Dynamic Particle Allocation to Solve Interactive POMDP Models for Social Decision Making

Rohith Dwarakanath Vallam
IBM Research - India
Bengaluru, Karnataka
rovallam@in.ibm.com

Sarthak Ahuja
Carnegie Mellon University
Pittsburgh, Pennsylvania
sarthaka@cs.cmu.edu

Surya Shravan Kumar Sajja
IBM Research - India
Bengaluru, Karnataka
suryasku@in.ibm.com

Ritwik Chaudhuri
IBM Research - India
New Delhi
charitwi@in.ibm.com

Rakesh Pimplikar
IBM Research - India
Bengaluru, Karnataka
rakesh.pimplikar@in.ibm.com

Kushal Mukherjee
IBM Research - India
New Delhi
kushmukh@in.ibm.com

Ramasuri Narayanam
IBM Research - India
Bengaluru, Karnataka
ramasurn@in.ibm.com

Gyana Parija
IBM Research - India
New Delhi
gyana.parija@in.ibm.com

ABSTRACT

In social dilemma settings, such as repeated Public Goods Games (PGGs), humans often come across a dilemma whether to contribute or not based on past contributions from others. In such settings, the decision taken by an agent/human actually depends not only on the belief the agent has about other agents and the environment, but also on their beliefs about others' beliefs. To factor in these aspects, we propose a novel formulation of computational *theory of mind* (ToM) to model human behavior in a repeated PGG using *interactive partially observable Markov decision processes* (I-POMDPs). Interactive particle filter (IPF) is a well-known algorithm used to approximately solve I-POMDP models for the agents to find their optimal contributions. Number of particles assigned to an agent in IPF can be translated into time and computational resources. Solving I-POMDPs in a time-memory efficient manner even in the case of small state spaces is a largely intractable problem. Also, maintaining a fixed number of particles assigned to each agent, over time, will be highly inefficient in terms of resource utilization. To address this problem, we propose a dynamic particle allocation algorithm for different agents based on how well they could predict. We validate our proposed algorithm through real experiments involving human agents. Our results suggest that dynamic particle allocation based IPF for I-POMDPs is effective in modelling human behaviours in repeated social dilemma setting while utilizing computational resources in an effective manner.

KEYWORDS

AAMAS; ACM proceedings; LaTeX; text tagging

ACM Reference Format:
Rohith Dwarakanath Vallam, Sarthak Ahuja, Surya Shravan Kumar Sajja, Ritwik Chaudhuri, Rakesh Pimplikar, Kushal Mukherjee, Ramasuri Narayanam, and Gyana Parija. 2019. Dynamic Particle Allocation to Solve Interactive POMDP Models for Social Decision Making. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

1 INTRODUCTION

In multi-agent settings, social dilemmas are characterized by situations in which there is a conflict between individual and collective interest. Repeated public goods game (PGG) is a generic social dilemma setting where agents try to maximize their cumulative utility, as the game is played over several rounds. Agents participating in a PGG, observe the historical contribution of others to learn their behaviors and infer what others will play and then decide what to play in the next round for utility maximization. Khalvati et al. [16] hypothesize that the other agents in the game form the environment and employ *partially observable Markov decision processes* (POMDP) framework to model a human agent. However, every agent takes decisions and builds their own models about others in an autonomous fashion. Further, while modeling an agent, it is necessary to consider what models others may be using and infer their actions based on those individual models. This human ability to model others and understand the models that others may have about him/her is well studied and generally referred as *Theory of Mind* (ToM) [24]. In this paper, we employ *interactive partially observable Markov decision processes* (I-POMDPs) to model ToM ability of agents ([11],[12], [25]). I-POMDPs generalize POMDPs to a multi-agent setting, where each agent maintains beliefs about an interactive state consisting of both the physical states of the environment and the decision making models of the other agents. However, solving I-POMDP models for optimal policy of an individual agent in a multi-agent setting is computationally intractable [8]. Intractability arises from uncertainty in the environment, noisy/partial observations, stochastic state transitions, complexity of the belief representation and complexity of the space of policies. These issues have been addressed through generalization of both particle filter and value iteration based methods for I-POMDP framework in [8] and [9] respectively.

Interactive particle filter (IPF) algorithm [8] is generalization of the particle filter algorithm to a multi-agent scenario. It is an effective approximation technique for implementing I-POMDPs. It uses a belief vector of particles to maintain a belief over other agents' models. Number of particles used for an agent in IPF is directly proportional to computing resources and time of execution [13]. Thus, the number of particles is analogous to the cognitive cycles that a agent puts into modelling other agents. An important aspect of human behavior in a PGG is their differential focus on the actions played by others. That is, humans unequally distribute cognitive cycles to model other agents (and models that other agents use). This asymmetric allocation of time/focus by humans can be based on the accuracy with which they are able to predict the contributions of other agents. This helps humans with their limited cognitive ability to maintain focus on more unpredictable agents (agents who cannot be predicted with high accuracy). IPF works with the same number of particles for every agent in all the rounds, however, we propose a modified IPF algorithm with dynamic particle allocation (DPA) to capture differential focus exhibited in human behaviour. It assigns or allocates appropriate number of particles for belief over each agent's model so that the agent's action can be predicted with higher accuracy and lower computational load. This approach allows us to assign larger number of particles for beliefs over models of agents with complex behaviour, whose actions are not easy to predict. This helps an I-POMDP based agent to take better actions in a PGG with limited computational resources.

To the best of our knowledge, our paper proposes the first computational ToM based algorithm which implements DPA between different participating agents during belief update calculations. Further, we adapt this algorithm for PGGs and conduct novel empirical studies with real humans to generate real world data. We use this data to analyse I-POMDP modeling framework for human agents in a social dilemma and also evaluate DPA while solving I-POMDPs. Under some mild assumptions on the state space, our results show that I-POMDP approximates human contribution in a PGG more accurately than POMDP models. This reinforces the fact that when DPA is used, agents leverage a better belief about the other agents to make a more well-reasoned contribution. Further, our results show that I-POMDP with DPA learns other human models faster than I-POMDP without DPA, which would help I-POMDP with DPA achieve a lower prediction error early in a game. This is achieved by dynamically allocating more particles to agents with higher prediction errors.

2 RELATED WORK

Social dilemmas expose strains between group-rationality and self-rationality [27]. Typically, cooperation improves the outcomes for all. However, the lure of free riding and other such parasitic strategies threatens the stability of any cooperative move. From a game theoretic perspective, the Nash equilibrium strategy [17] is that no one cooperates. However, in experiments with human agents [31], it has been seen that for repeated social dilemmas, cooperation emerges. There have been a number of theories proposed to explain this behavior including reciprocity[19][20], evolutionary mechanisms [21], and reinforcement learning [5][4]. This paper leverages the IPOMDP framework to create theory-of-mind enabled agents and examines their behavior in a social dilemma embodied as a canonical PGG.

