Collective Multiagent Sequential Decision Making Under Uncertainty

Duc Thien Nguyen and **Akshat Kumar** and **Hoong Chuin Lau**
School of Information Systems
Singapore Management University
{dtnguyen.2014,akshatkumar,hclau}@smu.edu.sg

Abstract

Multiagent sequential decision making has seen rapid progress with formal models such as decentralized MDPs and POMDPs. However, scalability to large multiagent systems and applicability to real world problems remain limited. To address these challenges, we study multiagent planning problems where the collective behavior of a *population* of agents affects the joint-reward and environment dynamics. Our work exploits recent advances in graphical models for modeling and inference with a population of individuals such as *collective graphical models* and the notion of *finite partial exchangeability* in lifted inference. We develop a collective decentralized MDP model where policies can be computed based on counts of agents in different states. As the policy search space over counts is combinatorial, we develop a sampling based framework that can compute open and closed loop policies. Comparisons with previous best approaches on synthetic instances and a real world taxi dataset modeling supply-demand matching show that our approach significantly outperforms them w.r.t. solution quality.

1 Introduction

Multiagent sequential decision making has seen rapid progress with the development of formal models such as decentralized MDPs and POMDPs (Bernstein et al. 2002). Dec-(PO)MDPs capture planning problems where agents act based on different partial information about the environment and about each other to maximize a global reward function. Applications of Dec-POMDPs include coordinating planetary rovers (Becker et al. 2004), multi robot coordination (Amato et al. 2015), and improving throughput in wireless networks (Pajarinen, Hottinen, and Peltonen 2014). However, scalability of algorithms to large scale problems has been limited due to high computational complexity.

To scale up Dec-(PO)MDP algorithms, several restricted class of models have been proposed with transition/observation independence (Nair et al. 2005; Kumar, Zilberstein, and Toussaint 2011), event driven interactions (Becker, Zilberstein, and Lesser 2004), and weak coupling (Spaan and Melo 2008; Witwicki and Durfee 2010). Recently, there is an increasing interest in settings where the identity of agents does not affect interactions among them (Varakantham, Adulyasak, and Jaillet 2014; Sonu, Chen, and Doshi 2015; Robbel, Oliehoek, and Kochenderfer 2016). Instead, interaction among agents is primarily influenced by the number of agents, similar to well known congestion games (Meyers and Schulz 2012). Planning in such anonymous agent settings has many applications in urban transportation (Varakantham et al. 2012).

To address such anonymous planning setting, our main contribution is a new framework called *collective* decentralized Dec-MDPs ($\mathbb{C}$Dec-MDPs) for collective multiagent decision making under uncertainty, and developing a model free sampling approach to optimize policies in this framework. Our framework is influenced by recent advances in the graphical models literature for inference with aggregate data such as *collective graphical models* (Sheldon and Dietterich 2011; Nguyen et al. 2016) and the notion of exchangeability (Niepert and Van den Broeck 2014) in lifted inference. We establish several basic properties of $\mathbb{C}$Dec-MDPs such as its agent count based sufficient statistic. As the space of counts is combinatorial, optimizing policies over counts is intractable. Therefore, we develop an inference-based algorithm for planning in $\mathbb{C}$Dec-MDPs. However, the standard planning-as-inference strategy where the planning problem is translated to that of inference in a graphical model suffers from poor convergence and local optima in our case (Toussaint, Harmeling, and Storkey 2006; Kumar, Zilberstein, and Toussaint 2015). Therefore, we develop a novel approach by combining the notion of fictitious play from game theory (Meyers and Schulz 2012), exchangeable variable models from lifted inference (Niepert and Van den Broeck 2014) and the inference based planning (Toussaint, Harmeling, and Storkey 2006). Furthermore, our approach is model free and requires only aggregated count-based samples from a simulator. Empirically, it scales to a real world supply-demand taxi matching problem with 8000 taxis, and provides significant quality improvements over previous best approaches.

Related work: Recently, (Robbel, Oliehoek, and Kochenderfer 2016; Sonu, Chen, and Doshi 2015) also develop models to exploit anonymity in multiagent planning. Their model assumes a pre-defined interaction graph among agents, whereas interaction among agents in our model is based on counts without any fixed graph. In Sonu et al., a plan is computed in interactive POMDP model for an in-

$\mathbb{I}_t^m(i) \in \{0,1\}$	if agent m is at state i at time t or $s_t^m = i$
$\mathbb{I}_t^m(i,j) \in \{0,1\}$	if agent m takes action j in state i at time t or $(s_t^m, a_t^m) = (i,j)$
$\mathbb{I}_t^m(i,j,i') \in \{0,1\}$	if agent m takes action j in state i at time t and transitions to state i' or $(s_t^m, a_t^m, s_{t+1}^m) = (i,j,i')$
$n_t(i) \in [0;M]$	Number of agents at state i at time t
$n_t(i,j) \in [0;M]$	Number of agents at state i taking action j at time t
$n_t(i,j,i') \in [0;M]$	Number of agents at state i taking action j at time t and transitioning to state i' at time $t+1$
$\boldsymbol{n}_{s_t}$	Count table $(n_t(i)\ \forall i \in S)$
$\boldsymbol{n}_{s_t a_t}$	Count table $(n_t(i,j)\ \forall i \in S, j \in A)$

Table 1: Summary of important notations; M denotes agent population size; individual agents indexed using m

dividual agent. Our goal is to find a policy for a team of agents. Closely related to our work are the anonymous planning based models proposed in (Varakantham, Adulyasak, and Jaillet 2014; Varakantham et al. 2012). In these models, only an approximate behavior of the agent population is determined by computing the "average" flow of agents. However, such an approximation can suffer from high error, as we also demonstrate empirically. To alleviate such issues, our $\mathbb{C}$Dec-MDP model provides an accurate representation of collective multiagent decision making taking into account the underlying stochasticity.

