Multiagent Planning via Partial Coordination in Markov Games

Undergraduate Honors Thesis Submitted to Brown University's Department of Computer Science

Written By: Daniel Ritter
Thesis Advisor: Michael Littman
Reader: Amy Greenwald
Support: Mark Ho

April 2021

1 Introduction

Multiagent systems have a broad range of applications, and have seen extensive research and interest in recent years. Both learning and planning algorithms for multiagent environments have been proposed for problems ranging from community energy allocation [11] to traffic congestion [12] to robotic swarm control [9]. However, planning and learning in multiagent systems can be difficult, particularly because the number of possible joint actions increases exponentially with the number of agents. In this work, I present a framework for centralized multiagent planning in Markov games called a *model game*. This framework decomposes a multiagent environment into a set of models, each of which only considers some subset of the total set of agents. These models are then combined into a solution for the complete environment. After describing this approach, I show, using a simple gridworld game, that it can match the reward achieved by planning in the complete environment in cases where agent rewards are sufficiently 'decoupled', as explained in Section 5. I also show that this method can scale to games too large to plan in directly, and still achieves high reward if the rewards are sufficiently decoupled.

2 Related Work

Several prior papers have explored different methods of scaling/decomposing multiagent planning problems. Guestrin, Koller, and Parr [6] frame the problem as one large, single-agent Markov Decision Process (MDP) and then decompose the joint value function into local approximations that only depend on a few

agents. These local value functions are then used for rapid, decentralized planning. However, their method only applies to fully cooperative games. The same authors then extend this approach [7] to handle cooperation between different agents at different timesteps, at the expense of requiring hand-crafted value rules to determine the decomposition of joint values across agents. There has also been a great deal of similar work in classical, deterministic planning [2, 3, 1]. The work of Brafman and Domshlak [1], in particular, bears some similarity to my approach, as the authors define precise notions of coupling between agents and show a relationship between the complexity of planning and the coupling factor of a problem. However, their coupling measures are designed for deterministic planning problems, and do not have immediate analogues in the probabilistic case I deal with here. Lastly, there has been some work on characterizing coupling between players in the game-theory literature [13], which defines coupling as a predefined transformation of a player's rewards that makes each player 'dependent' on the outcomes of some other players. This formulation is useful and relevant, but also quite distinct from the one I use here, where coupling is intrinsic to the environment or joint policy of agents.

3 Background

This section describes the underlying models central to my work.

3.1 Markov games

A Markov game is a generalization of the standard Markov Decision Process that allows for multiple agents to act in one environment. Formally, a Markov game is a tuple $\mathcal{G} = \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, T, R \rangle$, where $\mathcal{I} = \{1, 2, ..., N\}$ is an ordered set of agents; $\mathcal{S}$ is a set of states; $\mathcal{A} = \bigtimes_i \mathcal{A}_i$ is the joint action space—the cross-product of individual agent action spaces; $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a transition function; and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^N$ is a joint reward function. Critically, the rewards and transitions are both functions of the joint actions of all agents, which means that the environment from any single agent's point of view is non-stationary. A joint policy for a Markov game is a mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that defines a distribution over joint actions for each state. An individual policy is a mapping $\pi_i : \mathcal{S} \to \Delta(\mathcal{A}_i)$ that defines a distribution over actions for some particular agent i.

We can express the value of a joint policy $\pi(\vec{a} \mid s)$ as:

$$V^\pi(s) = \sum_{\vec{a}} \pi(\vec{a} \mid s)[R(s, \vec{a}) + \sum_{s'} T(s' \mid s, \vec{a}) V^\pi(s')], \tag{1}$$

where $V^\pi(s)$ is vector of length $|\mathcal{I}|$ representing the expected payoff for each agent.

Similarly, the state–action value function for π can be written as:

$$Q^\pi(s, \vec{a}) = R(s, \vec{a}) + \sum_{s'} T(s' \mid s, \vec{a}) V^\pi(s'). \tag{2}$$

