Exploiting Anonymity and Homogeneity in Factored Dec-MDPs through Precomputed Binomial Distributions

Rajiv Ranjan Kumar and Pradeep Varakantham
School of Information Systems, Singapore Management University
{rajivk,pradeepv}@smu.edu.sg

ABSTRACT

Recent work in decentralized stochastic planning for cooperative agents has focussed on exploiting homogeneity of agents and anonymity in interactions to solve problems with large numbers of agents. Due to a linear optimization formulation that computes joint policy and an objective that indirectly approximates joint expected reward with reward for expected number of agents in all state, action pairs, these approaches have ensured improved scalability. Such an objective closely approximates joint expected reward when there are many agents, due to law of large numbers. However, the performance deteriorates in problems with fewer agents. In this paper, we improve on the previous line of work by providing a linear optimization formulation that employs a more direct approximation of joint expected reward. The new approximation is based on offline computation of binomial distributions. Our new technique is not only able to improve quality performance on problems with large numbers of agents, but is able to perform on par with existing best approaches on problems with fewer agents. This is achieved without sacrificing on scalability/run-time performance of previous work.

Keywords

Multi-agent Systems, Dec-MDPs, Anonymity

1. INTRODUCTION

The worst case complexity of multi-agent planning under uncertainty increases exponentially with increase in number of agents [2]. Recent work in decentralised stochastic planning [15, 17] has highlighted the advantage of exploiting homogeneity[1] in agent models and anonymity in interactions to improve the scalability with respect to number of agents. Interaction anonymity indicates that interaction outcomes between agents are dependent on number of agents involved and not on actual identity of agents. For instance, in the navigation domains [6, 19], outcomes of collisions in narrow corridors are dependent only on the number of agents entering the narrow corridor simultaneously and not on the specific agents. Similarly, in the coordination problem introduced by [22] for Autonomous Underwater and Surface Vehicles (AUVs and ASVs), the value of coordinating in order to obtain underwater samples is dependent on the number of agents sampling simultaneously and not on which specific agents are sampling. In fact, most sensor network problems [5, 8] have coordination problems with anonymous interactions, where the confidence in tracking a target is dependent on the number of sensors and not on which specific sensors are tracking that target. Finally, in the context of coordinating defenders in security patrolling problems [14], the security provided for a potential location is typically dependent on the number of defender teams patrolling that location.

While homogeneity and anonymity have broad applicability in multi-agent systems, we specifically focus on multi-agent planning problems represented using the rich framework of Decentralized Markov Decision Process (Dec-MDP) [2]. Dec-MDP provides a framework to represent decentralized decision-making problems under transition uncertainty. However, solving a Dec-MDP to generate coordinated yet decentralized policies in environments with uncertainty is NEXP-Hard [2]. Researchers have typically employed three types of approaches to address this significant computational complexity: (1) approximate dynamic programming and policy iteration approaches [9, 10, 13, 3]; (2) exploit static and dynamic sparsity in interactions [1, 8, 20, 21, 7]; (3) exploit homogeneity in agent models and aggregate influence in interactions [17, 21, 7, 19].

We focus on the third category of approximation approaches, specifically exploiting anonymity in interactions and homogeneity in agent models. In Dec-MDPs, interaction anonymity refers to joint rewards and transitions being dependent on number of agents in certain state, action pairs and not on specific agents in certain state, action pairs. Specifically we extend on the work by Varakantham *et al.* [17]. Intuitively, Varakantham *et al.* achieve scalability with respect to number of agents due to three key factors:

- An updated factored Dec-MDP model called D-SPAIT that considers homogenous and anonymous reward and transition models; Complex reward and transition functions are approximated using piecewise constant or piecewise linear functions.
- Linear/quadratic optimization formulations for computing joint policies in decentralized MDPs; and finally

[1]Homogeneity refers to agents having the same decision models, i.e., state, action, transition and reward functions in domains with transition uncertainty.

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

- Optimizing joint reward for expected number of agents (instead of optimizing joint expected reward for all agents) in all state, action pairs. We refer to the optimization of joint reward for expected number of agents as the Expected Agent or EA objective. Due to law of large numbers, EA objective was shown to provide good performance on problems with large number of agents.

However, Varakantham *et al.*'s approach has two aspects that can be improved: (1) Reward functions are approximated using piecewise constant or piecewise linear components for linear or quadratic optimization to be applicable. (2) On problems with few agents, quality of the joint policies can deteriorate significantly.
Towards addressing these issues, we make the following key contributions in this paper:

1. We first define the expected joint reward objective, referred to as ER in the context of anonymous and homogenous interactions (reward and transition functions) in D-SPAIT. The key advantage of this objective is that reward and transition functions (however complex) do not have to be approximated.
2. We then provide a new approach that directly approximates the ER objective by using pre-computed binomial distributions. Specifically, pre-computed binomial distributions are employed to obtain an estimate on the probability that certain number of agents are in a certain state, taking a certain action.
3. We also show how this approach can be trivially extended to handle problems with multiple agent types and joint transition interactions.
4. Finally, we also provide detailed experimental results that demonstrate that the new approach is not only able to outperform Varakantham *et al.*'s algorithm, but also is able to perform on par or better than best known existing algorithms for benchmark problems in TI-Dec-MDPs.

2. BACKGROUND

In this section, since the approaches employed are based on dual formulation for solving MDPs, we first provide a brief background of the Markov Decision Process (MDP) model and the dual formulation for solving MDPs. We then describe the underlying model for decentralized stochastic planning that represents homogeneity and anonymity and the approach that exploits homogeneity and anonymity in interactions. Finally, we provide a brief description of binomial distributions.

2.1 MDPs

Markov Decision Processes (MDPs) [11] are used to represent decision problems in the face of transitional uncertainty. An MDP is defined using the tuple: $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, H, \alpha \rangle$. $\mathcal{S}$ represents the set of states, $\mathcal{A}$ represents the set of actions or decisions that can be taken in the states, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ represents the transition function and is a probability distribution over the destination states. $\mathcal{T}(s, a, s')$ is the probability that the state transitions from s to s' on taking action a and $\sum_{s'} \mathcal{T}(s, a, s') = 1, \forall s, a$. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ represents the reward matrix. $\mathcal{R}(s, a)$ is the reward obtained by taking action a in state s. α^0 represents the initial state distribution and hence for all $t > 0$, $\alpha^t(s) = 0$.

