Expectation Maximization for Average Reward Decentralized POMDPs

Joni Pajarinen[1] and Jaakko Peltonen[2]

[1] Department of Automation and Systems Technology, Aalto University, Finland
Joni.Pajarinen@aalto.fi

[2] Department of Information and Computer Science, Aalto University, Finland
Jaakko.Peltonen@aalto.fi

Abstract. Planning for multiple agents under uncertainty is often based on decentralized partially observable Markov decision processes (Dec-POMDPs), but current methods must de-emphasize long-term effects of actions by a discount factor. In tasks like wireless networking, agents are evaluated by average performance over time, both short and long-term effects of actions are crucial, and discounting based solutions can perform poorly. We show that under a common set of conditions expectation maximization (EM) for average reward Dec-POMDPs is stuck in a local optimum. We introduce a new average reward EM method; it outperforms a state of the art discounted-reward Dec-POMDP method in experiments.

Keywords: Dec-POMDP, average reward, expectation maximization, planning under uncertainty.

1 Introduction

Optimizing the behavior of several agents like robots [25,22] or wireless devices [7,18] is a crucial and hard problem, especially hard in an uncertain world where agents act using only noisy observations about the world and other agents. A decentralized partially observable Markov decision process (Dec-POMDP) can describe the optimal solution. Each agent gets observations on its own and decides its next action to optimize a shared goal. To plan actions, an agent must consider possible action-observation sequences of all agents, thus Dec-POMDP planning is computationally hard: finite-horizon Dec-POMDPs are NEXP-complete (doubly exponential), infinite-horizon Dec-POMDPs are undecidable [6].

In a Dec-POMDP, agents get a joint reward at each time step based on their actions and the world state. Finite-horizon Dec-POMDPs [22,14,23] maximize the sum of rewards over a fixed number of time steps and discounted infinite-horizon Dec-POMDPs [2,10,17] maximize the sum of discounted rewards over an infinite horizon; these objectives emphasize rewards closer to the first time steps, i.e., short-term effects of actions. However, in many Dec-POMDP problems it is natural to maximize *average reward* over an infinite horizon. In wireless networks [7] usual objectives are *average throughput* (average amount of transmitted data,

H. Blockeel et al. (Eds.): ECML PKDD 2013, Part I, LNAI 8188, pp. 129–144, 2013.

infinitely far into the future) or *average delay* (average time a data packet must wait). Such objectives emphasize short and long-term effects of actions equally. Usefulness of average rewards has been shown in robotics [25] and reinforcement learning [13]. Moreover, in finite-horizon and discounted reward methods the solution may depend heavily on the distribution for the first time step (initial belief), which may need to be designed by a domain expert. In many infinite-horizon problems a good initial belief depends on the optimal policy and vice versa (in wireless networks the amount of data in transmit buffers of devices depends on policy efficiency). In contrast, in an average-reward Dec-POMDP the solution does not depend on the initial belief, under certain conditions (see Section 3.1).

Optimizing average reward has been used in partially observable Markov decision processes (POMDPs) for one agent, and for special-case multiple agent problems, but solutions for generic multiple agent problems have not been given. Interaction of agents is essential e.g. in wireless network channel access [18]. We introduce a solution for multiple agents with partial observability: a Dec-POMDP method that optimizes average reward by a modified expectation-maximization (EM) algorithm. To our knowledge this is the *first general Dec-POMDP method for optimizing average reward.*

2 Related Work

We discuss related work on average reward Markov decision processes (MDPs), partially observable MDPs (POMDPs), and decentralized MDPs (Dec-MDPs). A fully observable POMDP or a single agent Dec-MDP is an MDP, a single agent Dec-POMDP is a POMDP, and a jointly fully observable Dec-POMDP is a Dec-MDP. The Dec-POMDP is the most general of these models. We know of previous work on average reward MDPs [13,21], average reward POMDPs [1,29,12], discounted reward POMDPs [20,8,2,17], transition and observation independent average reward Dec-MDPs [19], finite-horizon Dec-POMDPs [22,14,23], and discounted-reward Dec-POMDPs [24,5,4,3,2,10,17], but not on general average reward Dec-POMDPs. For *average reward MDPs*, policy iteration, value iteration, linear programming [21] and model-free methods [13] exist. Mahadevan et al. [13] showed average reward outperformed discounted reward in MDPs where an agent chose small short-term or large long-term rewards. Methods exist for *average reward POMDPs*: Li et al. [12] find memoryless policies, Yu et al. [29] use lower bound approximations, and Aberdeen [1] improves a finite state controller by gradient methods. For decentralized problems, Petrik et al. [19] show transition&observation independent *average reward Dec-MDPs* are NP-complete, and use bi-linear programming. Yagan et al. [28] minimize average cost in a transition&observation independent special-case Dec-POMDP where agents don't affect or sense the world state seen by other agents. In general Dec-POMDPs agents affect each other in complex ways. To our knowledge there is no research on general average reward Dec-POMDPs, but research on finite-horizon [22,14,23] and discounted reward Dec-POMDPs [24,5,4,3,2,10,17] exists.

Kakade et al. [9] showed MDP average reward could be approximated by discounting with large discount factor, but in our experiments real average reward optimization outperformed discounting.

3 Dec-POMDP

A Dec-POMDP is a solution to multi-agent planning under uncertainty about the world and other agents. It is defined by a set of N agents, the set of actions A, the set of states S, the set of observations O, the observation probability $P(\boldsymbol{o}|s', \boldsymbol{a})$, the state transition probability $P(s'|s, \boldsymbol{a})$, and real valued immediate reward function $R(s, \boldsymbol{a})$. Here $\boldsymbol{o}$ denotes the observations $o_1, \ldots, o_N$ and $\boldsymbol{a}$ the actions $a_1, \ldots, a_N$ of all agents. In each time step, the world starts from state s, each agent i takes action a_i, and the world transitions to the next state s' with probability $P(s'|s, \boldsymbol{a})$. Agents then make their observations $\boldsymbol{o}$ with probability $P(\boldsymbol{o}|s', \boldsymbol{a})$ and the action-observation cycle begins again. An agent does not sense actions, states or observations of other agents, so computational complexity of planning is high. In each time step the agents get immediate reward $R(s, \boldsymbol{a})$ depending on their actions $\boldsymbol{a}$ and the world state s. The finite-horizon objective is to maximize reward $E[\sum_{t=0}^{T} R_t(s, \boldsymbol{a})|\pi]$ where T is the horizon, π is the policy (consisting of the individual policies of all agents), and $R_t(s, \boldsymbol{a})|\pi$ is the reward at time step t following π. In the discounted reward case, expected discounted reward over an infinite-horizon $E[\sum_{t=0}^{\infty} \gamma^t R_t(s, \boldsymbol{a})|\pi]$ is maximized, with discount factor $0 < \gamma < 1$. With discounting, reward decreases geometrically with the horizon. Both finite-horizon and discounted reward objectives need an initial state probability distribution $b_0(s)$ called the initial belief.

Finite state controllers (FSCs) have been used as policy in POMDP [20,8,2,17] and infinite-horizon discounted reward Dec-POMDP [24,5,4,3,2,10,17] methods. The FSC of agent i consists of a set $\{q_i\}$ of FSC states q_i, an action probability distribution $P(a_i|q_i)$, and FSC state transition probability $P(q'_i|q_i, o_i)$. For simplicity, similar to the approach in [2], an agent starts in state $q_i = 1$. In each time step, agent i in state q_i takes action a_i with probability $P_{aq}^{(i)} = P(a_i|q_i)$. The world transitions to a new world state, the agent gets observation o_i about the world, and moves to a new FSC state q'_i with probability $P_{q'qo}^{(i)} = P(q'_i|q_i, o_i)$.

