Multiagent Planning with Trembling-hand Perfect Equilibrium in Multiagent POMDPs

Yuichi YABU, Makoto YOKOO, and Atsushi IWASAKI

Graduate School of ISEE, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka, 819-0395, Japan
yabu@agent.is.kyushu-u.ac.jp, {yokoo, iwasaki}@is.kyushu-u.ac.jp

Abstract. Multiagent Partially Observable Markov Decision Processes are a popular model of multiagent systems with uncertainty. Since the computational cost for finding an optimal joint policy is prohibitive, a Joint Equilibrium-based Search for Policies with Nash Equilibrium (JESP-NE) is proposed that finds a locally optimal joint policy in which each policy is a best response to other policies; i.e., the joint policy is a Nash equilibrium.
One limitation of JESP-NE is that the quality of the obtained joint policy depends on the predefined *default policy*. More specifically, when finding a best response, if some observation have zero probabilities, JESP-NE uses this default policy. If the default policy is quite bad, JESP-NE tends to converge to a sub-optimal joint policy.
In this paper, we propose a method that finds a locally optimal joint policy based on a concept called *Trembling-hand Perfect Equilibrium* (TPE). In finding a TPE, we assume that an agent might make a mistake in selecting its action with small probability. Thus, an observation with zero probability in JESP-NE will have non-zero probability. We no longer use the default policy. As a result, JESP-TPE can converge to a better joint policy than the JESP-NE, which we confirm this fact by experimental evaluations.

Key words: Multiagent systems, Partially Observable Markov Decision Process, Nash equilibrium, Trembling-hand perfect equilibrium

1 Introduction

Multiagent systems are increasingly being applied to such critical applications as disaster rescues, distributed unmanned air vehicles (UAV), and distributed sensor nets that demand robust, high-performance designs [1–3]. In these applications, we need to consider the uncertainty arising from various sources such as partial observability, imperfect sensing, etc. Multiagent Partially Observable Markov Decision Processes (Multiagent POMDPs) are emerging as a popular approach for modeling multiagent teamwork in the presence of uncertainty [4–6].

In a single-agent POMDP, a policy of an agent is a mapping of an agent's observation history to actions. The goal is to find an optimal policy that gives the

highest expected reward. In a Multiagent POMDP, the goal is to find an optimal joint policy of agents. Unfortunately, as shown by Bernstein *et al.* [7], the problem of finding an optimal joint policy for a multiagent POMDP is NEXP-Complete if no assumptions are made about the domain conditions. Therefore, a practical policy generation method requires to sacrifice optimality to some extent.

Nair *et al.* [8] propose an algorithm called Joint Equilibrium-based Search for Policy with Nash Equilibrium (JESP-NE) that computes the locally optimal joint policy within a practical runtime. JESP-NE finds a locally optimal joint policy in which each policy is a best response to other policies, i.e., a Nash equilibrium.

One limitation of JESP-NE is that the quality of the obtained joint policy depends on the predefined *default policy*. Assume that there are agent A and B. We first fix agent A's policy and find the best response of agent B. There is a chance that some observation of agent B has zero probability (with the fixed A's policy). In such a case, JESP-NE simply assigns default policy after observation with zero probability. This part of the policy does not affect the expected reward, since the probability that the part of the policy becomes active is zero. However, after finding the best response of agent B, we fix agent B's policy and find the best response of agent A. In this case, part of B's policy (where the default policy is assigned) can be active, since the policy of agent A will change. If the default policy is quite bad, the best response of agent A might be almost identical to the previous policy, so this part of B's policy remains inactive. As a result, JESP-NE converges to a sub-optimal joint policy, which is far from an optimal joint policy.

In this paper, we propose a method that finds a locally optimal joint policy based on a concept called *Trembling-hand Perfect Equilibrium* (TPE) [9]. In finding a TPE, we assume that an agent might make a mistake in selecting its action with small probability. Thus an observation that has zero probability in JESP-NE will have non-zero probability. Therefore, we assign a best response policy rather than the default policy in this part. As a result, the JESP-TPE can converge to a better joint policy than JESP-NE. The experimental results show that JESP-TPE outperforms JESP-NE in settings where the default policy is not good enough.

