Towards a formalization of teamwork with resource constraints

Praveen Paruchuri, Milind Tambe, Fernando Ordonez
University of Southern California
Los Angeles, CA 90089
{paruchur,tambe,fordon}@usc.edu

Sarit Kraus
Bar-Ilan University
Ramat-Gan 52900, Israel
sarit@macs.biu.ac.il

Abstract

Despite the recent advances in distributed MDP frameworks for reasoning about multiagent teams, these frameworks mostly do not reason about resource constraints, a crucial issue in teams. To address this shortcoming, we provide four key contributions. First, we introduce EMTDP, a distributed MDP framework where agents must not only maximize expected team reward, but must simultaneously bound expected resource consumption. While there exist single-agent constrained MDP (CMDP) frameworks that reason about resource constraints, EMTDP is not just a CMDP with multiple agents. Instead, EMTDP must resolve the miscoordination that arises due to policy randomization. Thus, our second contribution is an algorithm for EMTDP transformation, so that resulting policies, even if randomized, avoid such miscoordination. Third, we prove equivalence of different techniques of EMTDP transformation. Finally, we present solution algorithms for these EMTDPs and show through experiments their efficiency in solving application-sized problems.

1. Introduction

Teamwork is critical in a large number of multiagent domains, from simulated soccer teams to distributed sensors for monitoring, to future robotic teams on Mars [9, 8]. These teams must often adhere to certain resource constraints, e.g., when communicating, agents must consume only limited communication bandwidth, or when monitoring, each sensor agent must only consume limited energy (as much as it can replenish itself) [8].

This paper provides a novel formalization for teamwork in uncertain domains where agents must operate under resource constraints. Recently there has been a significant interest in formalizations of teamwork via distributed MDPs, where domain costs and uncertainties are treated as first class citizens [7, 11, 5]. Unfortunately, most of these previous formalizations fail to address teamwork with resource constraints. Incorporating resource constraints into distributed MDP frameworks is difficult, since they require maximizing expected team rewards while simultaneously limiting expected resource consumption. In particular, we focus on an important class of *soft* constraints, which are not easily modelled by explicitly representing resources and imposing infinite penalties in states violating constraints. For instance, bandwidth limit on communication may be soft, not hard and precise. Exceeding bandwidth limit within any single run is not a disaster; but if the team consumes more than its bandwidth limit on average, it jeopardizes the communications of other agents/applications on the same network. Similarly, given a replenishable sensor agent, it must limit its expected energy consumption to the amount it can be replenished in the next period. Exceeding this bound in one period is not a catastrophe (unless exceeded repeatedly). Let us consider that, the sensor has k units of energy initially and can replenish m units every time step. The amount of energy the sensor can use up is a soft constraint because spending more than m units occasionally is fine as long as it can make up the extra expended energy in the next few time steps. Constraints involving averaging a quantity, in general, are soft constraints because as long as the average is maintained there is no hard bound on how the resource should be used at each time step [8]. The importance of such soft constraints is seen by the continued work in operations research literature, which has developed single agent Constrained MDPs (CMDPs) for reasoning about expected resource consumption[1]. Indeed, hard constraints may lead to severe inflexibility in the tasks a team performs.

This paper takes a key step in enabling distributed MDP frameworks to reason about resource constraints, and in the process provides four key contributions. First, we propose EMTDP (extended MTDP[7]), where agents must optimize expected joint rewards while simultaneously bounding expected resource consumption. Second, we identify a novel coordination challenge in EMTDPs due to its distributed nature (a problem absent in CMDPs) and present an EMTDP transformation algorithm to address this challenge. Optimal policies in CMDPs can be obtained via linear programming and are randomized[1]. Unfortunately, in distributed settings, agents simply cannot execute such randomized policies without additional coordination, which in turn consumes its own resources. Thus, to enable execution by multiple agents, EMTDP policies must include appropriate communication, while bounding the expected communication costs (possibly to zero). We provide a novel, polynomial-time algorithm to transform an abstract *conjoined EMTPD*, which is similar to a CMDP, to an actual EMTDP where randomized policies can be executed by multiple agents with appropriate communication.