The goal is to obtain a policy, $\pi : \mathcal{S} \times H \to \Delta(\mathcal{A})$ such that the expected value, $\mathcal{V}^0(\alpha)$ is maximized over a given time horizon, H and a starting belief (i.e., probability distribution) of α over states. $\Delta(\mathcal{A})$ denotes a probability distribution over the set of actions $\mathcal{A}$ and more concretely, $\pi^t(s, a)$ gives the probability of taking action a in state s at time t and $\sum_a \pi^t(s, a) = 1$. $\mathcal{V}^0$ is defined as follows:

$$\mathcal{V}^0(\alpha) = \max_{\pi} \sum_{s \in \mathcal{S}} \mathcal{V}^0(s, \pi) \cdot \alpha^0(s), \text{ where}$$

$$\mathcal{V}^t(s, \pi) = \begin{cases} \sum_a \pi^t(s, a) \cdot \mathcal{R}(s, a) & \textbf{if} \quad t = H - 1 \\ \sum_a \pi^t(s, a) \cdot (\mathcal{R}(s, a) & \\ + \sum_{s'} \mathcal{T}(s, a, s') \cdot \mathcal{V}^{t+1}(s', \pi)) & \textbf{otherwise} \end{cases}$$

The optimal policy in the case of an MDP is a deterministic one. That is to say, for all time steps t and for all states s there exists one action, a such that $\pi^t(s, a) = 1$.

The dual formulation (primal formulation employs value function variables) for solving an MDP is given by:

$$\max_x \sum_{t,s,a} \mathcal{R}(s, a) \cdot x^t(s, a) \tag{1}$$

$$\text{s.t.} \sum_a x^{t+1}(s', a) - \sum_{s,a} x^t(s, a) \cdot \mathcal{T}(s, a, s') = \alpha^t(s), \qquad \forall s' \tag{2}$$

$$x^t(s, a) \geq 0, \qquad \forall s, a, t;$$

where $x^t(s, a)$ represents the number of times action a has been chosen at time t in state s and therefore, objective refers to expected value. The main set of constraints (Equation 2) ensures that flow out of a state s' is equal to flow coming into s'. Note that $\alpha^t(s) = 0$ for all $t > 0$ and s and α^0 is the starting belief distribution over states. The agent policy is obtained by normalizing $\{x^t(s, a)\}$, i.e., :

$$\pi^t(s, a) = \frac{x^t(s, a)}{\sum_{a'} x^t(s, a')}, \forall t, s$$

2.2 Model

We now describe the Decentralized Stochastic Planning with Anonymous InteracTions (D-SPAIT) model introduced by Varakantham *et al.* [17]. D-SPAIT combines the cooperative stochastic planning framework of factored Decentralized MDP (DEC-MDP) [1] with homogeneity and interaction anonymity from competitive game theory [12, 18]. Intuitively D-SPAIT can be viewed as representing a class of problems that is more general than the transition independent (or reward independent) Dec-MDP model and less general than the Dec-MDP model. However, D-SPAIT assumes that joint reward (or joint transition) can be represented as a sum (or product) of individual agent rewards (or individual agent transitions) that are each dependent on number of agents in relevant state, action pairs.

Similar to Dec-MDP, D-SPAIT also assumes full joint observability of the system state. For ease of understanding and to reduce the clutter in notation, we first define the model assuming all agents are of the same type, i.e., they are homogeneous and provide the key ideas of the approach corresponding to the simpler model. We then provide model and approach for the general case. Here is the tuple for a single type of agents:

$$\left\langle \mathcal{P}, \mathcal{S}, \mathcal{A}, I^R, I^{\varphi}, R, \varphi, (\alpha_i)_{i \in \mathcal{P}} \right\rangle$$

- $\mathcal{P}$ is the agent population. $\mathcal{S}$ and $\mathcal{A}$ represent the state and action sets respectively for any individual agent in $\mathcal{P}$. Given a single type of agents, states for any agent are from the same set $\mathcal{S}$ and therefore joint state space is $\mathcal{S}^{|\mathcal{P}|}$. Each agent is able to fully observe its local state and the joint state space is factored over the individual agent state space. We have the same situation with the action set, $\mathcal{A}$.
- $I^R(s_i, a_i)$ is the set of state, action pairs that have a reward interaction with an agent in state $s_i(\in \mathcal{S})$ executing action $a_i(\in \mathcal{A})$, i.e.,

  $$I^R(s_i, a_i) = \{(s_i', a_i'), (s_i'', a_i''), \cdots\}$$

  In essence, an agent in (s_i, a_i) will have reward interactions with other agents in state, action pairs belonging to the set $I^R(s_i, a_i)$.
- $I^{\varphi}(s_i, a_i)$ is the set of state, action pairs that have a transition interaction with an agent in s_i and executing a_i.
- R is the individual reward function that is dependent on the number of agents involved in reward interaction with the agent. Specifically, $R(s_i, a_i, d_{s_i,a_i})$ is the reward obtained in state s_i by taking action a_i when there are d_{s_i,a_i} agents in state, action pairs that have reward interactions with (s_i, a_i). Reward function is defined for all values of d_{s_i,a_i} $(0 \leq d_{s_i,a_i} \leq |\mathcal{P}|)$. If there are no reward interactions for a state, action pair (s_i, a_i):

  $$R(s_i, a_i, d_{s_i,a_i}) = R(s_i, a_i), \forall d_{s_i,a_i} \tag{3}$$

  Also, given a set of constants, $\{d_{s_i,a_i}\}_{i\in\mathcal{P}}$, the joint reward for all agents at a time step is expressed as the sum of individual rewards: $\sum_{i\in\mathcal{P}} R(s_i, a_i, d_{s_i,a_i})$. Note that this is not equivalent to complete reward independence, as there is dependence on numbers of other agents.
- φ is the individual transition function that is dependent on number of agents involved in transition interaction with the agent. Specifically, $\varphi(s_i, a_i, s_i', d_{s_i,a_i})$ is the probability of transitioning from state s_i to s_i' on taking action a_i when there are d_{s_i,a_i} agents in state, action pairs that have transition interactions with (s_i, a_i). Transition function is defined for all values of d_{s_i,a_i} $(0 \leq d_{s_i,a_i} \leq |\mathcal{P}|)$. If there are no transition interactions for a state, action pair (s_i, a_i):

  $$\varphi(s_i, a_i, d_{s_i,a_i}) = \varphi(s_i, a_i), \forall d_{s_i,a_i} \tag{4}$$