In the rest of this paper, we first show the multiagent POMDP model used in this paper (Section 2). Next we show an illustrative example, i.e., the multiagent tiger problem (Section 3). Then we describe JESP-NE (Section 4) and JESP-TPE (Section 5). Finally, we show the comparison between the JESP-NE and JESP-TPE through experimental evaluations (Section 6).

2 Model

We follow the Markov Team Decision Problem (MTDP) model [10] as a concrete description of a multiagent POMDP. Given a team of n agents, an MTDP is defined as a tuple: $\langle S, A, P, \Omega, O, R \rangle$. S is a finite set of world states $\{s_1, s_2, \ldots, s_m\}$. $A = \Pi_{1 \leq i \leq n} A_i$, where $A_1, \ldots, A_n$ are the sets of actions for agents 1 to n. A joint action is represented as $\langle a_1, \ldots, a_n \rangle$. $P(s_i, \langle a_1, \ldots, a_n \rangle, s_f)$, the transition

function, represents the probability that the current state is s_f, if the previous state is s_i and the previous joint action is $\langle a_1, \ldots, a_n \rangle$.

$\Omega = \Pi_{1 \leq i \leq n} \Omega_i$ is the set of joint observations where Ω_i is the set of observations for agent i. $O(s, \langle a_1, \ldots, a_n \rangle, \omega)$, the observation function, represents the probability of joint observation $\omega \in \Omega$, if the current state is s and the agents' previous joint action is $\langle a_1, \ldots, a_n \rangle$. We assume that an agent's observations are independent of others' observations. Thus the observation function can be expressed as: $O(s, \langle a_1, \ldots, a_n \rangle, \omega) = O_1(s, \langle a_1, \ldots, a_n \rangle, \omega_1) \cdot \ldots \cdot O_n(s, \langle a_1, \ldots, a_n \rangle, \omega_n)$.

Each agent i forms a belief state, $b_i^t \in B_i$, based on its observations seen through time t, where B_i represents the set of possible belief states for the agent. An agent relies on its state estimator function to update its belief state, given the latest observation. Finally, the agents receive a single, immediate joint reward $R(s, \langle a_1, \ldots, a_n \rangle)$ that is shared equally. This joint reward function is central to the notion of teamwork in a MTDP.

Each agent i chooses its actions based on its local policy, π_i, which is a mapping of its observation history to an action. **Since we assume agents act as a team, i.e., they are cooperative, we don't need to consider non-deterministic/probabilistic policies. Using a Non-deterministic policy (in other words, using a mixed strategy) makes sense if there exists some adversarial agent.** Thus, at time t, agent i will perform action $\pi_i(\boldsymbol{\omega}_i^t)$ where $\boldsymbol{\omega}_i^t = \omega_i^1, \ldots, \omega_i^t$ refers to an observation history of agent i. The important thing to note is that in this model, execution is distributed but planning is centralized. Thus agents don't know each other's observations and actions at runtime but they know each other's policies.

3 Example

We consider a multiagent version of the classic tiger problem introduced in [8]. Two agents are in front of the two rooms and its doors: "left" and "right". Behind one door lies a hungry tiger and behind the other lies untold riches but the agents do not know the position of either. Thus, $S = \{SL, SR\}$, indicating behind which door the tiger is present. The agents can jointly or individually open either door. In addition, the agents can independently listen for the presence of the tiger. Thus, $A_1 = A_2 = \{OpenLeft, OpenRight, Listen\}$. The transition function P, specifies that every time either agent opens one of the doors, the state is reset to SL or SR with equal probability, regardless of the action of the other agent (see Table 1). However, if both agents choose *Listen*, the state is unchanged. The observation function O will return either HL or HR with different probabilities depending on the joint action taken and the resulting world state (see Table 2). For example, if both agents listen and the tiger is behind the left door (state is SL), each agent receives observation HL with probability 0.85 and HR with probability 0.15. Reward function R returns a joint reward (see Table 3). For example, the injury sustained if they opened the door to the tiger is less severe if they open that door jointly than if they open the door alone.

