Increasing Security through Communication and Policy Randomization in Multiagent Systems

Praveen Paruchuri, Milind Tambe, Fernando Ordóñez, Sarit Kraus*
University of Southern California, Los Angeles, CA 90089,
{paruchur, tambe, fordon}@usc.edu
* Bar-Ilan University, Ramat-Gan 52900, Israel, sarit@cs.biu.ac.il

Abstract

Despite significant recent advances in decision theoretic frameworks for reasoning about multiagent teams, little attention has been paid to applying such frameworks in adversarial domains, where the agent team may face security threats from other agents. This paper focuses on domains where such threats are caused by unseen adversaries whose actions or payoffs are unknown. In such domains, action randomization is recognized as a key technique to deteriorate an adversarys capability to predict and exploit an agent/agent teams actions. Unfortunately, there are two key challenges in such randomization. First, randomization can reduce the expected reward (quality) of the agent team's plans, and thus we must provide some guarantees on such rewards. Second, communication within an agent team can help in alleviating the miscoordination that arises due to randomization, but communication is a scarce resource in most real domains. To address these challenges, this paper provides the following contributions. First, we recall the Multiagent Constrained MDP (MCMDP) framework that enables policy generation for a team of agents where each agent may have a limited (communication) resource. Second, since randomized policies generated directly for MCMDPs lead to miscoordination, we introduce a transformation algorithm that converts the MCMDP into a transformed MCMDP incorporating explicit communication actions. Third, we develop a non-linear program with non-convex constraints for the transformed MCMDP that randomizes team policy while attaining a threshold reward without violating the communication constraints. Finally, we experimentally illustrate the benefits of our work.

1 Introduction

Decision-theoretic models like the Multiagent Markov Decision Problem's (MMDPs) [7], Decentralized Markov Decision Problem's (Dec-MDPs) [5] and the Decentralized Partially Observable MDP's (Dec-POMDPs) [18] have been successfully applied to build agent-teams acting in uncertain environments. However, these teams must often work in an adversarial environment. For example, when patrolling, UAV (Unmanned Air Vehicles) teams might often be watched by adversaries such as unobserved terrorists [22] or robotic patrol units trying to detect intruders in physical security sites [8, 16].

Security, commonly defined as the ability of the system to deal with intentional threats from other agents [1], becomes a critical issue for these agent teams acting in such adversarial environments. In many realistic domains the agents cannot even explicitly model the adversary's actions and capabilities or its payoffs. However, the adversary can observe the agents' actions and exploit any action predictability in some unknown fashion. Consider the team of UAVs [3] monitoring a region undergoing a humanitarian crisis. Adversaries may be humans intent on causing some significant unanticipated harm — e.g. disrupting food convoys, harming refugees or shooting down the UAVs — the adversary's capabilities, actions or payoffs are unknown and difficult to model explicitly. However, the adversaries can observe the UAVs and exploit any predictability in UAV surveillance, e.g. engage in unknown harmful actions by avoiding the UAVs' route.

Given our assumption that the agent team acts in an adversarial domain where the adversary cannot be explicitly modeled, policy randomization becomes crucial for the teams to avoid the action predictability [22]. We also assume that the agent team is acting in accessible environments and hence can be modeled using MMDPs. We further make the following three assumptions about the adversary in our work. First, we assume that the adversary can also observe the agents' state exactly. The second assumption is that the adversary knows the agents policy, which it may do by learning over repeated observations. Policy randomization would then ensure that even if the adversary knows the agents' state exactly at each instant and also the agents' policy from that state, the adversary would still be unable to predict the agent's action correctly and hence significantly cut down the chances of unanticipated harm. The third assumption is that communication is encrypted and also communication being a private resource for the team, is unobservable for the adversary i.e. communication is safe. Even if the communication is observable it can be easily masked by using simple deception techniques like sending some meaningless data for non-communication acts, thus making it safe [26].

Although policy randomization avoids action predictability, simply randomizing an MDP policy as mentioned above can degrade the expected team reward significantly and hence we face a randomization-reward tradeoff. The difficulty in generating randomized policies that provide the appropriate randomization-reward tradeoff is further exacerbated by the fact that randomization creates miscoordination in team settings. Furthermore, for real world teams, resources are severely limited and teams must often adhere to certain resource constraints, e.g., when communicating, the UAV team must consume only limited communication bandwidth [21] allocated for it. Constraints involving averaging a quantity, in general, are soft constraints because as long as the average is maintained, there is no hard bound on the resource amount to be used at each timestep [25, 9]. In our example, we model bandwidth as a soft constraint [21] because exceeding bandwidth in any single run is not a disaster; but if the team consumes more than its bandwidth limit on an average, it jeopardizes the communications of other agents on the same network. The importance of such soft constraints is seen by continued work in operations research literature on constrained MDPs (CMDPs) that reason about expected resource consumption [2].

Our work focuses on increasing security using policy randomization for agent teams with bandwidth constraints while ensuring fixed reward thresholds. Although, such randomized policies have occurred as side effect [2] and turn out to be optimal in some stochastic games [17], work on intentional policy randomization has received focus only recently. For example, [22] intentionally randomizes MDP/POMDP policies for increasing security but their work provides solution for a specific case where there is no communication. Work that has been done on developing agent teams with resource constraints [2, 21, 10] has not paid attention to the issue of security in such teams. To address these concerns, we therefore solve a multicriterion problem that maximizes the team policy randomization while ensuring that the average bandwidth consumption is below a threshold and the team reward is above a threshold. The problem we solve is general enough and other soft resource constraints can be considered without any modifications to the structure of the problem.