  Also, given a set of constants, $\{d_{s_i,a_i}\}_{i\in\mathcal{P}}$, the joint transition for all agents at a time step is expressed as the product of individual transition: $\prod_{i\in\mathcal{P}} \varphi(s_i, a_i, d_{s_i,a_i})$. Note that this is not equivalent to complete transition independence, as there is dependence on numbers of other agents.
- α represents the initial belief distribution for any agent in $\mathcal{P}$.

The goal with Expected Agents (EA) objective is to find a joint policy, $\pi = (\pi_1, \ldots, \pi_n)$ (with one policy for each agent) over the given time horizon, H that maximizes the following value function

$$V^0(\pi) = \sum_{s_i,a_i,t,i} R\Big(s_i, a_i, \sum_{k,(s_k,a_k)\in I^R_{s_i,a_i}} x_k^t(s_k, a_k)\Big) \cdot x_i^t(s_i, a_i) \tag{5}$$

where $x_i^t(s_i, a_i)$ is the expected number of times action a_i is executed in state s_i for agent i at time t given individual agent policies π_i for all agents in the joint policy π. It should be noted that this objective is an extension of the objective employed in the dual formulation MDP of Equation 1. Specifically, they extend the objective to sum over multiple agents and account for the condition that reward for individual agents is dependent on number of other agents. The third argument for reward function is expected number of agents and hence represents the EA objective. Due to linearity of expectation, expected number of agents in a state, action pair is the sum of relevant (in this case, all agent, state, action tuples which have reward interactions with s_i, a_i) $\mathbf{x}$ values.

2.3 Approach

Varakantham *et al.* [17] provide scalable optimization models to solve D-SPAIT problems with different kinds of reward and transition functions. Specifically, they consider a modification of the dual formulation employed for solving MDPs. One of the main results of their paper shows that all agents of a type can have the same policy, i.e.,$x_i^t(s, a) = x_j^t(s, a), \forall s \in \mathcal{S}$ and $a \in \mathcal{A}$. The formulation for solving D-SPAIT problems with a single type that exploits this result is provided in Algorithm 1. There is a difference in both objective and constraints from the dual formulation for solving MDPs to account for multiple agents.

Algorithm 1: SolveDSPAIT-H-EA()

$$\max_{\mathbf{x}} \sum_{s_i,a_i,t} |\mathcal{P}| \cdot R\Big(s_i, a_i, |\mathcal{P}| \cdot \sum_{(s_i',a_i')\in I^R_{s_i,a_i}} x^t(s_i', a_i')\Big) \cdot x^t(s_i, a_i) \qquad \textbf{s.t.}$$

$$\sum_{a_i} x^t(s_i, a_i) - \sum_{s_i',a_i} x^{t-1}(s_i', a_i) \cdot \cdot \varphi(s_i', a_i, s_i, |\mathcal{P}| \cdot \sum_{(s_i',a_i')\in I^{\varphi}_{s_i',a_i}} x^t(s_i', a_i')) = \alpha^t(s_i), \forall s_i, t$$

$$x^t(s_i, a_i) \in [0, 1] \qquad \forall t, s_i, a_i$$

Since $x_i^t(s_i, a_i)$ will be the same for all agents i, we use $x^t(s_i, a_i)$ (instead of $x_i^t(s_i, a_i)$), a key difference from the objective definition in Equation 5. Furthermore, expected number of agents (due to linearity of expectation and same policy for all agents) in relevant state, action pairs now becomes $|\mathcal{P}| \cdot \sum_{(s_i',a_i')\in I^R_{s_i,a_i}} x^t(s_i', a_i')$.

Previously, scalable linear or convex formulations with binary variables were provided to solve the optimization model of Algorithm 1 for joint reward functions that are linear, Piecewise Constant (PWC) or Piecewise Linear and Convex (PWLC). Furthermore, they also provided a linear formulation with binary variables when there is a joint piecewise constant transition function. Given generality of PWC functions in approximating general functions, they were able to represent general joint reward/transition functions. We refer to the best performing formulation by Varakantham *et al.* as PWLD.

2.4 Binomial Distribution

Binomial distribution is a discrete probability distribution that is typically used to model number of successes in n independent yes/no experiments, where the probability of success in an experiment (i.e., getting a yes) is given by α. The probability of getting exactly k successes in n trials is given by the probability mass function:

$$f(k;n,\alpha) = \binom{n}{k} \cdot \alpha^k \cdot (1-\alpha)^{n-k} \quad (6)$$

$$\text{where } \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

In this paper, we employ binomial distribution to represent the probability that k agents will be in a certain state, action pair given that we have probability for an agent to be in that state, action pair (i.e., $x^t(s_i, a_i)$ in finite horizon Dec-MDPs). As will be explained later, such probabilities are useful in the computation of expected reward as we consider anonymous interactions, i.e., reward and transition functions are dependent on numbers of agents.