Our third contribution illustrates the equivalence of a series of EMTDP transformations in terms of expected rewards of optimal policies. Furthermore, we show that any EMTDP transformation must add non-linear, non-convex constraints into the optimization problem, yielding an optimization problem over a non-convex feasible region. There is no polynomial algorithm for finding a global optimal for a non-convex problem [10], in fact even finding a local optimum of a non-convex problem is in general not polynomial, with current non-linear solvers only able to guarantee convergence [6]. Thus, in contrast to CMDPs [1, 3] which can be solved via linear programming, we cannot provide polynomial results for EMTDPs in the most general sense. Hence our final contribution is a computationally efficient algorithm to obtain approximate solutions for EMTDPs with a guaranteed error bound. Experimental results for two separate domains are presented.

The rest of the paper begins with the *conjoined EMTDP*. An automated method of transformation to the actual EMTDP model follows, followed by our solution approach and computational results. Our results generalize to randomized policies in other settings as well e.g., in hostile settings, randomized policies may reduce predictability.

2. Conjoined EMTDP

Conjoined EMTDP is a useful tool for users, providing a layer of abstraction to model agent-teams with resource constraints in uncertain domains. However, optimal policies yielded by a *conjoined EMTDP* may not be executable by a team of agents, due to a lack of appropriate inter-agent coordination. As with other research on distributed MDPs [2], we introduce a 2-agent *conjoined EMTDP* for expository purposes. We deal with a fully observable environment. In particular, a 2-agent *conjoined EMTDP* is defined as a tuple: $\langle S, A, P, R, C1, C2, T1, T2, N, Q\rangle$. It consists of a finite set of states S. Given two individual actions a_l and a_m of the two agents in our team, the team's joint action $\hat{a} = (a_l, a_m) \in A$. $P = [p^{\hat{a}}_{ij}] (\equiv p(i, \hat{a}, j))$ is the transition matrix, providing the probability of transitioning from state i to 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. (C2 is similarly defined.) $T1 = [t1_k]$ and $T2 = [t2_k]$ are vectors of amounts of available 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. Communication costs are treated as joint costs to illustrate our ability to model shared resources such as bandwidth. Q is threshold on communication costs that can be used by the team of agents. A conjoined EMTDP is thus similar to a CMDP [1] with multiple agents.

The goal in the conjoined EMTDP is to maximize the total expected reward, while ensuring that the expected resource consumption is maintained below threshold. Formally, this requirement can be stated as a linear program, extending the linear program for CMDPs [3] 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.

$$\begin{array}{ll|c}
\textbf{Max} & \sum_i \sum_{\hat{a}} x_{i\hat{a}} r_{i\hat{a}} & \sum_{\hat{a}} x_{j\hat{a}} - \sum_i \sum_{\hat{a}} x_{i\hat{a}} p^{\hat{a}}_{ij} = \alpha_j \\
 & & \sum_i \sum_{\hat{a}} x_{i\hat{a}} c1_{i\hat{a}k} \leq t1_k \\
 & & \sum_i \sum_{\hat{a}} x_{i\hat{a}} c2_{i\hat{a}k} \leq t2_k \\
 & & \sum_i \sum_{\hat{a}} x_{i\hat{a}} n_{i\hat{a}} \leq Q \\
 & & x_{i\hat{a}} \geq 0
\end{array}$$

Figure 1 shows a small example of a *conjoined EMTDP*. There are two agents, A and B, with actions a_1, a_2 and b_1, b_2 respectively, leading to joint actions $\widehat{a1} = (a_1, b_1)$, $\widehat{a2} = (a_1, b_2)$, $\widehat{a3} = (a_2, b_1)$, $\widehat{a4} = (a_2, b_2)$. We also show the transition probabilities for each of the actions. For illustration, the reward and costs for $\widehat{a4}$ are also shown. States S4 through S7 are terminal states, and action $\mathrm{a}_1\mathrm{b}_1$ is an *abandon* action (to not play the game). Conjoined EMTDPs such as this yield randomized optimal policies e.g., the optimal policy here specifies that in S1, p($\widehat{a1}$) = 0, p($\widehat{a2}$) = 0.56, p($\widehat{a3}$) = 0.44 and p($\widehat{a4}$) = 0. If the two agents A and B are supplied this policy, they will fail in its execution, unless additional coordination occurs. In particular, suppose based on this policy for joint actions, agent A chooses its own actions such that p(a_1) = 0.56 and p(a_2) = 0.44. However, when A selects a_1, it has no guarantee that B will select b_2 — in fact, due to its own randomization, B may simultaneously select b_1. Thus, the team may jointly execute $\widehat{a1} = (a_1, b_1)$, even though the policy specifies p($\hat{a}$1) = 0.

