Sequential Optimality and Coordination in Multiagent Systems

Craig Boutilier
Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada V6T 1Z4
cebly@cs.ubc.ca

Abstract

Coordination of agent activities is a key problem in multiagent systems. Set in a larger decision theoretic context, the existence of coordination problems leads to difficulty in evaluating the utility of a situation. This in turn makes defining optimal policies for sequential decision processes problematic. We propose a method for solving sequential multiagent decision problems by allowing agents to reason explicitly about specific *coordination mechanisms*. We define an extension of value iteration in which the system's state space is augmented with the state of the coordination mechanism adopted, allowing agents to reason about the short and long term prospects for coordination, the long term consequences of (mis)coordination, and make decisions to engage or avoid coordination problems based on expected value. We also illustrate the benefits of mechanism generalization.

1 Introduction

The problem of coordination in multiagent systems (MASs) is of crucial importance in AI and game theory. Given a collection of agents charged with the achievement of various objectives, often the optimal course of action for one agent depends on that selected by another. If the agents fail to *coordinate* the outcome could be disastrous. Consider, for instance, two agents that each want to cross a bridge that can support the weight of only one of them. If they both start to cross, the bridge will collapse; coordination requires that they *each* "agree" which one of them should go first.

Coordination problems often arise in *fully cooperative MASs*, in which each agent shares the same utility function or *common interests*. This type of system is appropriate for modeling a team of agents acting on behalf of a single individual (each tries to maximize that individual's utility). In the bridge example above, it may be that neither agent cares whether it crosses first, so long as they both cross and pursue their objectives. In such a setting, coordination problems generally arise in situations where there is some flexibility regarding the "roles" into which agents fall. If the abilities of the agents are such that it makes little difference if agent $a1$ pursues objective $o1$ and $a2$ pursues $o2$, or vice versa, the agents run the risk of both pursuing the same objective—with consequences ranging from simple delay in goal achievement to more drastic outcomes—unless they coordinate. This issue arises in many team activities ranging from logistics planning to robotic soccer.

An obvious way to ensure coordination is to have the agents' decision policies constructed by a central controller (thus defining each agent's role) and imparted to the agents. This is often infeasible. Approaches to dealing with "independent" decision makers include: (a) the design of conventions or social laws that restrict agents to selecting coordinated actions [9, 15]; (b) allowing communication among agents before action selection [16]; and (c) the use of learning methods, whereby agents learn to coordinate through repeated interaction [5, 6, 8, 11].

Unfortunately, none of these approaches explicitly considers the impact of coordination problems in the context of larger sequential decision problems. If the agents run the risk of miscoordination at a certain state in a decision problem, how should this impact their policy decisions at *other states*? Specifically, what is the long-term (or sequential) *value* of being in a state at which coordination is a potential problem? Such a valuation is needed in order for agents to make rational decisions about whether to even put themselves in the position to face a coordination problem.

Unfortunately, there are no clear-cut definitions of sequential optimality for multiagent sequential decision processes in the general case. Most theoretical work on coordination problems assumes that a simple repeated game is being played and studies methods for attaining equilibrium in the stage game. In this paper, we argue that optimal sequential decision making requires that agents be able to reason about the specific coordination mechanisms they adopt to resolve coordination problems. With this ability, they can make optimal decisions by considering the tradeoffs involving probability of (eventual) coordination, the consequences of miscoordination, the benefits of coordination, the alternative courses of action available, and so on. We develop a dynamic programming algorithm for computing optimal policies that accounts not only for the underlying system state, but also the *state of the coordination mechanism being adopted*. Specifically, we show how the underlying state space can be expanded minimally and dynamically to account for specific coordination protocol being used.

With this definition of state value *given* a coordination mechanism, one can tackle the problem of defining good coordination mechanisms for specific decision problems that offer good expected value (we but will make a few remarks near the end of the paper on this point). Our framework therefore provides a useful tool for the design of conventional, communication and learning protocols [15].

We focus on fully cooperative MASs, assuming that a common coordination mechanism can be put in place, and that agents have no reason to deliberate strategically. However, we expect most of our conclusions to apply *mutatis mutandis* to more general settings. We introduce Markov decision processes (MDPs) and *multiagent MDPs* (MMDPs) in Section 2. We define coordination problems and discuss several coordination mechanisms in Section 3. In Section 4 we describe the impact of coordination problems on sequential optimality criteria, show how to expand the state space of the MMDP to reason about the state of the specific mechanisms or protocols used by the agents to coordinate, and develop a version of value iteration that incorporates such considerations. We illustrate the ability of generalization techniques to enhance the power of coordination protocols in Section 5, and conclude with some remarks on future research directions in Section 6.

2 Multiagent MDPs

2.1 Markov Decision Processes

We begin by presenting standard (single-agent) *Markov decision processes* (MDPs) and describe their multiagent extensions below (see [3, 13] for further details on MDPs). A fully observable MDP $M = \langle \mathcal{S}, \mathcal{A}, \Pr, R\rangle$ comprises the following components. $\mathcal{S}$ is a finite set of *states* of the system being controlled. The agent has a finite set of *actions* $\mathcal{A}$ with which to influence the system state. Dynamics are given by $\Pr : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$; here $\Pr(s_i, a, s_j)$ denotes the probability that action a, when executed at state s_i, induces a transition to s_j. $R : \mathcal{S} \rightarrow \Re$ is a real-valued, bounded *reward function*. The process is fully observable: though agents cannot predict with certainty the state that will be reached when an action is taken, they can observe the state precisely once it is reached.