3. UPDATED D-SPAIT

The only change considered here is with respect to the objective. The EA (Expected Agent) objective employed in D-SPAIT (Equation 5) is different from the objective employed in Dec-MDPs. Here, we provide an objective that better approximates the Dec-MDP objective by calculating expected reward (rather than the reward for expected number of agents employed in D-SPAIT). We refer to the following as the ER (Expected Reward) objective.

$$V^0(\pi) = \sum_{s_i,a_i,t,i} x_i^t(s_i,a_i) \cdot \sum_{d \le |\mathcal{P}|-1} Pr^R_{\mathbf{x}^t_{\mathcal{P}\setminus\{i\}}}(d,s_i,a_i) \cdot R(s_i,a_i,d+1) \quad (7)$$

$$= \sum_{i,s_i,a_i,t,d \le |\mathcal{P}|} Pr^R_{\mathbf{x}^t_{\mathcal{P}}}(d,s_i,a_i) \cdot R(s_i,a_i,d) \quad (8)$$

where $Pr^R_{\mathbf{x}^t_{\mathcal{P}\setminus\{i\}}}$ is the probability of d agents in the set $\mathcal{P}\setminus i$ being in state, action pairs where there are reward interaction with s_i, a_i at time t. Inner sum over d (in Equation 7) calculates the expected reward over different possible numbers of other agents (excluding i) in each state, action pair given that agent i will be in that state, action pair. Outer sum considers the probability of agent i being in that state, action pair. Equation 8 combines the two sum operations to provide a more compact expression.

In ER objective we employ reward on actual number of agents (and not on expected number of agents), so there is no need to approximate reward function using piecewise constant or piecewise linear function. In the next section, we also explain how we can optimize the ER objective using pre-computed binomial distributions without sacrificing on the scalability achieved with the EA objective.

4. OPTIMISING ER OBJECTIVE

In this section, we provide a mechanism to optimize the ER objective in a scalable manner. We focus specifically on reward interactions by assuming transition independence. However in Section 4.1, we provide details on the changes required to consider transition dependence. Here are the key major steps in deriving the updated linear formulation to handle ER:

- **Step 1:** Derive value function expression assuming all agents of the same type have same policies (albeit a mixed one).
- **Step 2:** Derive the connection between $Pr^R_{\mathbf{x}^t_{\mathcal{P}}}$ to binomial distribution.
- **Step 3:** Approximate computation of $Pr^R_{\mathbf{x}^t_{\mathcal{P}}}$ using pre-computed binomial distributions.
- **Step 4:** Combine the updates to value function and $Pr^R_{\mathbf{x}^t_{\mathcal{P}}}$ into a linear optimization model.

Step 1: For D-SPAIT, it was shown that all agents of the same type can have the same policy (typically randomized). In this paper, we also make a similar assumption thereby reducing the objective in Equation 8 to the following:

$$V^0(\vec{\pi}) = \sum_{s_i,a_i,t,d \le |\mathcal{P}|} d \cdot Pr^R_{\mathbf{x}^t_{\mathcal{P}}}(d,s_i,a_i) \cdot R(s_i,a_i,d) \quad (9)$$

In this new equation, we no longer sum over agents, i. Instead, since for all agents in a certain state, action pair, the expected reward would be the same, we multiply the expected reward equation by the number of agents, d.

Algorithm 2 provides the general optimization model for the ER objective with homogeneous agents. We define $\rho^{t,R}(s_i,a_i)$ as the probability of an agent being in state, action pairs relevant to s_i, a_i with respect to the reward function. Therefore,

$$\rho^{t,R}(s_i,a_i) = \sum_{(s'_i,a'_i) \in I^R(s_i,a_i)} x^t(s'_i,a'_i)$$

Step 2: We now provide details on the computation of $Pr^R_{\mathbf{x}^t_{\mathcal{P}}}$. Since $\rho^{t,R}(s_i,a_i)$ indicates the probability of being in relevant (with respect to reward function) state, action pairs for s_i, a_i , a key observation is that $Pr^R_{\mathbf{x}^t_{\mathcal{P}}}(.,s_i,a_i)$ will follow a binomial distribution. The twin parameters of this binomial distribution, namely n (number of trials) and α (probability of success of each trial) will be the number of agents, $|\mathcal{P}|$ and $\rho^{t,R}(s_i,a_i)$ respectively. Therefore, we have:

$$Pr^R_{\mathbf{x}^t_{\mathcal{P}}}(d,s_i,a_i) = \binom{|\mathcal{P}|}{d} \cdot \left(\rho^{t,R}(s_i,a_i)\right)^d \cdot \left(1-\rho^{t,R}(s_i,a_i)\right)^{|\mathcal{P}|-d}$$

To make this concrete, we now provide an example.

EXAMPLE 1. *Let us consider a two agent problem, where $I^R(s_i,a_i) = \{(s_i,a_i)\}$ and both agents are homogeneous (same models). Then $\rho^{t,R}(s_i,a_i) = x^t(s_i,a_i)$ and therefore, probability that an agent is not in (s_i,a_i) is $(1-x^t(s_i,a_i))$. So,*

$$Pr^R_{x^t_{\mathcal{P}}}(0,s_i,a_i) = (1-x^t(s_i,a_i)) \cdot (1-x^t(s_i,a_i))$$

$$Pr^R_{x^t_{\mathcal{P}}}(1,s_i,a_i) = 2 \cdot x^t(s_i,a_i) \cdot (1-x^t(s_i,a_i))$$

$$Pr^R_{x^t_{\mathcal{P}}}(2,s_i,a_i) = x^t(s_i,a_i) \cdot x^t(s_i,a_i)$$

Algorithm 2: SolveDSPAIT-H-ER()

$$\max_{\mathbf{x}} \sum_{s_i,a_i,t,d\leq|\mathcal{P}|} d \cdot Pr^R_{\mathbf{x}^t_{\mathcal{P}}}(d,s_i,a_i) \cdot R(s_i,a_i,d) \quad \textbf{s.t.}$$

$$\rho^{t,R}(s_i,a_i) = \sum_{(s'_i,a'_i)\in I^R(s_i,a_i)} x^t(s'_i,a'_i)$$

$$Pr^R_{\mathbf{x}^t_{\mathcal{P}}}(d,s_i,a_i) = \binom{|\mathcal{P}|}{d} \cdot \left(\rho^{t,R}(s_i,a_i)\right)^d \cdot \left(1-\rho^{t,R}(s_i,a_i)\right)^{|\mathcal{P}|-d}, \forall d,s_i,a_i \quad (10)$$

$$\sum_a x^t(s_i,a_i) - \sum_{s',a} x^{t-1}(s'_i,a'_i)\cdot\varphi(s'_i,a_i,s_i) = \alpha^t(s_i), \forall s_i,t \quad (11)$$

$$x^t(s_i,a_i) \in [0,1] \qquad \forall t,s_i,a_i$$

Step 3: Probability computation in Equation 10 is non-linear, so solving the optimization model of Algorithm 2 exactly has significant computational complexity. Therefore, we provide a mechanism to approximate the computation of the objective. Intuitively, we pre-generate binomial distribution probabilities for a fixed set of values (e.g., 0.05, 0.15, 0.25, 0.35...,) for each $\rho^{t,R}_{s_i,a_i}$. In the optimization model, we provide constraints that will ensure that a fixed value that is nearest to the actual value of $\rho^{t,R}_{s_i,a_i}$ is employed.