Thus, a *conjoined EMTDP*, i.e., 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. For example agent A could choose $\widehat{a3}$, and communicate its choice to agent B.

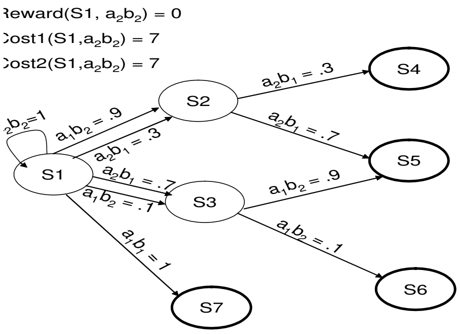


Figure 1. Original EMTPD

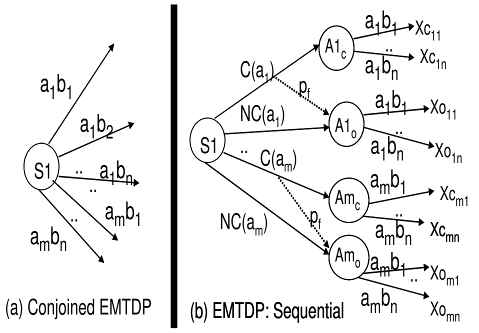


Figure 2. Transformation.

Unfortunately, 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.

3. From conjoined EMTDP to EMTDP

This section presents an automatic transformation of a *conjoined EMTDP* to an actual EMTDP, where the resulting optimal policies can be executed in multiagent settings, via appropriate communication (with communication costs within resource limits). One key assumption in these transformations is that the communication cost is directly proportional to the length of the message; thus small message sizes are preferred. We illustrate the key concepts in EMTPD transformations by focusing on one specific *sequential transformation*, given in Figure 2. Figure 2-a shows a portion of a *conjoined EMTDP*, where agent A with actions a_1 to a_m and B with actions b_1 to b_n act joinly ($a_i b_j$). Figure 2-b shows the transformation of this *conjoined EMTDP* into an actual EMTDP. 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); 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. (While we discuss the transformation in a two agent case for expository purposes, the n-agent case is discussed later.)

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 EMTDP translates into the addition of the following non-linear constraint into our original LP to solve an EMTDP. Specifically, in terms of the state action variables in the original LP, given any two states Ai_o and Ak_o, and any action b_j, it is necessary that:

$$\begin{aligned} &Xo_{ij}/(\sum_{u=1}^{n} Xo_{iu}) = Xo_{kj}/(\sum_{u=1}^{n} Xo_{ku}) \\ &\Rightarrow Xo_{ij} * (\sum_{u=1}^{n} Xo_{ku}) = Xo_{kj} * (\sum_{u=1}^{n} Xo_{iu}) \end{aligned} \tag{1}$$

Thus, to obtain an optimal policy in the actual EMTDP, we must solve an optimization problem which includes these non-convex constraints. Note that the non-linear constraints are only associated with states that are reached non-communication. The optimal policy for a transformed EMTDP 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 *conjoined EMTDP*. Figure 3 shows transformation of the *conjoined EMTDP* shown in Figure 1. (Note that the communication failure transitions are deleted to simplify the diagram). The non-linear constraints in this case affect only the actions taken from states s4 and s5, and ensure that P(b_j|S4) = P(b_j|S5) for j ∈ 1,2. This is because for agent B, state S4 and S5 are indistinguishable, as they are reached without A's communication.

As shown in Figure 4, there are other methods of transforming a *conjoined EMTDP* into an actual EMTDP. First,

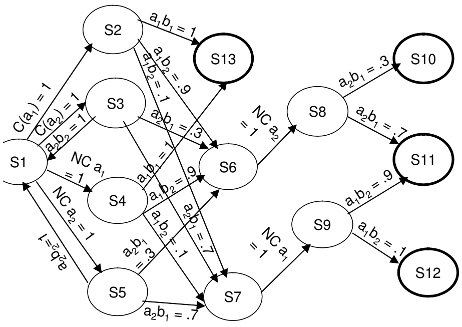


Figure 3. Transformed EMTDP

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.