An agent finding itself in state s^t at time t must choose an action a^t. The *expected value* of a course of action π depends on the specific objectives. A *finite horizon* decision problem with horizon T measures the value of π as $E(\sum_{t=0}^{T} R(s^t)|\pi)$ (where expectation is taken w.r.t. Pr). A *discounted, infinite horizon* problem measures value as $E(\sum_{t=0}^{\infty} \beta^t R(s^t)|\pi)$. Here $0 \leq \beta < 1$ is a discount factor that ensures the infinite sum is bounded.

For a finite horizon problem with horizon T, a *nonstationary* policy $\pi : S \times \{1, \cdots, T\} \rightarrow A$ associates with each state s and stage-to-go $t \leq T$ an action $\pi(s, t)$ to be executed at s with t stages remaining. An *optimal* nonstationary policy is one with maximum expected value at each state-stage pair. A *stationary* policy $\pi : S \rightarrow A$ for an infinite horizon problem associates actions $\pi(s)$ with states alone.

A simple algorithm for constructing optimal policies (in both the finite and infinite horizon cases) is *value iteration* [13]. Define the t-stage-to-go value function V^t by setting

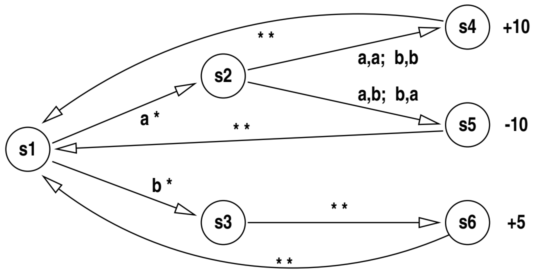

Figure 1: A Simple MMDP with a Coordination Problem.

$V^0(s_i) = R(s_i)$ and:

$$V^t(s_i) = R(s_i) + \max_{a\in\mathcal{A}}\{\beta \sum_{s_j\in\mathcal{S}} \Pr(s_i, a, s_j)V^{t-1}(s_j)\} \quad (1)$$

For a finite horizon problem with horizon T, we set $\beta = 1$ (no discounting) and during these calculations set $\pi(s_i, t)$ to the action a maximizing the right-hand term, terminating the iteration at $t = T$. For infinite horizon problems, the sequence of value functions V^t produced by value iteration converges to the optimal value function V^*. For some finite t, the actions a that maximize the right-hand side of Equation 1 form an optimal policy, and V^t approximates its value.

2.2 The Multiagent Extension

We now assume that a collection of agents is controlling the process. The individual actions of agents interact in that the effect of one agent's actions may depend on the actions taken by others. We take the agents to be acting on behalf of some individual; therefore, each has the same utility or reward function R. The system is fully observable to each agent.

We model this formally as a *multiagent Markov decision process* (MMDP). MMDPs are much like MDPs with the exception that actions (and possibly decisions) are "distributed" among multiple agents. An MMDP $M = \langle \alpha, \{A_i\}_{i\in\alpha}, \mathcal{S}, \Pr, R\rangle$ consists of five components. The set α is a finite collection of n agents, with each agent $i \in \alpha$ having at its disposal a finite set A_i of *individual actions*. An element $\langle a_1, \cdots, a_n\rangle$ of the *joint action space*, $\mathcal{A} = \times A_i$, represents the concurrent execution of the actions a_i by each agent i. The components $\mathcal{S}$, Pr and R are as in an MDP, except that Pr now refers to joint actions $\langle a_1, \cdots, a_n\rangle$.

Taking the joint action space to be the set of basic actions, an MMDP can be viewed as a standard (single-agent) MDP. Specifically, since there is a single reward function, the agents do not have competing interests; so any course of action is equally good (or bad) for all. We define optimal *joint policies* to be optimal policies over the joint action space: these can be computed by solving the (standard) MDP $\langle \mathcal{A}, \mathcal{S}, \Pr, R\rangle$ using an algorithm like value iteration.

Example An example MMDP is illustrated in Figure 1. The MMDP consists of two agents $a1$ and $a2$, each with two actions a and b that can be performed at any of the six states. All transitions are deterministic and are labeled by the joint actions that induce that transition. The joint action $\langle a, b\rangle$ refers to $a1$ performing a and $a2$ performing b, and others similarly (with * referring to any action taken by the corresponding agent). At the "source" state s_1, $a1$ alone decides whether the system moves to s_2 (using a) or s_3 (using b). At s_3, the agents are guaranteed a move to s_6 and a reward of 5 no matter what joint action is executed. At s_2 both agents must choose action a or both must choose b in order to move to s_4 and gain a reward of 10; choosing opposite actions results in a transition to s_5 and a reward of -10. The set of optimal joint policies are those where $a1$ chooses a at s_1 ($a2$ can choose a or b), and $a1$ and $a2$ choose either $\langle a, a\rangle$ or $\langle b, b\rangle$ at s_2.

The value function determined by solving the MMDP for the optimal joint policy is the *optimal joint value function* and is denoted $\widetilde{V}$. In the example above, an infinite horizon problem with a discount rate of 0.9 has $\widetilde{V}(s_1) = 29.9$, while for a finite horizon problem, $\widetilde{V}^t(s_1)$ is given by $10\lfloor \frac{t+1}{3} \rfloor$.

MMDPs, while a natural extension of MDPs to cooperative multiagent settings, can also be viewed as a type of stochastic game as formulated by Shapley [14]. Stochastic games were originally formulated for zero-sum games only (and as we will see, the zero-sum assumption alleviates certain difficulties), whereas we focus on the (equally special) case of cooperative games.