3.2 Solution Concepts

Because Markov games are non-stationary environments, and the actions of any one agent depend on those of the others, there is not a unique optimal policy as there is in a single agent MDP. Instead, the optimal behavior for an agent is defined in terms of its expectations for the behavior of other agents. One 'solution' to a Markov game can then be viewed as a policy that realizes a particular equilibrium among the agents. The relationship between an agent's expected reward and a particular kind of equilibrium can be formalized using the notion of a *solution concept*. A solution concept is a summary operator, $\bigotimes : (\mathcal{S} \times \mathcal{A} \to \mathbb{R}^{|I|}) \to \mathbb{R}^*$, that summarizes a state-action value function into a set of payoffs. We use the notation

$$\bigotimes_{\vec{a}} Q(s, \vec{a}),$$

where the solution operator returns payoffs based on the state–action values for the given state s. Solution operators are analogous to the max operator in single agent RL. In the multi-agent setting, however, solution operators may be associated with multiple possible values. As an example, the solution concept for a Nash equilibrium can be written as:

$$\texttt{Nash}_{\vec{a}} Q(s, \vec{a}) =$$
$$\{\mathbb{E}_{\vec{a} \sim \prod_i \pi_i} Q(s, \vec{a}) \mid \mathbb{E}_{a_i \sim \pi_i} Q_i(s, a_i, \vec{a}_{-i}) \geq \mathbb{E}_{a_i' \sim \pi_i'} Q_i(s, a_i', \vec{a}_{-i}), \forall i, a_i', -i\}.$$

There may be multiple policies that satisfy these constraints, which may also have different expected values. Selecting among equilibria within a particular class is then one of the additional challenges of multiagent environments. These solution-concept operators can be used in place of the max operator in the standard Bellman equations, giving rise to a variety of different multi-agent learning/planning updates. This procedure is equivalent to replacing the value function in Equation (2) with the solution operator applied to the Q-values of the next state. The Q/V functions defined for some solution concept $\bigotimes$ can then be written as

$$Q(s, \vec{a}) = R(s, \vec{a}) + \sum_{s'} T(s'|s, \vec{a}) \bigotimes_{\vec{a'}} Q(s', \vec{a}'), \tag{3}$$

and

$$V(s) = \bigotimes_{\vec{a}} [R(s, \vec{a}) + \sum_{s'} T(s' \mid s, \vec{a}) V(s')]. \tag{4}$$

3.3 Planning in Markov Games

Given a solution concept, $\bigotimes$, we can treat Equation 4 as an assignment rule and use it in a generalized form of value iteration. This algorithm is identical

to the single agent version, but each $V(s)$ is updated according to Equation 4 and $V(s)$ itself is a vector representing the values for each agent, rather than a scalar. The convergence guarantees for this algorithm depend on the particular solution concept and game. In past work [5, 10, 8], the authors have provided experiments with varying operators and some have proven convergence for certain kinds of operators or particular games (such as minimax solution concepts, coordination or zero-sum games). Greenwald and Hall [5] also showed empirical convergence on several games that do not fall into these categories.

Throughout the experiments in this work, I use a correlated equilibrium solution concept, which is not guaranteed to converge, but does so empirically in all games presented here. There are two main reasons I chose this solution concept. First, correlated equilibria are significantly more tractable to compute than a Nash equilibria in the general case. A correlated equilibrium can be solved as a linear program in polynomial time, while computing a Nash equilibrium is known to be PPAD-complete [4]. An approach like Friend-Or-Foe Q-Learning [10] can reframe the problem as a two-player, zero-sum game, which makes computing a Nash significantly more tractable. However, doing so requires assuming a priori that particular agents are either fully oppositional or fully cooperative. A correlated equilibrium avoids this assumption, at the cost of requiring access to (or at least estimates of) the Q values of all agents, rather than just one.

Secondly, the environments used in the experiments are variations of the Stag Hunt social dilemma, which has both payoff-dominant and risk-dominant equilibrium strategies. Because we are interested in coordination among agents to maximize reward, we want a solution concept that will converge to the payoff-dominant strategy. By solving for a correlated equilibrium, and specifying a utilitarian objective function (explained in more detail below), this solution concept will generally converge to the higher reward, cooperative strategy.

3.4 Correlated Equilibria as Linear Programs

Consider a normal form game with i players, $P = \{p_1 ... p_i\}$, and individual strategies $S_{p_1} ... S_{p_i}$. The set of joint strategies is then $S = \bigtimes_i S_{p_i}$. Let S_{-i} denote the set of joint strategies excepting one player p_i, and $\Delta(S)$ be a distribution over joint strategies. $\Delta(S)$ is then a correlated equilibrium if it satisifies the following constraints:

$$\sum_{s' \in S_{-i}} u^{p_i}_{s_j, s'} \Delta(s_j, s') \geq \sum_{s' \in S_i} u^{p_i}_{s_k, s'} \Delta(s_j, s'), \forall p_i \in P, \forall s_j, s_k \in S_{p_i} \qquad (5)$$

where $u^{p_i}_s$ is the utility for player i given joint strategy s. Essentially, no agent has any incentive to deviate from the equilibrium if every other agent follows it. These constraints can be transformed into a linear program of the following