Theory of mind [24] has received a great deal of attention in the computer science community over the last few years. As we observe the behavior of other people, we naturally attribute to them beliefs, goals, and other latent mental states that we cannot directly observe. Theory of mind aims to represent the latent state maintained by another agent based upon the observable behavior of that agent. There has been a vast body of work ([1]) that is related to the theory of mind, especially in the domain of robotics where AI agents attempt to understand the intentions of an human that it is collaborating with [28], while making the robots more social. Recently, the notion of the theory of mind has been extended beyond human-agent interactions. Rabinowitz et al. [26] talks about, the theory of mind in multi-agent setting, where every agent attempts to model the behavior of other agents using meta-learning from observations of their behavior. They have also shown to model other agents that use either random, algorithmic or deep reinforcement learning based methods. Other papers such as [3][23][2] use a Bayesian approach to model the belief and desire dependent actions of other agents as a partially observable Markov decision process (POMDP). The joint belief state and reward state is then reconstructed via Bayesian inference conditions on the agent's behavior. Gmytrasiewicz and Doshi [14], Wunder et al. [32] and Panella and Gmytrasiewicz [22] extend POMDPs to the multi-agent setting where multiple agents attempt to construct a model the of others. The construct is called interactive POMDPs (IPOMDPs). The complexity lies in the fact that an agent i must model how another agent j models it, and so on and so forth. While the complexity of these approaches increased very rapidly with the depth (or the number of levels) of modeling abstractions, [8] describes a method to find approximate solutions using particle filters.

3 PROBLEM SETTING AND MODEL FORMULATION

3.1 The multi-round PGG

We consider a setting where N agents are playing a multi-round PGG over a finite time horizon T. Let $\mathbf{P} = \{1, \cdots, P\}$ be the set of discrete actions from which each agent i can choose from in every round of the PGG. The utility of agent i in a round t is given by $u_i^t(a_i^t, a_{-i}^t) = e_i^t - a_i^t + \frac{\beta}{N}\sum_{k=1}^{N} a_k^t$ where e_i^t is the endowment given to agent i at the beginning of the round t and a_i^t is the action of the agent i in round t and a_{-i}^t is the actions of the all agents except agent i in round t. In this paper, we use the notation $-i$ to denote all agents except agent i. According to [18] a single-shot (i.e., for a fixed t) PGG game with utility function u_i^t, satisfies the social dilemma constraints if $e_i^t = 1, \forall i$, $a_i^t \in [0, 1]$ and $1 < \beta < N$. Thus, all the discrete actions in $\mathbf{P}$ correspond to fractions in the interval $[0, 1]$. The total discounted utility of agent i is given by $\mathbf{U}_i^T = \sum_{t=1}^{T} \gamma^{t-1} u_k^t(a_i^t, a_{-i}^t)$ where $\gamma \in (0, 1]$ is the discount factor for the multi-round PGG. It is assumed that each agent i plays the multi-round PGG so as to optimize $\mathbf{U}_i^T$ and the parameter β remains constant for the entire game. Throughout this section, we use the standard convention of using the indices i and j to denote specific agents and k to denote a specific action. We also provide detailed description of our notation in the supplementary material.

3.2 I-POMDP formulation of the the multi-round PGG

We model each agent i in the multi-round PGG as an I-POMDP agent wherein the agent has to choose its actions from the set $\mathbf{P}$ in every round. A finitely nested I-POMDP model of an agent i defined as the following:

$$\text{I-POMDP}_{i,\ell} = \langle IS_{i,l}, A, T_i, \Omega_i, O_i, R_i \rangle$$

where $IS_{i,\ell}$ denotes the set of interactive states defined as $IS_{i,\ell} = S \times \Theta_{j,\ell-1}$, for a strategy vector $\ell \geq 1$. In this framework

- $IS_{i,0} = S$ is the set of states of the physical environment,
- $\Theta_{j,\ell-1}$ is a class of $(\ell-1)^{th}$ level intentional models of agent j defined as

$$\theta_{j,\ell-1} = \langle b_{j,\ell-1}, A, \Omega_j, T_j, O_j, R_j, OC_j \rangle$$

where $b_{j,\ell-1}$ is the agent j's belief nested to the level $(\ell-1)$ and OC_j is j's *optimality criterion.*

Rest of the notation is standard as in [6]. Rewriting $\theta_{j,\ell-1}$ as $\theta_{j,\ell-1} = \langle b_{j,\ell-1}, \hat{\theta}_j \rangle$ where $\hat{\theta}_j \in \widehat{\Theta}_j$ includes all the elements of the intentional model other than the belief and is called the agent's j frame. In our context, the *action set* A is the joint action set of all the agents in PGG and it can be defined as $A = \bigtimes_{i=1}^{N} A_i$, where $A_i = \mathbf{P}$ is action set of each agent i. In this paper, we assume that: *each agent has a subjective opinion about how the whole group behaves (acts) collectively.* Since, each agent models the aggregate group behaviour, we define $\Psi_G = (\psi_{G1}, \psi_{G2}, \cdots, \psi_{GP})$ as a strategy of the whole group in a PGG. If we let $\mathbf{a} = \{a_1, a_2, \ldots, a_N\}$[1] represent the discrete actions of all N agents in an arbitrary round of our PGG, then the assumption stated above results in

$$\psi_{Gk} = P(a_i = k \mid \Psi_G) \quad \forall \text{ agents } i \in 1, 2, \cdots, N$$

i.e., ψ_{Gk} is the probability that the whole group chooses k^{th} action (from the set $\mathbf{P}$) in that round. Note that $\sum_{k=1}^{P} \psi_{Gk} = 1$ and the parameter $\boldsymbol{\psi_G}$ is unknown to the agents. Under this assumption, each agent i with an I-POMDP model will estimate how agent j models the group strategy vector Ψ_G. In order to facilitate this, we assume that the *observation set* of any agent i is factorizable into the action set of the other agents' action sets i.e., $\Omega_i = \bigtimes_{k=1, k\neq i}^{N} A_i$. This formulation assumes that actions of all agents are observed accurately by any agent without any noise in the observation. Since, each a_i has a finite set of outcomes from $\mathbf{P}$, we assume that a_i drawn from a discrete distribution conditional upon Ψ_G, i.e., $a_i \mid \Psi_G \sim \textsc{Discrete}(\Psi_G)\ \forall i = 1, \ldots, N$. Now, we assume that the group strategy parameter Ψ_G is drawn from a Dirichlet distribution whose parameters are given by $\boldsymbol{\alpha_G} = (\alpha_{G1}, \cdots, \alpha_{GP})$, i.e., $\Psi_G \mid \boldsymbol{\alpha_G} \sim \textsc{Dirichlet}(\boldsymbol{\alpha_G})$. After collecting our observation data as $\mathbf{a} = \{a_1, a_2, \ldots, a_N\}$ from a round of the PGG, we have observations $\mathbf{E} = (E_1, \cdots, E_P)$ where E_k denotes the number of occurrences of k^{th} action in $\mathbf{a}$. Application of Bayes rule results in $\Psi_G \mid \mathbf{a}, \boldsymbol{\alpha_G} \sim \textsc{Dirichlet}(\mathbf{E} + \boldsymbol{\alpha_G})$. This property motivates us to use $\boldsymbol{\alpha_G}$ as the *physical state space* (S) of our I-POMDP model, instead of Ψ_G. Once the true $\boldsymbol{\alpha_G}$ is learnt, a reasonably approximate value of Ψ_G can be recovered (for example: by computing the expected value of Dirichlet distribution with parameters $\boldsymbol{\alpha_G}$).

