Commitment-Driven Distributed Joint Policy Search

Stefan Witwicki and Edmund Durfee
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109
{witwicki,durfee}@umich.edu

ABSTRACT

Decentralized MDPs provide powerful models of interactions in multi-agent environments, but are often very difficult or even computationally infeasible to solve optimally. Here we develop a hierarchical approach to solving a restricted set of decentralized MDPs. By forming commitments with other agents and modeling these concisely in their local MDPs, agents effectively, efficiently, and distributively formulate coordinated local policies. We introduce a novel construction that captures commitments as constraints on local policies and show how Linear Programming can be used to achieve local optimality subject to these constraints. In contrast to other commitment enforcement approaches, we show ours to be more robust in capturing the intended commitment semantics while maximizing local utility. We also describe a commitment-space heuristic search algorithm that can be used to approximate optimal joint policies. A preliminary empirical evaluation suggests that our approach yields faster approximate solutions than the conventional encoding of the problem as a multiagent MDP would allow and, when wrapped in an exhaustive commitment-space search, will find the optimal global solution.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Design, Performance

Keywords

Coordination, Negotiation, Agent Modeling

1. INTRODUCTION

AAMAS'07 May 14–18 2007, Honolulu, Hawai'i, USA.

It is now well-established that finding joint policies for agents that are acting and interacting in a stochastic environment is a hard problem. Because in its general form the decentralized MDP problem is NEXP-complete [2], a variety of techniques have been proposed and developed for solving restricted varieties of the problem [1] [7] and for finding only approximately optimal solutions [10].

Our work described in this paper similarly considers restrictions and approximations, but adds to the arsenal of techniques a hierarchical approach that breaks the overall problem into a pair of coupled subproblems. One subproblem is the familiar one in which an agent formulates an optimal policy for its own local MDP. But paired with this is the subproblem of configuring the manner in which agents will handle their interactions such that the agents adopt appropriate inter-agent commitments and capture these in their local MDPs. Solving these subproblems in tandem can, in principle, enable agents to find effective and even optimal joint policies without explicitly searching the (huge) joint policy space.

In this paper, we investigate the viability of this new approach, with a particular eye on answering questions about how agents should model commitments among themselves and how these commitments can be folded into each agent's local policy-finding process. A key contribution that we make is in developing a novel construction of the agents' local MDPs that concisely captures inter-agent commitments as constraints on overall policies, in contrast to modeling commitments as rewards/penalties over states. Our evaluations show that our approach to incorporating commitments into local MDPs permits an efficient and semantically-accurate local encoding of commitments. Further, through this capability, we are able to empirically demonstrate that our approach supports an iterative, co-routining approach to joint policy formulation that can trade off joint policy quality for fast joint policy formulation.

The remainder of this paper is as follows. In the next section we briefly review the general decentralized MDP problem, and present the restricting assumptions (along with their rationales) that should hold in problems where our approach will be applicable. In Section 3, we examine the pitfalls of an existing solution method which motivates our novel approach. We next present the details, and evaluate the efficacy of the approach by contrasting it with encoding commitments as rewards/penalties, showing it to be much less error-prone in capturing the intended semantics. We also describe strategies for searching over the commitment space. In Section 4, we offer preliminary results suggesting

978-81-904262-7-5 (RPS)

the efficiency and "anytime" behavior that our approach engenders, in contrast to solving a more straightforward encoding of the problem as a multi-agent MDP.

2. PROBLEM DESCRIPTION

Our approach addresses coordination problems that arise among loosely-coupled agents that interact through features in their shared environment, where the nature of these interactions are known ahead of time to the agents. The kinds of problems where this and the assumptions described below hold arise in the DARPA Coordinators [13] application domain where one agent might establish the preconditions for another agent's action, or where successful achievement of objectives requires the simultaneous execution of actions by multiple agents. These types of multiagent coordination problems can be represented in the TAEMS language [3].

In this domain, the agents are attempting to maximize their collective rewards, which here for simplicity we will consider to be the sum of their individual rewards. Thus, it is rational for one agent to take actions that reduce its own reward if in doing so it sufficiently increases the local rewards of other agents. Because the agents are only loosely-coupled through the environment, the agents' transition models are semi-dependent, in that one agent's transition can affect special non-local features of another agent's state, thus affecting the other agents trajectory through its state space. These non-local features are separate from other local features changed through actions taken by the other agent, so that the effects of the two acting agents are independent from one another. That is, the effects of an agent's action are stochastically determined based on the agents' collective state, regardless of the contemporaneous actions being taken by other agents.

Furthermore, in this paper, we will assume that in carrying out its policy an agent will know all of the relevant features of the collective state, such as whether another agent has succeeded in a hoped-for enabling (precondition-establishing) action, or whether another agent has also begun the execution of a hoped-for simultaneous action. In other words, for now we take it for granted that inter-agent communication is sufficiently fast, cheap, and reliable that agents can maintain nonlocal awareness about the small subset of nonlocal world features that condition their (inter)actions. We will also assume that agents must reach terminal states in finite time: that due to (for example) utilities being associated with accomplishing (or failing to accomplish) a mission by some deadline, agents are working with finite horizons.