2 Collective Decentralized Dec-MDPs

The framework of $\mathbb{C}$Dec-MDP consists of the following:

- A finite planning horizon H.
- The number of agents M. An agent m can be in one of the states in the state space S. The joint state space is $\times_{m=1}^{M} S$. We denote a single state as $i \in S$.
- A set of action A for each agent m. We denote an individual action as $j \in A$.
- Let $(\boldsymbol{s}_{1:H}, \boldsymbol{a}_{1:H})^m = (s_1^m, a_1^m, s_2^m \ldots, s_H^m, a_H^m)$ denote the complete state-action trajectory of an agent m. We denote the state and action of agent m at time t using random variables s_t^m, a_t^m. Different indicator functions $\mathbb{I}_t(\cdot)$ are defined in table 1. We define the following counts given the trajectory of each agent $m \in M$:
  - $n_t(i,j,i') = \sum_{m=1}^{M} \mathbb{I}_t^m(i,j,i')\ \forall i,i' \in S, j \in A$
  - $n_t(i,j) = \sum_{m=1}^{M} \mathbb{I}_t^m(i,j) \qquad \forall i \in S, j \in A$
  - $n_t(i) = \sum_{m=1}^{M} \mathbb{I}_t^m(i)\ \forall i \in S$

  As noted in table 1, count $n_t(i,j)$ denotes the number of agents in state i taking action j at time step t; other counts are interpreted analogously. We denote count tables as $\boldsymbol{n}_{s_t} = (n_t(i)\ \forall i \in S)$ and $\boldsymbol{n}_{s_t a_t} = (n_t(i,j)\ \forall i \in S, j \in A)$; table $\boldsymbol{n}_{s_t a_t s_{t+1}}$ is defined analogously.
- We assume that an agent m has local full observability. The agent deterministically observes its local state, say $s_t^m = i$, at time t. In addition, it also observes the aggregate count $n_t(i)$ of other agents present at the state i.
- The transition function $\phi_{t,t+1} : S \times A \times S \times [0,M] \to \Re^+$ denotes the probability that an agent moves from state i at time t to i' when taking action j and there are $n_t(i)$ agents at state i: $\phi_t\big(s_{t+1}^m = i' | s_t^m = i, a_t^m = j, n_t(i)\big)$. The transition function is the same for all the agents.
- Each agent m has a non-stationary policy $\pi_t^m : S \times [0;M] \times A \to [0,1]$, with $\pi_t^m(j|i, n_t(i))$ denoting the probability of agent m to take action j given its observation $(i, n_t(i))$ at time t. We denote the policy over planning horizon of an agent m to be $\pi^m = (\pi_1^m, \ldots, \pi_H^m)$.
- An agent m receives a reward $R_t^m = R_t(i,j,n_t(i))$ dependent on the count $n_t(i)$ when taking action j at state i at time t. The reward function is same for all the agents.
- Initial state distribution, $P(i) \forall i \in S$, is same for all agents.

We only present here the simplest version where ϕ_t and R_t are dependent on $n_t(i)$. They can be extended to depend on $n_t(i,j)$. Similarly, we have assumed that all the agents are of the same type. Our model and algorithms can be extended to handle multiple agent types. The $\mathbb{C}$Dec-MDP model is not transition, reward or observation independent.

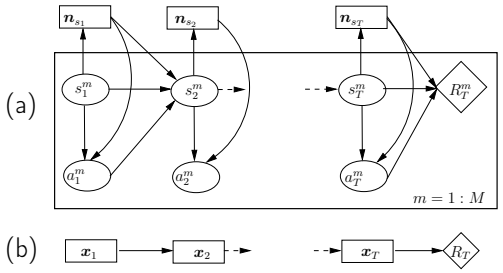

Figure 1: (a) DBN for T-step reward for $\mathbb{C}$Dec-MDP; (b) Deterministic Markov chain for T-step reward in the D-SPAIT model

Our model is motivated by the decentralized stochastic planning model (D-SPAIT) for anonymous agents proposed in (Varakantham, Adulyasak, and Jaillet 2014), and the framework of congestion games (Meyers and Schulz 2012). In our work, we explicitly model the distribution over counts $n(\cdot)$ of individuals and use this distribution as the basis for planning. In contrast, the D-SPAIT model is based on the concept of approximating the planning problem using *expected counts* of agents. Intuitively, if $\mathbb{E}[f(\boldsymbol{n})]$ denotes the planning objective over counts $\boldsymbol{n}$, then D-SPAIT model approximates this objective as $f\big(\mathbb{E}[\boldsymbol{n}]\big)$. Table 2 show the computation of such average flow; $x_{s_t}(i)$ denotes the expected number of agents in state i at time t. Computing policies based on such average flow leads to inaccurate estimation of the true objective function and lower quality policies, as we also demonstrate empirically. Fig. 1 shows DBNs for $\mathbb{C}$Dec-MDPs and the D-SPAIT model using the plate notation.

Motivating application: We now present a motivating application for $\mathbb{C}$Dec-MDPs based on the taxi supply demand

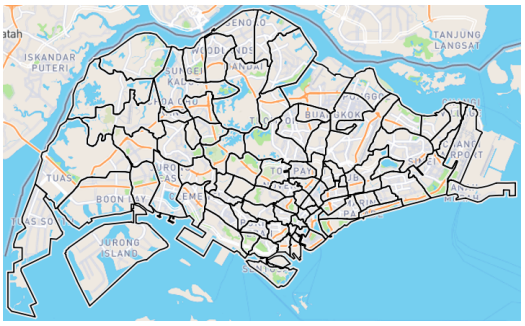

Figure 2: Zonal division of Singapore based on postal codes from (Cheng and Nguyen 2011)

problem introduced in (Varakantham et al. 2012). Figure 2 shows the map of Singapore divided into different zones. We are concerned with the problem of optimizing taxi drivers' policies such that the total profit of taxi fleet is maximized. Such a setting is useful in the case of autonomous taxi fleet operations for revenue maximization. We next describe a taxi driver's decision making. At time t, a taxi driver observes its current zone z and also the count of other taxis in zone z. The driver has two actions: decide to *stay* in the zone to look for passengers or *move* to another zone (out of 81 zones). If the driver stays in the current zone, its probability of picking up a passenger is dictated based on the current demand and the count of other taxis in the zone. E.g., if the demand is higher than the number of taxis, then the driver picks up a passenger with probability close to 1, else the probability is smaller than 1 (based on the ratio of taxis and the current demand). If the driver picks up a passenger, it moves to the passenger's intended destination. Such transition probabilities can be encoded into the transition function ϕ_t of the $\mathbb{C}$Dec-MDP. The reward a driver gets upon picking a passenger is the total profit of the trip (trip payment minus the fuel cost of moving). If the drive moves to another zone (without a passenger), it incurs the fuel cost for moving.