The *transition probability function* $T_i : S \times A \times S \rightarrow [0, 1]$ describes results of agent i's actions and is defined as $T_i(s, \mathbf{a}, s') = Pr\left(s' = \boldsymbol{\alpha'_G} \;\middle|\; s = \boldsymbol{\alpha_G}, \mathbf{a}\right)$ where

$$Pr\left(s' = \boldsymbol{\alpha'_G} \mid s = \boldsymbol{\alpha_G}, \mathbf{a}\right) = \begin{cases} 1 & \text{if } \quad \boldsymbol{\alpha'_G} = \boldsymbol{\alpha_G} + \mathbf{E} \\ 0 & \text{otherwise.} \end{cases}$$

The *observation function* $O_i : \Omega_i \times S \times A \rightarrow [0, 1]$ specifies probabilities of observations given agent's actions and resulting states. In our context it is defined as

$$O_i(p, \mathbf{a}, s') = Pr(o_i = p | s' = \boldsymbol{\alpha'_G}, \mathbf{a}) = \frac{\alpha'_{Gp}}{\sum_{k=1}^{P} \alpha'_{Gk}}.$$

The *reward function* $R_i : S \times A \rightarrow \mathbb{R}$ represents agent i's preferences and in our context, it is defined as

$$R_i(s, a) = 1 - a_i + \frac{\beta}{N}\left(\sum_{k=1}^{N} a_k\right) \text{ where } 1 < \beta < N.$$

4 THE INTERACTIVE PARTICLE FILTER ALGORITHM WITH DYNAMIC FOCUS ON AGENTS

In this section, we propose a modified (resource-efficient) version of the IPF algorithm [8] to approximately solve I-POMDPs in the PGG framework presented earlier. However, our algorithm can be used for a more general multi-agent setting other than PGG. Our version of IPF builds on a plain version of the algorithm proposed in [8] and dynamically focuses more resources on highly unpredictable agents and lesser resources on the more predictable agents. This augments the decision making process by enabling a more accurate inference of the intentions of other agents leading to better quality of decisions. In order to validate the proposed algorithm, we further carry out detailed experiments applying this resource-efficient algorithm on the multi-round PGG model described in the previous section. In this section, we use the index k to denote an agent. Now, we explain the details of the wrapper algorithm being executed by an agent k which is being modelled as an I-POMDP. In this section, we use the notation from [8]. In this wrapper algorithm for agent k, a loop is run for each discrete time period (round) till the horizon of the game. The following sequence of three events, keep repeating themselves at every discrete time instant until the end of horizon:

(i) At time $t-1$, agent k computes the near-optimal action based on its current belief by calling ApproxPolicy function (please refer to Section 9 and Appendix B of [8] for the algorithm), and then executes this action. ApproxPolicy computes an approximately optimal finite horizon policy tree given an initial belief using value iteration when $\ell > 0$.

(ii) At time t, agent k receives an observation from the environment.

(iii) At time t, agent k computes the revised belief based on the action it took at time $t-1$ and the observation it received at time t by executing the interactive particle filter algorithm with dynamic focus on agents (Ipf-Dpa) described in Algorithm 1.

[1] Considering the arbitrary nature of the PGG round, we drop the index t

Algorithm 1 IPF-DPA for approximating the belief update.

1: **function** IPF-DPA($\tilde{b}^{t-1}_{k,l}$, a^{t-1}_k, o^t_k, ℓ, $v^{t-1}_{k,-k,l}$, M, a^{t-1}_{-k}, $upWtS$)
2: **returns** $\left(\tilde{b}^t_{k,\ell},\ v^t_{k,-k,\ell}\right)$
3: $\tilde{b}^{tmp}_{k,\ell} \leftarrow \varnothing$; $\tilde{b}^t_{k,\ell} \leftarrow \varnothing$; $totWt \leftarrow 0$; $Q \leftarrow |\tilde{b}^{t-1}_{k,\ell}|$
▷ $|\cdot|$ represents number of elements of a set or a vector
Computation of Agent-specific Particle Allocation Weights
4: **for all** $is^{(n),t-1}_k = \langle s^{(n),t-1}, \theta^{(n),t-1}_{-k}\rangle \in \tilde{b}^{t-1}_{k,\ell}$ **do**
5: $Pr\left(A_{-k} \mid \theta^{(n),t-1}_{-k}\right) \leftarrow$ APPROXPOLICY$\left(\theta^{(n),t-1}_{-k}, \ell-1\right)$
6: $\widehat{a}^{(n),t-1}_{-k,\ell} \sim Pr\left(A_{-k} \mid \theta^{(n),t-1}_{-k}\right)$
7: $\widehat{s}^{(n),t} \sim T_k\left(S^t \mid a^{t-1}_k, \widehat{a}^{(n),t-1}_{-k,\ell}, s^{(n),t-1}\right)$
8: **if** ($upWtS$ == TRUE) **then**
9: $v^t_{k,-k,l} \leftarrow$ RECOMPAWTS$\left(k, \{\widehat{a}^{(n),t-1}_{-k,\ell}\}^Q_{n=1}, v^{t-1}_{k,-k,\ell}, a^{t-1}_{-k}, Q\right)$
10: **for** $m \leftarrow 1$ to N **and** $m \neq k$ **do**
11: $totWt \leftarrow totWt + v^t_{k,m,\ell}$
12: $\mathbf{upWtV}_m \leftarrow$ TRUE
Importance Sampling
13: **for all** $is^{(n),t-1}_k = \langle s^{(n),t-1}, \theta^{(n),t-1}_{-k}\rangle \in \tilde{b}^{t-1}_{k,l}$ **do**
14: **for all** $o^t_{-k} \in \Omega_{-k}$ **do**
15: **if** ($\ell == 1$) **then**
16: **for** $m \leftarrow 1$ to N **and** $m \neq k$ **do**
17: $q \leftarrow$ ROUND$(Q \times (v^t_{k,m,l}/totWt))$
18: $\tilde{b}^{(n),t}_{m,0} \leftarrow$ LEVEL0BELIEFUPDATE$(\tilde{b}^{(n),t-1}_{m,0}, \widehat{a}^{t-1}_m, o^t_m, q)$
19: $\theta^{(n),t}_m \leftarrow \langle b^{(n),t}_{m,0}, \widehat{\theta}^{(n)}_m\rangle$
20: $\theta^{(n),t}_{-k} = (\theta^{(n),t}_m)^{m=N}_{m=1,m\neq k}$
21: $is^{(n),t}_k \leftarrow \langle \widehat{s}^{(n),t}, \theta^{(n),t}_{-k}\rangle$
22: **else**
23: **for** $m \leftarrow 1$ to N and $m \neq k$ **do**
24: $q \leftarrow$ ROUND$(Q \times (v^t_{k,m,l}/totWt))$
25: $\tilde{b}^{(n),t}_{m,l-1} \leftarrow$ IPF-DPA$(\tilde{b}^{(n),t-1}_{m,\ell-1}, \widehat{a}^{t-1}_m, o^t_m, \ell-1, v^{t-1}_{m,-m,\ell-1}, q, a^{t-1}_{-k}, \mathbf{upWtV}_m)$
26: $\mathbf{upWtV}_m \leftarrow$ FALSE
▷ $\mathbf{upWtV}_m$ is the m^{th} element of **upWtV** and it set to FALSE to avoid calling RECOMPAWTS multiple times for the same level ℓ in a round.
27: $\theta^{(n),t}_m = \langle b^{(n),t}_{m,l-1}, \widehat{\theta}^{(n)}_m\rangle$
28: $\theta^{(n),t}_{-k} = (\theta^{(n),t}_m)^{m=N}_{m=1,m\neq k}$
29: $is^{(n),t}_k \leftarrow \langle \widehat{s}^{(n),t}, \theta^{(n),t}_{-k}\rangle$
30: weight $is^{(n),t}_k : w^{(n)}_t \leftarrow O_{-k}(o^t_{-k} | s^{(n),t}, a^{t-1}_k, \widehat{a}^{t-1}_{-k})$
31: adjust weight: $w^{(n)}_t \leftarrow w^{(n)}_t \times O_k(o^t_k | s^{(n),t} a^{t-1}_k, \widehat{a}^{t-1}_{-k})$
32: $\tilde{b}^{tmp}_{k,l} \overset{\cup}{\leftarrow} (is^{(n),t}_k, w^{(n)}_t)$
33: Normalize all $w^{(n)}_t$ so that $\sum_{n=1}^N w^{(n)}_t = 1$
Selection
Resample M particles with replacement $\{is^{(n),t}_k, n = 1\cdots M\}$ from the set $\tilde{b}^{tmp}_{k,\ell}$ according to the importance weights.
34: $\tilde{b}^t_{k,l} \leftarrow \{is^{(n),t}_k, n = 1\cdots M\}$