This paper provides three key contributions to solve the problem described. First, we recall MCMDP as multiagent MDP framework where agents reason not only about their rewards but also about resource constraints. We then introduce the entropy metric to quantify policy randomization for MCMDP and formulate a nonlinear program that maximizes policy randomization while ensuring threshold rewards. We then identify a novel coordination challenge that occurs due to randomized policies in multiagent settings, i.e agents miscoordinate if there are randomized policies in team settings. Second, we provide a novel polynomial time transformation algorithm that converts the MCMDP into a transformed MCMDP incorporating explicit communication actions to alleviate such miscoordination. Third, we developed a non-linear program with non-convex constraints for the transformed MCMDP that randomizes team policy while attaining a threshold reward without violating the communication constraints. We further show that the value of entropy for MCMDP and the transformed MCMDP remains the same for the same policy, thus showing that our transformation is correct. In our experimental section, we show results after evaluating the new non-linear program we developed for the transformed MCMDP. The rest of the paper begins with MCMDP and a non-linear program for it that captures policy randomization. An automated method of transformation is provided that converts this MCMDP to a transformed MCMDP. We then provide our solution approach to solve this new model. We then briefly describe the various transformations possible. Lastly, we provide experimental results that clearly show the interdependence between the important factors of our domain namely policy randomization, reward and bandwidth.

2 Randomization: MCMDP

MCMDP is a useful tool for users, providing a layer of abstraction to model agent-teams with resource constraints in uncertain domains. For purposes of this paper, the only resource being modeled is the bandwidth. We first recall a 2-agent MCMDP for expository purposes. A 2-agent MCMDP is defined as a tuple, $\langle S, A, P, R, C1, C2, T1, T2, N, Q \rangle$ where: S is a finite set of states. Given two individual actions a$_l$ and a$_m$ of the two agents in our team, the team's joint action $\hat{a} = (a_l, a_m) \in A$ i.e A represents the set of all possible joint actions. $P = [p_{ij}^{\hat{a}}](\equiv p(i, \hat{a}, j))$ is the transition matrix, providing the probability of transitioning from a source state i to a destination state j, given the team's joint action $\hat{a}$, $R = [r_{i\hat{a}}]$ is the vector of joint rewards obtained when an action $\hat{a}$ is taken in state i. $C1 = [c1_{i\hat{a}k}]$ is the vector to account for cost of resource k when action $\hat{a}$ is taken in state i by agent 1 i.e it models cost for individual resource of agent 1. (C2 is similarly defined.) $T1 = [t1_k]$ and $T2 = [t2_k]$ are vectors of thresholds on the availability of the individual resources k for agents 1 and 2 respectively. $N = [n_{i\hat{a}}]$ is the vector of joint communication costs incurred by the agents when an action $\hat{a}$ is taken in state i. Q is a threshold on communication costs that can be used by the team of agents. A MCMDP is thus similar to a CMDP [2] with multiple agents.

2.1 Randomization due to resource constraints

The goal in a MCMDP is to maximize the total expected reward, while ensuring that the expected resource (bandwidth here) consumption is maintained below threshold. Formally, this requirement can be stated as a linear program, extending the linear program for CMDPs [9] to a two agent case, as shown below. $x_{i\hat{a}}$ is the expected number of times an action $\hat{a}$ is executed in state i and α_j is the initial probability distribution over the state space.

$$\begin{array}{lll} \max & \sum_i \sum_{\hat{a}} x_{i\hat{a}} r_{i\hat{a}} & \\ \text{s.t.} & \sum_{\hat{a}} x_{j\hat{a}} - \sum_i \sum_{\hat{a}} x_{i\hat{a}} p_{ij}^{\hat{a}} = \alpha_j \quad \forall j \in S & \\ & \sum_i \sum_{\hat{a}} x_{i\hat{a}} c1_{i\hat{a}k} \leq t1_k, & \sum_i \sum_{\hat{a}} x_{i\hat{a}} c2_{i\hat{a}k} \leq t2_k \\ & \sum_i \sum_{\hat{a}} x_{i\hat{a}} n_{i\hat{a}} \leq Q, & x_{i\hat{a}} \geq 0 \quad \forall i \in S, \hat{a} \in A \end{array} \tag{1}$$

If x^* is the optimal solution to (1), optimal policy π^* is given by (2) below, where $\pi^*(s,\hat{a})$ is the probability of taking action $\hat{a}$ in state s.

$$\pi^*(s,\hat{a}) = \frac{x^*(s,\hat{a})}{\sum_{\hat{a} \in A} x^*(s,\hat{a})} \; . \tag{2}$$

It turns out that π^* is a randomized policy in the above case due to the resource constraints. Since, bandwidth is the only resource under consideration and is modeled as a team resource we set the individual resources and their thresholds i.e., $C1, C2, T1, T2$ to zero. Such randomization leads to miscoordination in team settings as shown in section 2.2. Further, the randomization occurred as sideeffect due to the communication constraint and hence not optimized for policy randomness as needed by our domain.

2.2 Miscoordination: Effect of Randomization in Team Settings

For illustrative purposes, Figure 1 shows a 2 state MCMDP with two agents A and B with actions a_1, a_2 and b_1, b_2 respectively, leading to joint actions $\hat{a1} = (a_1, b_1)$, $\hat{a2} = (a_1, b_2)$, $\hat{a3} = (a_2, b_1)$, $\hat{a4} = (a_2, b_2)$. We also show the transition probabilities, rewards and communication costs for each of the actions. The optimal policy for this MCMDP is to take joint actions $\hat{a}1$ and $\hat{a}4$ with 0.5 probability. Suppose, agent A chooses its own actions such that $p(a1) = .5$ and $p(a2) = .5$, based on the joint actions. However, when A selects $a1$, there is not guarantee that agent B would choose $b1$. In fact, B can choose $b2$ due to its own randomization. Thus, the team may jointly execute $\hat{a2} = (a_1, b_2)$, even though the policy specifies p($\hat{a}$2) = 0. Therefore, a MCMDP, a straightforward generalization of a CMDP to a multiagent case, results in randomized policies, which a team cannot execute without additional coordination. One simple solution is to add a communication action before each joint action. However, forcing a communication action before every single action can violate communication constraints, since communication itself consumes resources. Thus, a solution that limits communication costs is essential. Further, equation 1 maximizes the expected reward obtained for the MCMDP while we are interested in maximizing the randomness of our policy. Below, we first introduce an entropy measure to quantify randomness and then develop an algorithm that maximizes the measure while we threshold on reward and constrain the communication. However, the problem of miscoordination still remains which we solve in section 3.