Finally, we assume that transitions are stochastic. This means that even if there is some state of the world that an agent wants to bring about, its actions might not reliably coerce the world into the desired state. Because of this uncertainty, an agent will formulate a policy that considers all of the possible state trajectories, and what action the agent should take depending on what state is reached. Stochastic transitions not only complicate the local planning of an agent, but also complicate what agents can expect from (or promise to) each other. Specifically, *even if an agent has every intention of acting in support of another agent, it might not be able to guarantee to successfully complete the supporting action.*

2.1 Related Work

The study of various flavors of multiagent, decentralized MDPs and POMDPs has burgeoned over the last several years, and a survey of the entire area cannot fit within the space limitations of this paper. The reader interested in the area is encouraged to read more comprehensive treatments given in recent articles in this area such as [7].

Practical strategies for developing effective joint policies have been developed for particular problem subclasses that provide structure that can be exploited. Becker and colleagues groundbreaking Coverage Set algorithm [1] exploits independence among transitions (or rewards) whereby each agent can identify its optimal policy for the different policies that could be adopted by other agents (where the space of alternative policies is compactly represented), and then the joint optimal policy can be derived from these. Goldman and Zilberstein have developed a suite of techniques that exploit goal-oriented behavior in decentralized MDPs (and POMDPs) under different communication regimes [7]. Nair and colleagues, working in the realm of POMDPs, have explored the use of local search in their family of joint equilibrium-based search for policies (JESP) techniques, in which agents exchange local policy information and iteratively construct local policies that are the best responses to the policies currently subscribed to by others [10].

Our work has a flavor of each of the above. Like the work of Becker and colleagues, our inspiration is in reducing the search space for each agent in finding a local policy using a compact model of the space of policies that agents with whom it potentially interacts might adopt. The compact model we use is based on the specification of commitments, which as we show restrict the space of possible policies in ways that matter when it comes to interactions. Like Goldman and Zilberstein, our work exploits a notion of goal-oriented behavior, in the sense that different agents have control over establishing different features of the collective state, and that rewards accrue from reaching "final" states where the agents have together established important conditions. Like those introduced by Nair et al, our techniques conduct a local search, incrementally adjusting commitments to ascend a gradient up the global rewards, though we can also prune infeasible portions of the commitment configuration space. What distinguishes our work is our use of a local search through the space of commitment configurations rather than through policies, and our LP techniques for mapping commitment configurations into constraints on the local policy search.

The idea of forming commitments to interactions first, and then locally working around them, has a rich history in the multiagent systems literature, including work on (Generalized) Partial Global Planning [6] and on team planning [12] just to name a few. Compared to that work, our contribution is in applying these ideas in goal-oriented decentralized MDPs. More recently, commitment-based techniques for coordinating local MDP policies have been developed where commitments are enforced in local models by injecting additional rewards and penalties to particular (classes of) states [9]. As shown in this paper, such techniques do not always capture the semantics of the commitments as effectively as our approach.

2.2 Single-Agent MDPs

Each agent models its interactions with the system using a Markov Decision Process. We now briefly review the basic

MDP model and describe an extension that allows agents to coordinate through the use of commitments. A classical single-agent Markov Decision Process can be described by a 4-tuple $\langle S, A, P, R \rangle$ whose contents are as follows:

- S is a finite set of world states, over which there is a probability distribution α that specifies the start state. Each world state accounts for all features of the agent's local view of its environment.
- A is a finite set of actions available to the agent.
- The dynamics of the world are represented by a transition probability function $P : S \times A \times S \rightarrow [0, 1]$. Given that the agent performs action $a \in A$ in state s, it will transition to state s' with probability $P(s'|s, a)$.
- The reward function $R : S \times A \rightarrow \mathbb{R}$ defines a local reward $R(s, a)$ given to the agent upon taking action a in state s.

The solution of an MDP comes in the form of a policy π, which may be described as a mapping of states to probability distributions over actions ($\pi : S \times A \rightarrow [0, 1]$). An optimal policy π^* is defined as one that maximizes the agent's total expected discounted reward U:

$$U_\gamma(\pi, \alpha) = \sum_{t=0}^{\infty} \sum_i \sum_a \gamma^t \varphi_i(t) \pi_{ia} R(i, a), \quad (1)$$

where $\varphi_i(t)$ refers to the probability of being in state i at time t, $\gamma \in [0, 1)$ is the discount factor, and α is the agent's start state distribution.