It is clear from this application that identities of taxi drivers are not important for planning. A taxi driver's decision and state transition is based only on the aggregate count of other taxis and the total demand. Similarly, each taxi driver's observation can be different from each other based on their current zone making this a large multiagent planning problem. Thus, such problems can be modeled using the $\mathbb{C}$Dec-MDP model.

Policy representation: The benefit of models such as D-SPAIT and $\mathbb{C}$Dec-MDPs lies when agent population is large, and agent identity does not affect the reward or the transition function. E.g., in the taxi fleet operation optimization problem discussed earlier such aggregate interactions occur. Given large number of taxis ($\approx$ 8000), it is infeasible to compute a unique policy for each taxi. Therefore, similar to the D-SPAIT model, our goal is to compute a homogenous policy π for all the agents. As the policy is dependent on counts n_t, it allows for an expressive class of policies. As the space of counts can be large, we further allow for optimizing *piecewise* policies. That is, $\pi_t(j|i,\cdot)$ is a piecewise function over the space of all possible counts $n_t(i)$. This is similar to a controller based policy with each piece as a controller node (Hansen 1998).

$$
\begin{aligned}
x_{s_1}(i) &= M \times P(i), && \forall i \in S \\
x_{s_t a_t}(i,j) &= x_{s_t}(i) \times \pi(j|i, x_{s_t}(i)) && \forall i \in S, j \in A \\
x_{s_{t+1}}(i') &= \sum_{i,j} x_{s_t a_t}(i,j) \phi_t(i'|i,j,x_{s_t}(i)) && \forall i' \in S
\end{aligned}
$$

Table 2: Average approximation of agent flow

We define an *open loop* policy as a policy where action selection only depends on the current local state of the agent without any dependence on the count information. In a *closed loop* policy, action selection depends on counts also in addition to the agent's local state. Our proposed model free algorithm developed in the following sections can train both open and closed loop policies, whereas previous average flow based approaches are limited to open loop policy optimization.

3 Counts As Sufficient Statistic

We now establish several basic properties of the $\mathbb{C}$Dec-MDP model. For a fixed population M, let $\{(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T})^m \ \forall m\}$ denote the state-action trajectories of different agents sampled from the DBN in figure 1(a). Let $\boldsymbol{n}_{1:T} = \{(\boldsymbol{n}_{s_t}, \boldsymbol{n}_{s_t a_t}, \boldsymbol{n}_{s_t a_t s_{t+1}}) \ \forall t = 1:T\}$ be the combined vector of the resulting count tables for each time step t.

Theorem 1. *Count tables $\boldsymbol{n}_{1:T}$ are the sufficient statistic for a sample of M state-action trajectories from the $\mathbb{C}$Dec-MDP graphical model in figure 1(a).*

Proof. Let $(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T}) = \{(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T})^m \ \forall m\}$ denote the trajectories of all the agents. The joint-distribution $P(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T}; \pi)$ is defined as:

$$
\begin{aligned}
&= \prod_{m=1}^{M} \Bigg[\prod_{i \in S} P(i)^{\mathbb{I}_t^m(i)} \prod_{t=1}^{T-1} \prod_{i,j,i'} \Big[\pi_t(j|i, n_t(i))^{\mathbb{I}_t^m(i,j)} \\
&\phi_t(i'|i,j,n_t(i))^{\mathbb{I}_t^m(i,j,i')} \Big] \prod_{i,j} \pi_T(j|i,n_t(i))^{\mathbb{I}_t^m(i,j)} \Bigg]
\end{aligned}
$$

We can simplify the above expression by grouping together terms from all the agents. The resulting expression $f(\boldsymbol{n}_{1:T}; \pi)$ depends only on counts $\boldsymbol{n}_{1:T}$ as:

$$
\begin{aligned}
f(\boldsymbol{n}_{1:T}; \pi) = &\prod_{i \in S} P(i)^{n_1(i)} \prod_{t=1}^{T-1} \prod_{i,j,i'} \Big[\pi_t(j|i, n_t(i))^{n_t(i,j)} \\
&\phi_t(i'|i,j,n_t(i))^{n_t(i,j,i')} \Big] \prod_{i,j} \pi_T(j|i,n_t(i))^{n_t(i,j)} \qquad (1)
\end{aligned}
$$

Thus, count tables $\boldsymbol{n}_{1:T}$ are the sufficient statistic for the population sample as the joint-probability $P(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T}; \pi)$ is a function of counts $\boldsymbol{n}_{1:T}$. □

We next define a distribution directly over the count tables $\boldsymbol{n}_{1:T}$ as below:

Theorem 2. *The distribution $P(\boldsymbol{n}_{1:T};\pi)$ is defined as:*

$$P(\boldsymbol{n}_{1:T};\pi) = h(\boldsymbol{n}_{1:T}) f(\boldsymbol{n}_{1:T};\pi) \tag{2}$$

where $f(\boldsymbol{n}_{1:T};\pi)$ is given in (1)*. The function $h(\boldsymbol{n}_{1:T})$ counts the total number of ordered M state-action trajectories with sufficient statistic equal to $\boldsymbol{n}$, given as:*

$$h(\boldsymbol{n}_{1:T}) = \frac{M!}{\prod_{i\in S} n_1(i)!}\left[\prod_{t=1}^{T-1}\prod_{i\in S}\frac{n_t(i)!}{\prod_{i'\in S, j\in A} n_t(i,j,i')!}\right] \times \left[\prod_{i\in S}\frac{n_t(i)!}{\prod_{j\in A} n_t(i,j)!}\right] \times \mathbb{I}[\boldsymbol{n}_{1:T}\in\Omega_{1:T}] \tag{3}$$

Set $\Omega_{1:T}$ is the set of all allowed consistent count tables as:

$$\sum_{i\in S} n_t(i) = M \;\forall t \;; \sum_{j\in A} n_t(i,j) = n_t(i)\;\forall j, \forall t \tag{4}$$

$$\sum_{i'} n_t(i,j,i') = n_t(i,j)\;\forall i\in S, j\in A, \forall t \tag{5}$$

Proof is provided in the extended version. Function $h(\boldsymbol{n}_{1:T})$ and the constraint set $\Omega_{1:T}$ are based on similar concepts in CGMs (Sheldon and Dietterich 2011).

Joint-Value Function: We next show that the joint-value for a given policy π also depends on the count vector $\boldsymbol{n}$. Thus, making counts as the sufficient statistic for planning in $\mathbb{C}$Dec-MDPs.