Algorithm 2 Recomputation of weights based on comparison between ground truth and predictions made by each of the particles in the belief

1: **function** RECOMPAWTS(k, $\{\widehat{a}^{(n),t-1}_{-k,l}\}^Q_{n=1}$, $v^{t-1}_{k,-k,l}$, a^{t-1}_{-k}, Q)
2: **returns** $v^t_{k,-k,l}$
3: **for** $m \leftarrow 1$ to N and $m \neq k$ **do**
4: $\tau_m \leftarrow 0$
5: **for** $n \leftarrow 1$ to Q **do**
6: **for** $m \leftarrow 1$ to N and $m \neq k$ **do**
7: **if** $\widehat{a}^{(n),t-1}_{m,l} != a^{t-1}_m$ **then**
8: $\tau_m = \tau_m + 1$
▷ count number of incorrect predictions.
9: **for** $m \leftarrow 1$ to N and $m \neq k$ **do**
10: $v^t_{k,m,l} \leftarrow v^{t-1}_{k,m,l} \times e^{\eta\tau_m}$
▷ Exponentially Weighted ($\eta > 0$)

IPF-DPA requires an initial set of Q particles: $\tilde{b}^{t-1}_{k,\ell}$, that is approximately representative of the agent k's prior belief: $b^{t-1}_{k,\ell}$, along with the action at time $t-1$: a^{t-1}_k. We also need the current observation at time t: o^t_k with the level of belief nesting: $\ell > 0$. The n^{th} particle of the sample set $\tilde{b}^{t-1}_{k,\ell}$ is denoted by $is^{(n)}_k$, this particle represents the agent's possible interactive state, in which the other agents' belief may itself be a set of particles, i.e., $is^{(n)}_k = \langle s^{(n)}, \theta^{(n)}_{-k}\rangle$ where $\theta^{(n)}_{-k} = \langle \tilde{b}^{(n)}_{-k,\ell-1}, \widehat{\theta}^{(n)}_{-k}\rangle$. Note that $\tilde{b}^{(n)}_{k,0}$ is a probability distribution over the physical state space. IPF-DPA proceeds by propagating each particle forward in time. In order to perform the propagation, other agents' action must be known. We solve for other agents' model using APPROXPOLICY to find a distribution over its actions, from which its action is sampled (lines 5 – 6 in Algorithm 1). Then we propagate the physical state for each particle in time using the transition function of agent k in line 7.

In order to implement dynamic particle allocation, we maintain a set of values to keep track of how each agent is able to predict the actions of other agents. The variable $v^t_{k,m,\ell}$ defined for all agents $k, m \in \{1,\cdots,N\}$, for strategy level ℓ (initialized to 1) and time t: stores a numeric value, which is a measure of how well agent k is able to predict the actions of agent m when it is thinking at level ℓ at time t. It is necessary to update this variable for all agents in each round of the game. In the context of our PGG setting, it is assumed that other agents' actions a^{t-1}_{-k} are known accurately without any noise. Availability of the ground truth for observations allows us to evaluate the prediction accuracy of agents $-k$ by comparing the estimated action set $\{\widehat{a}^{(n),t-1}_{-k,\ell}\}^Q_{n=1}$ with a^{t-1}_{-k} using RECOMPAWTS in lines 8 – 9 of Algorithm 1. The function RECOMPAWTS (as described in Algorithm 2) calculates the values to be assigned to the variable $v^{t-1}_{k,-k,l}$. The weights represented by $v^{t-1}_{k,-k,l}$ are assigned by agent k to other agents $-k$ based on their unpredictability. From agent k's perspective: $v^{t-1}_{k,m,l}$ is an increasing function of the unpredictability of agent m. RECOMPAWTS function characterizes unpredictability of an agent in terms of the number of times a wrong prediction is made. For agent m, the value of this error count is iteratively

updated in τ_m. However, this approach of characterizing unpredictability can be performed through other metrics of calculating error like: root mean square error, absolute error, relative error etc. We carried out our experiments with both absolute error and the error count and obtained similar outcomes. We further use an exponential weighting scheme to update the weights of the agents in line 10 of Algorithm 2. Exponential weighting equips dynamic particle allocation with memory of errors from previous rounds and the normalization of $v^t_{k,-k,\ell}$ with $totWt$ (in lines 17 and 24 of Algorithm 1) accentuates the relative differences between prediction errors of different agents. The parameter η for the scheme is chosen empirically. In Algorithm 1, $upWtS$ ensures that weights get updated only once per agent per level per round.

Now we update each agent's model (line 14) based on other agents' possible observations. Because the model is intentional, we must update its belief state. If $\ell > 1$, updating the other agent's belief requires recursively invoking the IPF-DPA for performing its belief update (lines 23 – 29). This recursion in depth of the belief nesting terminates when the level of nesting becomes one, and a LEVEL0BELIEFUPDATE function is executed (lines 16 – 21). LEVEL0BELIEFUPDATE function is described in Figure 5 of [8]. The function LEVEL0BELIEFUPDATE returns an updated belief based on the transition function while considering other agents' actions as noise. However, we should note that the revised belief returned by LEVEL0BELIEFUPDATE algorithm consists of q number of particles for the belief based on the parameter q passed to the algorithm. To ensure that the output belief consists of q particles, we append the following sequence of actions at the end of LEVEL0BELIEFUPDATE.

if $(q == |\tilde{b}^t_{k,0}|)$ **then** done.
if $(q < |\tilde{b}^t_{k,0}|)$
remove $|\tilde{b}^t_{k,0}| - q$ particles uniformly at random from $\tilde{b}^t_{k,0}$.
if $(q > |\tilde{b}^t_{k,0}|)$
add $|\tilde{b}^t_{k,0}| - q$ particles uniformly at random through resampling with replacement from $\tilde{b}^t_{k,0}$.

