing operation in later stages as well. Table 1 shows the results of applying the localizing operations in each stage, where the deadline is 4, $\beta = 1$, and the success rate $q = 0.92$. The process for computing the EU and AoC values in each stage is as the following:

1. First, compute the state graph that represents the reachable state space using the centralized policy. Compute the EU of the centralized policy and its AoC (assuming the agents communicate at each stage).

2. Use the above state space and centralized policy, apply the *default* strategy to generate a conforming DP according to the centralized policy. Compute the EU and AoC for the default DP.

3. Starting from the first stage, localize the current stage. After the operation, for each state with a modified joint action, build the state space following that state and action (remember that the localizing operation introduces new state spaces unreachable in step 1.)

   After the localizing operation, the possible common belief states at the next stage would either be a singleton common belief state containing a terminal state (one of the states in T), or a common belief state that contains all non-terminating states in the next stage.

4. Then apply the default strategy starting from next non-terminating common belief state, using the modified state space. Then, compute the EU and AoC for this non-conforming DP (with the leading stages localized and the rest stages using the default strategy).

	EU	AoC
Centralized Policy (CP)	91.5202	2.3394
Default DP	91.5202	1.4123
Localizing the first stage	91.5218	1.3358
Localizing the first 2 stages	90.3096	0.3529
All stages localized	85.5874	0.0

Table 1: Localizing Operations

5. Continue the last 2 steps for the next stage, until the last stage is reached. In the end after all stages are localized, the AoC will drop to 0, reaching a completely local DP.

As we can see, when more stages are localized, the AoC of the resulting DP decreases, and in the end it drops to zero. The EU values of the result DPs also change – generally they have a decreasing trend but sometimes may increase. This is not surprising since in general performance decreases when agents communicate less (i.e., have more uncertainty), but there may be chances that the action agents have chosen may indeed be a better action since the original action is non-optimal.

The completely localized DP is an interesting DP that leads us to some important observations. Figure 18 shows a partial drawing of this DP. Each box corresponds to the current agent position (using the number within the box), and the arrow inside the box indicate the local action. The local policy for agent X is at the left and the local policy for agent Y is at the right.

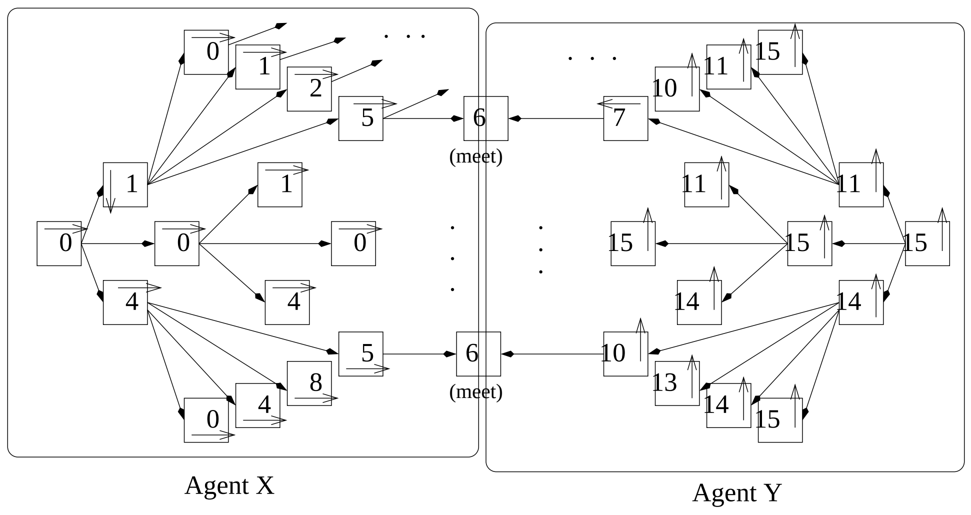

Figure 18: The Completely Localized DP

One thing we can immediately notice is that the intended path (and also the most probable path) from (0,15) ⟶ (1,11) ⟶(5,7) ⟶ (6,6) is preserved in the completely localized DP. This is not surprising since our localizing operation always chooses the action of the most probable state in the local belief. Thus, for each local action, the most probable outcome will keep its next local action as specified by the CP. Thus, this part of the state trajectory is not changed. In other words, the localizing operation preserves the "core" part of the reachable state space defined by the CP, and that forms the basis of the DPs, which consists of two completely local policies, one for each agent. From the agent's point of view, it can be regarded as if the other agent is always "cooperating", i.e. to take the action for the most probable outcome state no matter what the local outcome is. Such a policy is obviously local. This is interesting because the joint goal is changing without communication: there are different goals in response to different local outcome histories - the localized DP does not need communication to switch to a different goal, as if the agents have a secret communication channel.