We must select from these multiple transformations the one that provides the most optimal policies. Fortunately, we prove that none of our series of systematic transformation can lead to any improvement in expected rewards or in reducing communication costs. We begin with changing order of communications.

Lemma 1 *If in a given state S_i all communication of messages of the same length have equal cost and the communication cost is a shared cost, then changing the order of communication (e.g., whether agent A communicates as in figure 2-b vs agent B communicates as in Figure 4-a) has no impact on the expected reward of the optimal EMTDP policy obtained, regardless of the number of actions of agent A vs agent B in State S_i.*

Proof: We prove the result by showing that the optimal policy when agent A communicates first is a lower bound on the optimal policy when agent B communicates first, and vice versa.

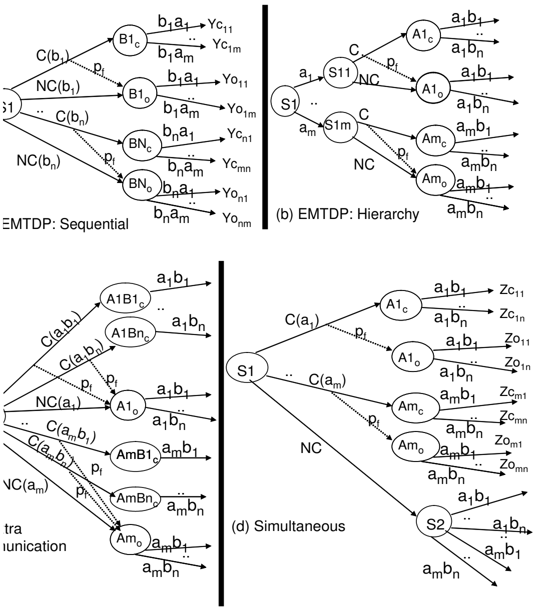


Figure 4. Methods of transformation

Suppose that the flow Xi yields the optimal policy when agent A communicates first (as in Figure 2-b). We now construct a feasible policy for the case when agent B communicates first (as in Figure 4-a) that has the same expected reward, which proves half of the result as the optimal policy for Figure 4-a will have a reward at least this big. Define $Y_{C_{ji}} = X_{C_{ij}}$ and $Y_{O_{ji}} = X_{O_{ij}}$; in essence, if in Figure 2-b, $a_i b_j$ has the flow Xi, then set the flow of $b_j a_i$ in Figure 4-a to Xi.

If we establish the above equivalences, then the following three observations can be made. First, the expected number of times any action $a_i b_j$ is executed after communication and after no communication is the same in Figure 2-b and Figure 4-a. Thus, there is no impact on future states. This means that the cost thresholds would be met even in the new scenario. Second, the cost of communication in the two cases becomes identical. In particular, in Figure 2-b, given a constant cost κ of communication, the total cost of communication is: $\kappa * \sum_{i=1}^{n} \sum_{j=1}^{m} Xc_{ij}/(1 - P_f)$. After setting of $Yc_{ji} = Xc_{ij}$, the communication cost in Figure 4-a is $\kappa * \sum_{j=1}^{m} \sum_{i=1}^{n} Yc_{ji}/(1 - P_f) = \kappa * \sum_{i=1}^{n} \sum_{j=1}^{m} Xc_{ij}/(1 - P_f)$.

Third, given that non-linear constraints in Figure 2-b are satisfied, we can show that non-linear constraints in Figure 4-a are also automatically satisfied. In particular, the non-linear constraints in Figure 2-b are derived from the probability constraint that for any two states Ai_o and Aj_o, where $i, j \in \{1..m\}$, and any action b_k where $k \in \{1..n\}$:

$P(b_k|Ai_o) = P(b_k|Aj_o)$ (Fig 2-b)
$\Rightarrow Xo_{ik}/\sum_{u=1}^{n} Xo_{iu} = Xo_{jk}/\sum_{u=1}^{n} Xo_{ju}$ (for $i, j \in \{1..m\}$)
$\Rightarrow$ for $k_1, k_2 \in \{1...n\}$,
$Xo_{ik_1}/Xo_{jk_1} = Xo_{ik_2}/Xo_{jk_2} = \sum_{u=1}^{n} Xo_{iu}/\sum_{u=1}^{n} Xo_{ju}$
$\Rightarrow Yo_{k_1 i}/Yo_{k_1 j} = Yo_{k_2 i}/Yo_{k_2 j}$
$\Rightarrow Yo_{k_1 i}/\sum_{j=1}^{m} Yo_{k_1 j} = Yo_{k_2 i}/\sum_{j=1}^{m} Yo_{k_2 j}$ (as $i, j \in \{1..m\}$)
$\Rightarrow$ for $k_1, k_2 \in \{1...n\}$, $P(a_i|Bk_{1o}) = P(a_i|Bk_{2o})$ (Fig 4-a)

These non-linear equations imply that the constraints on states Bk$_o$ in Figure 4-a are automatically satisfied. Thus by setting the quantities in Figure 4-a equal to the quantities in Figure 2-b, we have created a feasible policy for Figure 4-a which has the same expected reward as the optimal policy of Figure 2-b, thus the optimal policy for Figure 4-a is better. The proof that the optimal policy when B communicates first is a lower bound on the optimal policy when A communicates first is analogous. □

One immediate conclusion from Lemma 1 is that the agents no longer need to bother who should communicate first. The centralized policy generator can assign an arbitrary order. We are able to similarly prove equivalence results for the hierarchical, extra-communication and simultaneous transformation (remaining proofs are available at http://teamcore.usc.edu/paruchur-proofs.html). In each case, one of the agents must select an action without observation of its actual state, leading to non-linear constraints, e.g., the simultaneous transformation where in state S2, agents A and B must act simultaneously. Once again, non-linear constraints arise, and at least the following non-linear constraints hold in Figure 4-d (N is the number of times action NC is executed in state S1):

for $i \in \{1...m\}$ and $j \in \{1..n\}$, $Z_{ij} = P(a_i|S2) * P(b_j|S2) * N$
$\Rightarrow i, l \in \{1...m\}$, and $j, k \in \{1..n\}$, $Z_{ij}/Z_{ik} = Z_{lj}/Z_{lk}$
In addition, since P(b$_j$|S$_2$) = P(b$_j$|Ai$_o$)
$\Rightarrow \quad Z_{1j}/\sum_{u=1}^{n} Z_{1u} = Z_{ij}/\sum_{u=1}^{n} Z_{iu}$

Thus, non-linear constraints must be added in the simultaneous case also. Indeed, no matter what style of transformation is adopted, non-linear constraints must be added. This is because expressing probabilities of events in EMTDPs requires divisions via Xia variables. And regardless of the transformation that we choose for the EMTDP, 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. Thus:

o **Proposition 3:** It is necessary to add non-convex constraints to solve the actual EMTDP.

This result is contrary to expectations. In particular, since both MDPs and their resource constrained formulations (CMDPs) are solvable via linear programming, and MTDPs for observable environments are solvable via LPs, we earlier expected that EMTDPs (resource constrained versions of MTDPs) will also be solvable by LPs.

Since our sequential transformation has fewest numbers of states, we will use that as the basis of our work. We now present Algorithm 1 that achieves this sequential transformation of *conjoined EMTDP* into a real EMTDP automatically. (In fact our implementation creates a mathematical program 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. From Lemma 1 selecting A or B to communicate first is not relevant. 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 1. 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 EMTDP has $2 * |S| * m$ additional states inserted.

Finally, Proposition 6 states that the algorithmic transformation will indeed yield policies equivalent to the original *conjoined EMTDP* given sufficient communication resources.

o **Proposition 4:** If given a *conjoined EMTDP* E, and an optimal policy π that provides an expected reward r while meeting thresholds t, then a transformed EMTDP T(E) output by Algorithm 1 provides an optimal policy π', with an expected reward r while meeting thresholds t, if for all communication actions Ci, $Cost(Ci) = 0$, and $P_f(Ci) = 0$.

Proof sketch: Given a policy π, for every joint action of the two agents (a_i, b_j) to be taken in state s_i, we instead require in π' that a communication action a$_i$_c be taken in state s_i. Since $P_f(C_i) = 0$, a$_i$_c will lead to only one outcome state sa$_i$_c. In this new state execute (a_i, b_j). Given $Cost(Ci) = 0$, a$_i$_c will not change the expected reward or cost. Thus, we are able to create the required policy π'.