Theorem 3. *The joint-value function of a policy π over horizon H given by the expectation of total rewards of all the agents, $V(\pi) = \sum_m \sum_{T=1}^{H}\mathbb{E}[R_T^m]$, can be computed by the expectation over counts as:*

$$\sum_{\boldsymbol{n}\in\Omega_{1:H}} P(\boldsymbol{n};\pi)\left[\sum_{T=1}^{H}\sum_{i\in S, j\in A} n_t(i,j) R_T(i,j,n_t(i))\right] \tag{6}$$

Proof is in the appendix. Our goal in $\mathbb{C}$Dec-MDP is to compute the policy π that maximizes (6). Notice that the set of all the allowed counts $\Omega_{1:H}$ is combinatorially large, making the exact policy evaluation infeasible. Therefore, our approach would be to use a sampling based approach that can evaluate, and also optimize the policy π.

To optimize the policy π, one can translate the planning problem to that of inference in a mixture of dynamic Bayes nets (DBNs), similar to previous work (Kumar, Zilberstein, and Toussaint 2015), showing that likelihood maximization (LM) in such a mixture is equivalent to optimizing the policy π. The well known EM algorithm (Dempster, Laird, and Rubin 1977) and its monte-carlo variants (Vlassis and Toussaint 2009) can then be used for LM. However, upon implementation, we observed EM's convergence to be slow and to poor local optima. Empirically, in large population settings, EM updated the policy by very small amounts in each iteration, leading to slow convergence. Therefore, we next develop an EM variant called *Fictitious EM* (FEM) motivated by the concept of fictitious playing (FP) in congestion games (Meyers and Schulz 2012).

In fictitious playing, each individual agent m would run a policy optimizer, which is EM algorithm in our case, to maximize its own rewards given its local knowledge about the environment. In (Varakantham et al. 2012), such a fictitious play results in a policy update based on solving an MDP. However, such an MDP is based on the estimated mean of agent flow (or using the deterministic model in Fig.1(b)). In problems with tight transition dependence where the transition function ϕ_t is a nonlinear function of the agent count $n_t(i)$, such an expected flow based model is not sufficiently accurate. Empirically, we show that such an expectation-based FP approach of Varakantham et al. (2012), called FP-SAP, can become highly inaccurate in models with tight transition dependence among agents, and results in a poor policy.

4 Fictitious Play Based Policy Optimization

As the concept of fictitious play based EM (FEM) is based on optimizing a single agents's policy based on agent's local observations, we first need to compute the *individual* value function of an agent. Even such an individual value function is computationally challenging as it must take into accounts the effect of other agents summarized by counts $\boldsymbol{n}$, which is a combinatorial space. To make reasoning with counts tractable, we use several concepts based on monte-carlo sampling and the notion of *finite-partial exchangeability* from lifted inference (Niepert and Van den Broeck 2014).

The FEM algorithm's updates will require computing the individual value function $Q_t^m(i,j,n_t(i))$ for a fixed policy π, which is agent m's total expected reward from time step t with its observation as $(s_t^m = i, n_t(i), a_t^m = j)$.

$$Q_t^m(i,j,n_t(i)) = \mathbb{E}\left[\mathbb{I}\big(n_t'(i)=n_t(i), s_t^m=i, a_t^m=j\big)\sum_{T=t}^{H} R_T^m\right]$$

$$= \sum_{T=t}^{H}\sum_{\boldsymbol{n}'_{1:T}}\sum_{\boldsymbol{s}^m_{1:T},\boldsymbol{a}^m_{1:T}} \mathbb{I}\big(n_t'(i)=n_t(i)\big)\mathbb{I}_t^m(i,j) \times P(\boldsymbol{s}^m_{1:T},\boldsymbol{a}^m_{1:T},\boldsymbol{n}'_{1:T}) R_T\big(s_T^m, a_T^m, \boldsymbol{n}'_T(s_T^m)\big) \tag{7}$$

Notice that in the above expression we need to compute the probability $P(\boldsymbol{s}^m_{1:T},\boldsymbol{a}^m_{1:T},\boldsymbol{n}'_{1:T})$ which denotes the probability that the agent m follows the trajectory $(\boldsymbol{s}^m_{1:T},\boldsymbol{a}^m_{1:T})$ and the count vector is $\boldsymbol{n}'_{1:T}$. We next use results from lifted inference to compute this probability.

4.1 Exchangeability of joint-trajectories

We start by defining *full exchangeability* (Niepert and Van den Broeck 2014). A set of variables $\boldsymbol{X} = \{X_1,\ldots,X_k\}$ is fully exchangeable iff $P(X_1=x_1,\ldots,X_k=x_k)$ equals $P(X_1 = x_{\alpha(1)},\ldots,X_k = x_{\alpha(k)})$ for all permutations α of $\{1,\ldots,k\}$. E.g., a sequence of independent coin toss is fully exchangeable. Let $(\boldsymbol{s}_{1:T},\boldsymbol{a}_{1:T}) = \{(\boldsymbol{s}_{1:T},\boldsymbol{a}_{1:T})^m \;\forall m\}$ denote the T-step trajectories of all the agents. Clearly, $(\boldsymbol{s}_{1:T},\boldsymbol{a}_{1:T})$ is not fully exchangeable as an agent's next state depends on its previous state. A tractable generalization of full exchangeability is partial exchangeability (Diaconis and Freedman 1980a), which variables $(\boldsymbol{s}_{1:T},\boldsymbol{a}_{1:T})$ would satisfy.

Definition 1. *Let $\mathcal{D}_i$ be the domain of X_i, and let $\mathcal{T}$ be a finite set. A set of variables $\boldsymbol{X}$ is partially exchangeable w.r.t. the statistic $T : \mathcal{D}_1 \times \cdots \times \mathcal{D}_k \to \mathcal{T}$ if and only if:*

$$T(\boldsymbol{x}) = T(\boldsymbol{x}') \text{ implies } P(\boldsymbol{x}) = P(\boldsymbol{x}')$$

We next show the following for the $\mathbb{C}$Dec-MDP model.

Proposition 1. *The joint state-action trajectories of agents, $(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T})$, are partially exchangeable w.r.t. the count statistic $\boldsymbol{n}_{1:T} \in \Omega_{1:T}$.*

where $\Omega_{1:T}$ is the space of allowed counts satisfying constraints (4)-(5). This result follows directly from theorem 1. Next we use the exchangeability theorem that relates the joint-distribution $P(\boldsymbol{X})$ over variables $\boldsymbol{X}$ with the distribution over sufficient statistic.