Table 1. Transition function P: * corresponds with either *OpenLeft* or *OpenRight*

Action/Transition	$SL \rightarrow SL$ $(SR \rightarrow SR)$	$SL \rightarrow SR$ $(SR \rightarrow SL)$
$\langle *, * \ or \ Listen \rangle$	0.5	0.5
$\langle * \ or \ Listen, * \rangle$	0.5	0.5
$\langle Listen, Listen \rangle$	1.0	0.0

Table 2. Observation function O: * of state corresponds with either SL or SR, * of action corresponds with *OpenLeft* or *OpenRight*

State	**Action**	HL	HR	$Reset$
SL	$\langle Listen, Listen \rangle$	0.85	0.15	0.0
SR	$\langle Listen, Listen \rangle$	0.15	0.85	0.0
$*$	$\langle *, * \ or \ Listen \rangle$	0.0	0.0	1.0
$*$	$\langle * \ or \ Listen, * \rangle$	0.0	0.0	1.0

Table 3. Reward function R: c corresponds with listen cost, d corresponds with tiger cost, and $d/2$ means the cost when agents jointly open door to tiger.

Action/State	SL	SR
$\langle OpenRight, OpenRight \rangle$	$+20$	$-d/2$
$\langle OpenLeft, OpenLeft \rangle$	$-d/2$	$+20$
$\langle OpenRight, OpenLeft \rangle$	$-d$	$-d$
$\langle OpenLeft, OpenRight \rangle$	$-d$	$-d$
$\langle Listen, Listen \rangle$	$-c$	$-c$
$\langle Listen, OpenRight \rangle$	$+10$	$-d$
$\langle OpenRight, Listen \rangle$	$+10$	$-d$
$\langle Listen, OpenLeft \rangle$	$-d$	$+10$
$\langle OpenLeft, Listen \rangle$	$-d$	$+10$

4 JESP with Nash Equilibrium

In this section, we briefly explain Joint Equilibrium-based Search for Policies with Nash Equilibrium (JESP-NE) [8], which introduces the Nash equilibrium concept as a locally optimal approach. The Nash equilibrium is a solution concept of a game where no player has an incentive to unilaterally change strategy. The key idea in JESP-NE is finding the policy to maximize expected reward for other fixed agent's policies. This process is continued until the joint policy becomes a Nash equilibrium.

Algorithm 1 describes the JESP approach for n agents, given an initial belief state b and finite time horizon T. The key idea behind JESP is focusing on the decision problem for one agent at a time and keeping the policies of the other agent fixed; then the free agent faces a complex but normal single-agent POMDP. We repeat this process until we reach an equilibrium.

Within each iteration of JESP, agent i calls the BestResponse function (line 5 in Algorithm 1) to find an individual policy that maximizes the expected joint reward given that the policies of the other agents are fixed. The problem of finding such a best response policy is equivalent to solving a single-agent POMDP. However, a belief state that stores distribution as in the single-agent case, is not a sufficient statistic because agent i must also reason about the other agents' choice of actions, which, in turn, depend on their observation histories. Thus, at each time t, agent i reasons about the tuple $e_i^t = \langle s^t, \Pi_{j \neq i} \boldsymbol{\omega_j}^t \rangle$, where

Algorithm 1 JESP-NE(b, T)

initialize $V, \langle \pi_1, \ldots, \pi_n \rangle$
repeat
  $convergence \leftarrow true$
  for $i \leftarrow 1$ to n **do**
    $V', \pi_i' \leftarrow$ BESTRESPONSE$(b, \pi_{\neg i}, T)$
    if $\pi_i' \neq \pi_i$ **then**
      $convergence \leftarrow false$
      $V, \pi_i \leftarrow V', \pi_i'$
until convergence
return $V, \langle \pi_1, \ldots, \pi_n \rangle$

Algorithm 2 BESTRESPONSE(b, π_j, T)

$val \leftarrow$ GETVALUE$(b, b, \pi_j, 0, T)$
initialize π_i
$FINDPOLICY(b, b, \langle\rangle, \pi_j, 0, T)$
return val, π_1

$\Pi_{j \neq i} \boldsymbol{\omega_j^t}$ represents the observation histories of the other agents. Then multiagent belief state B for agent i given the distribution over initial state $b(s)$ is defined as:

$$B_i^t(e_i^t) = B_i^t(s^t, \Pi_{j \neq i} \boldsymbol{\omega}_j^t) = Pr(s^t, \Pi_{j \neq i} \boldsymbol{\omega}_j^t | \boldsymbol{\omega}_i^t, \boldsymbol{a}_j^{t-1}, b)$$