3 Coordination Problems and Coordination Mechanisms

The example MMDP above has an obvious optimal joint policy. Unfortunately, if agents $a1$ and $a2$ make their decisions independently, this policy may not be implementable. There are two optimal joint action choices at s_2: $\langle a, a\rangle$ and $\langle b, b\rangle$. If, say, $a1$ decides to implement the former and $a2$ the latter, the resulting joint action $\langle a, b\rangle$ is far from optimal. This is a classic coordination problem: there is more than one optimal joint action from which to choose, but the optimal choices of at least two agents are mutually dependent (we define this formally below). Notice that the uncertainty about how the agents will "play s_2" makes $a1$'s decision at s_1 rather difficult: without having a good prediction of the expected value at s_2, agent $a1$ is unable to determine the relative values of performing a or b at s_1 (more in this in Section 4).

In the absence of a central controller that selects a single joint policy to be provided to each agent, ensuring coordinated action choice among independent decision makers requires some *coordination mechanism*. Such a mechanism restricts an agent's choices among the potentially individually optimal actions, perhaps based on the agent's history. We describe some of these below, including learning, conventional and communication techniques.

In the remainder of this section, we focus on *repeated games*, returning to general MMDPs in the next section. An identical-interest repeated game can be viewed as an MMDP with only one state—joint actions are played at that state repeatedly. An immediate reward $R(a)$ is associated with each joint action. Our aim is to have the individual actions selected by each agent constitute an optimal joint action. Formally, a *stage game* G comprises action sets A_i for each agent i, joint action space $\mathcal{A}$, and reward function R. The stage game is played repeatedly.

Definition Joint action $a \in \mathcal{A}$ is *optimal* in stage game G if $R(a) \geq R(a')$ for all $a' \in \mathcal{A}$. Action $a_i \in A_i$ is *potentially individually optimal* (PIO) for agent i if some optimal joint action contains a_i. We denote by PIO_i the set of such actions for agent i.

Definition Stage game $G = \langle \alpha, \{A_i\}_{i\in\alpha}, R\rangle$ induces a *coordination problem* (CP) iff there exist actions $a_i \in \mathit{PIO}_i$, $1 \leq i \leq n$, such that $\langle a_1, \cdots, a_n\rangle$ is not optimal.

Intuitively, a CP arises if there is a chance that each agent selects a PIO-action, yet the resulting joint action is suboptimal.

CPs in repeated games can often be "reduced" by eliminating certain PIO-actions due to considerations such as dominance, risk (e.g., see the notions of risk-dominance and tracing used by Harsanyi and Selten to select equilibria [7]), or focusing on certain PIO-actions due to certain asymmetries. These reductions, if embodied in protocols commonly known by all agents, can limit choices making the CP "smaller" (thus potentially more easily solved), and sometimes result in a single "obvious" action for each agent. We do not consider such reductions here, but these can easily be incorporated into the model presented below.

A *coordination mechanism* is a protocol by which agents restrict their attention to a subset of their PIO-actions in a CP. A mechanism has a *state*, which summarizes relevant aspects of the agent's history and a *decision rule* for selecting actions as a function of the mechanism state. While such rules often select actions (perhaps randomly) from among PIO-actions, there are circumstances where non-PIO-actions may be selected (e.g., if the consequences of uncoordinated action are severe). Mechanisms may guarantee immediate coordination, eventual coordination, or provide no such assurances. To illustrate, we list some simple (and commonly used) coordination methods below. In Section 4, we will focus primarily on randomization techniques with learning. However, communication and conventional methods can be understood within the framework developed below as well.

Randomization with Learning This is a learning mechanism requiring that agents select a PIO-action randomly until coordination is achieved (i.e., an optimal joint action is selected by the group). At that point, the agents play that optimal joint action forever. We assume that actions are selected according to a uniform distribution.[1] The mechanism has $k + 1$ states, where k is the number of optimal joint actions: k states each denote coordination on one of the optimal actions, and one denotes lack of coordination. The state changes from the uncoordinated state to a coordinated state as soon as an optimal action is played. This requires that agents be able to observe actions or action outcomes.

We can model this protocol as a finite-state machine (FSM). The FSM for the CP at s_2 in Figure 1 is illustrated in

[1] In this and other mechanisms, reduction methods can be used to reduce the number of actions considered by each agent.

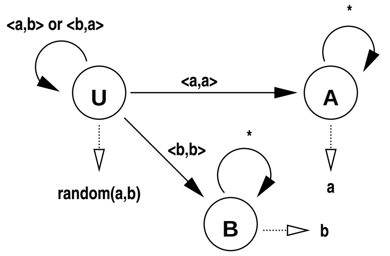

Figure 2: A simple FSM for the randomization mechanism: solid arrows denote state transitions, labeled by inputs (observed joint actions); dashed arrows indicate outputs (action choices).

Figure 2. When the agents are uncoordinated (state U), they each choose action a and b randomly. If the observed joint action is not coordinated, the remain in state U; but if they coordinate, they move to the appropriate state (A or B) and stay there (executing the corresponding action).

For many problems, we can view the mechanism as having only two states: coordinated (C) and uncoordinated (U). If C, we simply require that the agent memorize the action on which the group coordinated. For the purposes of computing expected value below, we often need only distinguish between C and U states (without regard to the actual action chosen). We note that randomization works quite well if there is a small group of agents with few actions to choose from; but as these sets grow larger, the probability of transitioning from U to C gets exponentially smaller. Randomization ensures eventual coordination, at a rate dictated by the number of agents and number of choices available to them.