Parameter q represents the appropriate number of particles needed by agent k to store the belief about agent m and this is determined by the current weights of the particles represented by $v^{t-1}_{k,m,l}$. This parameter is calculated in lines 17 and 24 in Algorithm 1. Parameter q is also used in the recursive call to IPF-DPA in line 25. IPF-DPA uses this parameter to update the variable M, which is used to dynamically manage the number particles assigned to an agent. The nested **for loops** in lines 13 and 14, generate $|\Omega_{-k}|Q$ appropriately weighted particles, but we resample M particles out of these (line 34 of Algorithm 1), using an unbiased resampling scheme. This feature enables efficient use of computational resources (which is analogous to the number of particles in our setup) by allocating more particles for unpredictable agents and fewer for the more predictable agents. We note that the exponential weighting scheme for adapting number of particles is applicable for each agent for each level in the nested belief hierarchy and the updation of weighting scheme is processed in a recursive manner.

Particle filter for a single agent performs sampling in proportion to likelihood of perceiving the observation from the state that the particle represents. This allows particle filters to focus the computational resources on regions with high likelihood. For IPF and IPF-DPA (multi-agent setting), in addition to using the agent's own observation for weighting, the other agent's observations also participate in the weighting process (lines 30 – 31 of Algorithm 1). This step should not confused with our dynamic particle allocation based on unpredictability of agents.

For a PGG setting, availability of ground truth about other's actions can be used to further improve IPF-DPA in two ways.

(i) Perform propagation of the physical state space using the ground truth a^{t-1}_{-k} instead of $\hat{a}^{(n),t-1}_{-k,\ell}$ in line 7. This approach will further improve the accuracy of IPF-DPA.
(ii) In the PGG setting, the observations $o^t_k = a^{t-1}_{-k}$ and $o^t_{-k} = \left\{o^t_j\right\}^N_{j=1,j\neq k}$ where $o^t_j = a^{t-1}_{-j}$. Availability of the exact observation set for all other agents can be used to replace the set of possible observations in the **for loop** in line 14.

In order to simplify the comparison between IPF and IPF-DPA and highlight the dynamic particle allocation aspect of our paper, we do not include the ideas presented above in Algorithm 1. IPF-DPA can be easily generalized for a different multi-agent setting other than PGG where a^{t-1}_{-k} may not be observable and hence unavailable as an input for the RECOMPAWTS function. For such scenarios, RECOMPAWTS function can be generalized to evaluate the prediction accuracy of an agent by comparing observation o^t_k with the expected value of the observation obtained from the probability distribution over Ω_k, conditional upon given agent's actions and resulting states. This probability distribution can be expressed as $Pr\left(\Omega^t_k \,\middle|\, \hat{s}^{(n),t}, a^{t-1}_k\right)$, where $\hat{s}^{(n),t}$ can be evaluated as shown in line 7 of Algorithm 1.

5 EXPERIMENTS

We conducted multiple 2-agent PGG games and 4-agent PGG games with humans to gather data to verify our PGG formulation and the algorithm proposed in the previous section. Apart from analysing the data gathered from PGG games involving humans, we also perform several numerical simulations to evaluate and analyse our model and algorithm. We use human-generated PGG data to perform two comparisons:

Comparison (a): Initially, we compare I-POMDPs (with $\ell = 1$) and POMDPs (equivalent to I-POMDPs with $\ell = 0$) for their capability to predict human contribution in a PGG. We do so by replacing one agent in a PGG involving humans with an I-POMDP (and POMDP) model. This I-POMDP (and POMDP) model will predict contributions from the other agents by observing their past behaviour. We use IPF algorithm to solve I-POMDPs and partially observable Monte Carlo planning (POMCP) algorithm [29] to solve POMDPs. We perform this comparison for both 2-agent and 4-agent PGGs. We also compare I-POMDPs and POMDPs by pitting them against each other in a 2-agent PGG.

Comparison (b): Then, we compare IPF and IPF-DPA for their capability to efficiently solve I-POMDPs and accurately predict human contribution when an I-POMDP replaces a human agent in the 4-agent PGG game.

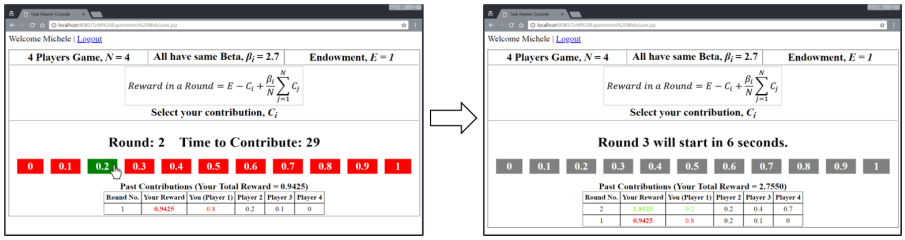

Figure 1: *UI used for conducting PGG games with human participation*

2 agent games (I-POMDP agent)	0.417
4 agent games (I-POMDP agent)	0.419
2 agent games (POMDP agent)	0.44666
4 agent games (POMDP agent)	0.4501

Figure 2: *Average RMSE of I-POMDP and POMDP agents for predicting human actions.*

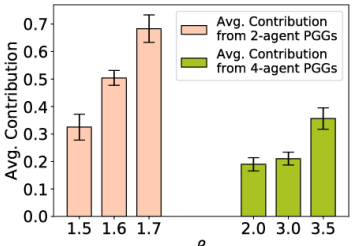

Figure 3

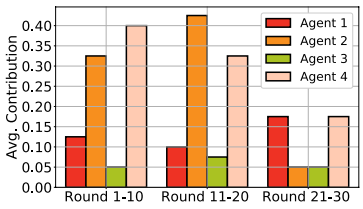

Figure 4

Avg. contributions of agents w.r.t. β and avg. contributions of agents over 30 rounds in a 4-agent PGG with $\beta = 3$.

Now, we describe the process of gathering human-generated data for PGG and examine the quality of the data to verify predictable trends. Then, these datasets are used for comparisons (a) and (b). We provide more detailed descriptions of our experiments, parameters chosen for solving I-POMDPs and detailed results in the supplementary material.

