A Scalable Framework to Choose Sellers in E-Marketplaces Using POMDPs *

Athirai A. Irissappane
Nanyang Technological University
Singapore
athirai001@e.ntu.edu.sg

Frans A. Oliehoek
University of Amsterdam
University of Liverpool
frans.oliehoek@liverpool.ac.uk

Jie Zhang
Nanyang Technological University
Singapore
zhangj@ntu.edu.sg

Abstract

In multiagent e-marketplaces, buying agents need to select good sellers by querying other buyers (called advisors). Partially Observable Markov Decision Processes (POMDPs) have shown to be an effective framework for optimally selecting sellers by selectively querying advisors. However, current solution methods do not scale to hundreds or even tens of agents operating in the e-market. In this paper, we propose the Mixture of POMDP Experts (MOPE) technique, which exploits the inherent structure of trust-based domains, such as the seller selection problem in e-markets, by aggregating the solutions of smaller sub-POMDPs. We propose a number of variants of the MOPE approach that we analyze theoretically and empirically. Experiments show that MOPE can scale up to a hundred agents thereby leveraging the presence of more advisors to significantly improve buyer satisfaction.

1 Introduction

In many domains, agents need to determine the trustworthiness (quality) of other agents before interacting with them. Specifically, in e-marketplaces, buying agents need to reason about the quality of sellers and determine which sellers to do business with (referred to as the *seller selection problem*). When buyers have no previous experience with sellers, they can obtain advice by querying other buyers (called *advisors*). However, some advisors may be untrustworthy and provide misleading opinions about sellers.

The Partially Observable Markov Decision Process (POMDP) is a framework for sequential decision making under uncertainty, suitable for e-markets, where buyers often need to make decisions with limited information about the sellers and advisors. Regan, Cohen, and Poupart (2001) propose the Advisor POMDP, for the seller selection problem, which, rather than trying to achieve the most accurate estimate of sellers, tries to select good sellers optimally with respect to its belief. Seller and Advisor Selection (SALE) POMDP (Irissappane, Oliehoek, and Zhang 2014) extends Advisor POMDP to additionally deal with trust propagation, by introducing queries about advisors. In principle, SALE POMDP enables maximizing buyer satisfaction by optimally trading off information gaining (querying advisors) and exploiting (selecting a seller) actions, and experiments have shown very good results in practice. Moreover, the approach is easily generalizable to more general problems with trust-propagation components, such as routing in Wireless Sensor Networks (WSNs) (Irissappane et al. 2015).

Unfortunately, the above approaches suffer from scalability issues. Finding optimal policies for POMDPs is, in general, computationally intractable (PSPACE complete) and solvers computing exact solutions do not scale to more than a handful of states (Cassandra, Kaelbling, and Littman 1994). While approximation algorithms have shown to supply good policies rapidly even for problems with very large state spaces (Silver and Veness 2010), the scalability of the SALE POMDP, which is based on one such method (Poupart 2005), is limited to about 10 agents (sellers and advisors). For larger number of agents, solution times grow and solution quality degenerates, precluding the SALE POMDP from exploiting the presence of more sellers and advisors.

This paper proposes a novel method, referred to as the *Mixture of POMDP Experts (MOPE)* approach, for dealing with very large trust-propagation problems such as SALE POMDPs with many sellers and advisors. The key idea is to divide the large seller selection POMDP problem into a multitude of computationally tractable smaller *(sub)-POMDPs*, each containing a subset of sellers and advisors. The actions of the *sub-POMDPs* (SPs) are then aggregated, to find the best action in the process of selecting a good seller.

The MOPE approach exploits the structure of the Dynamic Bayesian Network that represents the transition and observation probabilities of the SALE POMDP: query actions do not affect the actual states but only the agent's *beliefs* over the state factors, making it easier to decompose a large seller selection problem into smaller sub-problems that approximate the larger problem. Due to the improved scalability of MOPE, it can leverage the presence of more advisors to make more informed decisions about sellers. Extensive evaluation in a simulated e-marketplace demonstrates that MOPE can scale up to a hundred agents (millions of states and thousands of actions), outperforming the state-of-the-art POMCP (Silver and Veness 2010) method, while using less computation time. We also demonstrate that MOPE can bring scalability to other domains by showing results for wireless sensor networks with up to 40 neighboring nodes.

*An extended version of this paper is available on arXiv.

2 Background

This paper mainly relies on POMDPs, which can represent decision making problems under uncertainty in terms of states, actions, transitions, observations and rewards. We refer to Kaelbling et al. (1998), Spaan (2012) for a comprehensive introduction to POMDPs. Here, we briefly describe the Seller and Advisor Selection (SALE) POMDP (Irissappane, Oliehoek, and Zhang 2014), which is the main application for the technique we propose in this paper.

States. Each state is represented using the following state factors: the quality levels of each seller ($q_j \in \{high, low\}$), each advisor ($u_i \in \{trustworthy, untrustworthy\}$) and status of the transaction ($sat \in \{not\_started, satisfactory, unsatisfactory, gave\_up, finished\}$).