In implementing this approximation approach, we divide the space of feasible $\rho^{t,R}(s_i,a_i)$ values[2] into K intervals $\{[\check{d}_k(s_i,a_i),\hat{d}_k(s_i,a_i)]\}_{k\in K}$. We pre-generate binomial distribution for $|\mathcal{P}|$ agents for each interval k using a probability of success given by the midpoint of the interval, i.e., $\frac{\check{d}_k(s_i,a_i)+\hat{d}_k(s_i,a_i)}{2}$. $B^t_k(s_i,a_i,i)$ denotes the pre-computed binomial probability values corresponding to all values in interval k for i (out of $|\mathcal{P}|$) agents being in states relevant to s_i,a_i pair.

Step 4: Therefore, a key challenge in the context of the optimization problem is to identify the interval corresponding to the actual value of $\rho^{t,R}(s_i,a_i)$ using linear constraints. This can be enforced through the use of extra variables while preserving linearity of the constraints. Let $y^t_k(s_i,a_i)$ be a binary variable that indicates whether $\rho^t(s_i,a_i)$ belongs to the k^{th} interval. That is to say, if $\rho^t(s_i,a_i) \in [\check{d}_k(s_i,a_i),\hat{d}_k(s_i,a_i)]$ then $y^{t,k}(s_i,a_i)=1$ otherwise 0. The following linear constraints will be employed to identify the interval for a given $\rho^t(s_i,a_i)$:

$$\sum_k y^t_k(s_i,a_i) = 1 \quad (12)$$

$$\rho^{t,R}(s_i,a_i) \geq \sum_k y^t_k(s_i,a_i)\cdot\check{d}_k \quad (13)$$

$$\rho^{t,R}(s_i,a_i) \leq \sum_k y^t_k(s_i,a_i)\cdot\hat{d}_k \quad (14)$$

[2]By default the set of feasible values is [0,1] for finite horizon problems. However, with simple reachability analysis the interval be tightened, specifically the upper bound.

By setting values to $y^t_k(s_i,a_i)$, it is trivial to verify that interval for $\rho^{t,R}(s_i,a_i)$ is correctly identified with the above constraints.

Algorithm 3: SolveDSPAIT-H-ER-Binom()

$$\max_{\mathbf{x}} \sum_{s_i,a_i,t,k} V^t_k(s_i,a_i) \quad \textbf{s.t.}$$

$$V^t_k(s_i,a_i) \leq y^t_k(s_i,a_i)\cdot M \quad (15)$$

$$V^{t,k}(s_i,a_i) \leq \sum_i B^t_k(s_i,a_i,i)\cdot i\cdot R(s_i,a_i,i) \quad (16)$$

$$V^{t,k}(s_i,a_i) \geq \sum_i B^t_k(s_i,a_i,i)\cdot i\cdot R(s_i,a_i,i) - (1-y^{t,k}(s_i,a_i))\cdot M \quad (17)$$

$$\rho^{t,R}(s_i,a_i) = \sum_{(s'_i,a'_i)\in I^R(s_i,a_i)} x^t(s'_i,a'_i) \qquad \forall s_i,a_i,t$$

$$\sum_a x^t(s_i,a_i) - \sum_{s'_i,a_i} x^{t-1}(s'_i,a_i)\cdot \cdot\varphi(s'_i,a_i,s_i) = \alpha^t(s_i) \qquad \forall s_i,t \quad (18)$$

$$\sum_k y^t_k(s_i,a_i) = 1 \quad (19)$$

$$\rho^{t,R}(s_i,a_i) \geq \sum_k y^{t,k}(s_i,a_i)\cdot\check{d}_k \qquad \forall s_i,a_i,t \quad (20)$$

$$\rho^{t,R}(s_i,a_i) \leq \sum_k y^{t,k}(s_i,a_i)\cdot\hat{d}_k \qquad \forall s_i,a_i,t \quad (21)$$

$$\rho^{t,R}(s_i,a_i) \in [0,1] \qquad \forall t,s_i,a_i \quad (22)$$

We provide the linear formulation that combines these insights in Algorithm 3. Since expected reward depends on the intervals for probability, we introduce a new set of variables $V^t_k(s_i,a_i)$. Constraints 15-17 ensure the following:

- If $\rho^{t,R}(s_i,a_i)$ does not belong to interval k, then $V^t_k(s_i,a_i)$ is assigned a value of 0.
- If $\rho^{t,R}(s_i,a_i)$ belongs to interval k, then $V^t_k(s_i,a_i)$ is assigned the approximated expected reward computed from pre-computed binary distribution values, $B^t_k(s_i,a_i,i)$.

Rest of the constraints ensure flow preservation for the $\mathbf{x}$ variables (Constraint 18) and assignment of the right interval for $\rho^{t,R}(s_i,a_i)$.