5.1 Process of gathering human-generated data and the quality of the gathered data

Five 2-agent PGGs were conducted among 10 human agents. The datasets gathered from 2-agent PGG games were used for comparison (a). For every game, 2 human agents were randomly sampled from 10 human agents without repetition to avoid bias. Each human agent was assigned a unique ID which was kept private from other agents. Human agents took part in PGGs remotely. Each human agent was given 1 unit of the endowment at the beginning of each round and then they were asked to contribute a portion of their endowment in that round. If N is the total number of agents, then β/N is called the marginal per capita return (MPCR). For the 2-agent scenario, we conducted the PGG game for $\beta = \{1.5, 1.6, 1.7\}$. Values of β are chosen such that they preserve social dilemma. Every game lasted for 30 rounds to enable human agents understand the dynamics of the game. To further assist agents in understanding the dynamics of their game, they were informed about their utility function, the value of β, previous rewards and previous contributions from others through an user interface (UI). Each round of game was 30 seconds long to let human agents think and decide upon their contribution in each round based on past contributions from others. We also conducted three 4-agent PGG games with 12 human agents. The data gathered from 4-agent PGG games was used for comparisons (a) and (b) as described before. We followed the same procedure as we followed in conducting 4-agent games with slightly different values of $\beta = \{2, 3, 3.5\}$. For all simulations and experiments in this paper, we fix the discount factor $\gamma = 0.99$. Figure 1 gives a snapshot of the UI that each agent used in the game. Human-generated data from both 2-agent and 4-agent games show predictable trends in terms of their dependence on β. Higher value of β implies higher marginal per capita returns with lower cost of cooperation, thus agents are expected to contribute more. For relatively smaller β, the human agents tend to not contribute or contribute less. These trends can be observed in Figure 3, where average contribution of all agents in a PGG increases with increasing values of β. Free-riding is another common behavior exhibited in PGG games with higher number of agents (decrease in MPCR) [15]. Human-generated data from 4-agent games with $\beta = \{3, 3.5\}$ exhibited this behavior. We plot the avg. contribution of each agent for every ten rounds of the PGG ($\beta = 3$) in Figure 4. It can be observed that Agent 3 preferred not to contribute much in most of the rounds and received as much reward as possible by leveraging the public contribution of high contributing peers such as Agent 2 and Agent 4.

5.2 Analysis of data gathered from human agents for I-POMDP vs. POMDP

For comparison (a) mentioned above, we consider data from both 2-agent and 4-agent PGG games. In each game, we replace one agent with an I-POMDP model and the remaining are real human agents along with their actions in the 30 rounds of the PGG concluded before. I-POMDP agent takes new actions in each round by predicting the possible action of the human agent(s) based on the action taken by the human agent(s) in the past rounds. For the I-POMDP agent, we compute the Mean Square Error (MSE) between the predictions it makes and the actual values of the contribution levels of the human agent(s) in each round. The same experiment is carried out with a POMDP replacing one human agent. Similarly, we iterate over all the agents involved in the PGG by replacing one agent at a time with an I-POMDP (and POMDP) and perform the

same experiment. We repeat these experiments for all five humans who played 2-agent PGG games to get 5 x 2 = 10 experiments for each of I-POMDP and POMDP agents. We perform the same experiments for each of the three 4-agent PGG games where three agents are human agents from the PGGs conducted before, while the fourth agent is replaced by an I-POMDP (and POMDP) in the experiment. We repeat these experiments for all four humans who played 4-agent PGGs to get 4 x 3 = 12 experiments for each of I-POMDP and POMDP agents.

We have provided the average RMSE value of both I-POMDP and POMDP agents in predicting the real human actions in Figure 2. For 2 agent and 4 agent categories of games, the average is taken over all the games played in a category and over all the agents in every game. It is clear that I-POMDP is more accurate than POMDP in predicting human actions in PGGs. The reason for I-POMDP agent predicting the actions of the other agents more accurately is due to the fact that I-POMDP agent considers the other agents to be rational agents who are trying to maximize its rewards in every round based on the past actions taken in the game. On the contrary, in the case of the POMDP agent, it approximates other agent's actions as noise in the environment. Hence for the I-POMDP agent, a strategy level one ($\ell = 1$) belief makes it able to make better predictions of the contributions of other agents.

5.3 Analysis of data gathered from human agents for IPF vs. IPF-DpA

For comparison (b), we conduct a similar experiment by letting an I-POMDP agent replace a human agent in a 4-agent PGG game. I-POMDP observes and predicts actions of three other humans over 30 rounds in those games on behalf of the replaced agent. Here, we approximately solve for the predictions of I-POMDP using IPF and IPF-DPA as described in Algorithm 1. Figure 5 compares the improvement in running MSE achieved with IPF-DPA agent against that of a plain IPF after every round of a game, averaged over all the 4-agent games. Figure 6 compares MSE of predictions from IPF-DPA against the predictions from a plain IPF averaged over all the 4-agent games using a bar graph (representing the mean MSE) with error bars (representing standard deviation in MSE). In Figure 5, we use the below measure for calculating the relative improvement in MSE due to IPF-DPA.

$$\textit{Percentage MSE Improvement due to } \textsc{Ipf-Dpa} = \frac{|\textit{Running MSE of } \textsc{Ipf-Dpa} - \textit{Running MSE of } \textsc{Ipf}|}{\textit{Running MSE of } \textsc{Ipf}} \times 100$$

It can be observed that IPF-DPA agent achieves significant gains in MSE reduction especially in early stages in the game when compared to an IPF agent. Thus, we can infer that IPF-DPA learns other human models faster than plain IPF. However, these results are based on empirical studies and may vary depending on the complexity of the strategies employed by the human agent. IPF-DPA can be used to characterize the heterogeneity and complexity of human strategies in real world PGG settings by observing particle allocation to different agents over time. Figure 7 shows the number of particles allocated over 30 rounds for predicting the contribution levels for agents 2, 3 and 4 (top graph) and the running τ (cumulative average of number of wrong predictions by each agent) values for agents 2, 3 and 4 (bottom graph). Agent 1 (I-POMDP agent) predicts contributions from other agents using IPF-DPA. It can be observed that Agent 2's running τ decreases faster than Agent 3 and Agent 4, hence, a smaller number of particles are assigned for Agent 2. On the other hand, Agent 3 and Agent 4's running τ decreases slowly and hence more particles are assigned to Agent 3 and Agent 4 for generating better predictions. It should be noted that any attempt to decrease the particle allocation for an agent necessarily results in the increase in the allocation for some other agents with an equal or smaller allocation.

6 CONCLUSION AND FUTURE WORK

In this paper, we studied the complex problem of social decision making in multi-agent setting. In this context, we used the popular construct of social dilemma namely the public goods game to study the decision making task. We adopted the computational ToM model of I-POMDPs as our modeling paradigm and proposed a resource-efficient interactive particle-filter based algorithm for approximately solving the I-POMDPs which dynamically focuses more resources on highly unpredictable agents and lesser resources on the more predictable agents. The algorithm proposed in the paper is generic and is applicable to multiple domains where appropriate IPOMDP can be formulated. Also, the model proposed in this paper can analogously be scaled as proposed in some recent work ([30]) where the authors were able to exactly solve IPOMDPs for up to 2000 agents in less than 6 hours. In principle, we can adapt such approximations in our work and achieve similar scalability as our work is also based on the belief update of IPOMDPs. Further, we have implemented a parallelized version of IPF and IPF-DPA algorithms which basically helps us to do the computation of every particle in parallel. We conducted several real-world experiments and used that data to (a) illustrate the effectiveness of I-POMDP models over POMDP models in predicting human contributions in a PGG and (b) evaluate the effectiveness of dynamic particle allocation when compared with a static particle allocation, while solving an I-POMDP. For both (a) and (b), we show improvement in terms of prediction accuracy, when compared with conventional practices. Both (a) and (b) were carried out under mild assumptions on the state space and the results from (a) and (b) show that our assumptions are useful in obtaining valid approximations for I-POMDP solutions. Future research could examine dynamic focus based interactive Point-Based Value Iteration [10] to solve I-POMDPs.