Actions and Transitions. For query actions, $seller\_query_{(i,j)}$ $(SQ)_{(i,j)}$, i.e., ask advisor i about seller j and $advisor\_query_{(i,i')}$ $(AQ)_{(i,i')}$, i.e., ask advisor i about another advisor i', the states do not change. For BUY_j action, to buy from seller j, the state transitions to successful (sat = $satisfactory$) on buying from a good seller and unsuccessful (sat = $unsatisfactory$), otherwise. For do_not_buy (DNB) action, i.e., do not buy from any seller, the state transitions to sat = $gave\_up$.

Rewards. There is small cost for the query actions. A reward/penalty is associated with a successful/unsuccessful transaction. There is a penalty for taking DNB action, when there is a seller of high quality, otherwise a reward is given.

Observations. After SQ_{ij}, $AQ_{ii'}$ actions, an observation $o \in \{good, bad\}$, corresponding to the quality of seller j and $o \in \{trustworthy, untrustworthy\}$ corresponding to the quality of advisor i' is received, respectively. After BUY_j action, the agent can also receive an observation based on the actual quality of seller j, allowing to reuse the updated beliefs, in case of multiple transactions. The observation probabilities are such that trustworthy advisors give more accurate and consistent answers than untrustworthy ones.

When the SALE POMDP agent interacts with the environment, it maintains a *belief* $b \in B$, i.e., a probability distribution over states. If $b(s)$ specifies the probability of s (for all s), we can derive b' an updated belief after taking some action a and receiving an observation o using the Bayes' rule. We also assume an infinite horizon problem. A POMDP policy $\pi : B \rightarrow \mathcal{A}$, maps belief $b \in B$ to an action $a \in \mathcal{A}$. A policy π is associated with a value function $V(b)$, specifying the expected total reward of executing policy π starting from b, with discount factor γ. The main objective of the POMDP agent is to find an optimal policy π^*, which maximizes $V(b)$ (Eqn. 1). The value function can also be represented in terms of Q-functions, given by Eqn. 2, where, b^a_o is the belief state resulting from b after taking action a and receiving observation $o \in \mathcal{O}$.

$$V^*(b) = \max_{\pi} \mathbb{E}\Big[\sum_t \gamma^t R(s,a,s')|\pi, b\Big] = \max_{a \in A} Q^*(b,a) \quad (1)$$

$$Q^*(b,a) = \sum_{s \in S} b(s) R(s,a) + \gamma \sum_{o \in \Omega} p(o|b,a) V^*(b^a_o) \quad (2)$$

By computing the optimal value function, we can optimize the *long-term* rewards by picking maximizing actions. This stands in contrast to *myopic* approaches that maximize the immediate rewards R. Such approaches are unsuitable for seller selection: in order to correctly value the different query actions, one needs to reason about their impact on the future beliefs and the associated value of information. In order to compute V^* (approximately), state-of-the-art flat solvers such as SARSOP (Kurniawati, Hsu, and Lee 2008) can be used, but these provide very limited scalability (Oliehoek, Gokhale, and Zhang 2012), as the number of states grow exponentially with the number of agents n (i.e., sellers and advisors). Therefore, Irissappane, Oliehoek, and Zhang (2014) employ a solution method, factored Perseus (Poupart 2005), that exploits the factored representation of this domain, thus allowing to scale to roughly 10 agents. Beyond that solution times go up significantly while solution quality drops. Apart from the number of state factors themselves, a difficulty is that the number of actions grows with order $O(n^2)$ as the query actions involve pairs of agents.

3 A SingleExpert Baseline

In this paper, we propose techniques to exploit the structure present in (settings like) the SALE POMDP. Here, we introduce a baseline algorithm as an intuitive starting point for the more advanced method we propose in the next section.

This baseline is a method to apply the SALE POMDP to large problems. When faced with a SALE POMDP instance with many sellers and advisors, we can randomly select a subset of agents that is small enough to model and solve as a SALE POMDP and use the resulting policy. Since the q_j and u_i variables do not influence each other, defining such a *sub-POMDP (SP)* is trivial as it merely amounts to deleting all non-selected state variables as well as actions and observations that pertain to them. Also, the resulting model is a small SALE POMDP, thus we can find a good solution for it. While this voluntary restriction on the sellers one can consider and advisors one can ask is somewhat limiting, it is quite possible that it may lead to acceptable performance and it might be better than incorrectly reasoning about all of the agents. We call this approach the 'SingleExpert' approach, since the randomly selected SP acts as a (single) expert as to what action to take in the larger problem.