3.1 Average Reward Dec-POMDP

Intuitively average reward Dec-POMDPs optimize the policy to maximize average reward over an infinite horizon. Formally, they must maximize $R_{\text{average}} = E\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} R_t(s, \boldsymbol{a})|\pi\right]$. Unlike finite-horizon and discounted reward objectives, R_{average} does not need a parameter controlling effective planning horizon and depending on the underlying Markov chain does not need an initial belief. In Dec-POMDPs, an agent needs the full observation history to make optimal decisions [6]. As average reward Dec-POMDPs run the policy for arbitrarily long times, we use FSCs as policies taking a fixed amount of memory. For a set of FSCs (one per agent), the world state s and the FSC states q_i

together form a state of a Markov chain as follows: given the current state $(s, \boldsymbol{q})$, where $\boldsymbol{q} = q_1, \ldots, q_N$, the **probability for the next time step state** $(s', \boldsymbol{q}')$ is $P_{s'\boldsymbol{q}'s\boldsymbol{q}} = P(s', \boldsymbol{q}'|s, \boldsymbol{q}) = \sum_{\boldsymbol{a},\boldsymbol{o}} P(\boldsymbol{o}|s', \boldsymbol{a})P(s'|s, \boldsymbol{a}) \prod_i \left(P^{(i)}_{aq} P^{(i)}_{q'qo}\right)$. With initial belief $b_0(s)$ the **initial probability distribution** over $(s, \boldsymbol{q})$ is $P_0(s, \boldsymbol{q}) = b_0(s) \prod_i P(q_i)$ and $P_t(s, \boldsymbol{q})$ is the initial distribution projected t time steps into the future. The **expected immediate reward** for $P_t(s, \boldsymbol{q})$ is $\sum_{s,\boldsymbol{q},\boldsymbol{a}} P_t(s, \boldsymbol{q})R(s, \boldsymbol{a}) \prod_i P^{(i)}_{aq}$. The **optimization objective** is then $R_{\text{FSCs}} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{s,\boldsymbol{q},\boldsymbol{a}} \left(P_t(s, \boldsymbol{q})R(s, \boldsymbol{a}) \prod_i P^{(i)}_{aq}\right)$. Average reward Dec-POMDP problems can be grouped by properties of the above-described Markov chain. We consider fully stochastic policies, like FSCs with nonzero action and transition probabilities; the properties below don't depend on the policy as long as it is fully stochastic. Useful Markov chain classes (similar to [21]) are *Recurrent* - all states reachable from all states; *Periodic* - the greatest common divisor of the return time of one or more states is greater than one; *Aperiodic* - no state is periodic; *Unichain* - one set of recurrent states and a set of zero or more transient states; and *Multichain* - two or more closed irreducible sets of recurrent states and zero or more transient states. We focus on aperiodic problems. When the Markov chain is *aperiodic*, $P_t(s, \boldsymbol{q})$ converges to a stationary limiting distribution $P_*(s, \boldsymbol{q}) = \lim_{t\to\infty} P_t(s, \boldsymbol{q})$. Since rewards are bounded, R_{FSCs} becomes

$$R_{\text{FSCs,aperiodic}} = \sum_{s,\boldsymbol{q},\boldsymbol{a}} P_*(s, \boldsymbol{q})R(s, \boldsymbol{a}) \prod_i P^{(i)}_{aq} \ . \tag{1}$$

For *multichain* Markov chains, the limiting distribution depends on initial belief: if e.g. a robot can enter one of two hallways but cannot switch later, its start position affects the limiting distribution. For *unichain* Markov chains the limiting distribution does not depend on initial belief. Average reward unichain models are of practical interest: in a wireless network case, agents' transmission buffer sizes are the world state and transmission policies must be optimized to keep buffers as empty as possible; the reward is the negative sum of buffer sizes and the initial belief is the distribution over buffer sizes. Generally initial belief influences the best achieved policy so the belief should be optimized with the policy, but for unichain Markov chains we need not optimize initial belief since the optimal policy always yields the optimal limiting distribution.

4 Expectation-Maximization Planning

Expectation-maximization (EM) has been used to optimize finite state controllers (FSCs) for discounted reward in MDPs and POMDPs [27], Dec-POMDPs [10,17], and factored Dec-POMDPs [16]. In EM the idea is to scale rewards into probabilities and, by inference, find FSC parameters maximizing the reward likelihood. EM has been extended to problems with huge [16] and continuous [27] state spaces. **We now introduce an average reward EM method for aperiodic Dec-POMDPs**. (For experiments, we introduce a nonlinear programming based alternative in the Appendix.) We first use a traditional EM approach and show the result is stuck in a local optimum under certain conditions (see Section 4.1 for more details); we then use it as a foundation for a modified approach and introduce a practical EM method that yields good results.

An EM approach scales the real valued reward function into a binary reward variable r. Denote with $\hat{R}_{s\boldsymbol{a}} = \hat{R}(r = 1|s, \boldsymbol{a})$ the conditional probability for r to be one given actions $\boldsymbol{a}$ and state s. $\hat{R}_{s\boldsymbol{a}}$ is computed by scaling the real valued reward function using the minimum R_{min} and maximum R_{max} rewards: $\hat{R}_{s\boldsymbol{a}} = (R(s, \boldsymbol{a}) - R_{min})/(R_{max} - R_{min})$. In average reward Dec-POMDPs, FSC parameters θ are optimized to maximize average reward over time, scaled as above into a likelihood of a binary reward:

$$P(r = 1|\theta) = \lim_{T_M \to \infty} \sum_{T=0}^{T_M - 1} \frac{1}{T_M} \sum_{s,\boldsymbol{q},\boldsymbol{a}} \hat{R}_{s\boldsymbol{a}} P_T(s, \boldsymbol{q}) \prod_i P_{aq}^{(i)} \;, \tag{2}$$

where the horizon T_M is taken to the limit. It can be shown (2) corresponds to the original average reward objective (1); moreover the continuous expected average reward $R(\theta)$ can be extracted from the likelihood of the binary reward as $R(\theta) = P(r = 1|\theta)(R_{max} - R_{min}) + R_{min}$. Each EM iteration consists of an E- and M-step: the E-step computes alpha and beta messages with old FSC parameters to compute the log likelihood function, and the M-step finds new FSC parameters that maximize the log likelihood.

E-step. Based on the current policy parameters θ, the E-step computes alpha $\alpha_t^{s\boldsymbol{q}}$ and beta $\beta_t^{s\boldsymbol{q}}$ messages:

$$\alpha_0^{s\boldsymbol{q}} = P_0(s, \boldsymbol{q}) \;, \quad \alpha_t^{s'\boldsymbol{q}'} = \sum_{s,\boldsymbol{q}} P_{s'\boldsymbol{q}'s\boldsymbol{q}} \alpha_{t-1}^{s\boldsymbol{q}} \;, \tag{3}$$

$$\beta_0^{s,\boldsymbol{q}} = \sum_{\boldsymbol{a}} \hat{R}_{s\boldsymbol{a}} \prod_i P_{aq}^{(i)} \;, \; \beta_{t+1}^{s\boldsymbol{q}} = \sum_{s',\boldsymbol{q}'} P_{s'\boldsymbol{q}'s\boldsymbol{q}} \beta_t^{s'\boldsymbol{q}'} \;. \tag{4}$$