Having fixed the policy of agent j $(j \neq i)$, the best response for agent i can be computed using Algorithm 2. Following the model of the single agent value iteration algorithm, central to dynamic program is the value function over a T-step finite horizon. Value function $V_t(B^t, b)$ represents the expected reward that the team will receive, starting from the most recent belief state b and agent i following an optimal policy from the t-th step. Algorithm 3 constructs the optimal policy by maximizing the value function over possible action choices:

$$V_t(B_i^t, b) = \max_{a_i \in A_i} V_t^{a_i}(B_i^t, b) \tag{1}$$

Function $V_t^{a_i}$ can be computed using the GETVALUEACTION function as follows:

$$V_t^{a_i}(B_i^t, b) = \sum_{e^t} B_i^t(e_i^t) \cdot (R(s^t, \langle a_i, \pi_j(\boldsymbol{\omega_j}, T-t) \rangle)) + \sum_{\omega_i^{t+1} \in \Omega_i} Pr(\omega_i^{t+1} | B_i^t, a_i) \cdot V_{t+1}(B_i^{t+1})) \tag{2}$$

B_i^{t+1} is the belief state updated after performing action a_i and observing ω_i^{t+1} and is computed using the UPDATE function.

Algorithm 3 GETVALUE(B^t, b, π_j, t, T)

```
1: if t ≥ T then
2:    return 0
3: if V_t(B^t, b) is already recorded then
4:    return V_t(B^t, b)
5: best ← −∞
6: for all a_1 ∈ A do
7:    value ← GETVALUEACTION(B^t, a_i, b, π_j, t, T)
8:    V_t^{a_i}(B^t, b) ← value
9:    if value > best then
10:      best ← value
11: V_t(B^t, b) ← best
12: return best
```

5 JESP with Trembling-hand Perfect Equilibrium

In this section, we introduce Joint Equilibrium-based Search for Policies with Trembling-hand Perfect Equilibrium (JESP-TPE), which finds a locally optimal joint policy that corresponds to a Trembling-hand Perfect Equilibrium (TPE) [9].

5.1 Nash Equilibrium and Trembling-hand Perfect Equilibrium

JESP-NE obtains a joint policy that corresponds to a Nash equilibrium: i.e., each agent's policy is a best response to other agents' policies. In a Nash equilibrium, no agent has an incentive to unilaterally change strategy, assuming that no other agent changes its policies.

A TPE [1] is proposed as a refinement of the concept of a Nash equilibrium. In a TPE, we assume each agent's policy is a best response to other agents' policies, even if other agents might deviate from the given policies with small probability ϵ.

Definition 1 (Stochastic perturbed policy). ***A stochastic perturbed policy π_i^t for a deterministic policy π_i satisfies the following conditions. If a deterministic action for π_i in a situation corresponds with a^k, a probability that π_i^t gives an action a^k in the situation is $1-\epsilon$, and a probability that π_i^t gives an action $a^j (j \neq k)$ in the situation is $\epsilon/(|A_i|-1)$.***

Definition 2 (Trembling hand perfect equilibrium). ***For two agents, a combination of policies $\langle \pi_1, \pi_2 \rangle$ is a TPE if the following conditions hold for all deterministic policies π_1', π_2', and π_1^t, π_2^t, where π_1^t, π_2^t are***

[1] The definition of a trembling hand perfect equilibrium used in the game theory literature is quite complicated and there exist several alternative definitions [11]. Here, we use a rather simplified definition that matches our model.

Table 4. Policy of agents i and j with finite horizon $T = 2$

$t = 0$		$t = 1$	
observation	action	observation	action
(n/a)	$(Listen, Listen)$	HL	$(OpenRight, OpenRight)$
		HR	$(OpenLeft, OpenLeft)$
		$Reset$	$(OpenLeft, OpenRight)$

perturbed policies of π_1, π_2, ***respectively.***

$$JointReward(\langle \pi_1, \pi_2^t \rangle) \geq JointReward(\langle \pi_1', \pi_2^t \rangle) \tag{3}$$
$$JointReward(\langle \pi_1^t, \pi_2 \rangle) \geq JointReward(\langle \pi_1^t, \pi_2' \rangle) \tag{4}$$

As long as ϵ ***is small enough, if*** (3) ***and*** (4) ***are satisfied, the following conditions also hold.***

$$JointReward(\langle \pi_1, \pi_2 \rangle) \geq JointReward(\langle \pi_1', \pi_2 \rangle)$$
$$JointReward(\langle \pi_1, \pi_2 \rangle) \geq JointReward(\langle \pi_1, \pi_2' \rangle)$$

Thus, a TPE is also a Nash equilibrium, but not vice versa.