4.1 Anonymous Transition Interactions

To handle anonymous transition interactions, we have to modify the flow preservation constraint of Equation 11. The key difference from D-SPAIT is that we consider expected transition probability as opposed to transition probability for expected number of agents. Specifically, the new flow preservation constraint is given by:

$$\sum_a x^t(s_i,a_i) - \sum_{s',a} x^{t-1}(s'_i,a'_i)\cdot\Big[\sum_{d\leq|\mathcal{P}|} Pr^{\varphi}_{\mathbf{x}^t_{\mathcal{P}}}(d,s'_i,a_i)\cdot \cdot\varphi(s'_i,a_i,s_i,d)\Big] = \alpha^t(s_i), \forall s_i,t \quad (23)$$

$Pr^{\varphi}_{\mathbf{x}^t_{\mathcal{P}}}$ is computed in a similar way as $Pr^{R}_{\mathbf{x}^t_{\mathcal{P}}}$.

$$Pr^{\varphi}_{\mathbf{x}^t_{\mathcal{P}}}(d, s_i, a_i) = \binom{|\mathcal{P}|}{d} \cdot \left(\rho^{t,\varphi}(s_i, a_i)\right)^d \cdot \left(1 - \rho^{t,\varphi}(s_i, a_i)\right)^{|\mathcal{P}|-d}$$

$$\text{where} \quad \rho^{t,\varphi}(s_i, a_i) = \sum_{(s'_i, a'_i) \in I^{\varphi}(s_i, a_i)} x^t(s'_i, a'_i)$$

Once again, since probability computation is non-linear, we employ a similar approximation to the one for expected reward computation. We pre-generate binomial distribution values for certain fixed values of $\rho^{t,\varphi}(s_i, a_i)$. In the optimization model, we have constraints that ensure expected transition probability is computed corresponding to the nearest fixed value for the actual value of $\rho^{t,\varphi}(s_i, a_i)$. These constraints will be very similar to the ones introduced for expected reward computation. Due to space constraints, we do not provide it here.

It should be noted that when there are transition function based interactions, the optimal policy typically is a history dependent policy, i.e., action is dependent on the history of state, action pairs. Since computing history based policies reduces the scalability significantly, we focus primarily on policies that are only based on current state.

4.2 Extending to Multiple Types

To ensure ease of explanation, we have so far focussed on problems where all agents are of a single type. We now consider problems with multiple types. First, we consider modifications to the D-SPAIT tuple:

$$\left\langle \Gamma, \{\mathcal{P}_\tau\}_{\tau\in\Gamma}, \{\mathcal{S}_\tau\}_{\tau\in\Gamma}, \mathcal{A}, I^R, I^{\varphi}, \{R_\tau\}_{\tau\in\Gamma}, \{\varphi_\tau\}_{\tau\in\Gamma}, \{\alpha_\tau\}_{\tau\in\Gamma} \right\rangle$$

The first extension to D-SPAIT is to represent multiple types of agents, where agents of each type τ have a different state and action space, i.e., S_τ and A_τ along with different transition, φ_τ and reward functions, R_τ. $\mathcal{P}_\tau$ is the set of agents of type τ. When considering multiple types of agents each with a different state and action space, I^R and I^{φ} are defined for tuples of state, action and type. $I^R(s_\tau, a_\tau, \tau)$ is the set of state, action and type pairs of agents that will have reward interactions with an agent of type τ in state s_τ and taking action a_τ.

$$I^R(s_\tau, a_\tau, \tau) = \{(s_1, a_1, 1), (s_2, a_2, 2), \cdots\}$$

where s_τ is a state for a type τ agent. We will have a similar structure for I^{φ}.

The second set of changes are to the approach and we go through the same four steps employed in Section 4. In **Step 1**, we update the value function. As indicated earlier, all agents of the same type have the same policy. We will employ $\mathbf{x}^t_\tau$ to represent the flow variables corresponding to an agent of type τ. From Equation 8, we have:

$$V^0(\vec{\pi}) = \sum_{i, s_i, a_i, t, d \le |\mathcal{P}|} Pr^{R}_{\mathbf{x}^t_{\mathcal{P}}}(d, s_i, a_i) \cdot R(s_i, a_i, d)$$

Accommodating for the fact that all agents having same types have same policies, we have:

$$= \sum_{\tau, s_\tau, a_\tau, t} \sum_{\left[d_1 \le |\mathcal{P}_1|, \ldots, d_\tau \le |\mathcal{P}_\tau| \ldots d_{|\Gamma|} \le |\mathcal{P}_{|\Gamma|}|\right]} d_\tau \cdot \left[\prod_{\tau' \in \Gamma} Pr^{R}_{\mathbf{x}^t_{\mathcal{P}_{\tau'}}}(d_{\tau'}, s_\tau, a_\tau)\right] \cdot R_\tau(s_\tau, a_\tau, \sum_{\tau'} d_{\tau'}) \tag{24}$$

where

$$Pr^{R}_{\mathbf{x}^t_{\mathcal{P}_{\tau'}}}(d_{\tau'}, s_\tau, a_\tau) = \binom{|\mathcal{P}_{\tau'}|}{d_{\tau'}} \cdot \left(\rho^{t,R}_{\tau'}(s_\tau, a_\tau, \tau)\right)^{d_{\tau'}} \cdot \left(1 - \rho^{t,R}_{\tau'}(s_\tau, a_\tau, \tau)\right)^{|\mathcal{P}_{\tau'}| - d_{\tau'}} \tag{25}$$

$$\rho^{t,R}_{\tau'}(s_\tau, a_\tau, \tau) = \sum_{(s_{\tau'}, a_{\tau'}, \tau') \in I^R(s_\tau, a_\tau, \tau)} x^t_{\tau'}(s_{\tau'}, a_{\tau'}) \tag{26}$$

Equations 25 and 26 provide the required updates to **Step 2** in order to represent types.

For **Steps 3 and 4**, we can again employ this objective function in a linear optimization formulation by precomputing the binary distribution employed to calculate probability distribution[3] of Equation 25 and computing the interval for ρ variables using binary variables and linear constraints.