M-step. Let L_t denote a sequence of world states and FSC state, observation, and action variables of all agents from time $t = 0$ to T, so that $L_T = \{(s_t, q_{1,t}, \ldots, q_{N,t}, o_{1,t}, \ldots, o_{N,t}, a_{1,t}, \ldots, a_{N,t})\}_{t=0}^{T}$. Moreover, use $P_{\boldsymbol{o}s's\boldsymbol{a}}^{(t)}$ to denote $P(\boldsymbol{o}_{t+1}, s_{t+1}|s_t, \boldsymbol{a}_t)$ and $\hat{R}_{s\boldsymbol{a}}^{(t)}$ to denote $\hat{R}(r_t = 1|s_t, \boldsymbol{a}_t)$, and for agent i denote $P(a_{i,t}|q_{i,t})$ with $P_{aq}^{(i,t)}$ and $P(q_{i,t+1}|q_{i,t}, o_{i,t+1})$ with $P_{q'qo}^{(i,t)}$. Denote the set of current FSC parameters (action and transition probabilities) by θ and the set of new parameters by $\acute{\theta}$. In the M-step, EM maximizes the expected log likelihood $Q(\theta, \acute{\theta})$ denoted here with Q with respect to the new FSC parameters $\acute{\theta}$:

$$Q = \lim_{T_M \to \infty} \sum_{T=0}^{T_M - 1} \sum_{L_T} P_{\theta,T_M}^{r,L_T,T} \log P_{\acute{\theta},T_M}^{r,L_T,T} \;, \tag{5}$$

where

$$\log P^{r,L_T,T}_{\acute{\theta},T_M} = \log P(r=1, L_T, T|\acute{\theta}, T_M) = \log \hat{R}^{(T)}_{\boldsymbol{sa}} + \log P(s_0, \boldsymbol{q}_0)$$
$$+ \sum_{t=1}^{T} \log P^{(t-1)}_{\boldsymbol{o}s'\boldsymbol{sa}} + \sum_i (\sum_{t=0}^{T} \log \acute{P}^{(i,t)}_{aq} + \sum_{t=1}^{T} \log \acute{P}^{(i,t-1)}_{q'qo}) - \log T_M \quad (6)$$

is the log-probability to receive binary reward $r = 1$ after the latent sequence of actions and states L_T. (6) shows that the new FSC probabilities of agent i (action probability $\acute{P}^{(i,t)}_{aq}$ and FSC transition probability $\acute{P}^{(i,t-1)}_{q'qo}$) do not depend on the new distributions of other agents. For brevity, denote sets of sum indices as $V_{\neq i} = \{s, q_{j\neq i}, a_{j\neq i}\}$ and $W_{\neq i} = \{s, q_{j\neq i}, a_{j\neq i}, s', \boldsymbol{o}, \boldsymbol{q}'\}$.

We now construct $\tilde{Q}_i$, the part of Q affecting the new action probability $\acute{P}^{(i)}_{aq}$ for agent i. We use $\acute{P}^{(i,t)}_{aq}$ to denote $\acute{P}^{(i)}_{aq}$ at time step t and use $P(r = 1, a_{i,t} = a_i, q_{i,t} = q_i|T, \theta) = \sum_{W_{\neq i}} \alpha^{\boldsymbol{sq}}_t P_{\boldsymbol{o}s'\boldsymbol{sa}} P^{(i)}_{q'qo} \prod_{j\neq i} P^{(j)}_{aq} P^{(j)}_{q'qo} \beta^{s'\boldsymbol{q}'}_{T-t-1}$ to denote the probability of a binary reward $r = 1$ at time T, when at $t \leq T$ agent i takes action a_i and the FSC state is q_i. Since $\sum_{L_T} P^{r,L_T,T}_{\theta,T_M} \sum_{t=0}^{T} \log \acute{P}^{(i,t)}_{aq} = \sum_{t=0}^{T} \sum_{a_i,q_i} P(r = 1, a_{i,t} = a_i, q_{i,t} = q_i|T, \theta) \log \acute{P}^{(i)}_{aq}$, inserting (6) into (5) yields

$$\tilde{Q}_i = \lim_{T_M\to\infty} \sum_{T=0}^{T_M-1} \sum_{t=0}^{T} \sum_{W_{\neq i}} \frac{\alpha^{\boldsymbol{sq}}_t}{T_M} P_{\boldsymbol{o}s'\boldsymbol{sa}} P^{(i)}_{q'qo} \big(\prod_{j\neq i} P^{(j)}_{aq} P^{(j)}_{q'qo}\big) \beta^{s'\boldsymbol{q}'}_{T-t-1} \log \acute{P}^{(i)}_{aq} \ . \quad (7)$$

In (7), breaking the sum over t into $t = T$ and $t = 0, \ldots, T-1$ yields $\tilde{Q}_i = \sum_{a_i,q_i} P^{(i)}_{aq} \log \acute{P}^{(i)}_{aq} \lim_{T_M\to\infty} \sum_{T=0}^{T_M-1} \Big[\sum_{V_{\neq i}} \frac{\hat{R}_{\boldsymbol{sa}}}{T_M} \big(\prod_{j\neq i} P^{(j)}_{aq}\big) \alpha^{\boldsymbol{sq}}_T +$
$\sum_{t=0}^{T-1} \sum_{W_{\neq i}} \frac{\alpha^{\boldsymbol{sq}}_t}{T_M} P_{\boldsymbol{o}s'\boldsymbol{sa}} P^{(i)}_{q'qo} \big(\prod_{j\neq i} P^{(j)}_{aq} P^{(j)}_{q'qo}\big) \beta^{s'\boldsymbol{q}'}_{T-t-1} \Big]$.
Because $\acute{P}^{(i)}_{aq}$ is normalized over a_i, maximizing $\tilde{Q}_i$ with respect to $\acute{P}^{(i)}_{aq}$ yields $\acute{P}^{(i)}_{aq} = P^{(i)}_{aq} \lim_{T_M\to\infty} \frac{1}{C_{q_i}} \sum_{T=0}^{T_M-1} \Big[\sum_{V_{\neq i}} \hat{R}_{\boldsymbol{sa}} \prod_{j\neq i} P^{(j)}_{aq} \alpha^{\boldsymbol{sq}}_T$
$+ \sum_{t=0}^{T-1} \sum_{W_{\neq i}} \alpha^{\boldsymbol{sq}}_t P_{\boldsymbol{o}s'\boldsymbol{sa}} P^{(i)}_{q'qo} \prod_{j\neq i} P^{(j)}_{aq} P^{(j)}_{q'qo} \beta^{s'\boldsymbol{q}'}_{T-t-1} \Big]$,
where C_{q_i} is a normalizing constant.

Similarly to [26,10], we separate sums over alpha and beta messages using $\lim_{T_M\to\infty} \sum_{T=0}^{T_M-1} \sum_{t=0}^{T-1} \frac{\alpha^{\boldsymbol{sq}}_t \beta^{s'\boldsymbol{q}'}_{T-t-1}}{T_M} = \lim_{T_M\to\infty} \sum_{t=0}^{T_M-1} \alpha^{\boldsymbol{sq}}_t \sum_{\tau=0}^{T_M-1} \frac{\beta^{s'\boldsymbol{q}'}_{\tau}}{T_M}$, where $\tau = T - t - 1$. The action probability update becomes

$$\acute{P}^{(i)}_{aq} = P^{(i)}_{aq} \lim_{T_\alpha,T_\beta\to\infty} \frac{1}{C_{q_i}} \Bigg[\sum_{V_{\neq i}} \hat{R}_{\boldsymbol{sa}} \big(\prod_{j\neq i} P^{(j)}_{aq}\big) \sum_{T=0}^{T_\alpha-1} \alpha^{\boldsymbol{sq}}_T +$$
$$\sum_{W_{\neq i}} \sum_{t=0}^{T_\alpha-1} \alpha^{\boldsymbol{sq}}_t P_{\boldsymbol{o}s'\boldsymbol{sa}} P^{(i)}_{q'qo} \big(\prod_{j\neq i} P^{(j)}_{aq} P^{(j)}_{q'qo}\big) \sum_{\tau=0}^{T_\beta-1} \beta^{s'\boldsymbol{q}'}_{\tau} \Bigg] , \quad (8)$$

where we have used alpha and beta horizons, T_α and T_β, in place of T_M, to be used in later discussions.

4.1 Analysis: Stuck in a Local Optimum

We now prove that the action probability update of the traditional EM approach is stuck in a local optimum under certain conditions. The proof that the FSC transition probability updates are stuck is similar and is omitted.