First, let us describe a Nash equilibrium in the tiger problem. Table 4 shows a policy for agents i and j in a finite horizon, a listen cost, and a tiger cost ($T = 2$, $c = 2$, $d = 40$). This policy in Table 4 constitutes a Nash equilibrium, because each action of agent i is the best response against j. However, this policy does not satisfy a trembling-hand perfect equilibrium concept. Here, assume that agent j chooses *OpenLeft* by accident at the first step. Then both agents observe *RESET*. Thus agents i and j choose *OpenLeft* and *OpenRight*, respectively and receive rewards of -40: one agent encounters a tiger, while the other finds treasure. Therefore, this policy does not satisfy TPE when an agent may deviate from the given policy with small probability.

Next, we consider a policy that constitutes a TPE. If we change the tuple of the actions for $RESET$ as the observation with step $t = 1$ to $(Listen, Listen)$ as the tuple of the actions, all agents receive a -2 reward, which is greater than the other tuple's reward. Therefore, the changed policy constitutes a TPE. Notice that a TPE always constitutes a Nash equilibrium, but not vice versa.

5.2 JESP-TPE

This subsection describes the details of the proposed algorithm.

One limitation of JESP-NE is that the quality of the obtained joint policy depends on the predefined *default policy*, which is used for a branch with zero probability. Let us show an example. Assume there are two agents i and j, the finite horizon $T = 2$, the listen cost is $c = 30$, and the tiger cost is $d = 40$. The

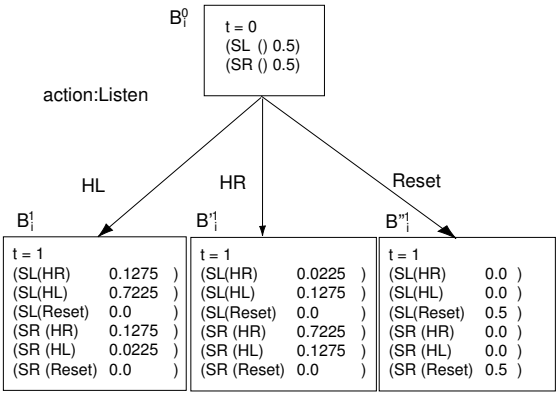

Fig. 1. Trace of tiger scenario

initial policy is: first *Listen*, then *OpenLeft* when *HR* is observed, *OpenRight* when *HL* is observed, and *OpenLeft* when *Reset* is observed. First, we calculate the best response for agent j as: first *Listen*, then *OpenLeft* when *HR* is observed, and *OpenRight* when *HL* is observed. In this case, observation *Reset* has zero probability. In such a case, JESP-NE simply assigns some default policy in this branch. Assume the default policy is always *Listen*. Next we calculate the best response for agent i as: first *Listen*, then *OpenLeft* when *HR* is observed, *OpenRight* when *HL* is observed, and *Listen* when *Reset* is observed. This joint policy is a Nash equilibrium. On the other hand, in JESP-TPE, when calculating the best response of agent j, we assume agent i might take an action that is not specified in the policy with a small probability. Thus, observation *Reset* has non-zero probability. Then, the best response becomes: first *Listen*, then *OpenLeft* when *HR* is observed, *OpenRight* when *HL* is observed, and *OpenLeft* when *Reset* is observed. Next we calculate the best response for agent i as: first *OpenLeft* (or *OpenRight*), then *OpenLeft* when *HR* is observed, *OpenRight* when *HL* is observed, and *OpenLeft* when *Reset* is observed. Here, since agent j chooses *OpenLeft* if *Reset* is observed, the expected utility improves if agent i performs *OpenLeft* (or *OpenRight*) at $t = 1$. Then the best response of agent j becomes: first *OpenLeft*, then *OpenLeft* when *HR* is observed, *OpenRight* when *HL* is observed, and *OpenLeft* when *Reset* is observed. This is a trembling-hand perfect equilibrium, as well as a Nash equilibrium. As shown in this example, by calculating the best response even though some branch that has zero probability, JESP-TPE can converge to a better joint policy than JESP-NE.