EXAMPLE 2. *Let us consider a two agent problem, where* $I^R(s_1, a_1, 1) = \{(s_1, a_1, 1), (s_2, a_2, 2)\}$ *and each agent is of a different type. Then,*

$$\begin{aligned}
\rho^{t,R}_1(s_1, a_1, 1) &= x^t_1(s_i, a_i) \\
\rho^{t,R}_2(s_1, a_1, 1) &= x^t_2(s_2, a_2) \\
Pr^{R}_{\mathbf{x}^t_{\mathcal{P}}}(0, s_1, a_1) &= (1 - x^t_1(s_1, a_1)) \cdot (1 - x^t_2(s_2, a_2)) \\
Pr^{R}_{\mathbf{x}^t_{\mathcal{P}}}(1, s_1, a_1) &= x^t_1(s_1, a_1) \cdot (1 - x^t_2(s_2, a_2)) + \\
&\quad + x^t_2(s_2, a_2) \cdot (1 - x^t_1(s_1, a_1)) \\
Pr^{R}_{\mathbf{x}^t_{\mathcal{P}}}(2, s_1, a_1) &= x^t_1(s_1, a_1) \cdot x^t_2(s_2, a_2)
\end{aligned}$$

It should be noted that many of the benchmark problems from Dec-MDP literature have two agents and therefore we can employ the probability expressions derived in Example 2.

4.3 Key Properties and Enhancements

In this subsection, we highlight key properties and enhancements of the new approximation technique that precomputes binomial distribution values.

- In the previous work by Varakantham *et al.*, complex reward functions (any non-linear function) had to be approximated using piecewise constant or piecewise convex, linear functions to provide scalable solutions. In this work, as we consider reward function directly, there is no need for approximation of the reward function and any complex reward function can be handled directly.

[3]It should be noted that the ρ value computed in Equation 26 will always be between 0 and 1, as we are computing probability for one agent of type τ' to be in state, action pair s_τ, a_τ for a type τ agent.

Horizon	Binom objective	Binom Simulated	Existing
50	154.94	155.05	154.94
60	185.71	185.70	185.71
70	216.48	216.43	216.47
80	246.86	246.82	247.24
90	278.02	277.97	278.01
100	308.79	308.83	308.78
1000	3077.63	3077.98	3078.00

Table 2: Recycling Robots.

Horizon	Binom objective	Bimom Simulated	Existing
2	0.00	0.00	0.00
3	0.13	0.13	0.13
4	0.43	0.43	0.43
5	0.90	0.88	0.89
6	1.49	1.49	1.49
10	4.68	4.68	4.68
100	94.35	94.35	94.26
1000	994.35	994.36	994.20

Table 3: Meeting in a grid 3x3.

Horizon	Binom objective	Bimom Simulated	Existing
5	0.00	0.00	0.00
6	0.00	0.00	0.00
7	0.71	0.71	0.71
8	1.66	1.66	1.67
9	2.61	2.61	2.68
10	3.66	3.64	3.68
20	13.65	13.64	13.68
30	23.65	23.65	23.68
40	33.65	33.65	33.68
50	43.65	43.65	43.68
100	93.65	93.66	93.68

Table 4: Meeting in a grid 8x8.

- In the previous sections, we provide general expressions assuming an agent in every state, action pair has interactions with agents in other state, action pairs. It should be noted that if the set $I^R(s_i, a_i)$ (or $I^R(s_\tau, a_\tau, \tau)$) is empty for a state, action pair, then there will be no prior computation of binomial distribution.
- Finally, while we do not explicitly address the curse of history, our approach can solve problems with reasonably long horizons (≤ 50). In problems with even longer horizons, it is always feasible to employ a rolling horizon method where our approach is employed to compute a policy for a horizon of 50 and then end state distribution is used for the start distribution of the second block of 50 time steps and so on.

5. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our approach on existing benchmark problems introduced by Spaan *et al.* [16, 6] and Dibangoye *et al.* [4].

We first compare all the algorithms with respect to solution quality on eight navigation problems by Spaan and Melo *et al.* [16, 6]. In all these problems, the number of agents varied between 2 and 4 and horizon was fixed at 250. For all the navigation problems for all horizon values, we were able to obtain the solution in less than 15 minutes[4]. For our approach, we provide both the objective value (referred to as BinomObjective) of our optimization problem and the Dec-MDP objective value computed using simulation (referred to as BinomSimulated) over 20000 trials[5] for the policy computed by our approach. Similarly for the PWLD algorithm by Varakantham *et al.*, we provide both PWLDObjective and PWLDSimulated. The optimal solution value (obtained by using either Melo *et al.* or Dibangoye *et al.*) is provided under the column titled "Existing". Here are the key observations from Table 1:

- On all the eight problems, our approach either provided the same or better solution quality (BinomSimulated) than the one obtained by PWLD.
- BinomObjective was either equal or very close to BinomSimulated in all the eight problems. This implies that the ER objective computed through pre-computed binomial distribution values is approximating the expected reward accurately.
- In all the eight navigation problems, our solution was very close to the optimal solution in problems where rewards range from -20 to 1.
- For PWLD, there was a significant difference between PWLDObjective and PWLDSimulated on at least 4 problems indicating an inaccurate approximation.

We also compared solution quality on a few other benchmark problems employed in Dibangoye *et al.* [4] where we have accurate parameter values. On these problems, we were unable to get any reasonable solution values for PWLD, primarily because piecewise constant approximation of reward functions in these problems was not accurate. Table 2, Table 3 and Table 4 provide the results for Binom for the recycling robots, meeting grid 3x3 and meeting grid 8x8 problems respectively in comparison with the optimal value. We were able to compute solutions within 15 minutes on all three problems and for all horizons. Here are the key observations with respect to solution quality:

- On all the three problems and for all horizon values, we were able to get optimal solution quality.
- On all the three problems and for all horizon values, BinomObjective was either equal or very close to BinomSimulated indicating a good approximation of expected reward.