form:

$$\sum_{s' \in S_{-i}} (u^{p_i}_{s_j,s'} - u^{p_i}_{s_k,s'})\Delta(s_j, s') \geq 0, \forall p_i \in P, \forall s_j, s_k \in S_{p_i}$$

$$\sum_{s \in S} \Delta(s) = 1$$

$$\Delta(s) \geq 0, \forall s \in S.$$

In addition to these constraints, the linear program can optimize for an objective function. There are several common objective functions, four of which are described by Greenwald and Hall [5]. In my experiments, I used social welfare as the objective function, which attempts to maximize the sum of all agent's utilities:

$$\max_{\Delta(s)} \sum_{p_i} \sum_{s \in S} \Delta(s) u^{p_i}_s.$$

In the context of a Markov game, we can treat a single state as a normal form game, using the state–action values of each agent as the utilities. An equilibrium for the state can be computed using the linear program described above. The expected values for each agent under that equilibrium can then be used to update their respective state–action values. It is important to note that the equilibrium solution to this linear program is not necessarily unique. As a result, solutions computed independently may end up being incompatible. Because of this fact, the equilibrium in my experiments is computed by a centralized planner, and then all agent's Q-values are updated according to that central equilibrium. This approach avoids coordination problems, where two agents might compute different but valid equilibria for the same state, leading to conflicting behaviors.

4 Model Games

As the number of agents in a Markov game grows, the size of the joint-action spaces increases exponentially. As a result, planning across many agents can rapidly become intractable. To avoid this intractability, I introduce the *model game* framework, which computes plans in a variety of partial models of the environment, and assigns the policies generated in those models to each agent. This approach allows for partial coordination among agents, enabling good performance in environments with separated tasks.

Formally, a model game, $\mathcal{MG} = \langle \mathcal{G}, \mathbb{G}_k, \bigotimes \rangle$, consists of a ground game $\mathcal{G}$, a set of *game models* $\hat{\mathcal{G}} \in \mathbb{G}_k$, and a solution concept, $\bigotimes$. The model game can be additionally augmented with a cost function, $C : \mathbb{G}_k^{|\mathcal{I}|} \to \mathbb{R}^{|\mathcal{I}|}$, which can represent costs associated with a particular choice of model for each agent (such as a penalty based on the complexity of different models).

A game model, $\hat{\mathcal{G}} \in \mathbb{G}_k$, is a partial model of the environment, $\langle \mathcal{I}_{\hat{\mathcal{G}}}, \mathcal{S}_{\hat{\mathcal{G}}}, \mathcal{A}_{\hat{\mathcal{G}}}, T_{\hat{\mathcal{G}}}, R_{\hat{\mathcal{G}}} \rangle$, where each element of the model is defined using up to k agents, $k \leq |\mathcal{I}|$. $\mathcal{I}_{\hat{\mathcal{G}}}$ is the set of agents included in the model, $\mathcal{S}_{\hat{\mathcal{G}}}$ is the set of states, $\mathcal{A}_{\hat{\mathcal{G}}}$ is the joint actions of the included agents, and $T_{\hat{\mathcal{G}}}, R_{\hat{\mathcal{G}}}$ are the modified transition and

$\mathcal{I}$	$\mathcal{S}$	$\mathcal{A}$	T	R
$\{1\}$	p_1	$\mathcal{A}_1$	$p_1 \times \mathcal{A}_1 \to \Delta(p_1)$	$p_1 \times \mathcal{A}_1 \to \mathbb{R}^1$
$\{2\}$	p_2	$\mathcal{A}_2$	$p_2 \times \mathcal{A}_2 \to \Delta(p_2)$	$p_2 \times \mathcal{A}_2 \to \mathbb{R}^1$
$\{3\}$	p_3	$\mathcal{A}_3$	$p_3 \times \mathcal{A}_3 \to \Delta(p_3)$	$p_3 \times \mathcal{A}_3 \to \mathbb{R}^1$
$\{1,2\}$	$\{p_1,p_2\}$	$\mathcal{A}_1 \times \mathcal{A}_2$	$\{p_1,p_2\} \times (\mathcal{A}_1 \times \mathcal{A}_2) \to \Delta(\{p_1,p_2\})$	$\{p_1,p_2\} \times (\mathcal{A}_1 \times \mathcal{A}_2) \to \mathbb{R}^2$
$\{1,3\}$	$\{p_1,p_3\}$	$\mathcal{A}_1 \times \mathcal{A}_3$	$\{p_1,p_3\} \times (\mathcal{A}_1 \times \mathcal{A}_3) \to \Delta(\{p_1,p_3\})$	$\{p_1,p_3\} \times (\mathcal{A}_1 \times \mathcal{A}_3) \to \mathbb{R}^2$
$\{2,3\}$	$\{p_2,p_3\}$	$\mathcal{A}_2 \times \mathcal{A}_3$	$\{p_2,p_3\} \times (\mathcal{A}_2 \times \mathcal{A}_3) \to \Delta(\{p_2,p_3\})$	$\{p_2,p_3\} \times (\mathcal{A}_2 \times \mathcal{A}_3) \to \mathbb{R}^2$