Algorithm 4 computes the expected reward for an action, and Algorithm 5 updates the state. Both functions consider the possibility that the other agents may make a mistake and set probabilities which the other agents deviate from the policy to $\epsilon/(|A_j|-1)$ (see lines 5 and 13). Fig. 1 shows different belief states (and possible transitions among them) for agent i in the tiger domain. In this notation, $(SL, (HR))$, which indicates the current world state, is SL, while agent j has

Algorithm 4 *GETVALUEACTION-TPE*$(B^t, a_i, b, \pi_j, t, T)$

$\delta_j \leftarrow \pi_j(b)$
$value \leftarrow 0$
for all $a_j \in A_j, e^t = \langle s^t, \boldsymbol{\omega}_j \rangle$ s.t. $B^t(e^t) > 0$ **do**
  if $a_j == \delta_j(\boldsymbol{\omega}_j, T-t)$ **then**
    $value \stackrel{+}{\leftarrow} B^t(s^t, \boldsymbol{\omega}_j) \cdot R(s^t, \langle a_i, a_j \rangle) \cdot (1-\epsilon)$
  else
    $value \stackrel{+}{\leftarrow} B^t(s^t, \boldsymbol{\omega}_j) \cdot R(s^t, \langle a_i, a_j \rangle) \cdot \frac{1}{|A_j|-1}\epsilon$
for all $\omega_i \in \Omega_i$ **do**
  $B^{t+1} \leftarrow UPDATE(B^t, a, \omega_i, \delta_j)$
  $prob \leftarrow 0$
  for all $a_j \in A_j, s^t, e^{t+1} = \langle s^{t+1}, \boldsymbol{\omega}_j \rangle$ s.t. $B^{t+1}(e^{t+1}) > 0$ **do**
    if $a_j == \delta_j(\boldsymbol{\omega}_j, T-t)$ **then**
      $prob \stackrel{+}{\leftarrow} B^t(s^t, \boldsymbol{\omega}_j) \cdot P(s^t, \langle a_1, a_2 \rangle, s^{t+1}) \cdot O_1(s^{t+1}, \langle a_1, a_2 \rangle, \omega_1) \cdot O_2(s^{t+1}, \langle a_1, a_2 \rangle, \omega_2) \cdot (1-\epsilon)$
    else
      $prob \stackrel{+}{\leftarrow} B^t(s^t, \boldsymbol{\omega}_j) \cdot P(s^t, \langle a_1, a_2 \rangle, s^{t+1}) \cdot O_1(s^{t+1}, \langle a_1, a_2 \rangle, \omega_1) \cdot O_2(s^{t+1}, \langle a_1, a_2 \rangle, \omega_2) \cdot \frac{1}{|A_j|-1}\epsilon$
  $value \stackrel{+}{\leftarrow} prob \cdot GETVALUE(B^{t+1}, t+1)$
return $value$

observed (HR). JESP-TPE explores the combinations of actions, observations, and histories that the JESP-NE does not consider under the assumption that no agent makes a mistake. Therefore, JESP-TPE can examine policies with belief states that JESP-NE no longer examines (see Fig. 1).

For example, JESP-TPE calculates the probability of 0.5 of episodes ($SL(Reset)$) and ($SR(Reset)$) on belief state ${B''}_i^1$, while JESP-NE calculates 0, as shown in Fig. 1, because agent j will probably choose *OpenLeft* or *OpenRight*, although the policy leads j to choose *Listen*. As a result, JESP-TPE searches for the policies on all belief states, while JESP-NE only searches for the policies on B_i^1 and ${B'}_i^1$.

Notice that Algorithm 4 defines value function $V_t^{a_i}$ that represents the expected reward that the agents will receive when agent i follows action a_i at the t-th step. Also, Algorithm 5 defines agent i's belief state B_i^{t+1} at the t+1-st step when i chooses action a_i and receives observation ω_i^{t+1} at the t-th step.

6 Experimental Results

In this section, we perform an empirical comparison of the JESP-NE and -TPE described in Sections 4 and 5 using the tiger scenario in terms of the expected reward of the obtained policies (see Section 3). We ran the algorithm for an initial state chosen randomly, and the initial/default policy is selected as *Listen* for all states. Probability ϵ, with which an agent makes a mistake, is set to a small value, i.e., $1.0E-13$ so that the expected reward is not affected.