The proof requires stochastic FSCs and that each closed irreducible state set has at least one state with a non-zero reward probability. These conditions are common. Firstly, the policy is usually stochastic, because a deterministic policy is always stuck, even in discounted POMDPs/Dec-POMDPs, because of the multiplicative nature of EM parameter updates. Secondly, the reward probability condition is common. If all sets of irreducible states have zero reward probability, then only transient states have non-zero reward probability. Therefore, the reward probability approaches zero at distant time steps and the need for taking long-term effects of actions into account, the motivation behind average reward, disappears. There may be multichain problems where some of the irreducible closed state sets have, and others do not have, non-zero reward probabilities. We are not aware of such problems, but this may need further investigation.

Note that the proof applies also to average reward POMDPs. The proof for POMDPs is obtained by just setting the number of agents to one. We do not claim the proof to hold in problems without stochastic controllers (e.g. it is possible to use EM in MDPs so that the action probability depends directly on the world state). In particular, we assume in the proof that $\sum_s \alpha_*^{s\boldsymbol{q}} > 0$ for all $\boldsymbol{q}$, which is true for stochastic controllers.

Preliminary. Recall that $\alpha_t = \{\alpha_t^{s\boldsymbol{q}}\}$ is a projection of the initial belief for t steps following the current policy. To measure difference between a probability distribution and the limiting distribution, we use the *total variation distance* D_{TV} [11], defined as the largest absolute difference of the probability of the same state in two distributions. The distance between distribution α_t at time step t and the limiting distribution α_* is $D_{TV}(\alpha_t, \alpha_*) = \max_{s,\boldsymbol{q}} |\alpha_t^{s\boldsymbol{q}} - \alpha_*^{s\boldsymbol{q}}|$. In aperiodic Markov chains, total variation distance decreases exponentially[1] with time t:

$$D_{TV}(\alpha_t, \alpha_*) \leq C_\epsilon \epsilon^t; 0 < \epsilon < 1 \ , \tag{9}$$

where $C_\epsilon > 0$ and ϵ are constants. In unichains the limiting distribution is unique, but in multichains it depends on the starting distribution. We will not denote the dependence on the starting distribution explicitly but we refer to it when necessary.

Theorem 1. *In unichain and multichain aperiodic Dec-POMDPs, the EM action probability update never changes finite state controller (FSC) parameter values, when each closed irreducible state set has at least one state for which a non-zero reward probability exists, and when the FSC policy is fully stochastic.*

[1] Theorem 4.9 in [11] shows this for aperiodic irreducible Markov chains. It is straightforward to modify the proof of the theorem to also apply to aperiodic unichains and multichains, which may have transient states in addition to irreducible communicating classes of states: the equilibrium distribution π in [11] is just replaced with a limiting distribution, which has a zero probability for each transient state.

Proof. We write (8) as $\acute{P}_{aq}^{(i)} = P_{aq}^{(i)}\Big[H_{aq}^{(i)} + J_{aq}^{(i)}\Big]$, where $H_{aq}^{(i)}$ is the expected sum of reward probabilities gained over all time in situations where agent i was in state q_i and took action a_i, scaled by a normalization term, and $J_{aq}^{(i)}$ is the expected sum of reward probabilities over the future from such situations, again scaled by the normalization term. We have $H_{aq}^{(i)} = \lim_{T_\alpha, T_\beta \to \infty} \frac{1}{C_{q_i}} \tilde{H}_{aq}^{(i)}$ and $J_{aq}^{(i)} = \lim_{T_\alpha, T_\beta \to \infty} \frac{1}{C_{q_i}} \tilde{J}_{aq}^{(i)}$ where $C_{q_i} = \sum_{a_i} P_{aq}^{(i)} (\tilde{H}_{aq}^{(i)} + \tilde{J}_{aq}^{(i)})$ is the normalization term, and we denoted $\tilde{H}_{aq}^{(i)} = \sum_{V_{\neq i}} \hat{R}_{s\boldsymbol{a}} \prod_{j \neq i} P_{aq}^{(j)} \sum_{T=0}^{T_\alpha - 1} \alpha_T^{s\boldsymbol{q}}$ and also denoted $\tilde{J}_{aq}^{(i)} = \sum_{q_{j\neq i}, a_{j \neq i}, s, s', \boldsymbol{q}'} \sum_{t=0}^{T_\alpha - 1} \alpha_t^{s\boldsymbol{q}} P_{s'\boldsymbol{q}' a_i s \boldsymbol{q}}^{(i)} \sum_{\tau=0}^{T_\beta - 1} \beta_\tau^{s'\boldsymbol{q}'}$. The term $P_{s'\boldsymbol{q}' s \boldsymbol{q} a_i}^{(i)} = \sum_{\boldsymbol{o}, a_{j \neq i}} P_{\boldsymbol{o} s' s \boldsymbol{a}} P_{q'qo}^{(i)} \prod_{j \neq i} P_{aq}^{(j)} P_{q'qo}^{(j)}$ is the probability that the world and agents will transition to states $(s', \boldsymbol{q}')$ given their current states $(s, q_{j \neq i})$ and a specific action a_i and controller state q_i of the ith agent. For convenience, define $\hat{J}_{aq}^{(i)}$ as

$$\hat{J}_{aq}^{(i)} = \lim_{T_\alpha, T_\beta \to \infty} \frac{T_\alpha T_\beta}{\hat{C}_{q_i}} \sum_{s_\tau, \boldsymbol{a}_\tau, \boldsymbol{q}_\tau} \hat{R}_{s_\tau \boldsymbol{a}_\tau} \alpha_*^{s_\tau, \boldsymbol{q}_\tau} \prod_j P_{a_\tau q_\tau}^{(j)} = \lim_{T_\alpha, T_\beta \to \infty} \frac{T_\alpha T_\beta}{\hat{C}_{q_i}} \cdot \mathrm{const}\ ,$$

here $\hat{C}_{q_i} = \sum_{a_i} T_\alpha T_\beta \sum_{s_\tau, \boldsymbol{a}_\tau, \boldsymbol{q}_\tau} \hat{R}_{s_\tau \boldsymbol{a}_\tau} \alpha_*^{s_\tau, \boldsymbol{q}_\tau} \prod_j P_{a_\tau q_\tau}^{(j)}$ is another normalizing term. We now prove that $J_{aq}^{(i)} = \hat{J}_{aq}^{(i)}$ and $H_{aq}^{(i)} = 0$. We will then show $\hat{J}_{aq}^{(i)}$ converges to a constant and that the action update is thus stuck.

To prove $J_{aq}^{(i)} = \hat{J}_{aq}^{(i)}$ we show that $|J_{aq}^{(i)} - \hat{J}_{aq}^{(i)}| = 0$. Expand the recursive form of beta messages as

$$\beta_\tau^{s, \boldsymbol{q}} = \sum_{s_\tau, \boldsymbol{q}_\tau, \boldsymbol{a}_\tau} P(s_\tau, \boldsymbol{q}_\tau | s_0 = s, \boldsymbol{q}_0 = \boldsymbol{q}) \hat{R}_{s_\tau \boldsymbol{a}_\tau} \prod_j P_{a_\tau q_\tau}^{(j)}\ ,$$

where $P_{s,\boldsymbol{q}}^{s_\tau, \boldsymbol{q}_\tau} = P(s_\tau, \boldsymbol{q}_\tau | s_0 = s, \boldsymbol{q}_0 = \boldsymbol{q})$ is the probability to arrive at world and controller states $s_\tau, \boldsymbol{q}_\tau$ in τ steps when starting from $s, \boldsymbol{q}$. Use the expanded form to compute an upper bound on $|J_{aq}^{(i)} - \hat{J}_{aq}^{(i)}|$:

$$\Big| \lim_{T_\alpha, T_\beta \to \infty} \Big[\frac{1}{C_{q_i}} \sum_{\substack{q_{j \neq i}, a_{j \neq i} \\ s, s', \boldsymbol{q}'}} \sum_{t=0}^{T_\alpha - 1} \alpha_t^{s\boldsymbol{q}} P_{s'\boldsymbol{q}' a_i s\boldsymbol{q}}^{(i)} \sum_{\tau=0}^{T_\beta - 1} \beta_\tau^{s'\boldsymbol{q}'} - \frac{T_\alpha T_\beta}{\hat{C}_{q_i}} \sum_{s_\tau, \boldsymbol{a}_\tau, \boldsymbol{q}_\tau} \Big($$

$$\hat{R}_{s_\tau \boldsymbol{a}_\tau} \alpha_*^{s_\tau, \boldsymbol{q}_\tau} \prod_j P_{a_\tau q_\tau}^{(j)} \Big) \Big] \Big| \leq \Big| \lim_{T_\alpha, T_\beta \to \infty} T_\alpha \Big[\frac{1}{C_{q_i}} \sum_{\substack{q_{j \neq i}, a_{j \neq i} \\ s, s', \boldsymbol{q}'}} \alpha_*^{s\boldsymbol{q}} P_{s'\boldsymbol{q}' a_i s\boldsymbol{q}}^{(i)}$$

$$\sum_{\tau=0}^{T_\beta - 1} \sum_{s_\tau, \boldsymbol{q}_\tau, \boldsymbol{a}_\tau} P_{s', \boldsymbol{q}'}^{s_\tau, \boldsymbol{q}_\tau} \hat{R}_{s_\tau \boldsymbol{a}_\tau} \prod_j P_{a_\tau q_\tau}^{(j)} - \frac{1}{\hat{C}_{q_i}} \sum_{\tau=0}^{T_\beta - 1} \sum_{s_\tau, \boldsymbol{a}_\tau, \boldsymbol{q}_\tau} \hat{R}_{s_\tau \boldsymbol{a}_\tau} \alpha_*^{s_\tau, \boldsymbol{q}_\tau} \prod_j P_{a_\tau q_\tau}^{(j)} \Big] \Big|$$

$$\leq \lim_{T_\alpha, T_\beta \to \infty} \Big[\sum_{\tau=0}^{T_\beta - 1} \sum_{s_\tau, a_\tau} \frac{\hat{R}_{s_\tau a_\tau} P_{a_\tau q_\tau}^{(i)} T_\alpha}{\min(C_{q_i}, \hat{C}_{q_i})} \Big| \sum_{\substack{q_{j \neq i}, a_{j \neq i} \\ s, s', \boldsymbol{q}'}} \alpha_*^{s\boldsymbol{q}} P_{s'\boldsymbol{q}' a_i s\boldsymbol{q}}^{(i)} P_{s', \boldsymbol{q}'}^{s_\tau, \boldsymbol{q}_\tau} - \alpha_*^{s_\tau \boldsymbol{q}_\tau} \Big| \Big]$$

$$\leq \lim_{T_\alpha, T_\beta \to \infty} \frac{T_\alpha}{\min(C_{q_i}, \hat{C}_{q_i})} \sum_{\tau=0}^{T_\beta - 1} C_\epsilon \epsilon^\tau = \lim_{T_\alpha, T_\beta \to \infty} \frac{T_\alpha}{\min(C_{q_i}, \hat{C}_{q_i})} \frac{C_\epsilon}{1 - \epsilon} = 0\ . \quad (10)$$

The last equality follows because $\min(C_{q_i}, \hat{C}_{q_i})$ approaches infinity quadratically: omitting all nonessential notation, C_{q_i} contains a double sum over terms $\alpha_t^{s\boldsymbol{q}} \beta_\tau^{s'\boldsymbol{q}'}$, from $t = 0$ to $T_\alpha - 1$ and from $\tau = 0$ to $T_\beta - 1$. Since the FSCs are fully stochastic, for each $\boldsymbol{q}$ the marginal limit probability is nonzero and thus one state $(s, \boldsymbol{q})$ must have nonzero limit probability $\alpha_*^{s\boldsymbol{q}}$ (and probability close to the limit for an infinite number of terms), i.e. it is a recurrent state. By assumption (see theorem) one recurrent state must have nonzero reward; such states are visited an infinite number of times, thus the double sum grows faster than $T_\alpha T_\beta \cdot \mathrm{const}$ for some constant. $\hat{C}_{q_i}$ has similar terms and also grows quadratically.

The first inequality in (10) comes from exponential decrease of $D_{TV}(\alpha_t, \alpha_*)^2$. In the second inequality we bounded terms $P_{a_\tau q_\tau}^{(j)}$ by 1 for $j \neq i$. The third inequality follows from using (9) to upper bound the term $\left| \sum_{q_{j\neq i}, a_{j\neq i}, s, s', \boldsymbol{q}'} \alpha_*^{s\boldsymbol{q}} P_{s'\boldsymbol{q}' a_i s\boldsymbol{q}}^{(i)} P(s_\tau, \boldsymbol{q}_\tau | s_0 = s', \boldsymbol{q}_0 = \boldsymbol{q}') - \alpha_*^{s_\tau \boldsymbol{q}_\tau} \right|$. To apply (9), $\alpha_*^{s\boldsymbol{q}} P_{s'\boldsymbol{q}' a_i s\boldsymbol{q}}^{(i)} P(s_\tau, \boldsymbol{q}_\tau | s_0 = s', \boldsymbol{q}_0 = \boldsymbol{q}')$ must converge in the limit $\tau \to \infty$ to $\alpha_*^{s_\tau \boldsymbol{q}_\tau}$, we show this. Define $P_0(s', \boldsymbol{q}' | a_i, q_i) = \sum_{q_{j\neq i}, a_{j\neq i}, s} \alpha_*^{s\boldsymbol{q}} P_{s'\boldsymbol{q}' a_i s\boldsymbol{q}}^{(i)}$ and $P_\tau(s_\tau, \boldsymbol{q}_\tau | a_i, q_i) = \sum_{s', \boldsymbol{q}'} P(s_\tau, \boldsymbol{q}_\tau | s_0 = s', \boldsymbol{q}_0 = \boldsymbol{q}') P_0(s', \boldsymbol{q}' | a_i, q_i)$.

In a *unichain*, the starting distribution does not affect the limiting distribution. Hence, $\lim_{\tau\to\infty} P_\tau(s_\tau, \boldsymbol{q}_\tau | a_i, q_i) = \alpha_*^{s_\tau \boldsymbol{q}_\tau}$. In a *multichain* the limiting distribution depends on the starting distribution, however, in $\alpha_*^{s\boldsymbol{q}}$ and thus in $P_0(s', \boldsymbol{q}' | a_i, q_i)$, all transient Markov chain states have zero probability (easy to verify from the definition of a transient state) and the probability mass is distributed among closed irreducible classes in the exactly same proportion as in $\alpha_*^{s_\tau \boldsymbol{q}_\tau}$. Further forward projection of the Markov chain does not change this probability mass distribution (as the irreducible classes are closed), thus, similarly to the unichain case, the Markov chain starting from $P_0(s', \boldsymbol{q}' | a_i, q_i)$ converges to $\alpha_*^{s_\tau \boldsymbol{q}_\tau}$. Next, we show that $H_{aq}^{(i)}$ is zero.