APPENDIX

Note on Algorithm 1

In order to perform other agent's action must be known. This is obtained by solving the other agent's model (using the algorithm APPROXPOLICY) to get its policy, and using its belief (contained in the particle) to find a distribution over its actions (line 3 in Fig. 1). The algorithm is provided in the reference([7],[8]). We also use the LEVEL0BELIEFUPDATE algorithm described in [8]. We modify this algorithm based on the number of particles needed (described in the paper).

Experimental Parameters

All the experiments were run on 2 high-end linux servers. The configuration of each server is :Intel(R) Xeon(R) CPU E5-2683 v4 @

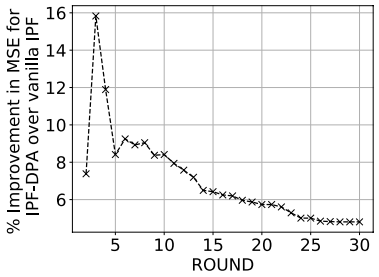

Figure 5 *Improvement in running MSE achieved with IPF-DPA agent against that of a plain IPF.*

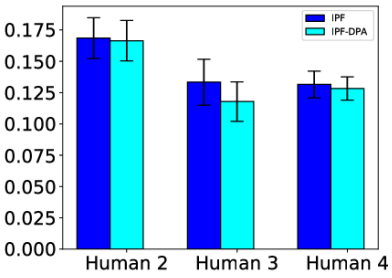

Figure 6 *Bar chart based MSE comparisons for each human agent's predictions when Human 1 is replaced by an I-POMDP and $\beta = 2$.*

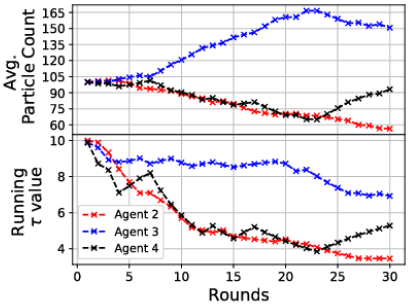

Figure 7 *Average particle allocation to agents and agent-specific running τ values for $\beta = 2$.*

IS_j^l	Set of interactive states of the IPOMDP
$IS_i^0 = S$	Set of physical states of the IPOMDP
$M_{j,l-1}$	Set of possible models of agent j
$\Theta_{j,l-1}$	Set of $l-1$ intentional models of agent j
Ω_j	Set of observations of agent j
$\theta_{j,l-1}$	Element of $\Theta_{j,l-1}$ that comprises of $< b_{j,l-1}, A, \Omega_j, T_j, O_j, R_j, OC_j >$
$\hat{\theta}_j$	Denotes all elements of $\theta_{j,l-1}$ except the belief $b_{j,l-1}$
$b_j^{l-1} \in \Delta(IS_j^{l-1})$	Agent j's belief nested to level $l-1$
$\tilde{b}_{k,l}^{t-1}$	Agent k's belief (represented as a set of particles) at level l in round $t-1$
a_k^{t-1}	Agent k's action in round $t-1$
o_k^t	Agent k's observation at the beginning of round t
$is_k^{(n),t-1}$	a particular interactive state represented by the particle n
$s^{(n),t-1}$	the physical state at time (or round) $t-1$ represented by the nth particle
$\theta_{-k}^{(n),t-1}$	the model of other agents according to agent k represented in the nth particle at time $t-1$
$v_{k,-k,l}^{t-1}$	Measure of how well agent k is able to predict the actions of agent $-k$ (all agents except k) when it is thinking at level l at time t.

Table 1 Important Notations

2.10GHz, 56 cores, 100 GB HDD, 100GB RAM. The code for IPF and IPF-DPA algorithms is written in Julia programming language. We use its libraries for POMDP solvers (https://github.com/JuliaPOMDP). In particular we use the Partially Observable Monte Carlo Planning (POMCP) solver (https://github.com/JuliaPOMDP/BasicPOMCP.jl) for solving the base POMDP model in the running of the IPF algorithm. NUM-ACTIONS = 5, NUM-AGENTS = 4, ROUNDS = 30, HORIZON = 3, NUM-PARTICLES-IPOMDP = 10, DISCOUNT-FACTOR = 0.9, NUM-PARTICLES-POMDP = 100.

Design of Empirical Experiments

Experimental set up for PGG with human subjects

- Volunteers for the experiments = 20
- Number of games needed is 10 for each 2-player, 4-player PGGs.
- Gathering a pool of subjects:
  - Step 1: People answered an open call to take part in our PGG experiment voluntarily.
  - Step 2: Let N be the number of subjects who enrolled for taking part. They will be subdivided in groups of 2 (or 4) for simultaneous PGGs.
- Dividing N subjects in groups of 2:
  - Step 1: Assign a random integer generated between 1 and N both inclusive to every subject (the ID of the subject).
  - Step 2: Generate 2 different random numbers without replacement between 1 and N. which corresponds to a group of 2 for the experiment.
  - Step 3: Re-assign a random integer generated between 1 and N-2 both inclusive to every remaining subject (the new IDs of the subjects).
  - Repeat step 2 and 3 until all the N – (N mod 2) subjects have been assigned in [N/2] PGG games.
- Informing each individual separately about the procedure of the game:
  - Step 1: A slide deck will be shared with the subjects 20 minutes prior to the beginning of the experiment which contains the game set up, the equation for PGG and procedure to compute the utility at the end of the round, task the subjects are expected to complete in each round.
  - Step 2: Subjects will be given 10 minutes to read the slide deck and will be given 10 more minutes if they have any questions.
- Gaming set up:
  - Step 1: The procedure below will be repeated for all groups.
  - Step 2: For every game assign a β randomly drawn from the set of values from the region of social dilemma. The game will go on for 30 rounds.
  - Step 3: At the game beginning, each subject will be given his/her ID number, and they will be informed how many competing players they are playing against and β value they will use in the utility function.
  - Step 4: At the beginning of each round, the following will be shown to each subject in the group: frequency distribution of contribution levels of the group, the per-round utility obtained, the aggregated utility, the per-round contribution amount of subject.
  - Step 5: At the beginning of the round, each subject will be given 30 seconds to enter their contribution amount for the next round.
  - Step 6: In case in if a subject fails to enter the contribution amount in step 3, subject's last entered contribution amount will be considered as the contribution level for the current round.
- Data to collected by the moderators from the game:
  - Step 1: Collect IDs, contribution levels of each subject for every round, utility amount of each subject for every round.
- Break between game 1 and the rest of the games:
  - Step 1: Moderators will take a 5 minute break after Game 1 to verify the quality of the data obtained are good for analysis.
- Completion of the rest of the games:
  - Step 1: Repeat the rest of the games simultaneously for all groups.
- Repetition of the games:
  - Step 1: For each group repeat the games 2 or 3 times, but assign a different ID to every subject in the same group for different games to remove any repetition bias from participating subjects.
- Games with 4 (or N/2) subjects:
  - Step 1a: A game will be initiated where 4 subject groups will take part in the game. The game will be repeated for 30 rounds following the procedures mentioned above. Different 4 subject groups can take part in simultaneous games. Repeat the games 2 times.