Fictitious play (FP) is a related learning technique commonly studied in game theory [4, 6] where each agent i observes the actions played in the past by other agents and plays a best response given the empirical distribution observed. We refer to [6] for details, but note that the state of the mechanism consists of "counts" of the PIO-actions played by other agents; thus FP has an infinite number of states. For fully cooperative games, FP converges to an optimal joint action if attention is restricted to PIO-actions and agents randomize over tied best responses [2, 12].[2] It also has the property that once a coordinated action is played, it is played forever. Unlike randomization, FP tends to lead to *faster* coordination as the number of agents and actions increase [2].

Lexicographic Conventions Conventions or social laws (e.g., driving on the right-hand side of the road) are often used to ensure coordination [9, 15]. *Lexicographic conventions* can be applied to virtually any CP. Given some commonly-known total ordering of both agents and individual actions, the set of optimal actions can be totally ordered in several different ways. Lexicographic conventions ensure immediate coordination, but can have substantial overhead due to the requirement that each agent have knowledge of these orderings of both agents and actions. This may be reasonable in a fixed setting, but may be harder to ensure over a variety of decision problems (e.g., involving different collections of agents). In contrast, the learning models described above can be viewed as "meta-protocols" that can be embodied in an agent once and applied across multiple decision problems.

Communication Finally, a natural means of ensuring coordination is through some form of communication. For example, one agent may convey its intention to perform a specific PIO-action to another, allowing the other agent to select a matching PIO-action. There are a number of well-known difficulties with devising communication and negotiation protocols, involving issues as varied as synchronization and noisy channels. We do not delve into such issues here. We assume that some agreed upon negotiation protocol is in place. Realistically, we must assume that communication has some cost, some risk of failure or misinterpretation, and delays the achievement of goals. As such, we model communication as actions in an MMDP which have effects not on the underlying system state, but on the "mental state" of the agents involved. Rather abstractly, we can say that the state of a communicative coordination mechanism for an agent i is its estimate of the "mental state" of other agents. For example, after negotiation, agent $a1$ may believe that $a2$ is committed to performing action b. The "mental state" of other agents will generally only be partially observable, and the state of the mechanism will be estimated by each agent.

4 Dynamic Programming with Coordination

4.1 Sequential Optimality and State Value

CPs arise at specific states of the MMDP, but must be considered in the context of the sequential decision problem as a whole. It is not hard to see that CPs like the one at s_2 in Figure 1 make the joint value function misleading. For example, $\widetilde{V}^1(s_2) = 10$ and $\widetilde{V}^1(s_3) = 5$, suggesting that $a1$ should take action a at s_1 with 2 stages-to-go. But $\widetilde{V}^1(s_2)$ assumes that the agents will select an optimal, coordinated joint action at s_2. As discussed above, this policy may not be implementable. Generally, the optimal joint value function $\widetilde{V}$ will overestimate the value of states at which coordination is required, and thus overestimate the value of actions and states that lead to them.

A more realistic estimate $V^1(s_2)$ of this value would account for the means available for coordination. For instance, if a lexicographic convention were in place, the agents are assured of optimal action choice, whereas if they randomly choose PIO-actions, they have a 50% chance of acting optimally (with value 10) and a 50% chance of miscoordinating (with value -10). Under the randomization protocol, we have $V^1(s_2) = 0$ and $V^1(s_3) = 5$, making the optimal decision at s_1, with two stages to go, "opting out of the CP:" $a1$ should choose action b and move to s_3.

Unfortunately, pursuing this line of reasoning (assuming a randomization mechanism for coordination) will lead the $a1$ to always choose b at s_1, no matter how many stages remain. If we categorically assert that $V^1(s_2) = 0$, we must have that $V^t(s_3) > V^t(s_2)$ for any stage $t \geq 1$. This ignores the

[2] Hence, it might best be described as a learning technique with randomization, rather than a randomization technique with learning.

fact that the coordination mechanism in question does not require the agents to randomize at *each* interaction: once they have coordinated at s_2, they can choose the same (optimal) joint action at all future encounters at s_2. Clearly, the $V^1(s_2)$ *depends on the state of the coordination mechanism.* If the agents have coordinated in the past, then $V^1(s_2) = 10$, since they are assured coordination at this final stage; otherwise $V^1(s_2) = 0$. By the same token, $V^t(s_2)$ depends on the state of the mechanism for arbitrary $t \geq 1$, as does the value of other states.

The optimal value function V is not a function of the system state alone, but also depends on the state of the mechanism. By expanding the state space of the original MMDP to account for this, we recover the usual value function definition. In this example, we define the *expanded MMDP* to have states of the form $\langle s, c \rangle$, where s is some system state and c is the state of the randomization mechanism. We use C and U to refer to coordinated and uncoordinated states of the mechanism, respectively (with C standing for either A or B in the FSM of Figure 2). Transitions induced by actions are clear: each action causes a system state transition as in the MMDP, while the coordination state changes from U to C only if the agents choose action $\langle a, a \rangle$ or $\langle b, b \rangle$ at s_2 (and never reverts to U). The coordination protocol also restricts the policies the agents are allowed to use at s_2. If they find themselves at (expanded) state $\langle s_2, U \rangle$, they must randomize over actions a and b. As such, the transition probabilities can be computed easily: $\langle s_2, U \rangle$ moves to both $\langle s_4, C \rangle$ and $\langle s_5, U \rangle$ with probability 0.5.[3]

The expanded MMDP can be viewed a combination of the original MMDP and the partially specified controller shown in Figure 2. The state space of the expanded MMDP is given by the cross-product of the MMDP and FSM state spaces, while the FSM restricts the choices that can be made when the agents are at state s_2 (for each state A, B or U of the FSM). Generally speaking, the protocol restricts action choices at the state where the CP arose, while optimal choices should be made at all other states. Notice that these choices are optimal *subject to the constraints imposed by the protocol (or finite-state controller).*

With this expanded state space, we can trace value iteration on our running example to illustrate how the agents reason about sequential optimality in a way that accounts for the CP and the coordination mechanism. We assume a finite horizon problem without discounting.