There are several common approaches for computing the optimal policy π^* of an MDP [11]. These include Dynamic Programming (i.e. policy iteration, value iteration), Monte Carlo methods, and reinforcement learning. In this paper, we find it convenient to discuss a Linear Programming (LP) approach. An MDP as described above can be formulated as a Linear Program:

$$\max \sum_{i,a} x_{ia} R(i, a) \left| \begin{array}{l} \forall j, \sum_a x_{ja} - \gamma \sum_{i,a} x_{ia} P(j|i, a) = \alpha_j \\ \forall i \forall a, x_{ia} \geq 0 \end{array} \right. \quad (2)$$

where α_j denotes to the probability of starting in state j and the x_{ia} variables, often called *occupancy measures*, denote the total expected discounted number of times action a is performed in state i [4] [8]. Upon solving this LP, we can easily compute the optimal policy π^* from the computed optimal occupancy measures:

$$\pi^*_{i,a} = \frac{x_{ia}}{\sum_b x_{ib}} \quad (3)$$

It is also very easy to compute the expected utility of this policy. This is accomplished by a dot product of the occupancy measures with the MDP reward model:

$$EU = \sum_i \sum_a x_{ia} R(i, a) \quad (4)$$

It may be desirable for optimal *deterministic* policies to be computed. This can be achieved by adding integer and linear constraints to the LP in Equation 9 and solving the resulting mixed-integer linear program (MILP). [5]

2.3 Modeling Peers

The conventional MDP is appropriate for a single agent acting alone in its environment. Here, however, we concern ourselves with a cooperative system of agents sharing an environment and interacting with one another as they interact with the environment. That is, both the state transitions and rewards for a given agent may vary depending on the actions of its peers. Even if the agent is rigidly following a deterministic policy and all state transitions are deterministic, the outcome may be affected by other agents. These transition and reward dependencies can be referred to as *non-local effects* [3].

We assume that, before constructing a policy, each agent is aware of its potential interactions with other agents. Furthermore, we seek to capture these non-local effects explicitly in the agents' local MDP models. Consider a decomposition of the agents fully-observable state space S into local features and non-local features affected solely by actions performed by other agents:

$$S = S_{Local} \times S_{NLE1} \times S_{NLE2} \times ... \quad (5)$$

The features represented by S_{NLEx} may be changed by actions performed by other agents at specified times, but may not be changed by actions performed by the local agent. The initial value of a features is known to both the local agent and the agent that may affect its value. And, when the value changes, so does the effective state of the local agent. Subsequently, the transition and reward models (P and R) must also capture the affected transitions through the factored state space and affected rewards, respectively. Although the agent's MDP model grows exponentially with the number of non-local commitments, as long as the system is loosely-coupled (i.e. exhibiting a low degree of inter-agent dependence), this representation is much more compact and efficient to solve than the full joint state and action space. In the evaluation section below, we contrast this approach with solving the full multi-agent MDP.

Figure 1 shows a graphical representation of 2 agents modeling non-local effects between each other. Here, there are two binary features (x and y) that are changed through actions taken by agent 1. Agent 2 models these in its state space as non-local features. Upon taking initial action a_6, agent 2's transition is dependent solely on whether or not agent 1 sets bit x. The "ghost" state 01?0 represents agent 2's certainty that its local feature bit d will be set, but its uncertainty about whether or not it will enter a state in which x is set. Similarly, bit y is affected by agent 1 during the next transition. In forming a policy, it is useful for agent 2 to know what actions agent 1 will take. In state s_{22}, for example, if agent 1 is likely to set y, then the best action for agent 2 is a_9. But if agent 1 will most surely not set y, agent 2 will be better rewarded if it takes action a_8. Agent 2 models uncertainty of the non-local effects (from agent 1's actions) through transition probabilities in the MDP transition model. Knowing the true values of the probabilities will allow agent 2 to maximize its expected local utility by coordinating its behavior with agent 1.

2.4 Explicitly Modeling Commitments

So that agent 1 and agent 2 can coordinate effectively, we define the notion of a *commitment*:

Definition 1. A **commitment** $C_{ix}(s) = \rho$ is a guaran-

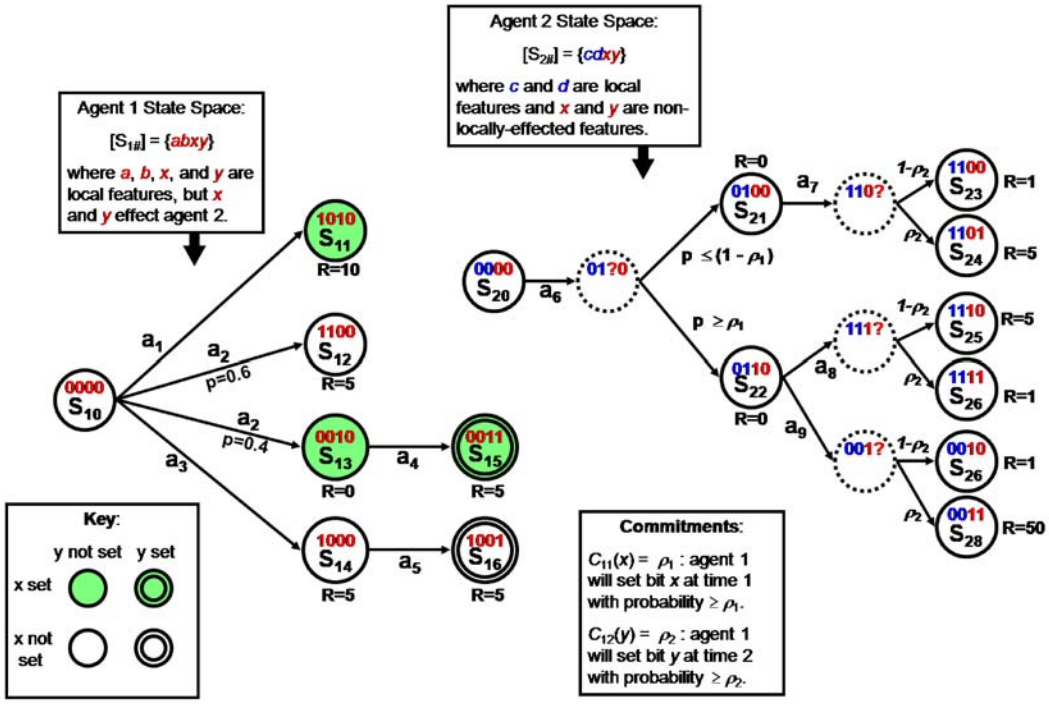

Figure 1: Simple local MDP models with non-local effects and commitments between 2 agents.

tee that agent i involved in non-local effect x will perform actions thereby affecting other agents such that the NLE feature value is effectively $S_{NLEx} = s$ with probability no less than ρ.

For every non-local effect, such a commitment may be formed by each supporting agent. The commitment can be thought of as a promise to enter with probability at least ρ some part of the state space whose feature values (s) are desirable to the dependent agents involved in the non-local effect. To a dependent agent, the commitment is a promise that its transitions and rewards will be restricted to some part of its state space with probability ρ or greater. Thus, from a practical standpoint, this information of commitments made by other agents effectively dictates the MDP transition probabilities for those non-locally dependent features. In this way, policies can be formed with adequate consideration taken of the expected behavior of the other agents in the system. For agent 2 in Figure 1, ρ_1 and ρ_2 model the probabilities with which agent 1 commits to setting bits x and y respectively, and hence agent 2's effective transition probabilities.

A strong assumption of this methodology is that the non-local effects of the system are known. This allows for an explicit representation of the commitment space (such as is shown in Figure 1). We know that agent 1 setting bit x affects the transition probabilities of agent 2. Furthermore, the compactness of the commitment space and efficiency of our commitment-based solution methods depend on the level of inter-agent dependency. For the general class of multi-agent MDP problems, the number of possible commitments can range from 0 to $n \times |S| \times |A|$, each of the these taking on a ρ value between 0 and 1. Discretizing ρ into $\bar{\rho}$ possible values then yields a commitment space as large as $\bar{\rho}^{n \times |S| \times |A|}$. Explicit modeling of commitments is thus most effective for weakly-coupled systems in which the space of possible commitment values is smaller.

3. SOLUTION METHODS

We seek to use the commitment methodology defined in the previous section to guide an agent to build a policy that coordinate its behavior with other agents. Toward this purpose, it is important that agents build policies that strictly adhere to their promised commitments.

3.1 Extra Rewards and Penalties

Let us first turn to a technique described by Musliner and colleagues [9] to enforce commitments in decentralized MDPs. Here, an extra reward $r \geq 0$ is added to for transitions into states in which a commitment is first satisfied and an extra penalty $p \leq 0$ is added for transition into states in which it can first be determined that the commitment will not be satisfied. That is, the local MDP reward model is tweaked so as to bias the MDP solver to find a policy that satisfies the commitment.

For the problem shown in Figure 1, let agent 1 commit to setting bit x with probability $\rho_1 = 1$. The reward/penalty approach will then add extra reward value r to states 0011 and 1010 (since these are the states that agent 1 enters right after setting bit x). A penalty p will then be added to the

reward values of states 1100 and 1001, since these are the states at agent 1's time horizon for which arrival means that bit x has never and will never be set. Notice that if $r > 10$ or if $p < -10$, action a_1 will be strictly preferred by agent 1. Running an MDP solver on this augmented MDP will invariably yield a_1 as the optimal action choice for agent 1 in state 0000. And so agent 1 will be able to satisfy its commitment whilst maximizing its local utility.

The reward/penalty methodology may be effective in some situations, but it is often difficult to set r and p appropriately. Consider commitment set $\{\rho_1 = 0.4, \rho_2 = 0.4\}$, indicating that agent 1 commits to setting x with probability 0.4 and y with probability 0.4. In order to help us select appropriate values for r and p, let us write the expected utility equations for the three policies, adding in r and p where appropriate:

$$\begin{aligned} EU_{rp}[a_1] &= r + p + 10 \\ EU_{rp}[a_2] &= 0.4(2r + 5) + 0.6(2p + 5) = 0.8r + 1.2p + 5 \\ EU_{rp}[a_3] &= r + p + 10 \end{aligned} \tag{6}$$

Puzzling over these equations should lead the reader to the inevitable conclusion drawn in Claim 1.