Naturally, the completely local policy incurs a lot of uncertainty regarding the global state. Thus, in some cases it might be beneficial to reduce this uncertainty and let the agents re-synchronize. Via communication, the agents emerge from the period of non-communicating local processing and discover the exact global state, which starts a new non-communicating period. In this perspective, communication is more like a "sensing" action often used in the study of single-agent planning in uncertain environments, such as the sensing action in Hansen's work [14], where an agent may decide if it is worthwhile to sense, i.e., gain more information about the environment in an anytime problem solving process. The difference here is that the environment now includes other agent(s) that may interact with this agent, therefore it is inaccurate to model the environment as some type of random process. Also, here the communication action is going to change the beliefs of *both* agents involved in communication thus cannot be regarded as a local action and cannot define local state transition rules for the communication action.

Typically, after several non-communicating steps due to the localizing operations, an agent's local belief may contain many states. It is quite possible that the most probable state only accounts for a small percentage of the total probability for all of the states in the current local belief. Clearly, the other states may prefer a different local action, and therefore by re-synchronizing, the agents discover the real global state (which is unlikely to be the most probable state), and take the local action according to the CP. However, since this incurs communication cost, re-synchronization should be used only when there is a potential utility gain large enough to offset the communication cost.

Re-synchronization can be viewed the reverse problem of localizing. In re-synchronization, agents start with a completely localized DP (and of course the CP) and decide if they can gain expected utility by using some communication, while in localizing operations they want to reduce communication at the expense of the expected utility. As such, we can study re-synchronization as the same problem of *selectively* localizing but starting from the reverse direction.

Namely, when doing the localizing operation, we analyze the potential loss of expected utility after localizing. That loss is the same gain of expected utility after re-synchronization in the re-synchronization process. Thus, we only need to slightly modify the localizing process described above.

1. First, compute the state graph that represents the reachable state space using the centralized policy.

2. Use the above state space and centralized policy, apply the *default* strategy to generate a conforming DP according to the centralized policy.

3. Starting from the first stage, find out if the LH sets have ambiguity or not. If an LH set has ambiguous local actions, first try to localize the LH set, and then compute the estimated loss of expected utility by first calculating the weighted average of the expected utility according to the CP for all of the states that have a different local action than the local action of the most probable state, and then multiplying that average by the loss percentage number in the re-synchronization criteria.

4. If the estimated loss is greater than the communication cost, then the agent should communicate (i.e., mapping this LH to *yes*). This is equivalent to choosing re-synchronizing when deciding if the agent should re-synchronize. Otherwise, apply the localizing operation for this LH, and for each state with a modified joint action, build the state space following that state and action (remember that the localizing operation introduces new state spaces unreachable in step 1.)

   After all LH sets are processed, we now have the possible common belief states at the next stage: it would either be a singleton common belief state containing a terminal state (one of the states in T) or a communicated state, or a common belief state that contains all non-communicated and non-terminating states in the next stage.

c	EU	AoC
0.0	91.5202	1.3901
1.0	91.5201	1.3896
2.0	91.0398	1.1008
5.0	90.3922	0.9310
10.0	90.3670	0.9248
20.0	85.5874	0.0

Table 2: Resynchronization

5. Apply the default strategy starting from next non-terminating common belief state, using the modified state space. Then, compute the EU and AoC for this non-conforming DP (with the leading stages localized and the remaining stages using the default strategy).