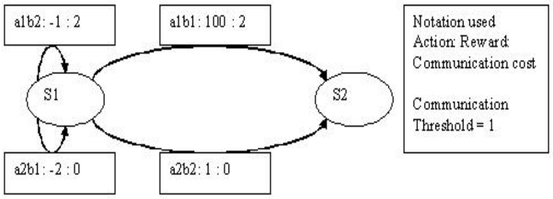


Figure 1: Simple MCMDP [(a1b1:100:2)- Action a1b1 gives reward 100 with communication cost 2]

2.3 Randomness of a policy

For a discrete probability distribution $p_1, p_2, ..., p_n$ the only function, upto a multiplicative constant, that captures the randomness is the entropy, given by the formula $H = -\sum_{i=1}^{n} p_i \log p_i$ [27]. For quantifying the randomness of a single agent MDP policy, we borrow the weighted entropy concept developed in [22]. For purposes of clarity we reproduce the formula here (π is the CMDP policy which defines a probability distribution over actions for each state s)-

$$H_W(x) = -\sum_{s \in S} \frac{\sum_{\hat{a} \in A} x(s,\hat{a})}{\sum_{j \in S} \alpha_j} \sum_{a \in A} \pi(s,a) \log \pi(s,a) = -\frac{1}{\sum_{j \in S} \alpha_j} \sum_{s \in S} \sum_{a \in A} x(s,a) \log \left(\frac{x(s,a)}{\sum_{\hat{a} \in A} x(s,\hat{a})} \right) .$$

Extending this formula for a 2-agent MCMDP is quite straightforward in the sense that instead of calculating the weighted entropy over a single agent policy we calculate it over the joint policy of both the agents for the 2-agent MCMDP. Hence, in the weighted entropy formula above, π refers to the joint policy of the agents.

2.4 Intentional Randomization: Maximal entropy solution

We can now obtain maximal entropy policies with a threshold expected reward meeting the communication requirements by replacing the objective of Problem (1) with the definition of the weighted entropy $H_W(x)$. (Note that the problem of miscoordination still remains which we will tackle in section 3). The reduction in expected reward can be controlled by enforcing that feasible solutions achieve at least a certain expected reward $E_{\min}$ and the communication constraint remains unchanged. The following problem maximizes the weighted entropy while maintaining the expected reward above $E_{\min}$ and a communication consumption below Q:

$$
\begin{array}{ll}
\max & H_W(x) \\
\text{s.t.} & \sum_{a\in A} x(j,a) - \sum_{s\in S}\sum_{a\in A} p(s,a,j)x(s,a) = \alpha_j \quad \forall j \in S \\
& \sum_{s\in S}\sum_{a\in A} r(s,a)x(s,a) \geq E_{\min}, \qquad \sum_{s\in S}\sum_{a\in A} x(s,a)n(s,a) \leq Q \\
& x(s,a) \geq 0 \qquad \forall s \in S, a \in A
\end{array}
\tag{3}
$$

where $E_{\min}$ is an input domain parameter (E_{min} can vary between 0 and E^* where E^* is the maximum expected reward obtained by solving (1)). Solving Problem (3) is our first algorithm to obtain a randomized policy that achieves at least $E_{\min}$ expected reward while meeting the communication constraints (Algorithm 1).

```
Algorithm 1 MAX-ENTROPY($(E_{\min}, Q)$)
Solve Problem (3) with $E_{\min}$ and Q, let $x_{E_{\min}}$ be optimal solution
return $x_{E_{\min}}$ (maximal entropy, expected reward $\geq E_{\min}$, communication required $\leq Q$)
```

Unfortunately, the problem of miscoordination introduced in section 2.2 still remains.

3 Solving Miscoordination: From MCMDP to Transformed MCMDP

This section presents an automatic transformation of a MCMDP to a transformed MCMDP, where the resulting optimal policies can be executed in multiagent settings, via appropriate communication (with communication costs within resource limits). We illustrate the key concepts in MCMDP transformations by focusing on one specific transformation namely the *sequential transformation*, given in Figure 2-a. While the transformation introduced is similar to [21], there are two key differences in that work and the present work: (i) The solution for policy randomization we develop for the transformed MCMDP needs a non-linear objective with non-linear constraints unlike earlier work where the reward maximization needed linear objective with non-linear constraints. Hence, the basic problem being solved is different. (ii) Maximizing entropy is the focus of our present work. The transformation requires addition of new states and actions and hence entropy would get affected. Our transformation has to ensure that maximizing entropy for the transformed MCMDP would be equivalent to the problem of maximizing entropy for the original MCMDP. We provided a mathematical proof later to show that indeed this property holds.