Claim 1. *There exists an MDP and a set of commitments for which*

1. *There exists a deterministic local policy that satisfies these commitments.*
2. *Using the reward/penalty methodology along with standard deterministic MDP solution techniques, no pair of the form $\{r > 0, p < 0\}$ will yield a policy that both adheres to the commitments and achieves optimal local utility (with respect to the commitment set).*

First, notice that a policy *does* exist which will satisfy the commitment set $\{\rho_1 = 0.4, \rho_2 = 0.4\}$. The deterministic local policy which does this is simply to perform action a_2 in state 0000. With probability 0.4, the agent will transition into 0010, satisfying the first half of its commitment and then into 0011, satisfying the second half of its commitment. The problem is that we cannot compute such a policy by adding extra rewards and penalties. One can prove that not only does there not exist an $\{r, p\}$ pair, but there does not even exist a $\{r_1, p_1, r_2, p_2\}$ tuple (giving the two commitments unique reward and penalty weights) that yields the desired policy.[1] It turns out that the *optimal joint policy* dictates that agent 2 should select action a_2 in order to maximize global utility.

This simple example shows the difficulties of setting r and p. Semantically, we are forced to assign value to satisfying the commitment vs. failing to satisfy it. This value is inherently tied to local policy values dictated by the MDP reward model. If r and p are too close to zero, a policy may be formulated that doesn't satisfy the commitment. But if r and p are too large (and small, respectively), then the agent may sacrifice some of its local quality so as to build a policy that will satisfy the commitment to a higher probabilistic degree than is required. Or still worse, as in our example, there may be no r and p values that can be used to compute a policy that satisfies the commitments even though such a policy exists.

[1]Due to space limitations, we omit this proof.

3.2 A Linear Programming Solution

To address the pitfalls of the rewards/penalties approach to commitment enforcement, we propose a novel alternative. Using extra rewards and penalties means manipulating the local rewards and utility visible to the agent, making some states and actions seem more appealing than others. But satisfaction of a commitment has nothing directly to do with local utility or local rewards. Agreeing to satisfy a commitment means, more accurately, constraining an agent's behavior. This is the basis for our new approach.

Let agent i assert commitment $C_{ix}(s) = \rho$. This commitment is fulfilled if and only if, upon termination, agent i has, with probability no less than ρ, reached a state for which, feature S_{NLEx} has value s. Assuming that the agent is executing a finite-time policy, checking that this commitment is fulfilled requires only checking that the probability distribution over states on the agent's finite-time horizon meets this commitment criterion. The Linear Programming methodology discussed in Section 2.2 is particularly obliging in this respect. Referring to the notation in Equation 2, we can calculate an expected count of the number of times state i is visited by summing over occupancy measures:

$$x_i = \sum_a x_{ia} \tag{7}$$

Moreover, we can constrain the occupancy measures in such a way that states (on the time horizon) with desired feature values are reached with the desired probability. The following constraint will ensure that the linear program computes a policy that fulfills commitment $C_{ix}(s_x) = \rho$:

$$\sum_{\{i|S_{NLEx}(i)=s_x\}} \sum_a x_{ia} \geq \rho \tag{8}$$

So, by adding one additional constraint per outgoing NLE commitment, we can formulate a new linear program for computing optimal local policies that adhere to a given set of commitments:

$$\max \sum_i \sum_a x_{ia} R(i,a) \left| \begin{array}{l} \forall j, \sum_a x_{ja} - \gamma \sum_{a,i} x_{ia} P(j|i,a) = \alpha_j \\ \forall x \sum_{\{i|S_{NLEx}(i)=s_x\}} \sum_a x_{ia} \geq \rho \\ \forall i \forall a, x_{ia} \geq 0 \end{array} \right. \tag{9}$$

A benefit of this LP approach is its ability to construct policies that capture commitments perfectly while still maintaining optimality. This makes our approach extremely robust when compared to the reward/penalty method. For the example in the previous section, our approach returns the optimal local policy which enforces the desired commitment.

It is possible that a certain set of commitments is infeasible for an agent. That is, there does not exist a local policy that will satisfy all commitments. In this case the LP solver will return "no solution" and the agent will know that it is overcommitted. Otherwise, the returned policy is guaranteed to satisfy the commitments. For the rewards/penalties methodology, on the other hand, if there does not exist a policy that satisfies all commitments, this methodology will nevertheless return a policy. Some post-processing of the policy is then needed to determine that not all of the commitments were satisfied.

3.3 Computing Near-optimal Joint Policies

We have shown how to compute optimal local policies given a set of commitments. The task of computing the optimal joint policy is now reduced to searching the space of commitments for the optimal commitment set. The paragraphs that follow describe a general strategy for searching over this space.

We envision an iterative heuristic search algorithm that selects a set of commitments, builds local policies around those commitments, estimates global quality, and repeats. As more iterations are performed, the goal is to achieve higher expected global quality through better sets of commitments. The building of local policies can be done using constrained Linear Programming as described in Section 3.2. We assume that global quality of the joint policy can be closely-approximated by some function of agents' local policy qualities.