Table 1: Example models for a three-player gridworld. The agent set $\mathcal{I}$ is $\{1,2,3\}$. The state space $\mathcal{S}$ is $\{p_1,p_2,p_3\}$, where p_i is agent i's (x,y) position on the grid. Here, k is limited to 2, so the ground game is not included as a model.

reward functions. If the set of states $\mathcal{S}_{\hat{\mathcal{G}}}$ is different from $\mathcal{S}$, the states in the ground game, then an additional function $\mathcal{F}: \mathcal{S} \to \mathcal{S}_{\hat{\mathcal{G}}}$ is necessary to map states in $\mathcal{S}$ to a corresponding state in $\mathcal{S}_{\hat{\mathcal{G}}}$. This function allows the policy learned in the model to be used in the true environment. Table 1 shows an example of the game models for a simple three-player game. The state space is defined in terms of the positions of the agents, so the function $\mathcal{F}$ for that example would simply return the positions of the agents that are in the given model, leaving out the others.

The example in Table 1 also illustrates an important point implicit to the structure of the model game. For this framework to be applicable, the effects of individual actions on immediate transitions and rewards have to be separable from the effect of the joint actions as a whole. This is necessary so that the transition and reward functions for the models can be defined only in terms of subsets of agents. For the gridworlds in Table 1 and in the experiments, this separability is clear. The position of an individual agent in the grid is determined primarily by its own actions, barring collisions, and so the transition function can be adjusted to include only the effects of certain agents, which amounts to ignoring possible collisions for those agents left out of the model. For an arbitrary Markov game, though, there is not a general process for creating these model transition and reward functions. Defining that process requires some access to the internal structure of the dynamics for a given environment. In environments where this information is not available, another option is to assume a predefined, fixed policy $\pi_{-\mathcal{I}_{\hat{\mathcal{G}}}}$ for all agents not included in the model. The transition and reward functions for $\hat{\mathcal{G}}$ can then be defined as the original functions conditioned on $\pi_{-\mathcal{I}_{\hat{\mathcal{G}}}}$.

4.1 Planning

Planning in an environment with a model game involves three steps: planning on individual game models, estimating the true values of combinations of game models, and computing a distribution over combinations of models.

4.1.1 Planning on Individual Game Models

The first step is to plan on each game model $\hat{\mathcal{G}} \in \mathbb{G}_k$. Because the game models are defined in terms of subsets of agents, the total cardinality of $\mathcal{G}_k$ will be $\sum_{j=1}^{k} \binom{|\mathcal{I}|}{j}$. Planning on a game model can be done using any planning algorithm for Markov games, as each game model is just a simplified version of the ground game. For each model, planning will generate a joint policy $\pi_{\hat{\mathcal{G}}} : \mathcal{S}_{\hat{\mathcal{G}}} \rightarrow \mathcal{A}_{\hat{\mathcal{G}}}$, defining actions for agents in $\mathcal{I}_{\hat{\mathcal{G}}}$. At the end of this step, each game model will have an associated joint policy.

4.1.2 Estimating Joint Policy Values

Because the game models leave out some agents present in the ground game, there is no guarantee that a policy that achieves a high reward in $\hat{\mathcal{G}}$ will also achieve that same reward in $\mathcal{G}$. To determine which model to use for each agent in the ground game, then, we have to estimate the true value of each model policy $\pi_{\hat{\mathcal{G}}}$ in $\mathcal{G}$.

To estimate this value, we have to look at each combination of models available to the agents (excluding those where an agent's model does not include the agent itself). Each agent has a total of $\sum_{j=0}^{k} \binom{|\mathcal{I}|-1}{j-1}$ models to choose from (the subsets up to size k of $\mathcal{I}$ that include a particular agent). The set of joint model combinations is then the cross product of those sets, giving a total of $(\sum_{j=0}^{k} \binom{|\mathcal{I}|-1}{j-1})^{|\mathcal{I}|}$ possible combinations. While this set is quite large, the estimated return of each joint strategy can be evaluated quickly. For each combination, I estimate its value with Monte Carlo sampling, averaging over a large number of trials in the ground game.