Algorithm 5 *UPDATE-TPE*$(B^t, a_i, \omega_i, \pi_j)$

for all e_i^{t+1} **do**
  $B^{t+1}(e^{t+1}) \leftarrow 0$
  for all $a_j \in A_j, s^t \in S$ **do**
    if $a_j == \delta_j(\boldsymbol{\omega}_j)$ **then**
      $B^{t+1}(e^{t+1}) \stackrel{+}{\leftarrow} B^t(e^t) \cdot P(s^t, \langle a_i, \ldots, a_j \rangle, s^{t+1}) \cdot O(s^t, \langle a_i, \ldots, a_j \rangle, \omega) \cdot 1 - \epsilon$
    else
      $B^{t+1}(e^{t+1}) \stackrel{+}{\leftarrow} B^t(e^t) \cdot P(s^t, \langle a_i, \ldots, a_j \rangle, s^{t+1}) \cdot O(s^t, \langle a_i, \ldots, a_j \rangle, \omega) \cdot \frac{1}{|A_j|-1} \epsilon$
normalize B^{t+1}
return B^{t+1}

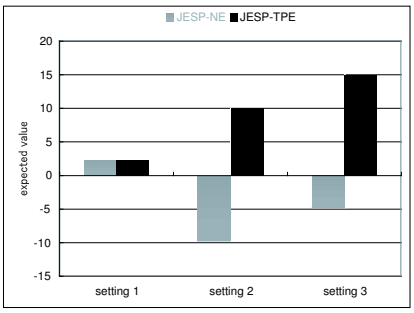

Fig. 2. Expected Reward obtained by the JESP-NE and the JESP-TPE

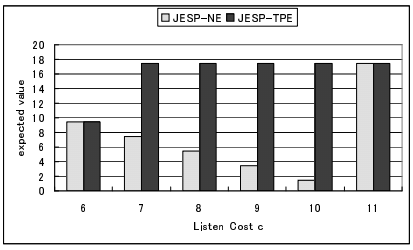

Fig. 3. Expected reward for listen cost c with tiger cost $d = 14$

Fig. 2 shows the expected reward of the two JESPs for three different settings where the tiger cost is 20. When we ran the algorithms with a listen cost of 8 and time horizon is 4 (setting 1), both expected rewards are 2.25. Next, we increase the listen cost to 14 (setting 2). JESP-TPE outperforms JESP-NE, i.e., the expected rewards are 10.0 and -9.70. Then we increase the finite horizon to 5, keeping the listen cost of 14 (setting 3). JESP-TPE again outperforms JESP-NE, i.e., the expected rewards are 15.0 and -4.75.

Let us discuss why our proposed algorithm outperforms the JESP-NE in terms of the expected reward under settings 2 and 3. When both algorithms compute the policy at the first iteration, action *Listen* at the first step is the best action. In JESP-NE, state ${B''}_i^1$ is never reached on Fig. 1 (JESP-TPE changes the policy on B_i^0 from *Listen* to either *OpenLeft* or *OpenRight*.

However, in setting 1, the expected reward in JESP-TPE is identical to JESP-NE. Recall that *Listen* is the best action at the first step. Thus, since the policies on ${B''}_i^1$ are already optimized, the policy does not change even in JESP-TPE. As a result, the expected reward in both algorithms is identical.

Fig. 3 shows the expected reward for a variety of listen costs, i.e., for $1 < c < d$ at $T = 4$ and $d = 14$. We achieve qualitatively similar results when the tiger cost is changed in the range of $[1, 100]$. Thus, we fix it to $d = 14$. For $c \leq 6$ and $c \geq 11$,

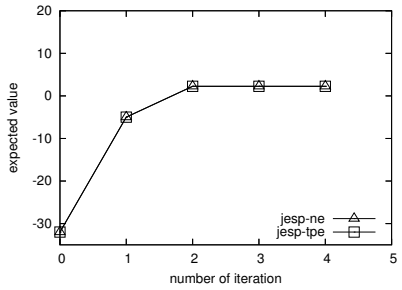

Fig. 4. Expected reward for number of iterations with $T = 4, c = 8$, and $d = 20$

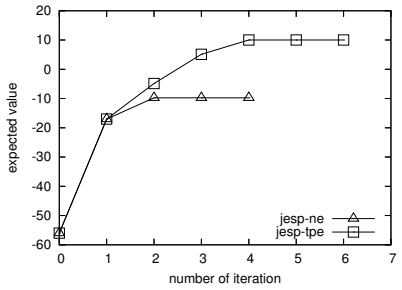

Fig. 5. Expected reward for number of iterations with $T = 4, c = 14$, and $d = 20$