We have $H_{aq}^{(i)} = \lim_{T_\alpha, T_\beta \to \infty} \frac{T_\alpha}{C_{q_i}} \sum_{V_{\neq i}} \hat{R}_{s\boldsymbol{a}} \prod_{j\neq i} P_{aq}^{(j)} \frac{1}{T_\alpha} \sum_{T=0}^{T_\alpha - 1} \alpha_T^{s\boldsymbol{q}} = \lim_{T_\alpha, T_\beta \to \infty} \frac{1}{C_{q_i}} T_\alpha \sum_{V_{\neq i}} \hat{R}_{s\boldsymbol{a}} \prod_{j\neq i} P_{aq}^{(j)} \alpha_*^{s,\boldsymbol{q}}$, because $\lim_{T_\alpha\to\infty} \frac{1}{T_\alpha} \sum_{T=0}^{T_\alpha - 1} \alpha_T^{s\boldsymbol{q}} = \lim_{T_\alpha\to\infty} \frac{1}{T_\alpha} T_\alpha \alpha_*^{s\boldsymbol{q}} = \alpha_*^{s\boldsymbol{q}}$. Because $\frac{T_\alpha}{C_{q_i}}$ becomes zero in the limit, by the same argument as $\frac{T_\alpha}{\min(C_{q_i}, \hat{C}_{q_i})}$ becomes zero in (10), and because other terms are finite in the limit, $H_{aq}^{(i)}$ is zero. Since $H_{aq}^{(i)}$ is zero and $J_{aq}^{(i)}$ converges to a constant, the probability update multiplies all action probabilities by the same constant; this concludes the proof and $\acute{P}_{aq}^{(i)} = P_{aq}^{(i)} \cdot \mathrm{const}$.

Theorem 1 may be surprising as the discounted reward EM methods [26,10,16] improve the policy in each EM iteration so that the discounted reward never decreases. Getting stuck is a consequence of the average reward setting, where the entire future must be fully taken into account. We next give a practical EM approach for average reward Dec-POMDPs that allows policy improvement.

[2] See http://users.ics.aalto.fi/jpajarin/avgrew/supplement.pdf for details.

4.2 A Practical EM Method

The average reward EM described above is always stuck in a local optimum. To force a change to FSC parameters in the M-step, one could try to use fixed instead of infinite horizons. Fixing a horizon induces an approximation error to parameter updates that decreases with a larger horizon. Discounted reward EM methods effectively fix both T_α and T_β to the same horizon by using discounted rewards. This has at least three drawbacks in average reward problems: 1) an initial belief is needed in optimization, 2) discounting rewards increases approximation error compared to uniform rewards, 3) limiting both T_α and T_β increases approximation error more than limiting only one of them.

We now give update rules with an infinite T_α, and propose to set only T_β to a fixed value which is doubled during optimization whenever the current policy value would decrease. This has several advantages. By not limiting T_α we do not need an initial belief in unichain problems and can compute the sum of alpha messages efficiently as detailed later in this Section. Furthermore, the approach allows to reduce the approximation error in parameter updates until the policy value increases. The adaptation of T_β is necessary not only because we know a priori that a too low T_β may not always yield increased value, but also because the approximation error that a specific T_β causes is problem dependent: the mixing rate of the Dec-POMDP determines how fast a distribution converges to the stationary distribution and this in turn determines how high the approximation error for a certain T_β is. In short, this kind of approach is necessary to adapt T_β to the specific Dec-POMDP problem.

Since $\lim_{T_\alpha \to \infty} \frac{1}{T_\alpha} \sum_{t=0}^{T_\alpha - 1} \alpha_t^{s\boldsymbol{q}} = \lim_{T_\alpha \to \infty} \alpha_{T_\alpha}^{s\boldsymbol{q}} = \alpha_*^{s\boldsymbol{q}}$, the **action probability update** is derived from (8) and becomes $\acute{P}_{aq}^{(i)} =$

$$\frac{P_{aq}^{(i)}}{C_{q_i}} \sum_{s, q_{j \neq i}} \alpha_*^{s\boldsymbol{q}} \sum_{a_{j \neq i}} \Big[\hat{R}_{s\boldsymbol{a}} \prod_{j \neq i} P_{aq}^{(j)} + \sum_{s', \boldsymbol{o}, \boldsymbol{q}'} P_{\boldsymbol{o}s's\boldsymbol{a}} P_{q'qo}^{(i)} \big(\prod_{j \neq i} P_{aq}^{(j)} P_{q'qo}^{(j)} \big) \sum_{\tau=0}^{T_\beta} \beta_\tau^{s'\boldsymbol{q}'} \Big] . \tag{11}$$

The **transition probability update** is derived similarly to the action probability update resulting in

$$\acute{P}_{q'qo}^{(i)} = \frac{P_{q'qo}^{(i)}}{C_{oq}^{(i)}} \sum_{s, q_{j \neq i}, \boldsymbol{a}, s', o_{j \neq i}, q'_{j \neq i}} \Big[\alpha_*^{s\boldsymbol{q}} P_{\boldsymbol{o}s's\boldsymbol{a}} P_{aq}^{(i)} \prod_{j \neq i} \Big(P_{aq}^{(j)} P_{q'qo}^{(j)} \Big) \sum_{\tau=0}^{T_\beta} \beta_\tau^{s'\boldsymbol{q}'} \Big] . \tag{12}$$

We propose the practical EM algorithm as follows: set T_β to an initial value (we use 32), then apply E- and M-steps in turn until the policy value does not increase or until any other stopping criterion is satisfied.

In the E-step the algorithm computes beta messages up to the horizon T_β using (4) and the limiting distribution $\alpha_*^{s\boldsymbol{q}}$ for alpha messages either projecting until convergence using (3) or by solving a system of linear equations. Because the EM algorithm gradually improves the policy, the limiting distribution from the previous EM iteration is likely close to the new limiting distribution. An efficient unichain implementation thus starts projecting from the limiting distribution of the previous EM iteration (in multichain problems projecting must start from the initial belief). This saves much computation: for example, in the “long fire fighting” experiment, iteration 1 needed 5000 projections, next iterations only 3-100 projections.

In the M-step the algorithm computes new FSC action and transition probabilities by (11) and (12). *After the M-step* the algorithm checks whether the value of a policy decreased: if it did, the algorithm multiplies T_β by 2 and recomputes the beta messages and performs the M-step again, until the value does not decrease (for $T_\beta \to \infty$ this would yield the original EM we derived; we limit T_β to a maximum of 32768). In the experiments, T_β needed duplication only rarely.

Our practical EM is better than naive bounding/discounting both alpha and beta. We efficiently compute exact infinite-horizon alphas, using the limiting distribution from the previous iteration as the start of propagation, whereas discounted EM would need to choose a discount factor and propagate alpha to large horizons. Our EM is intuitive and easy to implement.

5 Experiments

We evaluate the average reward on two different sets of benchmark problems. The first set consists of benchmark problems, used previously for evaluating discounted reward Dec-POMDP methods [10,3,2,16]. The second set consists of two new average reward benchmark problems, which emphasize long-term effects of actions.

For all problems, we compare the new expectation maximization (EM) average reward DEC-POMDP method (denoted “AvgEM”) of Section 4.2, against a baseline and loose upper bounds of performance. We use a uniformly random policy as baseline. For (loose) upper bounds we compute the optimal solution to the average reward MDP underlying the DEC-POMDP with linear programming [21]; this upper bound corresponds to agents that have full knowledge of the environment and each other. We also show AvgEM outperforms an alternative new non-linear programming approach (denoted “AvgNLP”) which we introduce in the Appendix. We compare AvgEM with a state of the art discounted reward EM (denoted “DiscEM”) method [10] on different discount factors 0.9, 0.99, and 0.999; we show that AvgEM outperforms DiscEM in average reward problems and has equal or better performance in benchmark problems from the discounted reward literature. Optimization of a controller using the EM methods, optimization of the random baseline, and optimization of the MDP upper bound were run in Matlab on a single processor core. Methods were stopped if the change in the policy value between iterations was under a small threshold. EM methods had a time limit of one hour. Non-linear programs were solved with the SNOPT solver on the publicly available NEOS server.