6. Continue the last 2 steps for the next stage, until the last stage is reached.

Clearly, if the unit communication cost c is tiny compared to the utility of the states, localizing operation would not occur unless the states to be localized has no utility at all. Similarly, if c is very large, localizing will always happen. Table 2 shows the resulting EU and AoC values for the same grid world problem when varying c from 0 to 20, when the agents consider the re-synchronization option in their policies:

Note that the AoC in the first line of Table 2 is slightly smaller than the AoC of the default DP, which is 1.4123. This is because some of the states do have 0 expected utility and thus become localized. The EU remains the same as the default DP, as predicted in this case. Also as predicted, the last line shows 0 AoC and the same EU as the completely localized DP, because such a large c prevented the agents from choosing synchronization.

When c increases, we can see the decreasing trend of AoC and also the decreasing trend of EU. Conversely, this means that for the re-synchronization process, when c decreases from 20 to 0 (reading the table in reverse order), the EU increases while AoC increases.

7.2 The TAEMS Problem

The multiagent planning problem described in Figure 2 as a TAEMS task structure can be mapped into a multiagent MDP by converting agent methods as actions, the possible sequences of the method outcomes as states, and each method outcome as a state transition. A global state is a vector consists of each agent's local states as components, while each agent defines its local states and transitions – its own local Markov process. Of course, the utility measure is based on the global states, and the assumption is that each agent only observes local method outcomes - agents have to communicate to discover the outcome of nonlocal methods.

Although the task structure looks complex, this problem is actually simpler than the grid world problem when transformed into an MDP, since there are only 4 methods for each agents and the constraints among the methods further reduce the size of the state space. Thus, we can study the DP transformation on this problem in more details.

We use the standard dynamic programming algorithm for solving MDPs to obtain the optimal centralized policy (CP). In Figure 19 we sketch this CP by showing the sequences of joint actions that might occur in an episode when this policy is use. As listed, there are five possible cases if this policy is adopted, each corresponds to different joint action sequences and different episodes. Basically, under this policy, A performs tasks A3 and A1, and when possible, perform A4 or A2; while B performs tasks B1 and B3, and when possible, perform B4. The first case corresponds to the method sequences [A3,A1] for agent A and [B1,B3] for agent B. This corresponds to the episodes where both A3 and A1 fail. Case (2) shows the method sequences [A3,A1,A4] and [B1,B3]: when A3 succeeds but A4 fails. Case (3) and (5) show the

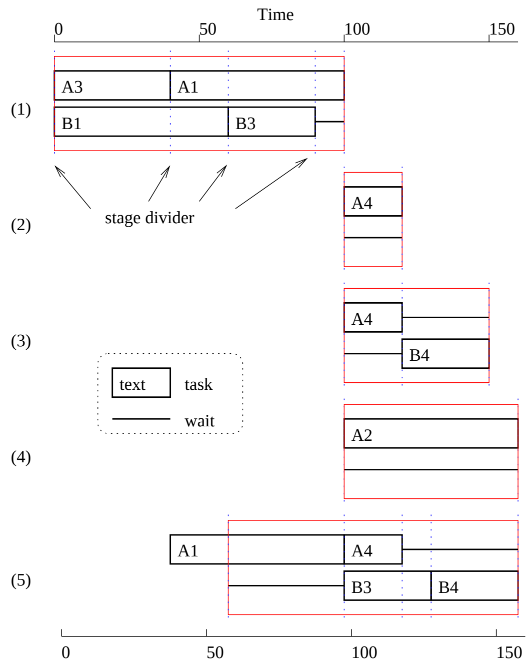

Figure 19: Joint Action Sequences of CP

method sequence [A3,A1,A4] and [B1,B3,B4]: when both A3 and A4 succeed. Case (4) shows the method sequence [A3,A1,A2] and [B1,B3]: when A3 fails but A1 succeeds. (Note: in the figure, the leading action sequences [A3,A1] and [B1,B3] are omitted in cases (2)-(4), and the leading [A3] and [B1] are omitted in case (5)). Based on the specifications given in Figure 19, this CP has a EU of 7.27425 and an AoC of 5.5425 (according to the trivial CP to DP transformation where the agents communicate in every stage).