We have focussed on a 2-agent case so far for simplicity. Currently we are investigating into the N-agent scenario. Our intuition is that the various transformations might have some tradeoffs.

Algorithm 1 Convert()

```
Input: $< S, A, P, R, C1, C2, T1, T2, N, Q >$
Output: $< S', A', P', C1', C2', T1', T2', N', Q' >$
Conversion()

Conversion(){
Initialize: $S' = S, A' = A, P' = P, R' = \phi, C1' = \phi, C2' = \phi, T1' = T1, T2' = T2, N' = \phi, Q' = Q$
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($S_{parent}, A_{parent}, S_{current}, A_{current}$){
$S' \leftarrow S' \bigcup S_{current}$
$A' \leftarrow A' \bigcup A_{current}$
$r'(S_{parent}, A_{current}), c1'(S_{parent}, A_{current}), c2'(S_{parent}, A_{current}), n'(S_{parent}, A_{current}) \leftarrow 0$
}

CommToDest($S_{parent}, A_{parent}, S_{current}, A_{current}$){
for all $s' \in S'$ do
    $p'(S_{current}, A_{parent}, s') \leftarrow p(S_{parent}, A_{parent}, s')$
$r'(S_{current}, A_{parent}) \leftarrow r(S_{parent}, A_{parent})$
$c1'(S_{current}, A_{parent}) \leftarrow c1(S_{parent}, A_{parent})$
$c2'(S_{current}, A_{parent}) \leftarrow c2(S_{parent}, A_{parent})$ }
```

4. Approximation algorithms

Two solution approaches exist for solving the EMTDP: one is the use of global optimization software, still limited to small size problems. The second is to use a non-linear solver to obtain a local optimum with no guarantee of global optimality. Here we propose a solution scheme for the EMTDP based on binary search which exploits the problem structure to obtain solutions that are guaranteed to be close to the global optimal even for large sized problems. At the heart of our binary search is the problem of finding a local optimal solution to a non-convex program, that is not a polynomial problem but can be solved efficiently with current algorithms for non-linear optimization.

If the upper(U) and lower(L) bounds on the optimal total reward are known, the binary search method looks for a solution to the EMTDP that additionally has an expected reward $\geq \frac{U+L}{2}$. This is achieved by using a non-linear optimization solver on a bounded EMTDP problem, formed by adding the constraint on expected reward to the EMTDP. The outcome of the non-linear solver will either be a point that satisfies the first order optimality conditions, if a feasible solution to the bounded EMTDP problem exists, or a proof that there are no points with a reward greater than $\frac{U+L}{2}$. In the first case we have found a solution with a expected reward $R \geq \frac{U+L}{2}$, thus the optimal expected reward is $\geq R$, and we can set the new lower bound to $L = R$. If there is no solution with reward higher than $\frac{U+L}{2}$, then the optimal expected reward must be lower than this value, and we can set the new upper bound to $U = \frac{U+L}{2}$.

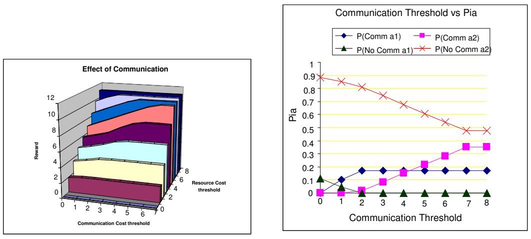
An initial upper bound on the expected reward is obtained by solving the linear programming problem obtained by removing the non-linear constraints of the EMTDP. An initial lower bound is the expected reward of any feasible solution to the EMTDP, for example the first local optimal obtained by the non-linear solver on the EMTDP.

o **Proposition 5**: A solution with total reward within ε of the optimal expected reward is attained after solving a bounded EMTDP with a non-linear solver $O(log\frac{1}{\varepsilon})$ times.