4 Mixture of POMDP Experts (MOPE)

While we argue that SingleExpert might have its merit, clearly, we want to develop methods that can exploit large pools of potential sellers and advisors. To accomplish this, we introduce the *Mixture of POMDP Experts (MOPE)* framework. SingleExpert exploits a particular property of trust propagation-like domains: constructing an SP is possible because the state variables encoding seller and advisor qualities do not affect each other and cannot be influenced by actions. In fact, interaction of these variables only arises in the agent's beliefs manifested as correlations induced by the coupling via observations. For example, if we query advisor i about seller j and receive observation bad, it not only increases the probability of the seller being low quality ($q_j = low$) and advisor being $u_i = trustworthy$, but also

Algorithm 1: Mixture of POMDP Experts (MOPE)

Input : $\mathcal{M}$, a large SALE POMDP

1 Randomly split $\mathcal{M}$ into SPs $\{\mathcal{M}_1, \ldots, \mathcal{M}_K\}$
2 Solve all SPs, yielding $\{V_1^*, \ldots, V_K^*\}$
3 **foreach** *TimeStep* t **do**
4 $\mathcal{V} \leftarrow \emptyset$; //the set of votes
5 **for** $k \in \{1 \ldots K\}$ **do**
6 $b_k \leftarrow$ DetermineLocalBelief(b, k)
7 $\mathcal{V} \leftarrow \mathcal{V} \cup \{$VoteFromSP$(k, b_k, V_k^*)\}$
8 $\bar{a} \leftarrow$ AggregateVotes$(\mathcal{V})$
9 Execute$(\bar{a})$
10 $o \leftarrow$ receiveObservation()
11 $b' \leftarrow$ GlobalBeliefUpdate$(b, \bar{a}, o)$

of $(q_j = high, u_i = untrustworthy)$. The MOPE framework aims to take this insight further by approximating such correlations using smaller clusters of variables, as in variational inference approaches (Koller and Friedman 2009), leading to the idea of representing the larger problem using a number of smaller SPs and leveraging their solutions.

4.1 MOPE Algorithm Overview

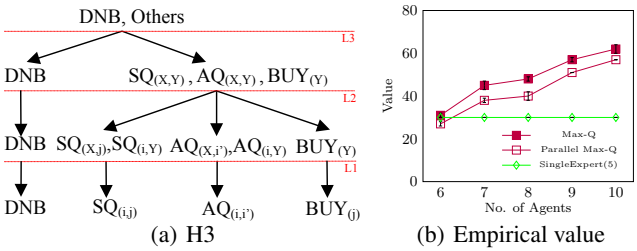
Algorithm 1 gives a brief overview of the MOPE framework. We first form the SPs by randomly selecting a subset of sellers and advisors (M_k in Line 1). Each SP is solved to obtain the optimal policy and thereby its maximum expected total reward V_k^* (Line 2)[1]. When SPs have the same agent composition (number of sellers and advisors), the found V_k^* can be reused amongst them. We define $\mathcal{V}$ as a set of votes v collected from each SP. Each vote $v = (a, q)$ is a set containing the action a suggested by the SP and its associated Q-value q. To maintain beliefs about all the state factors, it is possible to maintain the local beliefs in each SP, in parallel. However, doing so: 1) we need to deal with the actions not present in a SP as its local belief will be updated only if the SP contains the executed action $\bar{a}$; 2) we cannot properly take into account the influence of state factors not modeled in the SP on the belief, which may lead to inconsistent beliefs in different SPs. Thus, we propose to maintain and update the beliefs $b \in B$ at the global level, i.e., involving all state factors.

At each time step, for each SP, we first extract its current local belief b_k (Line 6) from the global belief b. Based on b_k, we obtain its vote v, i.e, its recommended action a and the associated q value (Line 7). The overall best action $\bar{a}$ is obtained (Line 8) by aggregating all the votes $v \in \mathcal{V}$. Action $\bar{a}$ is then executed (Line 9) and an observation o is received (Line 10), based on which the global beliefs are updated (Line 11). The following subsections give a more detailed description of the techniques used in the framework.

4.2 Dividing into Sub-POMDPs

We randomly select subsets of sellers and advisors from the whole population W to decompose $\mathcal{M}$ into a number of SPs. If SPA is the number of SPs that each agent can be a part of and APS is the number of agents each SP should contain, the total number of SPs necessary for the seller selection problem is given by $|W|$*SPA/APS. Also, APS is chosen such that the SPs can be computationally tractable.

[1]In practice, we may not solve the SPs optimally, and use the best policy and accompanying value function that we could find.



(a) H3 (b) Empirical value

Figure 1: (a) Voting hierarchy; (b) Max-Q vs Parallel Max-Q

4.3 AggregateVotes($\mathcal{V}$)

Here, we describe different ways to aggregate the votes $\mathcal{V}$.

Parallel Max-Q. Here, the best action $\bar{a}$ is selected as the action with the maximum Q-value ($\bar{a} = \arg\max_{a \in \mathcal{V}} q$), among those present in $\mathcal{V}$. Also, Parallel Max-Q maintains, in parallel, a set $\mathcal{B} = \{b_1, \ldots, b_K\}$ of local beliefs (corresponding to the SPs). We will use $\mathcal{B}$ as the global belief, in this case. GlobalBeliefUpdate(b,$\bar{a}$,o) is performed such that the beliefs b_k in each SP are updated using the Bayes' rule in parallel, when $\bar{a} \in \mathcal{A}_k$ (actions in M_k) and $o \in \mathcal{O}_k$.