3.1 Transformation Methods: Sequential and Others

Figure 2-a shows a portion of a MCMDP, where agent A with actions a$_1$ to a$_m$ and B with actions b$_1$ to b$_n$ act jointly (a$_i$b$_j$). Figure 2-b shows the transformation of this MCMDP into transformed MCMDP. This transformation is sequential in that one of the agents, in this case agent A, first chooses one of its actions a$_i$ and also decides whether to communicate this choice to its teammate, agent B. Thus, C(a$_i$) in Figure 2-b refers to A's selection and communication of action a$_i$ to B, incurring the cost of communication, and going to Ai$_c$ (with probability 1-p$_f$ where p_f is the probability with which communication may fail); while NC(a$_i$) results in state Ai$_o$, where agent A selected a$_i$ but decided not to communicate this choice to B to avoid communication costs. Note, since communication may fail with a probability p$_f$, C(a$_i$) may transition to Ai$_o$ with a probability p$_f$. Once in state Ai$_c$ or Ai$_o$, agent B chooses its action b$_j$, and the agents now jointly execute the action a$_i$b$_j$. When choosing its action, B observes which of the different Ai$_c$ state it is in, since any such state is reached only after A's communication. Unfortunately, agent B cannot distinguish between states Ai$_o$ reached without A's communication. Thus, B's action b$_j$ in such non-communication states must be taken without observing which of the m states A1$_o$ to Am$_o$ B is in. Thus, B will be unable to execute any randomized policy which requires it (agent B) to select an action b$_j$ with a different probability in a state say Ai$_o$ vs a state Ak$_o$. To avoid this problem, we require that for any two states reached after non-communication, the probability of B's action selection must be identical, i.e., for any action b$_j$ and states Ai$_o$ and Ak$_o$, P(b$_j$|Ai$_o$) = P(b$_j$|Ak$_o$). This restriction on probability of action execution in the transformed MCMDP translates into the addition of the following non-linear constraints into our Problem 3 applied for the transformed MCMDP, to solve the original MCMDP. Specifically, in terms of the state action variables, given any two states Ai$_o$ and Ak$_o$, and any action b$_j$, it is necessary that:

$$Xo_{ij}/(\sum_{u=1}^{n} Xo_{iu}) = Xo_{kj}/(\sum_{u=1}^{n} Xo_{ku}) \Rightarrow Xo_{ij} * (\sum_{u=1}^{n} Xo_{ku}) = Xo_{kj} * (\sum_{u=1}^{n} Xo_{iu}) \qquad (4)$$

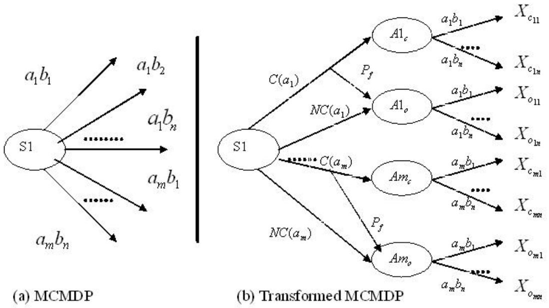


Figure 2: Transformation

Thus, to obtain an optimal randomized policy in the MCMDP, we must solve problem 3 for the transformed MCMDP with these non-convex constraints included in the problem. The optimal policy for a transformed MCMDP thus obtained will require a random selection at state S1 by agent A alone, and then in the next state (either Ai$_c$ or Ai$_o$) by agent B alone, thus avoiding the problem faced in the MCMDP. The non-linear constraints in the transformed MCMDP affect only the actions taken from states A1$_o$, A2$_o$,....,Am$_o$(from figure 2-b) and ensure that P(b$_j$|A1$_o$) = P(b$_j$|A2$_o$) = ... = P(b$_j$|Am$_o$) for j $\in$ 1,2....n. This is because for agent B, states A1$_o$, A2$_o$,....,Am$_o$ are indistinguishable, as they are reached without A's communication.

Apart from the addition of these non-linear constraints, the entropy function also undergoes change as the transformed MCMDP has new states and actions added to it. The entropy function for Figure 2-a would be the straightforward $H_W(x)$ as developed in section 2.1. In the transformed MCMDP, it would still be the $H_W(x)$ with a small change in the way entropy is calculated at each state. The entropy function at each state as calculated over the probability distribution of all actions at that state

is $H = -\sum_{i=1}^{n} p_i \log p_i$ where p_i is the probability of taking action a_i at that state. Therefore, for state S1 in figure 2-a the entropy is -
$H(S1) = -1 * (p(a_1b_1) * log(p(a_1b_1)) + ... + P(a_1b_n) * log(p(a_1b_n)) + ... + P(a_mb_1) * log(p(a_mb_n)) + ... + p(a_mb_n) * log(p(a_mb_n)))$.
If we notice state S1 of Figure 2-b, the probability with which agent A would take action say a_1 would be the sum of the probabilities with which it takes $C(a_1)$ and $NC(a_1)$. This is because whether agent 1 communicates that it would take action a_1 or does not communicate that it would take a_1 is internal to the system because of our assumption (as explained in introduction) that communication is safe. Therefore, only the fact that agent A will take action a_1 (independent of whether it is known to agent B) with certain probability is important to our entropy equation since the enemy gets to observe that as the policy of agent A. Therefore the new entropy function for state S1 in figure 2-b would be
$H(S1) = -1 * (p(C(a_1) + NC(a_1)) * log(p(C(a_1) + NC(a_1))) + ... + p(C(a_m) + NC(a_m)) * log(p(C(a_m) + NC(a_m))))$
instead of the entropy function
$H(S1) = -1 * (p(C(a_1)) * log(p(C(a_1))) + p(NC(a_1)) * log(p(NC(a_1))) + ... + p(C(a_m)) * log(p(C(a_m))) + P(NC(a_m) * log(p(NC(a_m))))$.

Hence to solve our original MCMDP we solve Problem 3 for the transformed MCMDP using the modified entropy function with the addition of non-linear constraints we described earlier. One interesting fact in Figure 2-a is that the entropy calculation would undergo such a change only for actions of agent A while no such addition of probabilities of C and NC actions is needed for agent B. Given that we now have a new entropy function (calculated using probability of an action of an agent as sum of communication and non-communication probabilities of that action), and also new states and transitions, it might not be necessary that optimizing the entropy function for the transformed MCMDP would automatically mean that we are increasing security for the original problem we were solving. We therefore prove the following lemma below for two cases of communication namely no communication and full communication. The lemma basically states that the under conditions of no communication or full communication the entropy obtained for the MCMDP and the transformed MCMDP would be the same if there are no changes in the reward thresholds to be met and the bandwidth constraints. Under conditions of limited communication, we experimentally verified over a large set of points and found the lemma still holds true although we do not provide a formal proof.