Proposition 2. *The distribution $P(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T})$ is defined as:*

$$P(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T}) = \sum_{\boldsymbol{n}_{1:T} \in \Omega_{1:T}} P(\boldsymbol{n}_{1:T}) \frac{\mathbb{I}_{\boldsymbol{n}_{1:T}}(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T})}{|S_{\boldsymbol{n}_{1:T}}|}$$

where $\mathbb{I}_{\boldsymbol{n}_{1:T}}(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T})$ denotes if $(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T})$ is consistent with statistic $\boldsymbol{n}_{1:T}$; $S_{\boldsymbol{n}_{1:T}}$ is the set of all possible joint-trajectories $(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T})$ having sufficient statistic $\boldsymbol{n}_{1:T}$.

This result is a direct corollary of the exchangeability theorem in (Diaconis and Freedman 1980b; Niepert and Van den Broeck 2014). Notice that $|S_{\boldsymbol{n}_{1:T}}|$ equals to the function $h(\boldsymbol{n}_{1:T})$ (3). Let $\mathbb{I}_{\boldsymbol{s}_{1:T}\boldsymbol{a}_{1:T}}(\boldsymbol{s}^m_{1:T}, \boldsymbol{a}^m_{1:T})$ denote if agent m's trajectory $(\boldsymbol{s}^m_{1:T}, \boldsymbol{a}^m_{1:T})$ is consistent with the joint-trajectory $\boldsymbol{s}_{1:T}\boldsymbol{a}_{1:T}$. Using this result, the joint probability $P(\boldsymbol{s}^m_{1:T}, \boldsymbol{a}^m_{1:T}, \boldsymbol{n}_{1:T})$ is:

$$\sum_{\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T}} P(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T}) \mathbb{I}_{\boldsymbol{n}_{1:T}}(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T}) \mathbb{I}_{\boldsymbol{s}_{1:T}\boldsymbol{a}_{1:T}}(\boldsymbol{s}^m_{1:T}, \boldsymbol{a}^m_{1:T})$$

In the above expression, we can use proposition 2 to compute $P(\boldsymbol{s}_{1:T}, \boldsymbol{a}_{1:T})$. Upon further simplification (with proof in appendix), we get the following result:

Theorem 4. *The joint probability $P(s^m_{1:T}, a^m_{1:T}, \boldsymbol{n}_{1:T})$ is given by the following expression:*

$$P(\boldsymbol{n}_{1:T}) \frac{n_1(s^m_1)}{M} \left[\prod_{t=1}^{T-1} \frac{n_t(s^m_t, a^m_t, s^m_{t+1})}{n_t(s^m_t)} \right] \frac{n_t(s^m_T, a^m_T)}{n_t(s^m_T)}$$

4.2 Individual value function

Based on theorem 4, we now show how to compute the value function. Substituting the expression for joint probability in theorem 4 into value function in (7), we have $Q^m_t(i, j, n_t(i))$ defined as following:

$$\sum_{T=t}^{H} \sum_{\boldsymbol{n}'_{1:T}, \boldsymbol{s}^m_{1:T}, \boldsymbol{a}^m_{1:T}} R_T\big(s^m_T, a^m_T, n'_{s_T}(s^m_T)\big) \mathbb{I}\big(n'_t(i) = n_t(i)\big) \mathbb{I}^m_t(i, j)$$
$$\times P(\boldsymbol{n}'_{1:T}) \frac{n'_1(s^m_1)}{M} \left[\prod_{t'=1}^{T-1} \frac{n'_{t'}(s^m_{t'}, a^m_{t'}, s^m_{t'+1})}{n'_{t'}(s^m_{t'})} \right] \frac{n'_T(s^m_T, a^m_T)}{n'_T(s^m_T)}$$

Exactly computing above expression is intractable due to the combinatorial space of counts $\boldsymbol{n}$. Therefore, we consider the Monte-Carlo approximation of above expression by sampling a set of sample $\{\boldsymbol{n}^\xi_{1:T} \sim P(\boldsymbol{n}'_{1:T})\}$, and consider the average over K samples as:

$$Q^m_t(i, j, n_t(i)) \approx \frac{1}{K} \sum_{\xi=1}^{K} \sum_{T=t}^{H} \sum_{\boldsymbol{s}^m_{1:T}, \boldsymbol{a}^m_{1:T}} R_T\big(s^m_T, a^m_T, n^\xi_{s_T}(s^m_T)\big) \times$$
$$\mathbb{I}\big(n^\xi_t(i) = n_t(i)\big) \mathbb{I}^m_t(i, j) \frac{n^\xi_1(s^m_1)}{M} \frac{n^\xi_T(s^m_T, a^m_T)}{n^\xi_T(s^m_T)} \times$$
$$\left[\prod_{t'=1}^{T-1} \frac{n^\xi_{t'}(s^m_{t'}, a^m_{t'}, s^m_{t'+1})}{n^\xi_{t'}(s^m_{t'})} \right] \tag{8}$$

We next show a simplified expression (with proof in the appendix) for efficiently computing the above:

Theorem 5. *The approximate function $\hat{Q}^m_t(i, j, n_t(i))$ in* (8) *can be computed as:*

$$\hat{Q}^m_t(i, j, n_t(i)) = \frac{1}{K} \sum_{\xi=1}^{K} \frac{n^\xi_t(i, j)}{M} \mathbb{I}\big(n^\xi_t(i) = n_t(i)\big) V^\xi_t(i, j)$$

where the function $V^\xi_t(i, j)$ is given as:

$$R_t(i, j, n^\xi_t(i)) + \sum_{T=t+1}^{H} \sum_{\boldsymbol{s}^m_{t:T}, \boldsymbol{a}^m_{t:T}} \mathbb{I}^m_t(i, j) R_T\big(s^m_T, a^m_T, n^\xi_{s_T}(s^m_T)\big)$$
$$\times \left[\prod_{t'=1}^{T-1} \frac{n^\xi_{t'}(s^m_{t'}, a^m_{t'}, s^m_{t'+1})}{n^\xi_{t'}(s^m_{t'})} \right] \frac{n^\xi_T(s^m_T, a^m_T)}{n^\xi_T(s^m_T)} \tag{9}$$

In the above result, it appears that computing the function $V^\xi(\cdot)$ is intractable due to the summation over $(\boldsymbol{s}^m_{t:T}, \boldsymbol{a}^m_{t:T})$. Fortunately, we show that it can be computed efficiently using dynamic programming.