Example For all stages $t > 0$, obviously $V^t\langle s_2, C \rangle > V^t\langle s_3, C \rangle$; so if the agents are in a state of coordination, $a1$ should choose action a at s_1 and "opt in" to the CP by moving to s_2. Matters are more complex if the agents are uncoordinated. For all stages $t < 8$, $V^t\langle s_2, U \rangle < V^t\langle s_3, U \rangle$. So with 8 or fewer stages remaining, $a1$ should choose to "opt out" (choose b) at $\langle s_1, U \rangle$. For all stages $t > 10$, however, $V^t\langle s_2, U \rangle > V^t\langle s_3, U \rangle$ (e.g., $V^{11}\langle s_2, U \rangle = 22.5$ while $V^{11}\langle s_3, U \rangle = 20$.)[4] Thus, $a1$ should "opt in" to the CP at $\langle s_1, U \rangle$ if there are 12 or more stages remaining.

This example shows how knowledge of the state of the coordination mechanism allows the agents to make informed judgments about the (long term) benefits of coordination, the costs of miscoordination, and the odds of (immediate or eventual) coordination. Because of the cost of miscoordination (and its 50% chance of occurrence), the agents avoid s_2 with fewer than eight stages to go. The safe course of action is deemed correct. However, with eight or more stages remaining, they move from $\langle s_1, U \rangle$ to $\langle s_2, U \rangle$: the 50% chance of coordination not only provides the agents with a 50% chance at the reward of 10, but also with a 50% chance at least two more passes through s_4. The *long term* benefits of coordination (with a sufficient horizon) make the risk worthwhile when compared to the safe alternative.

It is important to note that the state of the coordination mechanism must be taken into account at each (system) state of the MMDP. For instance, though the state of the mechanism can have no influence on what the agents do at state s_3 (there is only one "choice"), it is relevant to determining the value of being at state s_3.

In general, reasoning with coordination mechanisms allows one to account for the factors mentioned above. Naturally, the tradeoffs involving long term consequences depend on the decision problem horizon or discount factor. The key factor allowing computation of value in this case is an understanding of the coordination mechanism used to (stochastically) select joint actions in the presence of multiple equilibria, and the ability to associate a value with any state of the MMDP (given the state of the mechanism). Shapley's stochastic games [14] provide a related sequential multiagent decision model with a a well-defined value for game states. This value, however, is a consequence of the zero-sum assumption, which removes the reliance of state value on the selection of a (stage game) equilibrium. In particular, it does not apply to fully cooperative settings where CPs arise.

4.2 Value Iteration with State Expansion

Value iteration can be revised to construct an optimal value function and policy based on any given coordination mechanism. A straightforward version is specified in Figure 3. We discuss several optimizations below.

A list *CP* of state-game CPs and associated mechanisms is kept as they are discovered. A CP exists if the set of optimal joint actions at a state/stage pair (the Q-values in step 3(a)i) induces a CP in the sense defined earlier. Notice that CPs are defined using the value function V^t, not immediate reward. We assume that each CP is associated with a state and the collection of actions involved in the optimal joint actions. Any state s_i with a CP will have the availability of actions involved in the CP restricted by the state of the mechanism. The set *PA*(s_i) is the set of actions permitted at s_i given the mechanism state—this may include randomization actions as well (if s_i has no CP, this set is just $\mathcal{A}$); and agents can only use permitted actions (step 3(a)i). If a CP is discovered among the maximizing (permitted) actions at s_i, a new mechanism C is introduced and the state is split and replaced by all pairs of states $\langle s_i, c \rangle$ (where c is some state of C).

[3] More precisely, $\langle s_2, U \rangle$ transitions to states $\langle s_4, A \rangle$ and $\langle s_4, B \rangle$ with probability 0.25 each.

[4] The values $V^t\langle s_2, U \rangle$ and $V^t\langle s_3, U \rangle$ are equal for $8 \leq t \leq 10$.

1. Let $V^0(s) = R(s)$ for all $s \in \mathcal{S}$.
2. Set $t = 0$; $\mathcal{S}^1 = \mathcal{S}^0 = \mathcal{S}$; $CP = \emptyset$.
3. While $t < T$ do:
   (a) For each $s_i \in \mathcal{S}^{t+1}$
      i. For each $a \in PA(s_i)$, compute
      $Q^{t+1}(s_i, a) = R(s_i) + \{\sum_{s_j \in \mathcal{S}^t} \Pr(s_i, a, s_j) V^t(s_j)\}$
      ii. Let $Opt^{t+1}(s_i)$ be the set of actions with max Q-value
      iii. If $Opt^{t+1}(s_i)$ induces a *new* CP at s_i, introduce mechanism C: (a) add C to CP; (b) replace s_i in $\mathcal{S}^{t+1}$ with states $\langle s_i, c\rangle$, where c ranges over states of C; (c) Recompute $Q^{t+1}(\langle s_i, c\rangle, a)$ for each new state (according to rules of C); (d) return to step ii. to check for new CPs.
      iv. Let $V^{t+1}(s_i) = \max_a Q^{t+1}(s_i, a)$ and $\pi^{t+1}(s_i)$ be any maximizing action (if s_i was split, do this for all $\langle s_i, c\rangle$).
   (b) $t = t + 1$; $\mathcal{S}^{t+1} = \mathcal{S} \times CP$.