Lemma 1 *If in a state say $S1$ of MCMDP (Figure 3-a), the entropy is defined over the probability distribution of the actions over the state, then the entropy would remain the same in sequential transformation over the whole system of states generated.*

proof: For simplicity of proof, lets assume a two agent case where the bandwidth present is zero (no communication case) in the domain. In 3-a, we show the MCMDP where there are four joint actions obtained from the two individual actions a,b of agents 1 and 2. We now transform the MCMDP using our sequential transformation into a transformed MCMDP. We assume agent 1 decides on the communication/non-communication issue. Figure 3-b shows the transformed MCMDP. If the policy of the MCMDP and the transformed MCMDP is the same, then the flows and hence the path probabilities for the four corresponding paths in both figures are equal.

Entropy from 3-a: $Entropy1 = P_1logP_1 + P_2logP_2 + P_3logP_3 + P_4logP_4$

Entropy from 3-b: $Entropy2 = P_alogP_a + P_blogP_b + P_a * (P_{a1}logP_{a1} + P_{a2}logP_{a2}) + P_b * (P_{b1}logP_{b1} + P_{b2} * logP_{b2})$

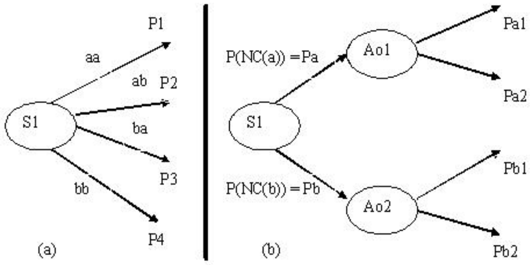
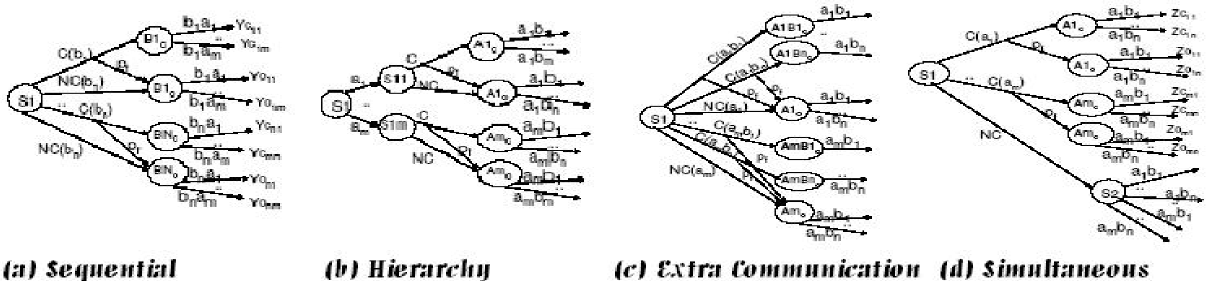


Figure 3: Illustrative Example

Lets consider the terms $P_a log P_a + P_a * (P_{a1} log P_{a1} + P_{a2} log P_{a2})$
$= P_a * (log P_a + P_{a1} log P_{a1} + P_{a2} log P_{a2})$
Since $P_{a1} + P_{a2} = 1$,
$= P_a * ((P_{a1} + P_{a2}) log P_a + P_{a1} log P_{a1} + P_{a2} log P_{a2}) = P_a * (P_{a1} * (log P_a + log P_{a1}) + P_{a2} * (log P_a + log P_{a2})$
$= P_a P_{a1} * log(P_a P_{a1}) + P_a P_{a2} * log(P_a P_{a2})$
Since the path probabilities are equal, $P_a P_{a1} = P_1$ and $P_a P_{a2} = P_2$. Hence proved equal to $P_1 log P_1 + P_2 log P_2$. Similar math applies to the other terms making it equal to $P_3 log P_3 + P_4 log P_4$. Therefore the entropies of both transformations is the same. The same reasoning as above follows if there is full communication also.

While we showed one particular method of transformation called the sequential transformation with one particular order of communication actions, as shown in Figure 4, there are other methods of transforming a MCMDP into a transformed MCMDP. First, as shown in Figure 4-a, the order of communication actions in the sequential transformation can be changed. If one agent has fewer actions than another (e.g., if $n < m$), such a change in the order of communication may improve the optimality of the resulting policy or reduce communication costs. Second, as shown in Figure 4-b, in a *hierarchical* transformation, an agent first decides which action to select, and only later whether to communicate this choice (C) or not (NC). By choosing an action first, an agent's communication decision may be improved, potentially improving policy optimality. Our third *extra-communication* transformation is similar to the sequential transformation, except that agent A chooses actions for itself and for agent B and communicates the choice of both to agent B. As discussed earlier, this would lead to extra overheads in communication. Finally, our *simultaneous* transformation, is shown in Figure 4-d. Here, while the choice of communication is done sequentially, no communication by A results in state S2; and in S2, agent A and B simultaneously and randomly select their actions. Additionally, combinations of these transformations are also feasible. Typically, we must select from these multiple transformations the one that provides the most optimal policies. However, in this paper, we just introduce the sequential transformation and its properties and leave such an analysis of various transformations for future work.