REFERENCES

[1] Stefano V Albrecht and Peter Stone. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence* 258 (2018), 66–95.

[2] Baker, Chris L, Jara-Ettinger, Julian, Saxe, Rebecca, and Joshua B. Tenenbaum. 2017. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour* (2017).

[3] Chris L. Baker, Rebecca Saxe, and Joshua B. Tenenbaum. 2011. Bayesian Theory of Mind: Modeling Joint Belief-Desire Attribution. In *CogSci*.

[4] Daan Bloembergen, Karl Tuyls, Daniel Hennes, and Michael Kaisers. 2015. Evolutionary Dynamics of Multi-agent Learning: A Survey. *J. Artif. Int. Res.* 53, 1 (May 2015), 659–697. http://dl.acm.org/citation.cfm?id=2831071.2831085

[5] Enrique Munoz de Cote, Alessandro Lazaric, and Marcello Restelli. 2006. Learning to Cooperate in Multi-agent Social Dilemmas. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '06)*. ACM, New York, NY, USA, 783–785. https://doi.org/10.1145/1160633.1160770

[6] Prashant Doshi and Piotr J. Gmytrasiewicz. 2005. Approximating State Estimation in Multiagent Settings Using Particle Filters. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '05)*. ACM, New York, NY, USA, 320–327. https://doi.org/10.1145/1082473.1082522

[7] Prashant Doshi and Piotr J Gmytrasiewicz. 2005. A particle filtering based approach to approximating interactive pomdps. In *AAAI*. 969–974.

[8] Prashant Doshi and Piotr J. Gmytrasiewicz. 2009. Monte Carlo Sampling Methods for Approximating Interactive POMDPs. *J. Artif. Int. Res.* 34, 1 (March 2009), 297–337. http://dl.acm.org/citation.cfm?id=1622716.1622725

[9] Prashant Doshi and Dennis Perez. 2008. Generalized Point Based Value Iteration for Interactive POMDPs.

[10] Prashant Doshi and Dennis Perez. 2008. Generalized Point Based Value Iteration for Interactive POMDPs. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 1 (AAAI'08)*. AAAI Press, 63–68. http://dl.acm.org/citation.cfm?id=1619995.1620007

[11] Prashant Doshi, Xia Qu, Adam Goodie, and Diana Young. 2010. Modeling recursive reasoning by humans using empirically informed interactive POMDPs. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 1223–1230.

[12] Prashant Doshi, Xia Qu, Adam S Goodie, and Diana L Young. 2012. Modeling human recursive reasoning using empirically informed interactive partially observable markov decision processes. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 42, 6 (2012), 1529–1542.

[13] Víctor Elvira, Joaquín Míguez, and Petar M Djurić. 2017. Adapting the number of particles in sequential Monte Carlo methods through an online scheme for convergence assessment. *IEEE Transactions on Signal Processing* 65, 7 (2017), 1781–1794.

[14] Piotr J. Gmytrasiewicz and Prashant Doshi. 2005. A Framework for Sequential Planning in Multi-agent Settings. *J. Artif. Int. Res.* 24, 1 (July 2005), 49–79. http://dl.acm.org/citation.cfm?id=1622519.1622521

[15] R Mark Isaac and James M Walker. 1988. Group size effects in public goods provision: The voluntary contributions mechanism. *The Quarterly Journal of Economics* 103, 1 (1988), 179–199.

[16] Koosha Khalvati, Seongmin A. Park, Jean-Claude Dreher, and Rajesh P. N. Rao. 2016. A Probabilistic Model of Social Decision Making Based on Reward Maximization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*. Curran Associates Inc., USA, 2909–2917. http://dl.acm.org/citation.cfm?id=3157382.3157423

[17] Paul AM Van Lange, Jeff Joireman, Craig D Parks, and Eric Van Dijk. 2013. The psychology of social dilemmas: A review. *Organizational Behavior and Human Decision Processes* 120, 2 (2013), 125–141.

[18] J. Ledyard. 1995. *Public Goods: A survey of experimental research*. In Handbook of Experimental Economics, J.H. Hagel and A.E. Roth, Eds. Princeton University Press, Princeton, NJ, USA.

[19] et al. Martin Nowak, Karl Sigmund. 1993. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game . *Nature* 364, 6432 (1993), 56–58.

[20] Martin Nowak and Karl Sigmund. 1998. Evolution of indirect reciprocity by image scoring. *Nature* 393, 6685 (1998), 573–577.

[21] Martin A. Nowak, Corina E. Tarnita, and Edward O. Wilson. 2010. The evolution of eusociality. *Nature* 466 (2010).

[22] Alessandro Panella and Piotr J. Gmytrasiewicz. 2015. Nonparametric Bayesian Learning of Other Agents? Policies in Interactive POMDPs. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1875–1876. http://dl.acm.org/citation.cfm?id=2772879.2773481

[23] Jan Pöppel and Stefan Kopp. 2018. Satisficing Models of Bayesian Theory of Mind for Explaining Behavior of Differently Uncertain Agents: Socially Interactive Agents Track. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 470–478.

[24] David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences* 1, 4 (1978), 515–526. https://doi.org/10.1017/S0140525X00076512

[25] Xia Qu, Prashant Doshi, and Adam Goodie. 2012. Modeling deep strategic reasoning by humans in competitive games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 1243–1244.

[26] Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew Botvinick. 2018. Machine Theory of Mind. *CoRR* abs/1802.07740 (2018). arXiv:1802.07740 http://arxiv.org/abs/1802.07740

[27] Anatol Rapoport. 1974. *Prisoner's Dilemma—Recollections and observations*. Springer.

[28] Brian Scassellati. 2002. Theory of Mind for a Humanoid Robot. *Autonomous Robots* 12, 1 (01 Jan 2002), 13–24. https://doi.org/10.1023/A:1013298507114

[29] David Silver and Joel Veness. 2010. Monte-Carlo planning in large POMDPs. In *Advances in neural information processing systems*. 2164–2172.

[30] Ekhlas Sonu, Yingke Chen, and Prashant Doshi. 2017. Decision-theoretic planning under anonymity in agent populations. *Journal of Artificial Intelligence Research* 59 (2017), 725–770.

[31] Siddharth Suri and Duncan J. Watts. 2011. Cooperation and Contagion in Web-Based, Networked Public Goods Experiments. *PLOS ONE* 6, 3 (03 2011), 1–18. https://doi.org/10.1371/journal.pone.0016836

[32] Michael Wunder, Michael Kaisers, John Robert Yaros, and Michael Littman. 2011. Using Iterated Reasoning to Predict Opponent Strategies. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2 (AAMAS '11)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 593–600. http://dl.acm.org/citation.cfm?id=2031678.2031702