To analyse the performance of Parallel Max-Q for a given decomposition $D = \{\mathcal{M}_1, \ldots, \mathcal{M}_K\}$ of SPs, we derive a lower bound on its performance. Specifically, we show that the expected sum of rewards V_D^{pmq} realized by parallel Max-Q for a decomposition D, is at least as much as the optimal value V_k^* realized by picking any SingleExpert $\mathcal{M}_k \in D$. For this, we need to make two assumptions: the decomposition $D = \{\mathcal{M}_1, \ldots, \mathcal{M}_K\}$ is non-overlapping (i.e., no two SPs $\mathcal{M}_i, \mathcal{M}_j$ contain the same seller or advisor state factors), and the true initial state distribution $b^0(s)$ is factored along the decomposition (i.e., $b^0(s) = b_1^0(s_1) \times b_2^0(s_2) \cdot \cdots \times b_K^0(s_K)$).

Theorem 1. *If the decomposition $D = \{\mathcal{M}_1, \ldots, \mathcal{M}_K\}$ is non-overlapping, and the true initial state distribution b^0 is factored along the decomposition, then the value realized by Parallel Max-Q is at least as much as the value of the best Single Expert: $V_D^{pmq}(\mathcal{B}^0) \geq \max_{k \in \{1,\ldots,K\}} V_k^*(b_k^0)$.*

The proof of Theorem 1 is described in detail by Irissappane, Oliehoek, and Zhang (2015). However, when SPs are overlapping, Parallel Max-Q at times, can perform worse than the Best SingleExpert due to inconsistent beliefs across SPs: consider 2 SPs with an overlapping (seller) and some non-overlapping advisors. If Parallel Max-Q selects SP 1 and executes a seller query about the (shared) seller, and receives observation 'bad', beliefs about the seller will be updated in SP 1 alone and not in SP 2. Parallel Max-Q will switch to SP 2 which has an (overestimated) higher value, leading to unnecessary information gaining actions (with associated costs), thereby affecting its performance.

Max-Q. To address the issue of inconsistent beliefs, here in Max-Q, the beliefs are not maintained in parallel, instead they are maintained and updated (using the Bayes' rule) at the global level, i.e., involving all state factors, as it helps to propagate information (about the sellers and advisors) across SPs. We empirically show the advantage[2] of Max-Q over Parallel Max-Q, for a decomposition D in Fig. 1(b) by plotting their values for different seller selection problems, comprising of $6-10$ agents. We also show the value of SingleExpert (randomly chosen 5 agent SP) in Fig. 1(b).

Majority Voting. As Parallel Max-Q and Max-Q select the action of the maximizing SP (with the maximum Q-value), they consider the value that the action will generate for a single sub-problem. It is likely that certain actions are more useful for many sub-problems and it is better to select the action with a higher value in all SPs than the action with the highest value in one single SP. Here, we formalize one such technique called the Majority Voting approach.

Algorithm 2 describes the Majority Voting approach in detail. Based on the votes $v \in \mathcal{V}$, we first count the number of SPs which suggested the action a (Line 3). We also determine the mean Q-value associated with each action a using the qvalsum[] and meanQs[] variables (Lines $4-6$).

While using the Majority Voting technique, we need to consider the fact that not every SP will have the same set of actions as it depends on which sellers and advisors are present in the SP. For instance, while each SP has the action DNB, the action, say $SQ_{(a12,s23)}$ will only be present in a SP containing both $advisor12$ and $seller23$. Thus, most $SQ_{i,j}$ and $AQ_{i,i'}$ actions might not be represented in any SP, and the ones present may be represented in just one SP.

To address this, we make use of the additional information present in the actions of each SP by formulating the concept of abstract actions. We consider three levels of abstractions (see Irissappane, Oliehoek, and Zhang (2015) for more details): Level L1 abstract actions, e.g., $SQ_{(X,s23)}$, $SQ_{(a12,Y)}$, $AQ_{(X,a30)}$, $AQ_{(a12,Y)}$, $BUY_{(Y)}$, DNB, where 'X', 'Y' denote an unbound variable; Level L2 abstract actions, $SQ_{(X,Y)}$, $AQ_{(X,Y)}$, $BUY_{(Y)}$, DNB; and Level L3 abstract actions, $DNB, Others \in \{SQ_{(X,Y)}, AQ_{(X,Y)}, BUY_{(Y)}\}$.

We also empirically investigate which abstract actions (among L1, L2, L3) need to be used by considering a number of voting hierarchies. In H1 hierarchy, only L1 abstract actions are considered and the best abstract action $\tilde{a}^*$ is chosen, after which the concrete action $\bar{a}$ is chosen. In hierarchy H2, first the best abstract action among the L2 abstract actions is determined, followed by the best L1 abstract action and the concrete action. In hierarchy H3 (shown in Fig. 1(a)), first the best L3 abstract action is determined followed by L2, L1 best abstract actions and then the concrete action.