In all the above transformations, one of the agents selects an action without observation of its actual state, leading to non-linear constraints, e.g., in simultaneous transformation at state S2, agents A and B act simultaneously. Once again, non-linear constraints arise and hence non-linear constraints must be added in the simultaneous case also. Indeed, irrespective of the style of transformation, non-linear constraints must be added. This is because expressing probabilities of events in MCMDPs requires divisions via Xia variables. And regardless of the transformation that we choose for the MCMDP, we need to express constraints using probabilities. Indeed, all transformations either involve sequential action selection or simultaneous, and we showed non-convex constraints in each case [21]. Thus:



(a) Sequential (b) Hierarchy (c) Extra Communication (d) Simultaneous

Figure 4: Other methods of transformation

o **Proposition 1:** It is necessary to add non-convex constraints to solve the actual MCMDP.

3.2 The Sequential Transformation Algorithm

Since sequential transformation is the basis of our work in this paper, we describe the transformation algorithm for it. We now present Algorithm 2 that achieves this sequential transformation of MCMDP into a transformed MCMDP automatically. (In fact our implementation creates problem 3 with the non-linear constraints as an output). The algorithm works by first adding intermediate states with (and without) communication in *SrcToComm* and then adding transitions from the intermediate states to the destination states in *CommToDest*. We assume that joint actions are processed in increasing order of the index i ($1 <= i <= m$) for a$_i$, and j for b$_j$ ($1 <= j <= n$). In *SrcToComm*, communication actions a$_i\_c$ leads to state sa$_i\_c$ with probability 1-P_{fn} (and state sa$_i\_nc$ with probability P_{fn}); and non-communication action a$_i\_nc$ deterministically transitions to state sa$_i\_nc$, where the first agent has decided not to communicate its choice to its teammate. Line 13 in the *Conversion* algorithm adds the constraints on probabilities of outgoing actions from sa$_i\_nc$ — because of transitivity of equality, it is sufficient to add probability constraints with respect to just the first non-communication state sa$_1\_nc$. From line 4 and line 7 of the algorithm, the number of probability constraints can be seen as (m-1)*n to be later translated into non-linear constraints using equation 4. Thus, this is a polynomial time algorithm, with a complexity of $O(|S|^2 * |A|)$, where $|A| = n * m$ gives us the number of joint actions. In the worst case, the resulting MCMDP has $2 * |S| * m$ additional states inserted. Given that the output of the transformation algorithm is a nonlinear program with nonlinear constraints our polynomial transformation algorithm does not add anything to the complexity of the problem.

4 Experimental Results

Based on the UAV example we described earlier, we first constructed a MCMDP with joint states, actions, transitions and rewards. We then transformed the MCMDP into a transformed MCMDP with the appropriate communication and non-communication actions. We then present results using the transformed MCMDP (Figure 5) to provide key observations about the impact of reward and communication thresholds on policy randomization. Figure 5-a shows the results of varying reward threshold (x-axis) and communication thresholds (y-axis) on the weighted entropy of the joint policies (z-axis).

Based on the figure, we make two key observations. First, with extreme (very low or very high) reward thresholds, communication threshold makes no difference on the value of the optimal policy. In particular, in extreme cases, the actions are either completely deterministic or randomized. On one extreme (maximum reward threshold), agents choose the best deterministic policy and hence communication makes no difference and entropy hits zero. At the other extreme, with low reward threshold (reward threshold 0) agents gain an expected weighted entropy of almost 2 (the maximum possible in our domain), since the agents can choose highest entropy actions and thus communication does not help. Second, in the middle range of reward thresholds, where policies are randomized, communication makes the most difference; indeed, the optimal entropy is seen to increase as communication threshold increases. For instance, when reward threshold is 7, the weighted entropy of the optimal policy obtained without communication is 1.36, but with high communication threshold of 6, the optimal policy provides a weighted entropy of 1.81.

Figure 5-b zooms in on one slice in Figure 5-a (reward threshold fixed at 7). It shows the changes in probability of communication and non-communication actions in the optimal policy (y-axis), with changes in communication threshold (x-axis). P(comm a$_i$) denotes the probability of executing the action to communicate a$_i$ (similarly for non-communication actions). The graph illustrates the following: when there is no communication in the system, action a_1 gets preferred over a_2 because of reward constraints. Action a_1 would have been chosen with probability 1 but for the fact that entropy needs to be maximized. As communication is increased, most communication is allocated to a_1 as opposed to a_2 because of the high reward to cost ratio for a_1. The interesting issue that arises here is, at the highest communication point even though after all the communication was used up but action a_1 accounted to only .4 of the total probability (i.e 1), the no communication action a_2 was chosen for the rest of the probability even though a_1 would have provided higher reward. This is due to our assumption that communication is safe, i.e both communication and non-communication actions appear the same to our adversary. If this assumption was not there, most possibly non communication of a_1 should have been chosen with higher probability. In the highest communication threshold case, increasing probability of NC(a_1) is actually detrimental to entropy since P(C(a_1)) + P(NC(a_1)) might then add up to near 1 making it more deterministic which seems counterintuitive. The other interesting issue is that, as communication threshold increases, the probability of communicative actions increase say P(a_1) increases from 0 to 0.4. At the same time, the probability of the non-communication actions decreases.

Table 1: Comparing Weighted Entropies.