5. Experimental Results on Two Domains

We first present results from our illustrative real EMTDP (Figure 3) to provide key observations about the impact of resource and communication thresholds on policy randomization. Figure 5-a shows the results of varying communication threshold (x-axis) and resource thresholds (y-axis) on the value of the optimal policies (z-axis). In this case, the small size of the problem enables optimal policies obeying resource and communication constraints to be obtained efficiently via a global optimizer. We make two key observations. First, with extreme (very low or very high) resource thresholds, communication threshold makes no difference on the value of the optimal policy. In particular, in extreme cases, the actions are deterministic. On one extreme (zero resource threshold), agents deterministically choose not to play the game at all (state S7) thus gaining a zero expected reward. At the other extreme, with high resources (resource threshold 8) agents gain an expected reward of 11.05, but the agents can choose actions deterministically and thus communication does not help. Second, in the middle range of cost thresholds, where policies are randomized, communication makes the most difference; indeed, the optimal value is seen to increase as communication threshold increases. For instance, when resource cost threshold is 6, the value of the optimal policy obtained without communication is 8.3923, but with high communication threshold of 7, the optimal policy provides a value of 10.7211.

Figure 5-b zooms in on one slice in Figure 5-a (when resource threshold is fixed at 7). It shows the changes in probability of communication actions 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). This graph illustrates that as the communication threshold increases, communication actions increase in probability, e.g., as the communication threshold increases from 0 to 8 the P(comm a_2) increases from 0 to 0.35. The trend is opposite for non-communication actions.



(a) (b)

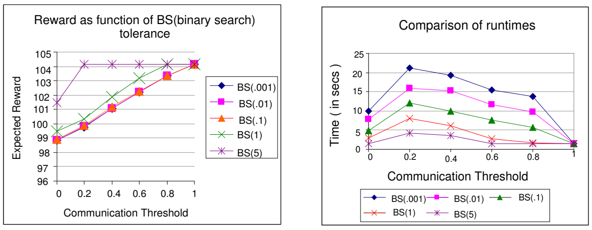
Figure 5. Effect of thresholds on rewards/policy.

Comm Threshold →	0	3	6
Conjoined	10.55	10.55	10.55
Deterministic	0	0	0
Miscoordination	No	No	No
EMTDP	6.99	8.91	10.55

Table 1. Comparing expected rewards.

Table 1 compares the expected rewards of different policies with changes in communication threshold for the example in Figure 1 (with resource threshold = 5). The first row shows three settings of the communication thresholds (0, 3, 6). Row 2 shows the expected rewards obtained by an optimal *conjoined EMTDP* policy. The expected reward (10.55) is an ideal upper-bound for benchmarking and the reward is unaffected by the communication threshold. Row 3 illustrates the results of a deterministic policy that can be executed within the resource constraints: giving a very low expected reward of zero. In this case, agents can only execute action a_1b_1. Row 4 shows the results, where agents take the optimal policy of the *conjoined EMTDP* and attempt to execute it without coordination, as discussed in Section 2. Unfortunately, in all cases, resource constraints are violated, because they execute action a_2b_2 which consumes significant resources. Finally, row 5 shows the expected reward of the actual EMTDP (Figure 3) for comparison. It is able to avoid the problems faced by policies in row 3 and 4. However, with communication threshold of 0, the EMTDP must settle for the optimal expected reward of 6.99; as the communication threshold increases, finally the EMTDP attains the reward obtained by the *conjoined EMTDP*.

Our second domain is inspired by the recent and planned Mars missions, whereby within a decade, significant numbers of rovers and UAVs may be deployed on Mars. We assume a team of two rovers, and several scientists, where each scientist has a daily routine of observations he/she wishes to conduct. Since there are many scientists, a rover can only use a limited amount of energy in serving one scientist. One experiment being conducted by a scientist is observing Martian rocks. The team of rovers must maximize the observation output within the energy budget provided to the scientist. This is a soft constraint because exceeding the energy bound on one day is not a catastrophic mission failure. However overutilizing the given energy budget frequently can interrupt other scientist's work. Uncertainty arises in this domain because a rover's action has only a 0.75 chance of succeeding in its observation. The EMTDP in this domain has 180 total states, leading to a non-linear program of 1500 variables, and 40 non-linear constraints over 200 variables.