In Algorithm 2, we maintain a separate set of votes $\mathcal{AV}$ for abstract actions $\tilde{a}$. In Line 8, we determine all the abstract actions that correspond to the regular action a contained in vote v. Subsequently, we increment their counts[$\tilde{a}$] and meanQs[$\tilde{a}$] (Lines 9-13). Then, in Line 14, the best abstract action $\tilde{a}^*$ is first selected, which is subsequently refined to determine the best concrete action. This refinement process depends on the employed voting hierarchy, e.g., when using the H1 hierarchy, $\bar{a} = \arg\max_{a \in \mathcal{A}(\tilde{a}^*)} \text{counts}[a] * \text{meanQs}[a]$, where $\mathcal{A}(\tilde{a}^*)$ denotes the set of concrete actions consistent with the abstract action $\tilde{a}^*$.

[2] Results are statistically verified by paired t-test (α=0.05).

Algorithm 2: AggregateVotes by Majority Voting

```
   Input : V, the set of votes
1  foreach v ∈ V do
2      (a, q) ← v ;                          //unpack vote
3      counts[a] += 1;
4      qvalsum[a] += q;
5  foreach a ∈ A do
6      meanQs[a] = qvalsum[a] / counts[a];
   //Count votes for abstract actions:
7  foreach v ∈ V do
8      AV = {(ã, q)} ← AbstractedVotes(v);
9      foreach (ã, q) ∈ AV do
10         counts[ã] += 1;
11         qvalsum[ã] += q;
12 foreach ã ∈ Ã do
13     meanQs[ã] = qvalsum[ã] / counts[ã];
14 ã* = arg max_ã(counts[ã] * meanQs[ã]);
15 ā = Refine(ã*, counts, meanQs);
16 return ā
```

4.4 Belief Update

Though we can maintain and perform exact belief updates at the global level, i.e., involving all state factors, using the Bayes' rule, such exact inference is complex and does not scale to more than 10 agents. Therefore, we propose to employ the approximate inference methods. In particular, we apply Factored Frontier (FF) (Murphy and Weiss 2001), which maintains the belief as the product of marginals of the state factors x_i: $b(s) = \prod_{i=1}^{|s|} \hat{b}(x_i)$. Thus the beliefs for each SP can directly be constructed by combining the marginals of those state factors that are a part of the sub-POMDP. While FF is a simple algorithm, and other choices are possible, it does allow influence of variables to propagate through the network and our experiments suggest that FF performs quite well.

5 Experiments

Here, we empirically investigate the solution quality and scalability of the proposed Mixture of POMDP experts (MOPE) technique in the e-marketplace domain. We are primarily interested to see if the added scalability can actually translate into additional value from the buyer's perspective.

Experimental Setup. We analyze different design considerations for MOPE (SPA=4 SPs per agent and APS=5 agents per SP with a uniform composition for all SPs comprised of 1 seller and 4 advisors, such that we can reuse V*, as described in Sec. 4.1) and compare it with: 1) the original SALE POMDP. We assume uniform initial beliefs and compute the SALE POMDP optimal policy using Symbolic Perseus (Poupart 2005).; 2) SingleExpert(5), i.e., a randomly selected 5-agent SP, serving as the lower bound; 3)

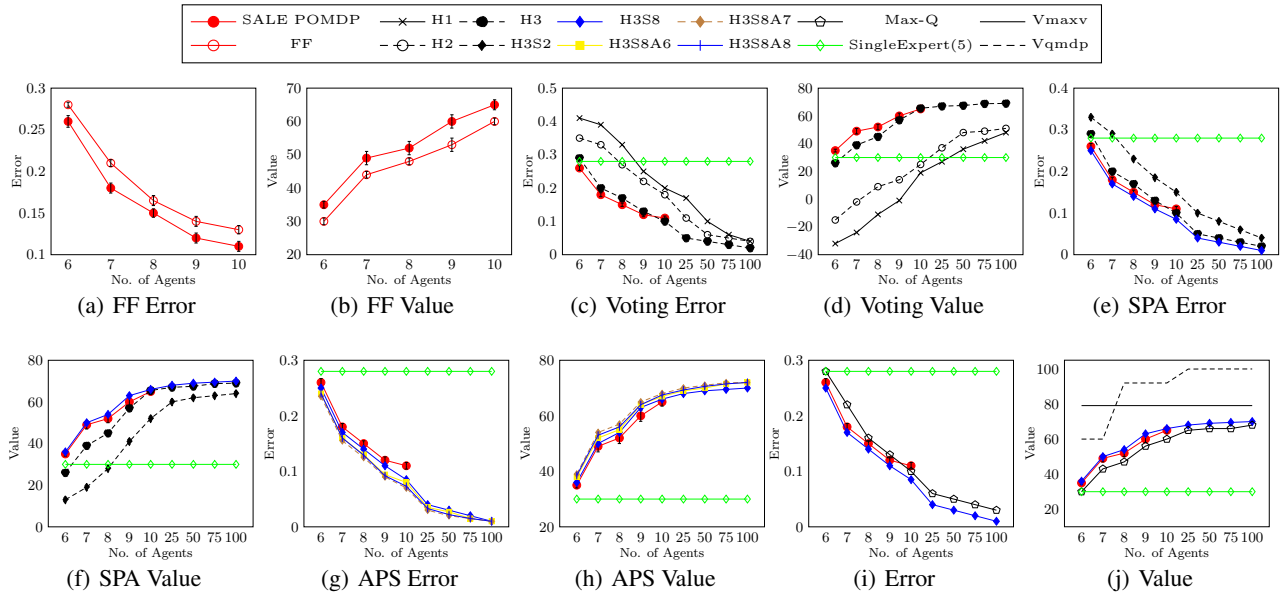


Figure 2: (a-b) Influence of FF algorithm; (c-d) Performance comparison of Majority Voting technique based on voting hierarchies; (e-f) Influence of SPA; (g-h) Influence of APS; (i-j) Comparison with Max-Q, V_{maxv}, V_{qmdp}