Figure 3: Value Iteration with State Expansion

To illustrate, suppose the value function V^t induces the following choices at s_i:

	a	b	c
a	10	0	0
b	0	10	0
c	0	0	7

If randomization is used to coordinate on a/b, expected value is 5 (and the mechanism requires agents to randomize over their PIO-actions). In contrast, the Q-value of $\langle c, c\rangle$ is better than that of attempting to coordinate, thus the value of s_i is defined as 7 if the agents are uncoordinated (and 10 if they are coordinated). Notice that new CPs may be introduced at the same state and the process can be iterated.[5] In this problem, each state s is split into three states: $\langle s, A\rangle$ (agents have coordinated on joint action $\langle a, a\rangle$ at s_i), $\langle s, B\rangle$ (coordinated on $\langle b, b\rangle$), and $\langle s, U\rangle$ (have not coordinated w.r.t. a and b).

If a mechanism has been introduced for the same state and actions at an earlier stage, a new mechanism is not generated. Value (and policy choice) is defined by comparing the value of actions *not* involved in the CP and the value of behaving according to the rules of the mechanism (step 3(a)iv). At the next iteration all states are split according all mechanisms introduced, since this may be required to predict the value of reaching state s_i. If multiple CPs exist, each underlying system state is expanded many times in this (naive) algorithm.

Implicit in this discussion is that assumption that the transitions induced by a coordination protocol over the expanded state space are well defined: this will generally involve extending the underlying system dynamics by rules involving mechanism state evolution. The mechanism designer must provide such rules (as discussed in Section 3.2).

An important optimization is to have the algorithm only expand states with mechanisms whose state is required to predict value. This can be effected rather easily. If *system* state s_i transitions to state s_j, and s_j has been split in $\mathcal{S}^t$ to involve some mechanism in *CP*, s_i must be split in state $\mathcal{S}^{t+1}$. But if s_i moves only to states that are unaffected by some (or all) CPs, s_i need not be split using the state of those CPs. This

[5] However, the splitting must eventually terminate.

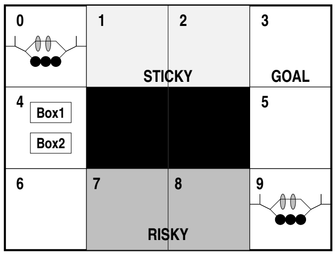

Figure 4: A More Complex Coordination Problem.

allows one to only refer to the state of a mechanism when it is necessary for predicting value: the state space need not be split uniformly.

Other optimizations of the algorithm are possible. For example, one can "cluster" together states of the coordination mechanism together that provide for the same optimal action and value at a given state. For instance, though FP has an infinite number of distinct states, for any finite number of stages-to-go, only a finite number of distinction are relevant (much like state abstraction methods used in MDPs and reinforcement learning [1, 3]). Finally, we note that modeling communcation protocols requires introducing communication actions, in addition to the state-splitting mechanism above.

4.3 Examples

We describe the results of applying the algorithm to several small test problems in this section. We focus here on the use of the simple randomization mechanism described above.

Testing a finite horizon version of the problem in Figure 1 shows that a single CP exists (at state s_2). The state space is eventually expanded so that each state is split into two (referring to coordination or lack of it at s_2). The optimal decision at $\langle s_1, U\rangle$ is to "opt out" with fewer than eight stages to go and "opt in" with eight or more stages remaining. The infinite horizon version of this problem gives rise to stationary policies. When the discount rate $\beta = 0.9$ (or higher), $a1$ "opts in" at $\langle s_1, U\rangle$; but for $\beta = 0.85$ (or lower), $a1$ "opts out" and avoids the CP—because of discounting, the delay in expected payoff of coordination ensures that "opting in" is not worth the cost. With $\beta = 0.9$, the value of opting in is 17.14 and opting out is 16.54 (assuming the agents act optimally thereafter), while with $\beta = 0.85$, the value of opting in is 8.62 and opting out is 9.36 (within tolerance 0.001).

A more complex example is illustrated in Figure 4. Two agents have independent tasks. Agent $a1$ moves box $b1$ and $a2$ moves $b2$ to the goal state repeatedly. Once a box is dropped at the goal, a reward is received and a new box appears in the original location (so the problem is a continuous, infinite horizon MMDP). While the objectives are independent, both agents are rewarded with the same constant reward whenever either of their boxes is delivered. The optimal policies are not independent however. The dark shaded region at the bottom is "risky:" if both agents are in the region, a large (variable) penalty is given. They must coordinate their moves to ensure that no more than one agent is in the risky region at any time. The agents' actions are stochastic: they can move in any (feasible) compass direction but with probability 0.1 they fail to move (they can also stay in place intentionally). Complicating the problem is the fact that the light shaded region is "sticky:" the agents' moves are more prone to failure (with varying probability). If stickiness is low, the optimal policy is for both agents to traverse the top of the grid repeatedly. But if stickiness is relatively high (or the problem is heavily discounted, making speedy delivery more important), one or both agents will want to traverse the risky area, in which case coordination is needed. The problem has 900 nominal states (though a number of these are not reachable) and 25 joint actions.