Next, to generate the DPs, we start from the initial common belief state, and calculate the local history matrix M. If all LH sets (rows and columns of H) have unambiguous local actions, there is no communication needed and the next common belief state is simply N, and we move to the next stage. Otherwise, we will have a few operations that we can apply, with each corresponds to a different DP: first, there is the default communication strategy, then the hill-climbing strategy (if it produces different communication decisions). Also, to produce alternative DPs, the T1 terminating operation may be applied to the current common belief state *if* the difference between the current (terminating) reward and the default value is smaller than the default AoC times a constant (the cost per communication). Intuitively, this criteria means that it is not worthwhile to continue the problem solving when the cost of communication outweighs the potential utility gain by further actions. Also, we may choose to apply T2 operations on the next common belief states produced by the default strategy and the hill-climbing strategy: marking a communicating next belief state terminal *if* its default AoC is greater than the probability of reaching it, and its default value is below the default value of the current belief state. And finally we may apply the merging operations on the sets of next belief states generated by the default strategy and the hill-climbing strategy. In our case, for each communicating LH row (or column), merge the row (column) to the compatible row (column) with the best default value, if such a compatible row (column) exists. Thus, we have several alternative sets of possible next common belief states in the next stage. This search process continues - at the next stage, we again apply those different techniques (where applicable) to expand the search for alternative DPs.

As a result, we obtained 160 different DPs for the optimal CP mentioned before, with 12 DPs containing only combinations of default strategy and hill-climbing strategy - these are the conforming DPs. We evaluated the EU and AoC of each DP, and Figure 20 shows how their EU and AoC values are scattered.

To give a comparison and also some details about the results, the following table lists the EU and AoC of various policies:

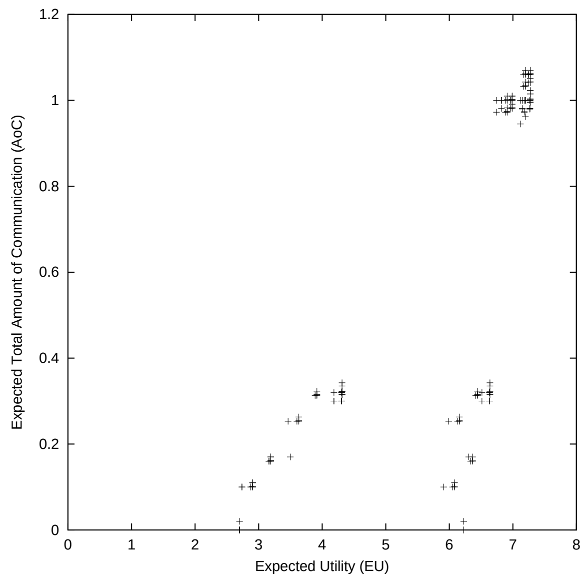

Figure 20: EU/AoC of Generated DPs

	EU	AoC
Optimal CP	7.27425	5.5425
Default DP	7.27425	1.07
Conforming (12 DPs)	7.27425	0.996 – 1.07
All (160 DPs)	2.7 – 7.27425	0 – 1.07

Immediately, we notice that even the default DP (i.e., the DP uses the default strategy only) reduces communication greatly compared to the AoC of the centralized policy (5.5425). This is done without any loss to EU. But the AoC can be further reduced, even to 0, while still having good EU (the rightmost data point on the EU axis has a EU of 6.228). However, the conforming DPs offer only a very limited range of AoC compared to nonconforming DPs, which reduce AoC but degrade the EU.

An interesting pattern shown in Figure 20 is that the data points are somewhat "clustered". The points can be roughly divided into 3 clusters: top-right, lower-right, and lower-left. Note the AoC gap from 0.4 to 0.9, and the EU gap from 4.5 to 5.5. By looking at the details of the DPs we notice that gaps are largely due to the effect of some non-conforming operations, such as marking some belief states terminal, i.e., to stop the problem solving at an earlier stage and therefore abandon all further communications, or the merging of one local outcome to another, therefore causes a significant change of AoC and EU. In other words, some operations are the deciding factors of EU and AoC changes. Thus, by looking at these operations we can detect the "critical points" in DP, and therefore help us in designing coordination strategies for the agents.

As an example, let us examine the DP corresponding to the data point (EU=6.228, AoC=0):

1. At time 0, A starts task A3 and B starts task B1. The next stage begins at time 40 when A3 finishes, and has 3 possible states. Both agents do not have ambiguity toward the next local action, so the next belief state contains these 3 states.