POMCP (Silver and Veness 2010), an online planning approach which requires a number of random simulations (we use $10,000$ simulations per selected action) to estimate the potential for long-term reward; 4) an *optimistic heuristic* value V_{maxv}, which is the value obtained by running many simulations of MOPE (Majority Voting with H3 hierarchy and SPA=8) on 'ideal' global problems (i.e., on 100 agent problems with good sellers and trustworthy advisors). We consider such a heuristic as we know that beginning with a most favourable state (which in our case is the presence of good sellers and trustworthy advisors in the SPs, as they have a higher probability of resulting in successful transactions), always results in best performance while executing a POMDP policy; 5) the QMDP value V_{qmdp}, which is the value obtained by considering the states to be fully observable in the next time step (Littman, Cassandra, and Kaelbling 1995). Though majority of our (query) actions do not have value while computing V_{qmdp}, we still consider the QMDP value as it can serve as an upper bound.

W	6	7	8	9	10	25	50	75	100
$\|S\|$	2^6	2^7	2^8	2^9	2^{10}	2^{25}	2^{50}	2^{75}	2^{100}
$\|A\|$	27	38	45	59	75	486	1971	4456	7941

Table 1: Size of the seller selection problem

We conduct experiments in a simulated e-marketplace, where buyers need to choose sellers as successful transaction partners. We measure the average *error* $\in [0, 1]$ in terms of the percentage of 'unsuccessful transactions' (buying from a bad seller or taking the DNB action in the presence of a good seller) and *value*, i.e, the discounted (0.95) reward in the process of choosing a seller. The buyer pays a cost of 1 for querying advisors about other advisors, 10 for querying about a seller, gains 100 for choosing a good seller or taking DNB when no seller is of good quality, loses 100 for choosing a bad seller or taking DNB when there is a good seller. The number of sellers is 20% of the whole population W and number of advisors is 80% among which 20% are untrustworthy. All the results are values averaged over 500 iterations from the point of view of a single buyer. We consider single transaction settings, where the buyer has no previous experience with the seller.

To analyze the scalability, we increase the number of agents W in the e-marketplace from $6 - 100$ (size of the corresponding seller selection problem, is given in Table 1) and measure the performance of the approaches in Fig. 2-3. As SALE POMDP does not scale effectively to more than 10 agents (ran out of time while computing the policy), its performance is not shown for W>10 in the figures.

Influence of FF. Fig. 2(a-b) shows the influence of using FF for the belief update. We see that while the approximation introduced by FF leads to a reduction in value compared to using exact belief updates, the difference is quite small.

Analysis of the Different MOPE Design Schemes. In Fig. 2(c-d), we analyse the performance of the H1, H2 and H3 hierarchies while using the Majority Voting scheme for selecting the best action in the MOPE approach. We see that H3 hierarchy outperforms[2] H1 and H2. We see that for (most) cases where SALE POMDP is able to provide an answer, it is performing slightly better than H3. This is expected since it does a full POMDP reasoning over the entire state space. However, for larger problems, the difference in performance becomes negligible and when including more advisors, H3 finds policies that lead to significantly smaller errors and higher payoffs. SingleExpert(5) achieves a constant performance as it always considers a group of 5 agents

to make decisions. The performance of all other approaches increase with the number of agents as there are more advisors to seek information about the sellers.

In Fig. 2(e-f), we analyse the influence of the number of SPs per agent (SPA), using H3 Majority Voting (with default SPA=4). Fig. 2(e-f) show that performance of H3 increases with SPA, i.e., H3S8 (SPA=8) shows the best performance and H3S2 (SPA=2) shows the least performance, because the total number of SPs considered increase, resulting in more informed decision making. We see that H3 and H3S8 outperform SALE POMDP for 10 agents, suggesting that the quality of Symbolic Perseus degrades for larger problems. Fig. 2(g-h) show the influence of the number of agents per SP (APS) for the H3S8 technique. H3S8A6 (APS=6), H3S8A7 (APS=7) and H3S8A8 (APS=8) outperform H3S8 (default APS=5) as increasing APS improves performance by reasoning over a larger state space. Importantly, we see how this enables MOPE to accumulate a significantly higher value (72 for H3S8A7, 100 agents) than the best SALE POMDP value (65). We expect that the lower performance of H3S8A8 compared to H3S8A7 is caused by a relative degradation of the solution quality of the (larger) SPs. But, H3S8A6, H3S8A7 and H3S8A8 involve a greater policy computation time than H3S8.