Comm Threshold →	0	3	6
MCMDP	1.9	1.9	1.9
Deterministic	0	0	0
Miscoordination	Yes	Yes	No
Transformed MCMDP	1.6	1.83	1.9

Table 1 compares the weighted entropies of different joint policies with changes in communication threshold for the same example we showed our results on earlier (using a fixed reward threshold of 5). In the first row we show the three settings of the communication thresholds (0,3 and 6 respectively) we use for deriving the entropy values for the various cases in the table. Row 2 shows the entropies obtained by an optimal MCMDP policy. The entropy (1.9) is an ideal upper-bound for benchmarking

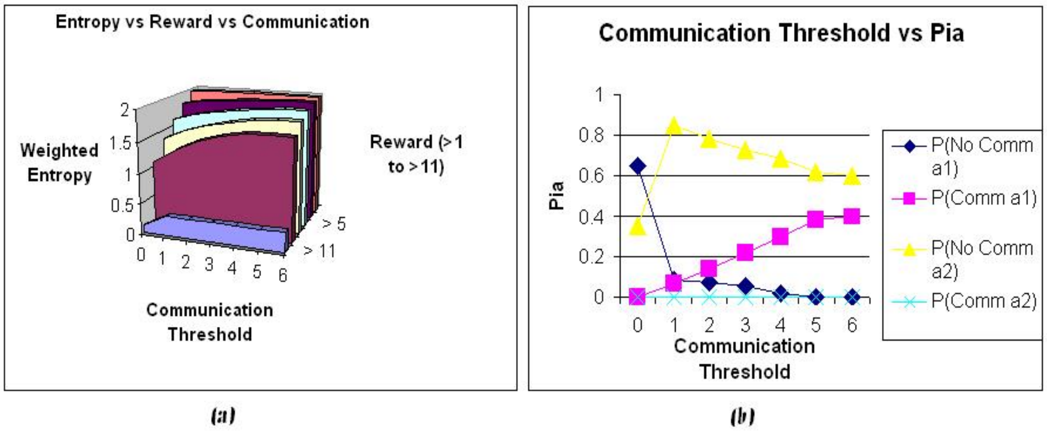


Figure 5: Effect of thresholds

and the entropy is unaffected by the communication threshold. Row 3 illustrates that deterministic policies exist in our domain but their entropy be 0 and hence there would be no security. Row 4 shows the results, where agents take the optimal policy of the MCMDP and attempt to execute it without coordination. Unfortunately, communication constraints are violated in columns 1 and 2. Only when communication resource of 6 units is available the MCMDP policy becomes executable without any miscoordination. Finally, row 5 shows the entropy of the transformed MCMDP for comparison. It is able to avoid the problems faced by policies in row 3 and 4. However, with communication threshold of 0, the transformed MCMDP must settle for an entropy of 1.6; as the communication threshold increases, it finally settles at an entropy of 1.9 which also shows why the MCMDP policy(row 1) becomes executable when communication threshold is 6.

5 Summary and Related Work

This paper focuses on security in multiagent teams acting in observable domains with communication resource constraints. The issue of security arises here because of intentional threats that are caused by unseen adversaries, whose actions and capabilities are unknown, but the adversaries can exploit any predictability in our agent's policies. Policy randomization with guaranteed rewards meeting the communication constraints becomes critical in such domains. To this end, this paper provides three key contributions. First we recall the MCMDP framework where agents not only maximize their expected team rewards but also bound the expected team consumption of the communication resource. We then developed a non-linear program for this MCMDP that maximizes policy randomization while bounding communication consumption at the same time providing guarantee on the expected team reward obtained. We then show how randomized policies in team settings lead to miscoordination and hence the policies obtained from our non-linear program can be inexecutable. Our second contribution is the introduction of a novel transformation algorithm called the sequential transformation where we can explicitly incorporate communication and non-communication actions. Thus problems may be formulated using our abstract transformed MCMDP and our transformation ensures that the resulting randomized policies avoid miscoordination. We also show the existence of many other such transformations. Third, we showed that despite the fully observable domains, transformed MCMDPs necessitate programs using our non-convex constraints. We then solved our non-linear program with the non-convex constraints on our UAV domain, initially modeled as a MCMDP on which we applied our transformation algorithm to obtain the transformed MCMDP. From these experiments, we showed the various tradeoffs involved between the three key factors namely entropy, reward and communication resources. Finally, while our techniques are applied for analyzing randomization-reward-communication tradeoffs, they could potentially be applied more generally to analyze different tradeoffs between competing objectives in MCMDPs.

Decision-theoretic literature has focused on maximizing total expected reward [24, 5, 12] but maximizing policy randomization as a goal has received little attention in the literature. Randomization is mostly seen as a means or side-effect in attaining other objectives, e.g., in resource-constrained MDPs [2] or limited memory POMDP policy generators [14, 20, 15, 23, 6]. In [21] coordination of multiple agents executing randomized policies in a MDP team setting is discussed, but there randomization occurs as a side-effect of resource constraints. The work in [22] explicitly emphasizes on maximizing policy entropy but no resource constraints are considered. In contrast, our work focuses on policy randomization while explicitly ensuring that the communication constraints of the team are met. The effect of communication in multiagent teams has been analyzed extensively [28, 4, 11, 19]. However, none of this work focuses on using communication to counter the miscoordination arising due to randomized policies in team settings. Further we model communication as a resource with a cost which is independent of the reward i.e communication costs and rewards cannot be compared and the focus is to make optimal usage of the limited communication unlike heuristic techniques developed earlier for adding communication actions. Significant attention has been paid to learning in stochastic games, where agents must learn dominant strategies against explicitly modeled adversaries [17, 13]. Such dominant strategies may lead to randomization, but randomization itself is not the goal. Our work in contrast does not require any model of the adversary and under this worst case assumption hinders any adversary's actions by increasing the policy's weighted entropy. Thus, we focus on agent teams using Decentralized MDPs with communication constraints doing intentional policy randomization.

Acknowledgments : This research is supported by NSF grants #0208580, #0222914 & ISF #8008. It is also supported by the United States Department of Homeland Security through Center for Risk and Economic Analysis of Terrorism Events (CREATE). Sarit Kraus is also affiliated with UMIACS.

References

[1] Call for papers: Safety and security in multiagent systems. In *AAMAS*, 2005.

[2] E. Altman. *Constrained Markov Decision Process*. Chapman and Hall, 1999.

[3] R. Beard and T. McLain. Multiple UAV cooperative search under collision avoidance and limited range communication constraints. In *IEEE CDC*, 2003.