Benchmark problems from the discounted reward literature. The first six problems in Table 1 (denoted “Disc. Prob.“) have been used to evaluate discounted reward methods [17], but as we evaluate methods by average reward, the earlier evaluations based on discounted reward are not directly comparable. The problems are: DecTiger (2,3,2), Recycling robots (4,3,2), 2x2 Grid meeting (16,5,2), Wireless network (64,2,6), Box pushing (100,4,5), and Mars rovers (256,6,8), where for each problem we list (number of states, number of actions, number of observations). For each problem AvgEM, AvgNLP, and DiscEM optimized different size FSCs in parallel over 10 random FSC initializations. Table 1 shows also the average reward for the random policy and for the MDP upper bound. AvgEM performs well, in “recycling robots” it is even close to the full-knowledge upper bound. AvgEM outperforms AvgNLP and performs as well as “DiscEM 0.9”. “DiscEM 0.9” outperforms “DiscEM 0.99” and “DiscEM 0.999” demonstrating that, in these problems, good results are already obtained with a small discount factor. Next, we will discuss two new average reward problems with long-term effects of actions.

Wireless network with overhead ($|S| = 64$, $|A_i| = 2$, $|O_i| = 6$). In the wireless networking problem, [16] two wireless agents try to keep their transmit buffers, modeled with four states, as empty as possible. Each buffer gets data from a two-state source model. Buffer fullness is modeled as few states at rough intervals; insertions/transmissions have a probability to change the buffer state. If both agents transmit simultaneously both transmissions fail and data is not removed from the buffers. The world state is the cross product of the transmit buffers and source models, in total 64 states. In [16] the objective corresponded to minimizing delay. In the new problem, successful transmissions are rewarded, corresponding to maximizing throughput. In real wireless networks, decisions are made at 10 microsecond intervals; to reflect this, we multiplied the probability to transition from one buffer state to another and the probability to insert data into a buffer with 0.01. As overhead from packet headers etc. is proportionally smaller for larger packets, the new wireless problem allows transmission of more data, when the buffer is fuller: for buffer size x, $y = 2x/(x + 1)$ data units are transmitted (probability to change buffer state is proportional to y).

Long fire fight ($|S| = 27$, $|A_i| = 2$, $|O_i| = 2$). In the fire fighting problem [15] two robots try to extinguish three houses and receive negative reward for higher house fire levels (see [15] for details). In the new *long fire fighting* version a house can also start burning on its own with probability 0.1. To make a single Dec-POMDP time step correspond to a shorter time in the real application, we multiplied all transition probabilities between fire levels with 0.01. In this version a fire takes longer to put out, and it takes longer for fire levels to increase.

Table 1 shows results for the wireless network with overhead (denoted “Long Wirel.“) and the long fire fight (denoted “Long FF”) problems (FSC size was fixed to 3). In both problems AvgEM converged rapidly and got highest average reward. Figure 1 shows convergence of the EM methods. Results for the discounted method DiscEM agree with the observations in Section 4.2 about the negative effect of discounting alpha and beta messages. DiscEM converges with a low discount factor 0.9 to suboptimal solutions and with a large 0.999 factor too slowly. Interestingly, in fire fighting “DiscEM 0.9” convergences to a bad local optimum where both agents only try to extinguish the middle house, showing the necessity of adapting optimization parameters to the specific Dec-POMDP

Table 1. Expected average reward of a uniformly random policy ("Random"), a MDP based upper bound ("MDP"), the average reward nonlinear programming method ("AvgNLP"), the discounted expectation maximization method ("DiscEM") for discount factors 0.9, 0.99, and 0.999, and the average reward expectation maximization method ("AvgEM") in benchmark problems used in discounted method research [10,3,2,16] ("Disc. Prob.") and in new average reward benchmarks ("Avg. Prob."). A result is bolded, when the 95% confidence interval of the best result contains the result or vice versa. AvgEM outperforms AvgNLP, performs as well or better as DiscEM in discounted reward problems, and outperforms DiscEM in the average reward problems.

Disc. Prob.	Random	MDP	AvgNLP	DiscEM 0.9	DiscEM 0.99	DiscEM 0.999	AvgEM
DecTiger	−46.22	20.00	**−2.00**	**−1.375**	**−1.80**	−2.19	**−1.79**
Rec. robots	0.45	3.27	1.24	**3.08**	**3.08**	2.59	**3.08**
2x2 Grid	0.25	1.00	0.28	0.80	**0.83**	0.56	0.75
Wireless	−3.04	−1.46	−3.00	**−1.96**	−2.07	−2.86	−2.05
Box pushing	−1.37	20.35	−0.19	3.69	3.45	0.28	**3.75**
Mars rovers	−1.21	2.88	1.05	**1.77**	0.80	−0.315	**1.55**
Avg. Prob.	Random	MDP	AvgNLP	DiscEM 0.9	DiscEM 0.99	DiscEM 0.999	AvgEM
Long Wirel.	0.0063	0.0099	**0.0089**	0.0081	0.0085	0.0066	**0.0093**
Long FF	−1.85	−0.20	−3.00	−4.00	−1.095	−1.44	**−0.91**

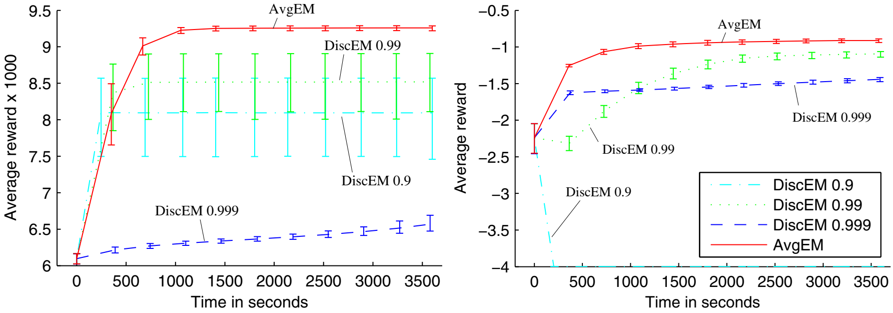

Fig. 1. Expected average reward of discounted reward EM (DiscEM) with 3 discount factors and our average reward EM method (AvgEM), for "wireless network with overhead" (left) and "long fire fighting" (right). Error bars are 95% confidence intervals from bootstrapping.

problem. In fact, for most EM iterations AvgEM held parameter T_β (see Section 4.2) between 32 and 512 in "wireless network with overhead" and at 32 in "long fire fight".

6 Conclusions

Average reward is a useful criterion for planning under uncertainty with multiple agents; it has real-life importance in wireless networks and other domains.

We showed that traditional expectation maximization is stuck in average reward Dec-POMDPs (and POMDPs) under certain conditions and provided a new EM based method for average reward Dec-POMDPs. Our new EM method yields good performance, outperforming a state of the art discounted reward EM method in average reward problems. We also introduced two average reward benchmark problems, long fire fighting and wireless network with overhead. **To our knowledge this is the first general Dec-POMDP method for optimizing average reward.**

Acknowledgements. The work was supported by the Academy of Finland, decisions 252845, 271394, and 251170 (Finnish CoE in Computational Inference Research COIN). The work was also supported by Nokia. J. Peltonen belongs to COIN and Helsinki Institute for Information Technology HIIT. J. Pajarinen carried out most of the work at the Department of Information and Computer Science, Aalto University.