Finally, we compare against the approach of Varakantham *et al.* [17] referred to as PWLD on the navigation problems from [6] with 200 agents. We consider three different reward configurations, primarily ones which are piecewise constant so as to ensure PWLD does not have to approximate the rewards. Both approaches generated the solutions within 2 hours. The three reward configurations are as follows: (i) Reward Configuration 1: Piecewise Linear and Concave; (ii)

[4]Configuration for the machine on which experiments were run: Windows 7 Enterprise, 64Bit OS, Interl(R) Core(TM) i5-3470, 3.2GHz 4GB RAM

[5]When we calculate the simulated value using multiple rounds of 20000 samples, the change in value was $\leq \pm 0.1$

Map	Horizon	PWLDObjective	PWLDSimulated	BinomObjective	BinomSimulated	Existing
Map 1	250	12.013	8.820	10.94	10.947	12.059
Map 2	250	10.576	7.373	10.238	10.246	11.108
Map 3	250	13.646	11.339	12.884	12.877	13.837
Pentagon	250	15.363	14.189	14.697	14.65	16.016
CIT	250	10.576	10.575	10.576	10.583	11.128
ISR	250	15.363	14.444	14.49	14.413	14.407
MIT	250	5.856	5.756	6.352	6.362	6.705
SUNY	250	10.576	10.575	10.576	10.57	11.149

Table 1: Navigation domains with 2-4 agents. Comparison with Melo *et al.*s approach.

Map	Reward Configuration 1		Reward Configuration 2		Reward Configuration 3	
	PWLD Simulated	Binom Simulated	PWLD Simulated	Binom Simulated	PWLD Simulated	Binom Simulated
Map 1	725.41	744.77	906.77	932.67	1745.32	1757.27
Map 2	154.26	231.79	-46.29	0	149.87	163.51
Map 3	88.44	114.8	-24.46	1.07	55.17	76.24
Map 4	-44.83	74.47	-70.43	0	-146.97	5.53
Pentagon	602.92	606.23	-7.7	8.46	1316.04	1324.8
CIT	786.57	820.42	566.04	597.75	1511.76	1524.58
ISR	564.79	607.24	-38.7	11.74	1364.78	1377.53
MIT	1258.2	1257.14	881	917.67	1764.05	1773.65
SUNY	564.91	607.28	-27.46	12.47	1313.26	1321.99

Table 5: Navigation domains with different reward configurations and 200 agents. Comparison with Varakantham *et al.*s approach (PWLD).

Reward Configuration 2: Piecewise Linear and Convex; (iii) Reward Configuration 3: Multimodal piecewise linear.

In the experiments, we used same number of intervals for ρ in our approach as number of piecewise components used in the PWLD approach. Comparison of simulated values of policies generated using PWLD and our binomial approach is shown in table 5. On all the maps and reward structures, our approach provided solution which is better or at least as good as PWLD. Our approach would perform even better than PWLD if we use non-linear reward configurations that do not have piecewise constant/linear components, as PWLD would have to further approximate the rewards.

In conclusion, our approach provides better performance than the approach by Varakantham *et al.* in problems with fewer and large number of agents. Furthermore, on problems with few agents (2-4 agents), our approach provided comparable solution quality to optimal solution approaches.

REFERENCES

[1] R. Becker, S. Zilberstein, V. Lesser, and C. Goldman. Solving transition independent decentralized Markov decision processes. *Journal of Artificial Intelligence Research*, 22:423–455, 2004.

[2] D. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[3] D. S. Bernstein, E. A. Hansen, and S. Zilberstein. Bounded policy iteration for decentralized POMDPs. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 1287–1292, Edinburgh, Scotland, 2005.

[4] J. S. Dibangoye, C. Amato, and A. Doniec. Scaling up decentralized mdps through heuristic search. *arXiv preprint arXiv:1210.4865*, 2012.

[5] A. Kumar, S. Zilberstein, and M. Toussaint. Scalable multiagent planning using probabilistic inference. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2140–2146, Barcelona, Spain, 2011.

[6] F. S. Melo and M. Veloso. Decentralized mdps with sparse interactions. *Artificial Intelligence*, 175(11):1757–1789, 2011.

[7] H. Mostafa and V. Lesser. Offline planning for communication by exploiting structured interactions in decentralized mdps. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 2, pages 193–200. IET, 2009.

[8] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 133–139, 2005.

[9] F. A. Oliehoek. Decentralized pomdps. *Reinforcement Learning*, pages 471–503, 2012.

[10] S. Omidshafiei, A. akbar Agha-mohammadi, C. Amato, and J. P. How. Decentralized control of partially observable markov decision processes using belief space macro-actions. In *ICRA'15*, 2015.

[11] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[12] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.

[13] S. Seuken and S. Silberstein. Improved memory-bounded dynamic programming for decentralized POMDPs. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, pages 344–351, 2007.

[14] E. Shieh, M. Jain, A. X. Jiang, and M. Tambe. Efficiently solving joint activity based security games. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 346–352. AAAI Press, 2013.

[15] E. Sonu, Y. Chen, and P. Doshi. Individual planning in agent populations: Exploiting anonymity and frame-action hypergraphs. In *International Conference on Automated Planning and Scheduling*, 2015.

[16] M. T. Spaan and F. S. Melo. Interaction-driven markov games for decentralized multiagent planning under uncertainty. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pages 525–532. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[17] P. Varakantham, Y. Adulyasak, and P. Jaillet. Decentralized stochastic planning with anonymity in interactions. In *Proc. of the AAAI Conference on Artificial Intelligence*, pages 2505–2512, 2014.

[18] P. Varakantham, S.-F. Cheng, G. Gordon, and A. Ahmed. Decision support for agent populations in uncertain and congested environments. In *26th AAAI Conference on Artificial Intelligence (AAAI-12)*, pages 1471–1477, 2012.

[19] P. Varakantham, J. Y. Kwak, M. Taylor, J. Marecki, P. Scerri, and M. Tambe. Exploiting coordination locales in distributed POMDPs via social model shaping. In *Nineteenth International Conference on Automated Planning and Scheduling*, page 313âĂŞ320, 2009.

[20] P. Velagapudi, P. Varakantham, K. Sycara, and P. Scerri. Distributed model shaping for scaling to decentralized pomdps with hundreds of agents. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 955–962. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[21] S. J. Witwicki and E. H. Durfee. Influence-based policy abstraction for weakly-coupled dec-pomdps. In *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling*, pages 185–192, 2010.

[22] Z. Yin and M. Tambe. Continuous time planning for multiagent teams with temporal constraints. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 465–471, 2011.