(a) (b) Intel Pentium 4 CPU 2.40 GHz

Figure 6. Binary Search with various tolerances

For this problem, we apply our approximate *binary search* method (global optimization in this case is expensive). We illustrate that this method enables us to tradeoff time for precision, paving the way towards practical methods to solve large EMTDPs. In particular, Figure 6-a plots the rewards obtained, varying the communication threshold (x-axis) and plotting the upper and lower bounds on the optimal reward for various levels of tolerance ε(y-axis). The plot of BS(ε) shows the upper bound on the reward for a tolerance ε; the lower bound for all tolerances coincides with the line BS(0.001). Thus, for BS(5) and a communication threshold of 0.6, the optimal solution is between the lower bound of 102.2 and the upper bound of 104.1. When we reduce ε to 0.001, the upper and lower bounds converge, providing us very precise bounds on expected rewards. In general, as the communication threshold increases, the rovers improve their expected reward, via increased communication. Figure 6-b shows that the price of precision in expected rewards is the total run time. Here we plot run-time (y-axis) vs communication threshold (x-axis). As we decrease ε, the run time is seen to increase, e.g., at the communication threshold of 0.2, ε of 0.001 runs at about 22 seconds, 4.5 times slower than when ε is 5.

One key observation is that with the exception of communication threshold of 0, the running times decrease as the communication threshold is increased. This is because a problem that has more resources for communication obtains a local optimal solution that is closer to the initial upper bound we can obtain, and thus the binary search performs less iterations. The special case of zero communication threshold forces some variables to be zero thus reducing the size of the non-convex problem. If our system is severely constrained in communication, *we are better off assuming that there is no communication*, because the marginal improvement in reward is offset by the longer computational effort to determine the optimal solution.

6. Summary and Related Work

This paper provides a novel formalization of teamwork with resource constraints. It provides a distributed MDP framework called EMTDP, where agents must not only maximize their expected team reward, but they must simultaneously bound their expected resource consumption. Second, we introduce an automated algorithm for EMTDP transformation. Thus problems may be formulated using the more abstract conjoined EMTDP and our transformation ensures that resulting policies, even if randomized, avoid miscoordination. We also prove equivalence of different EMTDP transformation strategies. Third, we illustrated that despite fully observable environments, EMTDPs *necessitate* non-linear programs using non-convex constraints. Finally, we provide an approximation algorithm for solving EMTDPs with guaranteed error bounds and illustrate its efficiency on a problem of 1500 variables, and 40 nonlinear constraints. Our results are applicable in other settings where multiple agents must coordinate over randomized policies, e.g., in hostile settings randomized policies may provide unpredictability.

In terms of related work, distributed POMDP research [7, 2, 5] has focused on maximizing the total expected reward, but not on resource bounds, the focus of this paper. Current POMDP research would include resources as part of the reward, but that may lead to undesirable behaviors as the agents try to minimize resource consumption at the expense of their true objective. Furthermore, the issue of policy randomization has not been addressed. Within single agent MDPs, CMDPs enable reasoning about resource constraints[1, 3]. However, generalizing CMDPs to multiagent domains requires coordination of randomized policies, the key contribution in this paper. Indeed, recent research on applying distributed MDPs for multiagent teams [4] complements our research in two ways. First, they do not address the central question in this paper of coordination of randomized policies. In addition, while we focus on consumable resources such as fuel, time etc, [4], they do not focus on such resources. Another area of related work is the Mathematical Programming literature, which has significant amount of research on global optimization algorithms, none of which has polynomial complexity in general [10]. Our binary search approach exploits the structure of the problem by constructing upper and lower bounds to find a solution strategy with better complexity guarantees.

Acknowledgements : This research was supported by NSF grant#0208580 and #IIS-0222914. Sarit Kraus is also affiliated with UMIACS.

References

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

[2] R. Becker, V. Lesser, and C.V. Goldman. Transition-independent decentralized markov decision processes. In *AAMAS*, 2003.

[3] D. Dolgov and E. Durfee. Constructing optimal policies for agents with constrained architectures. In *UMich, CS Technical Report*, 2003.

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

[5] C. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *AAMAS*, 2003.

[6] X. Liu and J. Sun. Robust primal-dual interior point algorithm for nonlinear programs. *SIAM J. Optimization*, 2004.

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

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

[9] M. Tambe. Towards flexible teamwork. *JAIR*, 7:83–124, 1997.

[10] S.A. Vavasis. Complexity issues in global optimization: a survey. In R. Horst and P.M. Pardalos, editors, *Handbook of Global Optimization*, pages 27–41. Kluwer, 1995.

[11] P. Xuan and V. Lesser. Multi-agent policies: from centralized ones to decentralized ones. In *AAMAS*, 2002.