Theorem 6. *The function $V^\xi(\cdot)$ is equal to the value function of an MDP with state-space S^m, action space A^m, and transition and reward function defined as below for the given count vector sample $\boldsymbol{n}^\xi$:*

$$\phi^{\boldsymbol{n}^\xi}_t(i'|i, j) = \frac{n^\xi_t(i, j, i')}{n^\xi_t(i, j)};\ \pi^{\boldsymbol{n}^\xi}_t(j|i) = \frac{n^\xi_t(i, j)}{n^\xi_t(i)} \tag{10}$$

$$P^{\boldsymbol{n}^\xi}_1(i) = \frac{n^\xi_1(i)}{M};\ R^{\boldsymbol{n}^\xi}_t(i, j) = R_t(i, j, n^\xi_t(i)) \tag{11}$$

where $\phi^{\boldsymbol{n}^\xi}, R^{\boldsymbol{n}^\xi}_t$ are the transition and the reward function, $\pi^{\boldsymbol{n}^\xi}_t$ represents the fixed policy and P_1 is the initial state distribution. As a result of theorem 6, given a sample $\boldsymbol{n}^\xi$, we can define an MDP for an individual m, and compute $\hat{Q}$ function for this MDP by dynamic programming as follows:

$$V^\xi_H(i, j) = R_H(i, j, n^\xi_H(i)) \tag{12}$$

$$V^\xi_t(i, j) = R_t(i, j, n^\xi_t(i)) + \sum_{i' \in S, j' \in A} \phi^{\boldsymbol{n}^\xi}_t(i'|i, j) \pi^{\boldsymbol{n}^\xi}_{t+1}(j'|i') V^\xi_{t+1}(i', j')$$

$$Q^\xi_t(i, j, n^\xi_t(i)) = \frac{n^\xi_t(i, j)}{M} \times V^\xi_t(i, j) \tag{13}$$

$$\hat{Q}^m_t(i, j, n_t(i)) = \frac{1}{K} \sum_{\xi | n^\xi_t(i) = n_t(i)} Q^\xi_t(i, j, n^\xi_t(i)) \tag{14}$$

4.3 Sampling based fictitious EM

Based on the results developed in previous section, we now describe our FEM algorithm. We consider the fictitious play setting in which each agent would try to optimize its own reward given other agents' policy. We can model planning for each fictitious agent m as a POMDP planning problem where the state-space is the joint-space $\times_{m=1}^{M} S$, action space is agent m' action space A. The observation of the agent at time step t is its local state s_t^m and the counts $n_t(s_t^m)$ or $o_t^m = \langle s_t^m, n_t(s_t^m)\rangle$. The reward and the transition function of the agent are the same as in $\mathbb{C}$Dec-MDP model in section 2. As the individual planning problem is a POMDP, we can use the existing EM algorithm for POMDPs to optimize the policy π (Toussaint, Harmeling, and Storkey 2006). Notice that the state-space in this POMDP is the joint-state space of all the agents, which is combinatorial. Therefore, directly using the POMDP updates in (Toussaint, Harmeling, and Storkey 2006) is not feasible. To address the tractability issues, we showed in earlier sections how to compute the expectations $\hat{Q}$ in theorem 6. Briefly, outline of the EM algorithm is:

- E-step: Compute expectations $\hat{Q}_t^m(i, j, n_t(i))\forall i \in S, j \in A, n_t(i) \in [0, M]$ by sampling from the count distribution $P(\boldsymbol{n}_{1:H}; \pi)$ in (2) for a fixed policy π from previous iteration.
- M-step: Maximize the following w.r.t. $\pi^\star$ $\forall t, i, n_t(i)$:

  $\sum_{j\in A} \hat{Q}_t^m(i, j, n_t(i); \pi) \log \pi_t^\star(j|i, n_t(i))$

  subject to $\sum_j \pi_t^\star(j|i, n_t(i)) = 1$

The well-known solution of the above M-step is:

$$\log \pi_t^\star(j|i, n_t(i)) = \frac{1}{C}\hat{Q}_t^m(i, j, n_t(i); \pi) \qquad (15)$$

in which C is the normalization constant. Notice that in the most general form, we need a policy update for each count $n_t(i) \in [0, M]$, which may not be scalable if the agent population M is large. We can therefore use a piecewise policy by dividing the overall count range $[0, M]$ into multiple sub-ranges, and use the same policy for each sub-range. For our experiment, we use such a piecewise closed loop policy. The pseudo-code of EM algorithm is presented in algorithm 1. This EM algorithm is not guaranteed to monotonically increase the policy value as it is based on fictitious play and sampling based approximation. However, empirically, we observed that it often converged to good policies.

Algorithm 1: FEM: Collective Sampling based Fictitious EM

Algorithm `FEM()`
  Initialize: $\beta \leftarrow$learning rate
  $\hat{Q}_t(i, j, n_t(i)) \leftarrow 0\ \forall t, i \in S, j \in A, n_t(i) \in [0, M]$
  $\pi_t(j|i, n_t(i)) \leftarrow \frac{1}{|A|}\ \forall t, i, j, n_t(i)$
  repeat
    `E-step`
    `M-step`
  until ***convergence***
  return $\boldsymbol{\pi}$
Procedure `E-step`
  Sample $\boldsymbol{n}^\xi \sim P(\boldsymbol{n}_{1:T}; \pi)\ \forall \xi = 1$ to K
  for *each count sample* ξ **do**
    Compute $Q_t^\xi(i, j, n_t^\xi(i))\forall i, j, \xi$ using (12)-(13)
  $\hat{Q}_t(i, j, n_t(i)) \leftarrow (1-\beta)\hat{Q}_t(i, j, n_t(i)) +$
  $\beta(1/K)\sum_{\xi|n_t^\xi(s)=n_t(i)} Q_t^\xi(i, j, n_t^\xi(i)), \forall i, j, n_t(i)$
Procedure `M-step`
  $\pi_t(j|i, n_t(i)) \leftarrow \frac{1}{\sum_{j'} \hat{Q}_t(i,j',n_t(i))}\hat{Q}_t(i, j, n_t(i))$

5 Experiment

We compare our proposed sampling based fictitious EM approach with three other competing methods—Soft-Max Based Flow Update (SMFU), Fictitious Play for Symmetric Agent Populations (FP-SAP) from (Varakantham et al. 2012), and the MIP based solver in (Varakantham, Adulyasak, and Jaillet 2014). We test on synthetic instances modeling congestion aware robot navigation in a grid, and a real world dataset modeling a supply-demand matching problem for a fleet of taxis in a city. For EM, we compute both the closed loop and open loop policies. As previous approaches (FP-SAP, SMFU, MIP) are based on average flow approximation, they cannot compute closed loop policies. Each data point is an average of 10 instances. As our policy evaluation is based on sampling, we also report 95%-confidence intervals over 200 samples. For each approach, iteration limit was 500, with convergence occurring much earlier within a time limit of 0.5 hour; MIP had 2 hour limit.