4.1.3 Computing a Model Equilibrium

Given the estimated returns for each joint model combination, the choice of models for an agent can be viewed as a normal-form game, where the set of individual strategies is the collection of the agent's available models. The estimated values are then used as the payoffs in this game. To determine a distribution over the joint model combinations, compute an equilibrium over this stage game. Any equilibrium concept could be applied, but in the experiments we use a correlated equilibrium with a utilitarian objective function, to maximize the reward among agents who can cooperate.

To apply the final result to the ground game for testing, we sample from the joint-model equilibrium, and then use the policy from each agent's assigned model to determine their actions.

5 Measures of Task Coupling

The performance of the model game's solution is dependent on the structure of the rewards in the environment. If joint policies that achieve high reward require the cooperation of many agents, that is, if the rewards are highly coupled,

then the model game's solution will not necessarily provide high reward. In the opposite case, where cooperation is only necessary between a few agents, the model game's solution can perform well. Formalizing this notion of coupling is then useful in analyzing the circumstances and environments in which the model game will find high performing solutions. However, finding a single way to completely express this kind of agent interaction is difficult. Instead, I describe a potential way to formalize some aspects of coupling, and describe the benefits and drawbacks of it.

One approach to capturing the importance of coordination between agents is to define it using the relationship between the values of the model game and ground game solutions. We can write this measure as a minimization problem over k, where k is the maximum number of agents that can be included in one model:

$$\min k : V^{\pi_{\text{ground}}}(s_0) = V^{\pi_{M_k}}(s_0), \tag{6}$$

where π_{M_k} is the joint policy produced by the model game when limited to models of k agents or fewer, and π_{ground} is the joint policy produced by planning in the ground game. The coupledness of an environment (and whatever solution concept has been chosen) is then defined as the smallest value of k, that is, the minimal number of coordinating agents, that can still match the values of the ground-game solution.

Using the same quantities, we can also write a related form of Equation 6 based on the concept of the *price of anarchy* in standard game theory. The price of anarchy is defined for a set of equilibria S_{eq}, set of strategies S, and objective function f as

$$\frac{\max_{s \in S} f(s)}{\min_{s \in S_{eq}} f(s)}. \tag{7}$$

This ratio captures the amount of lost utility (based on f) between the optimal centralized strategy and the worst-case strategy in some set of equilibria. Substituting in values for the model and ground games in place of f, we can write a similar ratio between the ground game and model game:

$$\max_i \frac{\max_{\pi_{\text{ground}}} V_i^{\pi_{\text{ground}}}(s_0)}{\min_{\pi_{M_k}} V_i^{\pi_{M_k}}(s_0)}. \tag{8}$$

This version represents the proportion of reward lost between the ground game and model game solutions for a particular value of k and agent i, instead of measuring the minimal k needed to match the ground game value. By maximizing across i, we get the worst case loss of reward for any agent in the environment. Rather than being defined with respect to an optimal central controller, as the price of anarchy is, the ratio is specified in terms of the equilibrium solutions available in the ground game. Both these formulations provide methods for quantifying the coupledness of a particular environment/solution concept pair and relating it to the rewards produced by the model game. However, computing either one requires access to the set of valid ground game solutions, which

can be intractable to compute. Different notions of coupledness that are policy agnostic, or approximations of these quantities that are tractable to compute, would then be more useful in assessing the potential efficacy of a model game.

6 Experiment 1

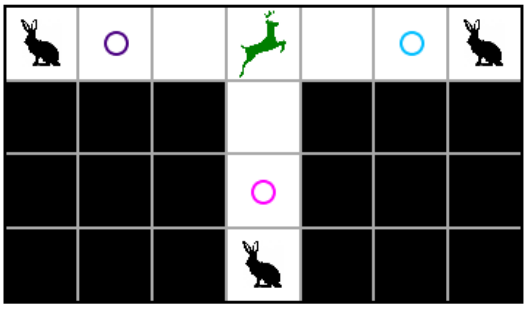
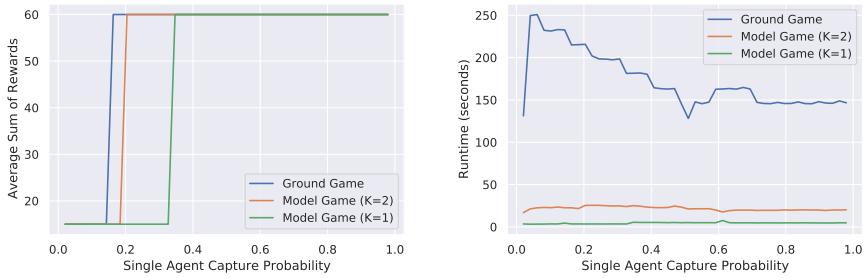