The search algorithm begins by initializing the commitment set. One simple approach is to assign random ρ values to each commitment. This initialization may, however, start the search in a very strange part of the commitment space with ρ values that have no relation to probabilities in the agents' MDP models. A slightly more sophisticated method by which to initialize commitments is to allow agents to act greedily. That is, set all ρ values to 0 and build local policies. Next, inspect each local policy, computing what the actual probabilities of satisfying various commitments turned out to be.[2] Let these probabilities form the initial commitment set. In this way, we determine what agents can accomplish if they are alone in the world, and then search for potential interactions that the agents missed out on by acting alone.

On subsequent iterations of our search algorithm, we construct the next commitment set by modifying the previous commitment set. This can be accomplished by random perturbation of the ρ values biased by heuristics such as suspected agent overcommitment or undercommitment. Upon estimating the global quality, one may find that the agents took a step down the hill, achieving a lower overall expected utility than on the previous iteration. Furthermore, a commitment set may be reached that is infeasible. That is, an agent may be overcommitted to the point that no matter how much local utility is sacrificed, it is impossible to satisfy the agent's outgoing commitments. Schemes may be devised for deciding when to step back to a previous set of commitments.

4. EMPIRICAL EVALUATION

This section presents some preliminary empirical results from using our constrained local policy formulation approach within the larger context of searching the joint policy space. As shown in Figure 2, we devised a scalable problem to test the commitment methodology. This problem involves n agents linked together by transition-dependent non-local effects. As shown, agent 1 affects what agent 2 "senses" via a transition that changes the value of a local feature. In turn agent 2 affects agent 3 when it changes one of its features. And agent 3 affects agent 4; and so on, forming a chain of possible enablements all the way to agent n. Notice that the relevant actions $\{A, B, C\}$ of an intermediate agent i correspond to how it can interact with the agent $i+1$. A is the *greedy* action and may only help the agent to achieve high local utility. B and C, on the other hand, are *enabling* actions. With high probability, B and C respectively continue or start a chain of enablements, allowing the next agent to achieve higher utility ($R = 10i$ as opposed to $R = 0$). Due to stochasticity of the environment, the enablement only occurs with with probability 0.9.

Notice that the reward structure is designed so that agents further along in the potential chain get a larger payoff for being enabled than do earlier ones. The same sequential increase is prescribed for the reward received by agents that choose to continue an enablement chain (by performing B in state 0001). These increasing rewards reflect the general cooperative paradigm that it is in the agents' best interests to work together and to form enablement chains with other agents. The MDPs shown in Figure 2 define the local views of each agent. Note, however that the MDPs for intermediate agents vary slightly from one another due to the randomly initialized reward variables g_i and e_i. For each agent, these values are redrawn randomly from uniform distributions $[1, 10]$ and $[1, 5]$ respectively. This causes some agents to be more self-centered and some to be more accommodating to their peers. These MDPs may define an arbitrarily large system of agents. The global utility of the system can be calculated as the sum of local utilities, equal to the sum of rewards that all agents receive.

We begin by examining a 3-agent *commitment chain* problem in which the reward variables are set as follows:
$\{g_1 = 10, g_2 = 5, e_1 = 1, e_2 = 3\}$.
This problem is small enough that we can fully explore the joint policy space. Agent 1 has 2 possible policies:
$\{\langle \text{perform } A_1 \rangle, \langle \text{perform } C_1 \rangle\}$.
Agent 2 has 4 possible policies:
$\{\langle \text{perform } A_2 | state = 0001, \text{perform } A_2 | state = 0000 \rangle,$
$\langle \text{perform } A_2 | state = 0001, \text{perform } C_2 | state = 0000 \rangle,$
$\langle \text{perform } B_2 | state = 0001, \text{perform } A_2 | state = 0000 \rangle,$
$\langle \text{perform } B_2 | state = 0001, \text{perform } C_2 | state = 0000 \rangle\}$.
Agent 3 has only one possible policy, since it has only one action. Out of the $2 \times 4 = 8$ joint policies, the optimal can be calculated to be: $\pi^* = \langle C_1, B_2|0001, C_2|0000 \rangle$.[3] The global expected utility of this joint policy is calculated below in Equation 10:

$$\begin{aligned} U_1(\pi^*) &= 0.9e_0 \\ U_2(\pi^*) &= 0.9(10 \cdot 2 + 0.9(2 \cdot 2)) + 0.1(0.9(e_2)) \\ U_3(\pi^*) &= 0.9(10 \cdot 3) \\ EU(\pi^*) &= \sum_i U_i(\pi^*) = 0.9 + 21.51 + 27 = 49.41 \end{aligned} \tag{10}$$

This joint policy corresponds to the commitment set $C^* = \langle \rho_1 = 0.9, \rho_2 = 0.9 \rangle$, indicating that both agent 1 and agent 2 will perform the necessary enabling actions with the highest probability, thereby forming a full enablement chain. Applying the constrained Linear Programming approach to compute deterministic local policies given optimal commitment set C^*yields local policies $\langle C_1 \rangle$ and $\langle B_2|0001, C_2|0000 \rangle$. Together, these thus form the optimal joint policy π^*. As shown, given the optimal set of commitments, our LP algorithm computes for each agent its contribution to the optimal joint policy. This is because all interactions are captured via commitments and our approach computes optimal local

[2]This is accomplished by a simple summation of appropriate occupancy measures computed by the linear program.