2. At time 40, A starts task A1 and B continues B1. The next stage is when B1 finishes, with 9 possible next states, and merging is applied to B1's q=0 outcome (which contains 3 states). The remaining 6 states form the next common belief state, and no communication is needed.

3. B1 finishes (at time 60), A continues A1 and B starts B3. In the next stage (when B3 finishes), A continues A1 and B waits, so the 12 next states become the next belief state.

4. B3 finishes now (time 90). Next stage, when A1 finishes, there are 36 possible next states based on the 12-state current common belief state. The merging operation is applied to 8 of the next states, and the rest

of 28 states become the next belief state.

5. A1 finishes now, and agent A chooses either A2 or A4, depending on the previous outcome of A3 and A1: if the q(A1) = 6 or if q(A1) = 2 and q(A3) = 0, choose A2, otherwise A4. Clearly this is a local decision, so there is no ambiguity. B simply waits (idle). The next stage has 56 states, and 50 of them are merged, so the next belief state contains only 6 states, all of them terminal states. So the problem solving finishes at the end of the stage.

Essentially, this DP contains 3 merging operations, and as a result the agents have unambiguous local action throughout the problem solving. In typical planning language, it means that the agents simply ignore unexpected local outcomes (such as errors) and stick to a local plan, thus save communication. This divides the cooperation into two seemingly independent local processes, but in fact it is an indication of possible constraint relaxations in the plan: the nonlocal interrelationships are silently established and therefore can be relaxed in each agent's planning.

8 Summary

In this paper we propose a method for deriving decentralized multiagent policies from centralized ones. In the past, the design of decentralized policies has been limited to the use of *ad hoc* heuristic methods, and the results are often domain-specific. By using this method, we now have a domain-independent, systematic way of developing decentralized policies. Furthermore, this method provides a bridge between centralized policies and decentralized policies, thus allows us to connect the research in these areas and find more insights. The use of nonconforming operations in generating new CPs is particularly interesting, because it offers a new perspective in viewing and designing CPs - a perspective from the need of the decentralized agents.

This method also helps us gain some interesting insights on design decentralized policies, for example:

- First of all, communication can be significantly reduced, sometimes even without degradation of utility. There are many techniques and chances for making tradeoffs between the expected total utility and the amount of communication to be used in multiagent cooperation. Ultimately, the problem of meta-level communication decisions can be regarded as a version of the one of the most fundamental problems in AI: the value of information.

- Also, from our calculation of belief states, we notice that when a mutually agreed-upon communication policy is in place, even when there is no communication, agents can infer nonlocal information by noticing that there is no communication. This is an important feature that may be very useful in designing coordination mechanisms for agents with low communication bandwidth.

- The expected utility should not be the sole criteria for CPs. In fact, we have to consider how easily decomposable a CP is - this affects the resulting amount of communication for the derived DPs and affects the implementability of the CP.

- Communication are often used to resolve uncertainty. But here we showed that the necessity for communication is not uncertainty. It is ambiguity. Of course, ambiguity is a result of uncertainty, but uncertainty alone does not demand communication.

- Communication is very much tied to the coordination mechanisms, especially in terms of the dynamics of commitments in multiagent planning. We can develop communication policies based on the coordination mechanisms, and conversely, infer the coordination mechanism by observing the communication patterns.

- Finally, we notice that the effectiveness of nonconforming operations depends heavily on the shape and texture of the problem solution space. There may be some relationship between the state space topology and the suitable kinds of solution policies. It might be also possible to exhibit phase transition phenomena, i.e., some topologies are much easier to generate nearly decomposable DPs while others are much more difficult. This also calls for more study on the structural characteristic of MDPs: new nonconforming operations may be developed to exploit MDPs with particular features.

Designing good decentralized policies is a very challenging task. Yet, it is very important to understand the reasoning, situation assessing, planning, and decision-making in a decentralized, situated agent. One important direction for future work is to further enforce the assumption of decentralization. Specifically, in this paper we assume that the knowledge of the global state space is available to every agent in the system. In the future, we plan to relax this assumption and examine how to generate decentralized policies when each agent only has its own, partial view of the global state space – apparently a much more realistic reflection of actual systems. One possibility is to investigate the applicability of factored MDPs [13] as a way of decomposing a large MDP state space - we are hoping that in those cases, not only the need for communication may be further reduced, but the CPs themselves may also be more compact to represent and easier to decompose.