[4] R. Becker, V. Lesser, and S. Zilberstein. Analyzing Myopic Approaches for Multi-Agent Communication. In *Proceedings of IAT*, pages 550–557. IEEE Computer Society, September 2005.

[5] R. Becker, S. Zilberstein, V. Lesser, and C. Goldman. Transition-independent decentralized markov decision processes. In *Proceedings of AAMAS*, 2003.

[6] D. Bernstein, E. Hansen, and S. Zilberstein. Bounded policy iteration for decentralized pomdps. In *IJCAI*, 2005.

[7] C. Boutilier. Sequential optimality and coordination in multiagent systems. In *IJCAI*, 1999.

[8] D. Carroll, K. Mikell, and T. Denewiler. Unmanned ground vehicles for integrated force protection. In *SPIE Proc. 5422*, 2004.

[9] D. Dolgov and E. Durfee. Approximating optimal policies for agents with limited execution resources. In *Proceedings of IJCAI*, 2003.

[10] D. Dolgov and E. Durfee. Resource allocation and policy formulation for multiple resource-limited agents under uncertainty. In *Proceedings of ICAPS*, 2004.

[11] M. Ghavamzadeh and S. Mahadevan. Learning to communicate and act in cooperative multiagent systems using hierarchical reinforcement learning. In *AAMAS*, 2004.

[12] C. V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *AAMAS*, pages 137–144, 2003.

[13] J. Hu and P. Wellman. Multiagent reinforcement learning: theoretical framework and an algorithm. In *ICML*, 1998.

[14] T. Jaakkola, S. Singh, and M. Jordan. Reinforcement learning algorithm for partially observable markov decision problems. *Advances in NIPS*, 7, 1994.

[15] L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. In *Technical Report, Brown University*, 1995.

[16] P. J. Lewis, M. R. Torrie, and P. M. Omilon. Applications suitable for unmanned and autonomous missions utilizing the tactical amphibious ground support (tags) platform. *http://www.autonomoussolutions.com/Press/SPIE%20TAGS.html*, 2005.

[17] M. Littman. Markov games as a framework for multi-agent reinforcement learning. In *ML*, 1994.

[18] R. Nair, D. Pynadath, M. Yokoo, M. Tambe, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI*, 2003.

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

[20] R. Parr and S. Russel. Approximating optimal policies for partially observable stochastic domains. In *Proceedings of IJCAI*, 1995.

[21] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. Towards a formalization of teamwork with resource constraints. In *AAMAS*, 2004.

[22] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. Security in multiagent systems by policy randomization. In *AAMAS*, 2006.

[23] P. Poupart and C. Boutilier. Bounded finite state controllers. In *NIPS*, 2003.

[24] D. Pynadath and M. Tambe. The communicative multiagent team decision problem: analyzing teamwork theories and models. *JAIR*, 2002.

[25] M. H. Rahimi, H. Shah, G. S. Sukhatme, J. Heidemann, and D. Estrin. Studying the feasibility of energy harvesting in a mobile sensor network. In *ICRA*, 2003.

[26] A. Serjantov. On the anonymity of anonymity systems. *PhD Dissertation, University of Cambridge*, 2004.

[27] C. Shannon. A mathematical theory of communication. *The Bell Labs Technical Journal*, pages 379–457,623,656, 1948.

[28] P. Xuan and V. Lesser. Multi-Agent Policies: From Centralized Ones to Decentralized Ones. *Proceedings of AAMAS*, pages 1098–1105, 2002.

Algorithm 2 CONVERT()

```
**Input**:$< S, A, P, R, N, Q >$
**Output**:$< S', A', P', N', Q' >$
Conversion()
Create Problem 3 from **Output**.

Conversion(){
Initialize: $S' = S, A' = A, P' = P, R' = \phi, N' = \phi, Q' = Q$
**for all** $s \in S$ **do**
  **for all** $(\hat{a} = (a_i, b_j)) \in A$ **do**
    **if** $sa_i\_nc \notin S'$ **then**
      SrcToComm$(s, \hat{a}, sa_i\_nc, a_i\_nc)$
      $p'(s, \hat{a}, sa_i\_nc) \leftarrow 1$
      **if** $(|p(s, < a_i, * >, *) > 0| > 1)$ **then**
        SrcToComm$(s, \hat{a}, sa_i\_c, a_i\_c)$
        $n'(s, a_i\_c) \leftarrow Communication\_Model$
        $p'(s, a_i\_c, sa_i\_c) \leftarrow 1 - P_f$
        $p'(s, a_i\_c, sa_i\_nc) \leftarrow P_f$
    **if** $i \neq 1$ **then**
      $prob(b_j|sa_i\_nc) = prob(b_j|sa_1\_nc)$
    CommToDest$(s, \hat{a}, sa_i\_nc, a_i\_nc)$
    **if** $(|p(s, < a_i, * >, *) > 0| > 1)$ **then**
      CommToDest$(s, \hat{a}, sa_i\_c, a_i\_c)$
    **for all** $s' \in S'$ **do**
      $p'(s, \hat{a}, s') \leftarrow 0$
}

SrcToComm$(S_{parent}, A_{parent}, S_{current}, A_{current})${
$S' \leftarrow S' \bigcup S_{current}$
$A' \leftarrow A' \bigcup A_{current}$
$r'(S_{parent}, A_{current}), n'(S_{parent}, A_{current}) \leftarrow 0$
}

CommToDest$(S_{parent}, A_{parent}, S_{current}, A_{current})${
**for all** $s' \in S'$ **do**
  $p'(S_{current}, A_{parent}, s') \leftarrow p(S_{parent}, A_{parent}, s')$
$r'(S_{current}, A_{parent}) \leftarrow r(S_{parent}, A_{parent})$
```