[3]This joint policy dictates that agent 1 perform action C in its initial state and that agent 2 perform action B in state 0001 and C in state 0000.

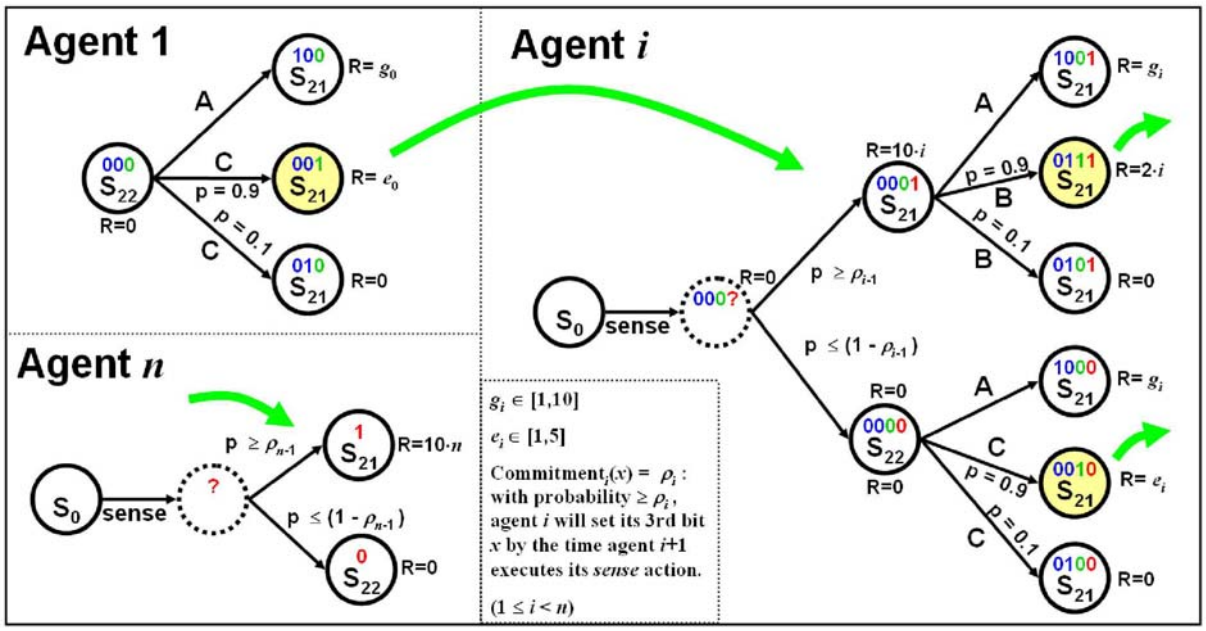

Figure 2: n-agent *Commitment Chain* problem.

policies given a set of commitments. So, given that the optimal set of commitments can be found, our methodology is capable of producing the optimal joint policies as would be computed by solving the full multiagent MDP (MMDP).

A benefit of this distributed approach is that computing local policies is much less computationally intensive. The *Commitment Chain* problem has n agents, $n-2$ of which each have 3 actions and 4 state features. So the joint state and action spaces for the full multiagent MDP are on the order of 2^{4n} and 3^n respectively. As the number of agents grows, formulating and solving the multiagent MDP rapidly becomes intractable. However, our distributed approach can be used to compute a joint policy relatively quickly by solving n local MDPs with 2^4 states and 3 actions each. The optimality of such a joint policy is dependent, however, on the set of commitments, of course. Even if the commitments are suboptimal, our techniques can quickly enable agents to formulate a joint policy that realizes the commitments (if they can all be realized), making our approach useful in time-limited domains.

Using a heuristic search algorithm that searches through the space of commitments, one can take advantage of the trade-off of time versus optimality that our methodology provides. We show the results of one such joint policy search experiment in Figure 3. This plot shows joint policy expected utility values for 25 iterations of commitment-space search (averaged over 50 runs) as described in Section 3.3, using an iterative algorithm similar to simulated annealing.[4]

[4]The line marked "agent-estimated" utility is computed using the estimated utility over which the algorithm is performing its hill-climbing. As shown, this generally underestimates the actual expected joint policy value ("effective utility"), since the effective commitment probability values are typically higher than those promised by the agents.