References

1. Aberdeen, D.: Policy-gradient algorithms for partially observable Markov decision processes. Ph.D. thesis, Australian National University (2003)
2. Amato, C., Bernstein, D.S., Zilberstein, S.: Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. Autonomous Agents and Multi-Agent Systems 21(3), 293–320 (2010)
3. Amato, C., Bonet, B., Zilberstein, S.: Finite-state controllers based on Mealy machines for centralized and decentralized POMDPs. In: AAAI, pp. 1052–1058. AAAI Press (2010)
4. Bernstein, D.S., Amato, C., Hansen, E.A., Zilberstein, S.: Policy iteration for decentralized control of Markov decision processes. Journal of Artificial Intelligence Research 34(1), 89–132 (2009)
5. Bernstein, D.S., Hansen, E.A., Zilberstein, S.: Bounded policy iteration for decentralized POMDPs. In: IJCAI, pp. 1287–1292. IJCAI (2005)
6. Bernstein, D., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of Markov decision processes. Mathematics of Operations Research, 819–840 (2002)
7. Bianchi, G., Fratta, L., Oliveri, M.: Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs. In: PIMRC, vol. 2, pp. 392–396. IEEE (1996)
8. Ji, S., Parr, R., Li, H., Liao, X., Carin, L.: Point-based policy iteration. In: AAAI, vol. 22, pp. 1243–1249. AAAI Press (2007)
9. Kakade, S.: Optimizing average reward using discounted rewards. In: Helmbold, D.P., Williamson, B. (eds.) COLT 2001 and EuroCOLT 2001. LNCS (LNAI), vol. 2111, pp. 605–615. Springer, Heidelberg (2001)
10. Kumar, A., Zilberstein, S.: Anytime planning for decentralized POMDPs using Expectation Maximization. In: UAI, pp. 294–301. AUAI Press (2010)
11. Levin, D., Peres, Y., Wilmer, E.: Markov chains and mixing times. American Mathematical Society (2009)

12. Li, Y., Yin, B., Xi, H.: Finding optimal memoryless policies of POMDPs under the expected average reward criterion. European Journal of Operational Research 211(3), 556–567 (2011)
13. Mahadevan, S.: Average reward reinforcement learning: Foundations, algorithms, and empirical results. Machine Learning 22(1), 159–195 (1996)
14. Oliehoek, F.: Value-Based Planning for Teams of Agents in Stochastic Partially Observable Environments. Ph.D. thesis, Informatics Institute, University of Amsterdam (February 2010)
15. Oliehoek, F., Spaan, M., Whiteson, S., Vlassis, N.: Exploiting locality of interaction in factored DEC-POMDPs. In: AAMAS, vol. 1, pp. 517–524. IFAAMAS (2008)
16. Pajarinen, J., Peltonen, J.: Efficient planning for factored infinite-horizon DEC-POMDPs. In: IJCAI, pp. 325–331. AAAI Press (2011)
17. Pajarinen, J., Peltonen, J.: Periodic finite state controllers for efficient POMDP and DEC-POMDP planning. In: NIPS, pp. 2636–2644 (2011)
18. Pajarinen, J., Hottinen, A., Peltonen, J.: Optimizing spatial and temporal reuse in wireless networks by decentralized partially observable Markov decision processes. IEEE Transactions on Mobile Computing (2013) (preprint)
19. Petrik, M., Zilberstein, S.: Average reward decentralized Markov decision processes. In: IJCAI, pp. 1997–2002 (2007)
20. Poupart, P., Boutilier, C.: Bounded finite state controllers. In: NIPS, pp. 823–830. MIT Press (2004)
21. Puterman, M.L.: Markov decision processes: discrete stochastic dynamic programming. Wiley (2005)
22. Seuken, S., Zilberstein, S.: Formal models and algorithms for decentralized decision making under uncertainty. Autonomous Agents and Multi-Agent Systems 17(2), 190–250 (2008)
23. Spaan, M., Oliehoek, F., Amato, C.: Scaling up optimal heuristic search in DEC-POMDPs via incremental expansion. In: IJCAI. AAAI Press (2011)
24. Szer, D., Charpillet, F.: An optimal best-first search algorithm for solving infinite horizon DEC-POMDPs. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 389–399. Springer, Heidelberg (2005)
25. Tangamchit, P., Dolan, J., Khosla, P.: The necessity of average rewards in cooperative multirobot learning. In: ICRA, vol. 2, pp. 1296–1301. IEEE (2002)
26. Toussaint, M., Harmeling, S., Storkey, A.: Probabilistic inference for solving (PO)MDPs. Tech. rep., University of Edinburgh (2006)
27. Toussaint, M., Storkey, A.: Probabilistic inference for solving discrete and continuous state Markov decision processes. In: ICML, pp. 945–952. ACM (2006)
28. Yagan, D., Tham, C.: Coordinated reinforcement learning for decentralized optimal control. In: ADPRL, pp. 296–302. IEEE (2007)
29. Yu, H., Bertsekas, D.P.: Discretized approximations for POMDP with average cost. In: UAI, pp. 619–627. AUAI Press (2004)

Appendix: Non-linear Programming for Average Reward Dec-POMDPs

A non-linear programming (NLP) approach has been used in recent discounted reward POMDP and Dec-POMDP research [2]. To study whether a NLP approach is suitable for average reward cases, we introduce a new NLP based method as an alternative to the expectation-maximization approach that we recommend. We do not claim that the method below is the only possible NLP approach to average reward Dec-POMDPs, but to our knowledge no other NLP methods for average reward Dec-POMDPs have been presented so far, therefore we use our method below as a first proxy.

Motivated by the linear programming solution for average reward MDPs [21] we use the same basic idea that the limiting distribution remains the same over successive time steps. Note that the discounted reward NLP approach in [2] uses the Bellman equation to recursively define the optimal value function over world and FSC states, but the approach requires a discount factor and is not directly applicable to average reward problems. Instead we use the limiting distribution as the basis for optimization.

Table 2. Non-linear program for an aperiodic unichain average reward Dec-POMDP. The program maximizes the immediate reward of the limiting distribution $P_*(s, \boldsymbol{q})$, which corresponds to maximizing the average reward. The program solves for the FSC parameters $P^{(i)}_{q'qo}$ and $P^{(i)}_{aq}$ of each agent i.

Variables: $P_*(s, \boldsymbol{q})$ and for each agent i: $P^{(i)}_{q'qo}$, $P^{(i)}_{aq}$

Optimization goal: Maximize $\sum_{s,\boldsymbol{a}} R_{s\boldsymbol{a}} \sum_{\boldsymbol{q}} P_*(s, \boldsymbol{q}) \prod_i P^{(i)}_{aq}$

Subject to the following **constraints**:

$P_*(s', \boldsymbol{q}') - \sum_{s,\boldsymbol{q}} P_{s'\boldsymbol{q}'s\boldsymbol{q}} P_*(s, \boldsymbol{q}) = 0\,,\ \sum_{s,\boldsymbol{q}} P_*(s, \boldsymbol{q}) = 1\ , \qquad P_*(s, \boldsymbol{q}) \geq 0\ \forall s\ \forall \boldsymbol{q}$

$\sum_{a_i} P^{(i)}_{aq} = 1\ \forall q_i\ ,\ P^{(i)}_{aq} \geq 0\ \forall q_i\ \forall a_i\ ,\ \ \sum_{q_i{}'} P^{(i)}_{q'qo} = 1\ \forall q_i\ \forall o_i,\ P^{(i)}_{q'qo} \geq 0\ \forall q_i\ \forall o_i\ \forall q_i'$

Table 2 shows the non-linear program for solving aperiodic unichain average reward Dec-POMDPs. In Table 2 we have kept the notation for probability distributions used throughout the paper, one may use functions instead of distributions for notational purposes. We now discuss the program from top to bottom. **Variables:** The limiting distribution $P_*(s, \boldsymbol{q})$ and the FSC parameters of each agent i, $P^{(i)}_{q'qo}$ and $P^{(i)}_{aq}$, are the variables to solve for. **Optimization goal:** The optimization goal of the non-linear program is to maximize the average reward $\sum_{s,\boldsymbol{a}} R_{s\boldsymbol{a}} \sum_{\boldsymbol{q}} P_*(s, \boldsymbol{q}) \prod_i P^{(i)}_{aq}$. **First constraint:** The first constraint $P_*(s', \boldsymbol{q}') - \sum_{s,\boldsymbol{q}} P_{s'\boldsymbol{q}'s\boldsymbol{q}} P_*(s, \boldsymbol{q}) = 0$ forces $P_*(s, \boldsymbol{q})$ to be a limiting distribution. **Other constraints:** The remaining constraints force the probability distributions to be positive and to sum to one. In the experiments non-linear programs were solved with the SNOPT solver on the publicly available NEOS server.