Figure 1: Experiment 1 Gridworld

This experiment involves a very simple gridworld environment based on the stag-hunt social dilemma in game theory. In this environment, pictured in Figure 1, the agents (shown as colored circles), can move into adjacent, non-diagonal squares. If any agent moves into the same square as a hare or a stag, they 'capture' it and receive a reward, causing the episode to terminate. Both the stag and the hare are fixed in place. If an agent captures a hare, it receives 5 points. If one captures a stag, it receives 20 points. Agents in the squares immediately surrounding a stag will also receive the 20 points if any one agent captures it. Lastly, if two agents try to move into the same square, the collision is resolved probabilistically, with each agent having a 50 percent chance to move to the square or stay in place.

While moving onto a square containing a hare is deterministic, barring collisions, moving onto a square containing a stag is not. When an agent attempts to capture the stag, it only succeeds in moving into the stag's square with some probability p, defined as part of the environment. This probability roughly correlates with the coupledness of rewards in the environment. If p is quite small, then it requires many agents attempting to capture the stag for them to succeed. If p is large, then a single agent is like to capture it alone.

By varying p, we can make the environment more or less coupled, and examine the performance of the model game across different conditions. The results of this experiment are shown in Figure 2a. The model game was used with k values of 1 and 2. As a baseline, I used the result of planning directly in the



(a) Sum of Undiscounted Rewards vs. Capture Probability

(b) Runtime vs. Capture Probability

Figure 2: Results for Experiment 1. Value iteration was run until convergence on each game model and the ground game ($\gamma = .5$). Each combination of game models was run for 500 episodes in the ground game to estimate their true returns. The average rewards were estimated from 200 test episodes after planning was complete.

ground game. For each model, and for the ground environment, value iteration with a correlated equilibrium solution concept was used for planning. At very low probabilities, it is ideal for each agent—in both the ground game and the game models—to go directly to its nearby hare because the odds of any one agent getting the stag are so low that the expected value of going for the stag, even with all 3 agents cooperating, is still lower than that of capturing the hare. The opposite holds in the high probability domain, where, regardless of whether or not there is cooperation, it is still ideal for each agent to go to the stag.

The interesting region lies from around $p = .20$ to $p = .40$, and demonstrates the differences in planning between the ground game and model game. The ground game planner switches to a cooperative strategy around $p = .20$, seen in the jump in the summed rewards from 15 to 60. At this point, all 3 agents cooperating can capture the stag consistently enough to outweigh the 5 points they can gain from the hare. The model games, however, do not switch to a cooperative strategy until a higher probability. The reason for this difference is because, from the model game's perspective, there is only ever coordination between k agents, while the ground game can coordinate between all of the agents. The $k = 1$ and $k = 2$ model games then change to a cooperative strategy only when the probability is high enough for 1 or 2 agents, respectively, to capture the stag consistently. In sufficiently decoupled conditions, the model game can match the performance of the ground game, and do so in a drastically smaller amount of time. Figure 2b shows the runtime for each probability value, and the model game is significantly faster than the ground game in every case. For environments where tasks are known to be relatively decoupled, planning with a model game can provide a boost in efficiency with minimum loss in value.

6.1 Coupling

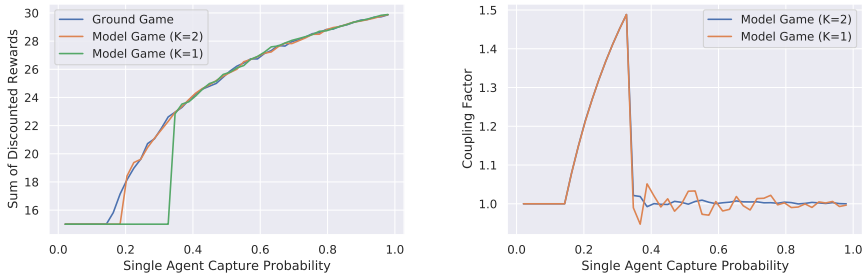
Because this three player environment is small enough to compute the ground game solution, we can evaluate the coupling formulation described in Section 5. Figure 3a shows the discounted sum of rewards for the ground and model games. The value is equivalent to $V^{\pi}(s_0)$, which can be written as $\sum_{t=0}^{\infty} \gamma^t \times R_t$. The formulation in Equation 6 then corresponds to the sections of Figure 3a where the model game matches the expected value of the ground game. The games with probabilities from .00 to around .15, for example, would have a coupling factor of 1 because the $k = 1$ model game can achieve the same value as the ground game. The games with probabilities from .15 to .20 would have a coupling factor of 3, because neither model game achieves the same value. Those from .20 to .40 would have a value of 2, since the $k = 2$ model game matches the ground game, but $k = 1$ does not. After .40, the coupling factor would be 1, as $k = 1$ is again able to match the ground game value.