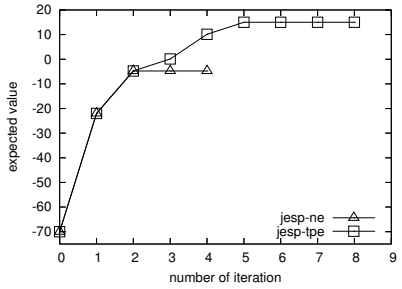

Fig. 6. Expected reward for number of iterations with $T = 5, c = 14$, and $d = 20$

the expected reward in both algorithms is identical, because they converge at the same equilibrium. For $7 \leq c \leq 10$, JESP-TPE significantly outperforms the JESP-NE, since they converge at different equilibria. In summary, our results suggest that, when the listen cost c is about half of tiger cost d, JESP-TPE is likely to outperform JESP-NE. On the other hand, when c is very small or very large, the expected rewards of both algorithms are about the same.

Furthermore, we examined the trajectories of JESP-NE and -TPE until they converge to an equilibrium. Figs. 4-6 show the expected reward for each iteration. The trajectories in Fig. 4 are exactly identical, since both algorithms converge to the same equilibrium. Fig. 5 shows that JESP-TPE outperforms JESP-NE except for the expected reward at the first iteration. JESP-NE converges to an equilibrium at the second iteration, while JESP-TPE does so at the fourth iteration and reaches a better joint policy. Fig. 6 shows a qualitatively similar result to Fig. 5 in setting 3, i.e., the expected reward starts at the same value, and as the iteration continues, JESP-TPE eventually outperforms the JESP-NE.

7 Conclusion

Multiagent POMDPs provide a rich framework to model uncertainties and utilities in complex multiagent domains. Since the problem of finding an optimal joint policy for a multiagent POMDP is NEXP-Complete, we need to sacrifice optimality to some extent. In this paper, we proposed a method that finds a locally optimal joint policy that corresponds to a Trembling-hand Perfect Equilibrium (JESP-TPE), while the existing approach, JESP-NE, finds a joint policy that corresponds to a Nash equilibrium. TPE is stable even if we assume that an agent might make a mistake in selecting its action with small probability, and it is less dependent on the quality of a predefined default policy. The experimental results showed that JESP-TPE outperforms JESP-NE in settings where the default policy is not good enough. In future works, we would like to examine our algorithm in more practical real-world problems.

References

1. Randal W. Beard and Timothy W. McLain. Multiple uav cooperative search under collision avoidance and limited range communication constraints. In *Proceedings of the 42nd Conference Decision and Control*, pages 25–30. IEEE, 2003.
2. Ranjit Nair and Milind Tambe. Hybrid BDI-POMDP framework for multiagent teaming. *Journal of Artificial Intelligence Research*, 17:171–228, 2002.
3. Victor Lesser, Charles Ortiz, and Milind Tambe. *Distributed Sensor Networks: A Multiagent Perspective*. Kluwer, 2003.
4. Ping Xuan and Shlomo Zilberstein Victor Lesser. Communication decisions in multiagent cooperation. In *In Proceedings of the Fifth International Conference on Autonomous Agents*, pages 616–623, 2001.
5. Claudia V. Goldman and Shlomo Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Joint Conference on Agents and Multiagent Systems (AAMAS-03)*, pages 137–144, 2003.
6. Ranjit Nair, Milind Tambe, and Stacy Marsella. Role allocation and reallocation in multiagent teams: Towards a practical analysis. In *Proceedings of the Second International Joint Conference on Agents and Multiagent Systems (AAMAS-03)*, pages 552–559, 2003.
7. Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. The complexity of decentralized control of markov decision processes. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 32–37, 2000.
8. Ranjit Nair, Maayan Roth, Makoto Yokoo, and Milind Tambe. Communication for improving policy computation in distributed pomdps. In *Proceedings of the Third International Joint Conference on Agents and Multiagent Systems (AAMAS-04)*, pages 1098–1105, 2004.
9. Reinhard Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4:25–55, 1975.
10. David V. Pynadath and Milind Tambe. The communicative multiagent team decision problem: Analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16:389–423, 2002.
11. Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.