Comparison with Max-Q, POMCP, V_{maxv} and V_{qmdp}. In Fig. 2(i-j), we compare the performance of H3S8 along with Max-Q (SPA=8, APS=5 and using the FF algorithm). We have shown the error and value for the POMCP approach in Table 2 separately, to retain the clarity in Fig. 2(i-j). We see that H3S8 outperforms both Max-Q and POMCP. As the number of agents increases, performance of POMCP decreases, as it requires higher number of simulations to sample the beliefs and histories about the agents. Also, POMCP does not scale well with the number of actions (large in these problems). We have not shown the POMCP results for $W > 25$ due to the complexity of the simulations.

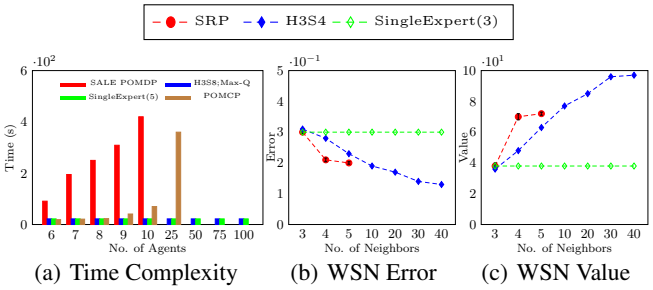
Fig. 2(j) also shows that V_{maxv}, i.e., the value obtained by H3S8 for problems where all sellers are of good quality and advisors are trustworthy is greater than the value obtained by H3S8 for normal problems (with low quality sellers and untrustworthy advisors). V_{qmdp}, the QMDP value, looks like a piecewise function because of the same number of sellers in some problems. Thus, Fig. 2(j) shows the lower bound, i.e., value of SingleExpert(5), optimistic heuristic value V_{maxv}, and the upper bound V_{qmdp} for a 5-agent decomposition.

W	6	7	8	9	10	25	50	75	100
error	0.60	0.62	0.70	0.80	0.81	0.90	-	-	-
value	-46	-54	-68	-90	-92	-110	-	-	-

Table 2: Performance of POMCP

Fig. 3(a) shows the policy computation time for each seller selection problem involving 6 to 100 agents. For POMCP we measure the simulation time per episode. We see that time taken by H3S8, SingleExpert(5), Max-Q is less than SALE POMDP and POMCP. Their constant time $22s$ is due to using the same 5-agent policy for all SPs.

Performance in WSN domain. We also apply the MOPE approach (H3S4 Majority Voting with APS=3, SPA=4) to improve the scalability of the SRP model (Irissappane et al. 2015) in the WSN domain and compare it with: the original SRP and SingleExpert(3) with 3 agents. We use the same simulation settings as in (Irissappane et al. 2015). Fig. 3(b-c) shows that SRP performs better than H3S4 for $3-5$ neighbors. However, it cannot provide solutions for more than 5 neighbors, while H3S4 can scale up to 40 neighbors, generating much higher value. Also, the policy time is $73s$ for H3S4 and $736s$ for the SRP model for 5 neighbors.



(a) Time Complexity (b) WSN Error (c) WSN Value

Figure 3: (a) Comparison of policy time; (b-c) Performance in wireless sensor networks

6 Related Work

There is extensive literature on scalable POMDP methods. Point-based value iteration (Pineau et al. 2003) and bounded policy iteration (Poupart and Boutilier 2003) exploit the presence of good quality policies that can be represented by a small number of value vectors, but are limited to thousands of states (Poupart and Boutilier 2004). In structured domains, factored POMDPs improve scalability by exploiting compact representations (Feng and Hansen 2001; Guestrin, Koller, and Parr 2001b; Poupart 2005; Veiga et al. 2014) as also applied in the regular SALE POMDP model.

Some approaches use a similar concept of decomposing a (PO)MDP into smaller sub-problems. Meuleau et al. (1998) assume that sub-problems are *very weakly coupled*: each sub-problem corresponds to an independent subtask whose state/action spaces do not directly influence the other tasks. In contrast, MOPE divides a single large POMDP problem into SPs, which can contain overlapping state variables/actions. Similar to our work, Williams and Young (2007) consider a more general decomposition, but they rely on domain specific heuristics, while we investigate several general methods to aggregate the recommendations from all SPs. Yadav et al. (2015) also propose an approach which decomposes a POMDP into SPs, but these are formed in a very different way: by sampling *values* for sub-sets of hidden state factors. A major difference between all these works and ours, is that their sub-problems directly follow from the domain. In contrast, in our approach, the number of sub-problems can be chosen to control the time vs. quality trade-off.

Decomposition has also been a popular technique in multiagent planning approaches (Guestrin, Koller, and Parr 2001a; Becker et al. 2003; Nair et al. 2003; Witwicki

and Durfee 2010; Oliehoek, Witwicki, and Kaelbling 2012; Amato and Oliehoek 2015). However, in all these cases, structure is exploited that is particular to the multiagent setting; when applied to single-agent problems these methods do not directly offer any additional benefits.