Figure 3b shows the other formulation based on the price of anarchy (Equation 8).[1] The ratio is one (or very close to it) at all the regions where the model games match the rewards of the ground game, and spikes severely in the probability region where the ground game outperforms the model games, showing how those games are more tightly coupled and require more coordination. The drop down to one between .30 and .40 happens at the point where it becomes worthwhile even for a single agent to go for the stag. At that point, the policy for every model is to go for the stag. No matter what combination of models is picked, then, every agent will go for the stag, and the resulting value will match that of the ground game solution.

7 Experiment 2

Because model games only consider partial models of the ground game, which are generally logarithmically smaller than the true environment (assuming $k < |\mathcal{I}|$), they can scale to significantly larger games that include greater numbers of agents. Experiment 2 illustrates this idea. The environment dynamics, goals, rewards and planning process are the same as described in Experiment 1, but the game itself is significantly larger (shown in Figure 4). There are 5 agents, rather than three, and 22 possible positions, rather than 10. The optimal strategy (in terms of maximizing the sum of rewards) is for each pair of agents on the left and right to try and capture their respective stags, and for the agent in the middle to try and capture either stag. The non-cooperative strategy is for each agent to capture its respective hare. Experiment 1 had around 700 states, with

[1] Technically, the denominator in Equation 7 involves enumerating the set of correlated equilibria and then separately maximizing over those for a particular agent's value function. Here, I simply used the equilibrium computed to maximize the sum of all agent's rewards. However, this game is simple enough that the strategy that maximizes the sum of rewards is also optimal for each individual agent. The values in the numerator are based on the estimated payoffs in the true environment, which explains why there is a small amount of variance in the graph.



(a) Sum of Discounted Rewards vs. Capture Probability

(b) Price of Anarchy Coupling vs. Capture Probability

Figure 3: Discounted Sum of Rewards ($\gamma = .5$) and Coupling Factor for Experiment 1

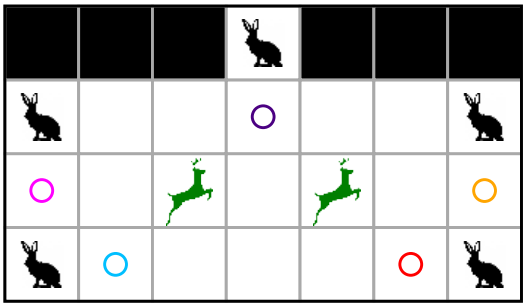
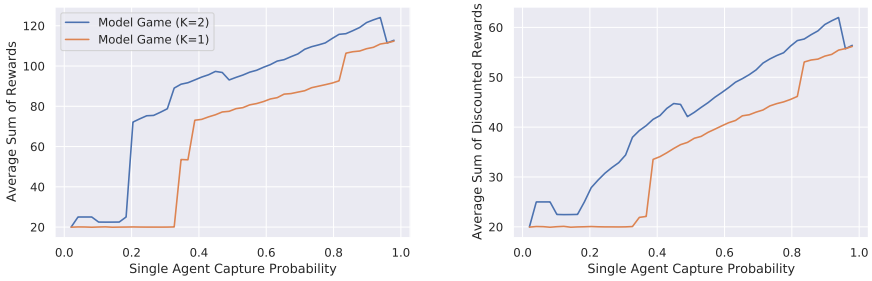


Figure 4: Experiment 2 Gridworld

a joint action space size of 125. This game has approximately 3.16 million states and a joint action space size of 3125. Planning in the complete environment, given its size, is completely intractable. While we cannot compare to the ground game solution in this case, we can see the same trend in the sum of rewards for the model game with k set to 1 and 2. These results are shown in Figure 5. In the lower probability regions (excepting very low probabilities where even the ground game solution would go for the hare), both model games achieve relatively low reward, choosing an uncooperative strategy. As the probability increases, and the environment becomes less coupled, the rewards increase, as it becomes more and more worthwhile for agents to cooperate, even within game models. Given significantly decoupled rewards, then, a model game can handle much larger games, with far greater numbers of agents, by breaking down the exponentially large joint-action space into relevant subgames.