Robots moving to a goal: In this setting, the task for a population of robots (=20) is to move from a initial location to a specific goal location in a grid. Each grid edge has a capacity (=4). When total number of agents simultaneously crossing an edge is less than its capacity, then each agent has a higher probability of moving to the next location (=0.8); this probability decreases sharply (=0.1) if total agents crossing the edge are more than the capacity. Each robot receives a reward 1 when in the goal state, otherwise the reward is zero. For closed loop EM, we use a piecewise policy with 5 pieces. These set of experiments are designed to test coordination among agents when any congestion leads to sharp decrease in movement probabilities.

Figure 3(a) shows the *normalized* solution quality of different approaches for varying grid sizes; for $n \times n$ grid, the plan horizon was $2n$ or the maximum manhattan distance. From this result, we can clearly observe that our EM approach is significantly better than previous approaches. Closed loop EM provides more than 20% higher quality solutions than SMFU consistently across all the grid sizes. We observed that SMFU was the best among the three previous approaches; the MIP solver could not scale to more than 7×7 grid. The open loop EM also provided about 5%-10% higher quality than SMFU. We highlight that SMFU can not optimize closed loop policies because of the deterministic approximation of agent flow. Figure 3(b) shows the effects of increasing plan horizon for a fix grid size of 5×5. We again observe similar result with EM variants providing better quality than previous approaches.

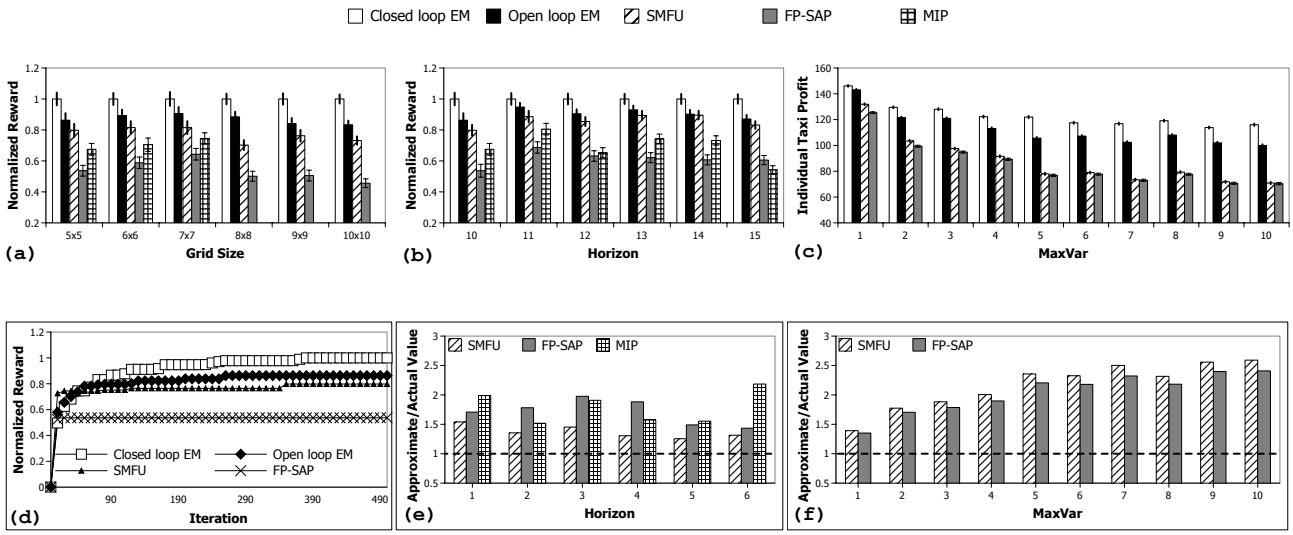

Figure 3: Experimental results comparing EM with SMFU, FP-SAP and the MIP solver

Figure 3(d) shows the convergence w.r.t. iterations of different approaches for 5×5 grid. SMFU and FP-FSAP converge to their local solutions quickly within 20 iterations, while the open loop EM and closed loop EM converge after more than 200 iterations. At the earlier iterations, open loop EM solutions have higher quality as training a closed loop policy requires more iterations. However, upon convergence closed loop policy is significant better than all the other approaches. This further highlights the advantage of optimizing closed loop policies in our model versus previous approaches which are unable to optimize closed loop policies.

Figure 3(e) shows the quality of approximate objective computed by SMFU, FP-SAP and MIP. Let obj be the objective computed by an algorithm. As obj is based on an average approximation, it is not the true evaluation of the underlying policy. We therefore compute the true evaluation $\mathrm{obj}^\star$ of the underlying policy using our sampling approach. In fig. 3(e), we show the ratio $\mathrm{obj}/\mathrm{obj}^\star$. If the average approximation employed in previous approaches was accurate, then this ratio should be close to 1. However, this is not the case in fig. 3(e). The average objective is highly inaccurate for all the instances. This further motivates $\mathbb{C}$Dec-MDPs to correctly model the behavior of an agent population.

Taxi Suppy-Demand Matching We next test on the large scale real-world taxi problem described in section 2, introduced previously in (Varakantham et al. 2012). The dataset contains the actual movement traces of 8000 taxis roaming in Singapore divided into 81 zones as shown in figure 2 for one year. For more details about problem settings (such as the transition function), we refer to (Varakantham et al. 2012). We have a planning horizon of 48 (half an hour intervals over 24 hrs). The goal is to compute policies for taxis to maximize the total profit of the fleet. The policy should balance the movement of taxis with the expected demand in each city zone at different time periods. If more taxis are present in a zone than the aggregate demand in that zone, then unhired taxis incur the cost when seeking passengers. Therefore, a good policy would direct taxis to different city zones to match demand with supply.