We give a brief summary of the results in this domain with the following specific parameter settings: a reward of 5 is given for each box delivered; a penalty of -20 is given whenever both agents are in the risky area; stickiness (the probability of not moving) is 0.7 in the sticky region; and $\beta = 0.95$. With these settings, the optimal joint policy (roughly) requires that one agent move across the top of the grid and one move across the bottom.[6] Generally, if an agent is closer to the top it will move across the top; but if both agents are close (and equally close) to the bottom, they must coordinate (since either could move to the top).

CPs arise at eight states of the MMDP. Thus there are eight coordination mechanisms needed to solve this problem, expanding the state space by a factor of 256 (no distinctions need be made among coordinated choices, so each mechanism has only two states). We focus on two MMDP states where CPs arise and their interaction: $s_{4,4} = \langle h1, h2, 4, 4\rangle$, where both agents are located at grid cell 4 each holding boxes, and $s_{6,6} = \langle h1, h2, 6, 6\rangle$, which is similar, but with both agents at location 6. The optimal joint policy at $s_{4,4}$ requires one agent to move up (to traverse the sticky region) and the other to move down (to traverse the risky region) on the way to the goal. The optimal policy at $s_{6,6}$ is similar: one agent should move up, the other right. The optimal joint value function has $\widetilde{V}(s_{4,4}) = 11.54$ and $\widetilde{V}(s_{6,6}) = 11.83$.

If the agents have coordinated at all other states where CPs arise, we have the following optimal values for the four states of the expanded MMDP corresponding to each of $s_{4,4}$ and $s_{6,6}$ (here we use u_4 to denote that the agents have not coordinated at $s_{4,4}$, and c_4 to denote that they have coordinated at $s_{4,4}$; similarly for $s_{6,6}$):

	u_4u_6	u_4c_6	c_4u_6	c_4c_6
$s_{4,4}$	10.4419	11.3405	11.5356	11.5356
$s_{6,6}$	7.1866	11.8339	7.3983	11.8340

In both states ($s_{4,4}$ or $s_{6,6}$), if the agents are uncoordinated, the optimal policy requires them to randomize, regardless of the state of the other coordination mechanism. Notice that the values for most of the expanded states where the agents are uncoordinated are less than the corresponding values for the optimal joint policy (which is identical to the expected values at the states where c_4c_6 holds), as expected. The one exception is at $s_{4,4}$: when c_4 holds, expected value is identical whether or not c_6 holds, since the optimal policy will never take the agents from $s_{4,4}$ to $s_{6,6}$. In contrast, when u_4 holds, the status of c_6 has a dramatic impact on expected value: if the agents are uncoordinated at $s_{4,4}$ they will randomize and with probability 0.25 both choose to move down (hence to $s_{6,6}$). Their state of coordination at $s_{6,6}$ is thus important to predicting expected value. Being uncoordinated at $s_{6,6}$ has very low value, since randomization has a good chance of moving both agents to the risky area—the risk is worthwhile, however, so randomization is the optimal choice at $s_{6,6}$.[7] Also when the agents are coordinated at $s_{6,6}$, the status of c_4 has a rather small effect on value. Because coordination at $s_{6,6}$ ensures that one agent takes the "sticky" route to the goal region, the agents get "out of synch" and the odds of them both reaching the pickup location (cell 4) at the same time (within a reasonable time frame) is quite small. Hence, whether or not the agents are coordinated at $s_{4,4}$ has little impact on expected value at $s_{6,6}$.

Randomization is an important aspect of this problem. If the agents were to choose from among their PIO actions independently, but deterministically, without reasoning about the consequences of miscoordination, they can end up in cycles that never reach the goal state.

5 Generalization of Coordination Decisions

One difficulty with the algorithm above is the potential for uninhibited state expansion, and the corresponding computational cost. In the simple experimental domain with two agents collecting boxes in a grid world, eight CPs occurred across the 900 problem states, requiring the state space to be increased by a factor of 256 (to 230,400 states). Fortunately, in many circumstances we can introduce a single coordination mechanism to deal with multiple, related CPs. In the grid problem, for example, once the agents coordinate at a state by one agent moving up and the other down, they can maintain these "roles" at other states exhibiting similar CPs.

We do not propose a method for constructing such generalizations automatically—this could use, say, generalization techniques from reinforcement learning [1]—but we illustrate potential benefits with the simple example shown in Figure 5. It is similar to the MMDP in Figure 1 except that miscoordination at s_2 has a larger penalty, and an analogous "low cost" CP has been added. If a single mechanism is used for both CPs (at s_2 and s_7), once coordination is attained at s_7, it is automatic at s_2. As in the original MMDP, with fewer than 12 stages-to-go, the optimal action at $\langle s_1, U\rangle$ is to "opt out" and take the sure reward 5. With 12 or more stages remaining, the optimal action at $\langle s_1, U\rangle$ is $\langle a, a\rangle$: the agents move to the low risk CP and try to coordinate there. Never do the agents move to s_2 in an uncoordinated state. Even though there is no immediate benefit to moving to s_7, it gives the agents an opportunity to "train," or learn to coordinate with minimal risk. Once they coordinate, they immediately exploit this learned protocol and choose $\langle a, b\rangle$ at $\langle s_1, C\rangle$ (thereby moving to $\langle s_2, C\rangle$). Reasoning about the long term

[6] If the penalty is negligible or if the stickiness is even higher, the agents will both tend to move across the bottom, perhaps with one waiting for the other. If the stickiness is negligible, then both agents will traverse the top of the grid.