(a) Average Sum of Undiscounted Rewards vs. Probability

(b) Average Sum of Discounted Rewards vs. Probability

Figure 5: Average Sum of Rewards vs. Capture Probability for Experiment 2

8 Discussion

The model-game planning framework can produce effective solutions in large, partially-cooperative environments, provided that tasks/rewards are sufficiently separated and require coordination between small numbers of agents. However, this work has shown only a basic version of this framework to showcase its costs and benefits. There are numerous potential extensions and improvements to this idea that are worth briefly sketching out.

For example, the model game currently requires a centralized planner that computes plans for every model and assigns a model to each agent. However, many multiagent problems require decentralized planning methods, where agents plan independently to solve some common task. Extending model games to operate in a decentralized fashion would broaden their applicability to a significantly larger class of problems. This extension could take several different forms. One option would be to have each agent maintain a set of individual game models, and have some kind of message-passing scheme to coordinate on which model to use (or even to transfer models between agents). This proposal would create some problems with coordinating plans between agents, like those mentioned in the section on correlated equilibria, but could still work with certain equilibria or games that converge in decentralized settings.

The current approach for estimating the values of joint combinations of models could also be made more efficient. The space of possible model combinations grows quite rapidly, and so evaluating every one (which is the current approach) could be intractable in very large games. Methods for pruning the set of game models could avoid this problem. As an example, in the stag-hunt gridworld, agents would likely only need to coordinate with agents close by in the grid, and so limiting game models to those where two agents close to each other are included might be an effective heuristic.

Model games can be used for scaleable and efficient planning in environments with well-separated tasks. Their efficiency comes from considering only partial

coordination between agents, bypassing planning in the complete environment. Using a gridworld game based on the stag-hunt social dilemma, I showed how the performance of a model game compared to planning in the true environment, as well as how the model game can scale well beyond standard planning algorithms. Although there is not yet a clear formal measure for game coupledness separate from an already learned policy, for environments where this task separation seems to hold, model games can be an efficient and effective solution.

References

[1] Ronen Brafman and Carmel Domshlak. "From One to Many: Planning for Loosely Coupled Multi-Agent Systems". In: Jan. 2008.

[2] Ronen Brafman et al. "Planning Games". In: *Proceedings of the International Joint Conference on Artificial Intelligence*. 2009.

[3] Ronen Brafman et al. "Transferable Utility Planning Games". In: Jan. 2010.

[4] Constantinos Daskalakis, Paul Goldberg, and Christos Papadimitriou. "The Complexity of Computing a Nash Equilibrium". In: *SIAM Journal on Computing* (2008).

[5] Amy Greenwald and Keith Hall. "Correlated-Q Learning". In: *Proceedings of the Association for the Advancement of Artificial Intelligence*. 2003.

[6] Carlos Guestrin, Daphne Koller, and Ronald Parr. "Multiagent Planning with Factored MDPs". In: *In NIPS-14*. The MIT Press, 2001, pp. 1523–1530.

[7] Carlos Guestrin, Shobha Venkataraman, and Daphne Koller. "Context-Specific Multiagent Coordination and Planning with Factored MDPs". In: *Eighteenth National Conference on Artificial Intelligence*. Edmonton, Alberta, Canada: American Association for Artificial Intelligence, 2002, pp. 253–259. ISBN: 0262511290.

[8] Junling Hu and Michael Wellman. "Nash Q-Learning for General-Sum Stochastic Games". In: *Journal of Machine Learning Research* (2003).

[9] Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. *Guided Deep Reinforcement Learning for Swarm Systems*. 2017. arXiv: 1709.06011 [cs.MA].

[10] Michael Littman. "Friend-Or-Foe Q-Learning in General-Sum Games". In: *Proceedings of the International Conference on Machine Learning*. 2003.

[11] Amit Prasad and Ivana Dusparic. "Multi-agent Deep Reinforcement Learning for Zero Energy Communities". In: *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)* (2019), pp. 1–5.

[12] Sejoon Lim and D. Rus. "Stochastic distributed multi-agent planning and applications to traffic". In: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 2873–2879. DOI: 10.1109/ICRA.2012.6224710.

[13] Mo Wei and Jose Cruz. "Role of cooperation in coupling game theory". In: *International Journal of Control - INT J CONTR* 80 (Apr. 2007), pp. 611–623. DOI: 10.1080/00207170601126701.