Previous work only considers a fixed expected taxi demand in each city zone. To make the problem more realistic, we address stochastic taxi demand. While sampling demand, we multiply the given expected demand in a zone z with $v_z \sim \hat{\mathcal{N}}(1, \sigma_z)$, where $\hat{\mathcal{N}}$ is a truncated normal distribution between $[0, 2]$. We generate several problem settings by sampling the variance σ_z uniformly from $[0, \mathrm{MaxVar}]$ and varying MaxVar from 1 to 10 as shown on the x-axis in fig. 3(c). Intuitively, with higher value of σ_z, multiplier v_z tends to follow a uniform distribution over $[0, 2]$; with lower value of σ_z, v_z is close to constant (≈ 1). Figure 3(c) shows the solution quality (average profit per taxi per day) of different approaches for varying MaxVar. We can clearly observe again that both closed loop EM and open loop EM significantly outperform other approaches (the MIP solver did not scale to these problems). Notably, when the MaxVar parameter increases, it increases the stochasticity in the problem. With increasing stochasticity, average approximation approaches (SMFU, FP-SAP) performed poorly against EM. This further highlights the weakness of average approximation. This insight is also confirmed by fig. 3(f) which shows the accuracy of the average approximation ($\mathrm{obj}/\mathrm{obj}^\star$). The accuracy of approximation decreases as MaxVar parameter increases from 1 to 10 on the x-axis.

6 Conclusion

In this work, we developed a new model for collective decision making by a group of agents. Our model can represent planning problems where the collective behavior of agents influences model dynamics. Such problems often arise in real world settings such as urban transportation. We established several basic properties of our model such as its count

based sufficient statistic and the value function. To compute the policy maximizing expected reward, we developed a novel sampling based model free approach combining fictitious play from game theory and the notion of finite exchangeability. Our approach is scalable to large real world problems such as taxi fleet optimization. It holds significant potential to apply multiagent planning to real world problems. Empirically, on synthetic instances of robots moving to a goal and a real world dataset modeling taxi supply-demand matching, our approach significantly outperformed previous best approaches.

7 Acknowledgments

This research project is supported by National Research Foundation Singapore under its Corp Lab @ University scheme and Fujitsu Limited. First author is also supported by A*STAR graduate scholarship.

References

Amato, C.; Konidaris, G.; Cruz, G.; Maynor, C. A.; How, J. P.; and Kaelbling, L. P. 2015. Planning for decentralized control of multiple robots under uncertainty. In *IEEE International Conference on Robotics and Automation, ICRA*, 1241–1248.

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

Becker, R.; Zilberstein, S.; and Lesser, V. 2004. Decentralized Markov decision processes with event-driven interactions. In *Proceedings of the 3rd International Conference on Autonomous Agents and Multiagent Systems*, 302–309.

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

Cheng, S., and Nguyen, T. D. 2011. Taxisim: A multiagent simulation platform for evaluating taxi fleet operations. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, 14–21.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical society, Series B* 39(1):1–38.

Diaconis, P., and Freedman, D. 1980a. De Finetti's generalizations of exchangeability. *Studies in Inductive Logic and Probability* 2:233–249.

Diaconis, P., and Freedman, D. 1980b. Finite exchangeable sequences. *The Annals of Probability* 8(4):745–764.

Hansen, E. A. 1998. Solving POMDPs by searching in policy space. In *International Conference on Uncertainty in Artificial Intelligence*, 211–219.

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

Kumar, A.; Zilberstein, S.; and Toussaint, M. 2015. Probabilistic inference techniques for scalable multiagent decision making. *Journal of Artificial Intelligence Research* 53(1):223–270.

Meyers, C. A., and Schulz, A. S. 2012. The complexity of congestion games. *Networks* 59:252–260.

Nair, R.; Varakantham, P.; Tambe, M.; and Yokoo, M. 2005. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *AAAI Conference on Artificial Intelligence*, 133–139.

Nguyen, D. T.; Kumar, A.; Lau, H. C.; and Sheldon, D. 2016. Approximate inference using DC programming for collective graphical models. In *International Conference on Artificial Intelligence and Statistics*, 685–693.

Niepert, M., and Van den Broeck, G. 2014. Tractability through exchangeability: A new perspective on efficient probabilistic inference. In *AAAI Conference on Artificial Intelligence*, 2467–2475.

Pajarinen, J.; Hottinen, A.; and Peltonen, J. 2014. Optimizing spatial and temporal reuse in wireless networks by decentralized partially observable Markov decision processes. *IEEE Trans. on Mobile Computing* 13(4):866–879.

Robbel, P.; Oliehoek, F. A.; and Kochenderfer, M. J. 2016. Exploiting anonymity in approximate linear programming: Scaling to large multiagent MDPs. In *AAAI Conference on Artificial Intelligence*, 2537–2543.

Sheldon, D. R., and Dietterich, T. G. 2011. Collective graphical models. In *Advances in Neural Information Processing Systems*, 1161–1169.

Sonu, E.; Chen, Y.; and Doshi, P. 2015. Individual planning in agent populations: Exploiting anonymity and frame-action hypergraphs. In *International Conference on Automated Planning and Scheduling*, 202–210.

Spaan, M. T. J., and Melo, F. S. 2008. Interaction-driven markov games for decentralized multiagent planning under uncertainty. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, 525–532.

Toussaint, M.; Harmeling, S.; and Storkey, A. 2006. Probabilistic inference for solving (PO)MDPs. Technical report, University of Edinburgh, Edinburgh, UK.

Varakantham, P.; Adulyasak, Y.; and Jaillet, P. 2014. Decentralized stochastic planning with anonymity in interactions. In *AAAI Conference on Artificial Intelligence*, 2505–2511.

Varakantham, P. R.; Cheng, S.-F.; Gordon, G.; and Ahmed, A. 2012. Decision support for agent populations in uncertain and congested environments. In *AAAI Conference on Artificial Intelligence*, 1471–1477.

Vlassis, N., and Toussaint, M. 2009. Model-free reinforcement learning as mixture learning. In *Annual International Conference on Machine Learning*, 1081–1088.

Witwicki, S. J., and Durfee, E. H. 2010. Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *International Conference on Automated Planning and Scheduling*, 185–192.