Acknowledgments

Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Air Force Materiel Command, USAF, under agreement number F30602-99-2-0525 and by the National Science Foundation under Grant number IIS-9812755. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Air Force Research Laboratory, National Science Foundation, or the U.S. Government.

References

[1] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Transition-independent decentralized markov decision processes. In *Proceedings of the Second International Conference on Autonomous Agents and Multi-agent Systems (AAMAS-03)*, Melbourne, Australia, 2003.

[2] D. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, November 2002.

[3] D. Bernstein and S. Zilberstein. Generalized dynamic programming for decentralized pomdps. In *Proceedings of the Sixth European Workshop on Reinforcement Learning*, France, 2003.

[4] Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. The complexity of decentralized control of markov decision processes. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, 2000.

[5] C. Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings of the Sixteenth International Joint Conferences on Artificial Intelligence (IJCAI-99)*, July 1999.

[6] Iadine Chades, Bruno Scherrer, and Francois Charpillet. A heuristic approach for solving decentralized pomdp: Assessment on the pursuit problem. In *Proceedings of the Seventeenth ACM Symposium on Applied Computing (SAC-2002)*, 2002.

[7] R. Davis and R. G. Smith. Negotiation as a metaphor for distribuited problem solving. *Artificial Intelligence*, 20(1):63–109, 1983.

[8] Keith S. Decker and Victor R. Lesser. Quantitative modeling of complex computational task environments. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 214–217, 1993.

[9] E. H. Durfee and T. A. Montgomery. Coordination as distributed search in a hierarchical behavior space. *IEEE Transactions on Systems, Man, and Cybernetics*, 21, 1991.

[10] S. Fujita and V. Lesser. Centralized task distribution in the presence of uncertainty and time deadlines. In *Proceedings of the Second International Conference on Multi-Agent Systems*, pages 87–94, 1996.

[11] Piotr J. Gmytrasiewicz and Edmund H. Durfee. A rigorous, operational formalization of recursive modeling. In *Proceedings of the First International Conference on Multi-Agent Systems*, 1995.

[12] C. V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Conference on Autonomous Agents and Multi-agent Systems*, Melbourne, Australia, 2003.

[13] Carlos Guestrin, Shobha Venkataraman, and Daphne Koller. Context specific multiagent coordination and planning with factored mdps. In *AAAI Spring Symposium on Collaborative Learning Agents*, Stanford, California, March 2002.

[14] E. Hansen. Cost-effective sensing during plan execution. In *Proceedings of the Twelth National Conference on Artificial Intelligence*, 1994.

[15] N. R. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 1993.

[16] G. A. Kaminka and M. Tambe. Robust multi-agent teams via socially-attentive monitoring. *Journal of Artificial Intelligence Research*, 12:105–147, 2000.

[17] V. R. Lesser. A retrospective view of fa/c distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 21, 1991.

[18] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 1990.

[19] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proc. 11th International Conf. on Machine Learning*, pages 157–163, 1994.

[20] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, 2003.

[21] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

[22] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, 1994.

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

[24] Z. Rabinovich, C. V. Goldman, and J. S. Rosenchein. Non-approximability of decentralized control. Technical report, Leibniz Center for Computer Science, 2002.

[25] Jiaying Shen, Victor Lesser, and Norman Carver. Minimizing communication cost in a distributed bayesian network using a decentralized mdp. In *Proceedings of Second International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, pages 678–685, Melbourne, Australia, July 2003.

[26] Ping Xuan and Victor Lesser. Multi-agent policies: from centralized ones to decentralized ones. *Proceedings of the First International Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, 2002.

[27] Ping Xuan, Victor Lesser, and Shlomo Zilberstein. Communication decisions in multi-agent cooperation: Model and experiments. In *Proceedings of the Fifth International Conference on Autonomous Agent (AGENTS 01)*, pages 616–623, Montreal, Canada, 2001.