[7] Though with higher penalties, it is not.

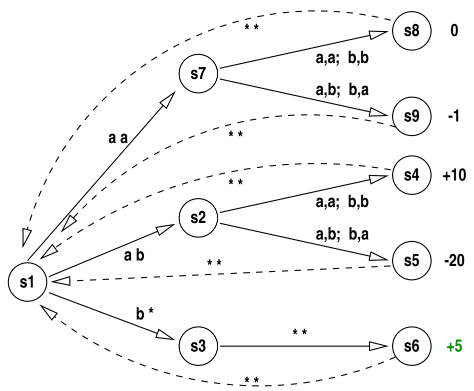

Figure 5: An MMDP with Similar Coordination Problems

prospects of coordination and its costs, the agents realize that risk-free training is worthwhile.

If we retain the original penalty of -10 at s_5, this reasoning fails: there is essentially less risk involved in training at the high stakes CP, so the agents will never move to s_7 to train.

The infinite horizon problem is similar. With a discount rate of 0.95, the optimal policy requires the agents to move to s_7 until they coordinate, at which point they repeatedly move to s_2. Interestingly, adding the "training states" increases the expected reward accrued by the agents. Without the training states, $V(\langle s_1, U\rangle) = 46.68$ since the agents accept the risk of getting several -20 rewards to ensure coordination. With the training states, they can learn to coordinate without the severe penalties, and $V(\langle s_1, U\rangle) = 49.57$.

6 Concluding Remarks

We have introduced a novel method of defining value functions (and consequently, optimal policies) for multiagent decision problems that accounts for specific means of coordination. We also defined a value iteration algorithm for computing optimal policies that recognizes and reasons about CPs.

Further experimentation is needed with other coordination mechanisms and their impact on policy value. We have described experiments in this paper using randomization, and have begun to investigate communication methods, and hope to explore other models like FP. We intend to introduce economic models (such as auctions) so that agents may integrate reasoning about their activity in markets into their decision processes. We must explore automated generalization methods further; it has the potential to substantially reduce the required number of mechanisms, alleviate computational difficulties, and increase objective policy value.

We would also like address the problem of designing robust, computationally effective and value-increasing coordination protocols in the framework. In a certain sense, such an undertaking can be viewed as one of designing *social laws* [15]. It is also related to the issues faced in the design of protocols for distributed systems and the distributed control of discrete-event systems [10]. But rather than designing protocols for specific situations, metaprotocols that increase value over a wide variety of CPs would be the target. The framework developed here can also help decide whether sophisticated protocols are worthwhile. For instance, a lexicographic protocol induces immediate coordination with a measurable (in our model) increase in expected value over (say) a randomization method. This increase can then be used to decide whether the overhead of incorporating a lexicographic convention (e.g., ensuring agents have common orderings) is worthwhile. Similar remarks can be applied to the design of agents (e.g., is communicative ability worthwile given the class of decision problems they will face).

Acknowledgements

This research was supported by the DARPA Co-ABS program (through Stanford University contract F30602-98-C-0214), NSERC Research Grant OGP0121843, and IRIS Phase-III Project BAC. Thanks to Ronen Brafman for discussion of these issues.

References

[1] D. P. Bertsekas and J.. N. Tsitsiklis. *Neuro-dynamic Programming*. Athena, Belmont, MA, 1996.

[2] C. Boutilier. Learning conventions in multiagent stochastic domains using likelihood estimates. *Proc. 12th Conf. on Uncertainty in Artif. Intel.*, pages 106–114, Portland, OR, 1996.

[3] C. Boutilier, T. Dean, and S. Hanks. Decision theoretic planning: Structural assumptions and computational leverage. *Journal of Artif. Intel. Research*, 1998. To appear.

[4] G. W. Brown. Iterative solution of games by fictitious play. In T. C. Koopmans, editor, *Activity Analysis of Production and Allocation*. Wiley, New York, 1951.

[5] D. Fudenberg and D. K. Levine. Steady state learning and Nash equilibrium. *Econometrica*, 61(3):547–573, 1993.

[6] D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*. MIT Press, Cambridge, MA, 1998.

[7] J. C. Harsanyi and R. Selten. *A General Theory of Equilibrium Selection in Games*. MIT Press, Cambridge, 1988.

[8] E. Kalai and E. Lehrer. Rational learning leads to Nash equilibrium. *Econometrica*, 61(5):1019–1045, 1993.

[9] D. K. Lewis. *Conventions, A Philosophical Study*. Harvard University Press, Cambridge, 1969.

[10] F. Lin and W. Wonham. Decentralized supervisory control of discrete-event systems. *Info. Sciences*, 44:199–224, 1988.

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

[12] D. Monderer and L. S. Shapley. Fictitious play property for games with identical interests. *Journal of Economic Theory*, 68:258–265, 1996.

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

[14] L. S. Shapley. Stochastic games. *Proc. National Academy of Sciences*, 39:327–332, 1953.

[15] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. *Proc. 10th National Conf. on Artif. Intel.*, pages 276–281, San Jose, 1992.

[16] G. Weiß. Learning to coordinate actions in multi-agent systems. *Proc. 13th International Joint Conf. on Artif. Intel.*, pages 311–316, Chambery, FR, 1993.