Here we apply heuristic search strategies to an instance of

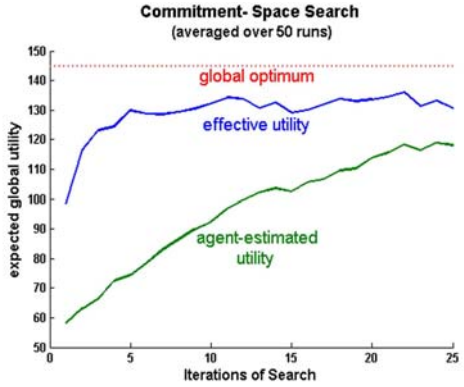

Figure 3: 5-agent *Commitment Chain* Results

the *Commitment Chain* problem with 5 agents, which has a commitment space containing commitments of the form $\langle \rho_1, \rho_2, \rho_3, \rho_4 \rangle$. The commitments are initialized by first locally optimizing under the assumption that no other agent will perform the desirable enablement. On each subsequent iteration of search, one of the commitment values is randomly selected, and perturbed randomly according to an overcommitment heuristic[5]. The random perturbations de-

[5]We use a simple heuristic that estimates an overcommitment level based on how high the ρ value is, whether or not the agent's local utility has gone up or down from the previous iteration, and if it was able to compute a policy at all. The overcommitment level is then used to set the direction and magnitude of the random value perturbations.

crease in intensity across the 25 iterations by use of an inverse exponential cooling schedule. Notice that as the algorithm progresses, expected global utility tends to increase. And after just 25 iterations, the average joint utility has come very close to the optimal value of 144.63. [6]

5. DISCUSSION AND FUTURE WORK

Our research addresses the difficult task of developing coordinated policies for systems of interacting agents. By imposing reasonable restrictions on the problem domain such as loose-coupling, agent awareness of non-local effects, finite horizons, and additive global utility, we have presented a commitment-based methodology for approximating the optimal joint policy. In particular, we have introduced a novel approach to commitment enforcement using constrained Linear Programming. We have shown our approach to be more robust than others in capturing the intended commitment semantics while maintaining local optimality. While constrained Linear Programming is guaranteed to produce optimal local polices that satisfy a set of commitments (if such policies exist), we have shown that standard MDP solution methods with extra rewards and penalties do not have such guarantees.

We have described how our constrained local policy formulation method can form an inner loop to an iterative exploration over the space of commitments. We shown empirically that given a limited amount of time, such a search may be used to approximate optimal joint policies. Iterative commitment-space search has the added benefit of being able to be run for arbitrary amounts of time. The longer it is run, the better policies that we expect to be computed. This "anytime" feature allows for a trade-off of time and solution quality that is particularly useful for domains with computational limitations.

Our empirical results suggest that locally-constrained commitment-based search is an efficient and effective methodology for computing joint policies. Although the limited data presented here demonstrates only that commitment-based search may be applied to a scalable, structured problem, we strive for stronger claims concerning the efficacy of our approach as well as understanding better the applicability of alternative approaches (such as extra rewards/penalties). To this end, ongoing work includes a more thorough empirical study, the development of alternative commitment-space search strategies, and specification of a wider range of test problems. We are concurrently developing algorithms for a large-scale multiagent domain which may serve as a testbed for the methodologies presented here.

6. ACKNOWLEDGMENTS

This material is based upon work supported by Honeywell International, and by the DARPA/IPTO COORDINATORs program and the Air Force Research Laboratory under Contract No. FA8750-05-C-0030. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government. The authors would like to thank David Musliner, Mark Boddy, Robert Goldman, and Jianhui Wu for their valuable contributions to this work.

[6]The optimal joint policy for this instance of *Commitment Chain* is known to be that which prescribes all enabling actions (B_i and C_i) to all agents in all respective states. Given the assigned g_i and e_i variables, the value of this policy can easily be computed to be 144.63 (plotted as the "global optimum" dotted line). We omit the calculations here due to space limitations.

7. REFERENCES

[1] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman. Solving transition independent decentralized Markov Decision Processes. *Journal of Artificial Intelligence Research*, 22:423–455, 2004.

[2] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov Decision Processes. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 32–37, San Mateo, CA, 2000. Morgan Kaufmann Publishers.

[3] K. Decker. TAEMS: A framework for environment centered analysis & design of coordination mechanisms. In *Foundations of Distributed Artificial Intelligence, Chapter 16*, pages 429–448. G. O'Hare and N. Jennings (eds.), Wiley Inter-Science, 1996.

[4] D'Epenoux. A probabilistic production and inventory problem. *Management Science*, 10:98108, 1963.

[5] D. A. Dolgov and E. H. Durfee. Stationary deterministic policies for constrained MDPs with multiple rewards, costs, and discount factors. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, August 2005.

[6] E. Durfee and V. Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5):1167–1183, September 1991.

[7] C. V. Goldman and S. Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.

[8] L. Kallenberg. *Linear Programming and Finite Markovian Control Problems*. Math. Centrum, Amsterdam, 1983.

[9] D. J. Musliner, E. H. Durfee, J. Wu, D. A. Dolgov, R. P. Goldman, and M. S. Boddy. Coordinated plan management using multiagent MDPs. In *Working Notes of the AAAI Spring Symp. on Distributed Plan and Schedule Management*, March 2006.

[10] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *AAAI05*, pages 133–139, 2005.

[11] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[12] M. Tambe. Teamwork in real-world dynamic environments. In *Proceedings of the Second International Conference on Multi–Agent Systems (ICMAS)*. MIT Press, 1996.

[13] DARPA/IPTO COORDINATORS Program. http://www.darpa.mil/IPTO/Programs/coordinators/