MOPE can be interpreted as a type of ensemble method (Dietterich 2000). In particular, there is a resemblance to random forests (Breiman 2001): the way that they randomly select features is not unlike our random selection of state factors (seller and advisor variables).

7 Conclusion and Future Work

We propose the Mixture of POMDP Experts (MOPE) technique to address the scalability issues in solving large seller selection (SALE) POMDP problems for e-marketplaces. MOPE works by dividing the large POMDP problem into computationally tractable smaller sub-POMDPs and then aggregates the actions of the sub-POMDPs. Extensive evaluation shows that MOPE achieves a reasonable approximation to the SALE POMDP for small problems and can scale up to a hundred agents by effectively exploiting the presence of more advisors to generate significantly higher buyer satisfaction. We also show that MOPE improves the scalability of a POMDP model in the sensor network domain. Our agenda for future work is to analyse more sophisticated ways (e.g., using community detection) of dividing the sub-POMDPs rather than random partitioning.

Acknowledgments

This work is supported by the A*STAR SERC grant (1224104047), NWO Innovational Research Incentives Scheme Veni #639.021.336, and the Institute for Media Innovation.

References

Amato, C., and Oliehoek, F. A. 2015. Scalable planning and learning for multiagent POMDPs. In *AAAI*.

Becker, R.; Zilberstein, S.; Lesser, V.; and Goldman, C. V. 2003. Transition-independent decentralized Markov decision processes. In *AAMAS*.

Breiman, L. 2001. Random forests. *Machine learning* 45(1):5–32.

Cassandra, A. R.; Kaelbling, L. P.; and Littman, M. L. 1994. Acting optimally in partially observable stochastic domains. In *AAAI*.

Dietterich, T. G. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857. 1–15.

Feng, Z., and Hansen, E. A. 2001. Approximate planning for factored POMDPs. In *European Conference on Planning*.

Guestrin, C.; Koller, D.; and Parr, R. 2001a. Multiagent planning with factored MDPs. In *NIPS*.

Guestrin, C.; Koller, D.; and Parr, R. 2001b. Solving factored POMDPs with linear value functions. In *ICAPS*.

Irissappane, A. A.; Zhang, J.; Oliehoek, F. A.; and Dutta, P. S. 2015. Secure routing in wireless sensor networks via POMDPs. In *IJCAI*.

Irissappane, A. A.; Oliehoek, F. A.; and Zhang, J. 2014. A POMDP based approach to optimally select sellers in electronic marketplaces. In *AAMAS*.

Irissappane, A. A.; Oliehoek, F. A.; and Zhang, J. 2015. Scaling POMDPs for selecting sellers in e-markets—extended version. *ArXiv e-prints* arXiv:1511.09147.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101(1):99–134.

Koller, D., and Friedman, N. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

Kurniawati, H.; Hsu, D.; and Lee, W. S. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *RSS*.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning policies for partially observable environments: Scaling up. In *ICML*.

Meuleau, N.; Hauskrecht, M.; Kim, K.-E.; Peshkin, L.; Kaelbling, L. P.; Dean, T. L.; and Boutilier, C. 1998. Solving very large weakly coupled Markov decision processes. In *AAAI*.

Murphy, K., and Weiss, Y. 2001. The factored frontier algorithm for approximate inference in DBNs. In *UAI*.

Nair, R.; Tambe, M.; Yokoo, M.; Pynadath, D.; and Marsella, S. 2003. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI*.

Oliehoek, F. A.; Gokhale, A. A.; and Zhang, J. 2012. Reasoning about advisors for seller selection in e-marketplaces via POMDPs. In *Workshop on Trust in Agent Societies*.

Oliehoek, F. A.; Witwicki, S.; and Kaelbling, L. P. 2012. Influence-based abstraction for multiagent systems. In *AAAI*.

Pineau, J.; Gordon, G.; Thrun, S.; et al. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI*.

Poupart, P., and Boutilier, C. 2003. Bounded finite state controllers. In *NIPS*.

Poupart, P., and Boutilier, C. 2004. VDCBPI: an approximate scalable algorithm for large POMDPs. In *NIPS*.

Poupart, P. 2005. *Exploiting structure to efficiently solve large scale Partially Observable Markov Decision Processes*. Ph.D. Dissertation, University of Toronto.

Regan, K.; Cohen, R.; and Poupart, P. 2001. The advisor-POMDP: A principled approach to trust through reputation in electronic markets. In *Privacy, Security and Trust*.

Silver, D., and Veness, J. 2010. Monte-Carlo planning in large POMDPs. In *NIPS*.

Spaan, M. T. J. 2012. Partially observable Markov decision processes. In *Reinforcement Learning*, volume 12. Springer. 387–414.

Veiga, T. S.; Spaan, M. T.; Lima, P. U.; Brodley, C. E.; and Stone, P. 2014. Point-based POMDP solving with factored value function approximation. In *AAAI*.

Williams, J. D., and Young, S. 2007. Scaling POMDPs for spoken dialog management. *IEEE Transactions on Audio, Speech, and Language Processing* 15(7):2116–2129.

Witwicki, S. J., and Durfee, E. H. 2010. Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *ICAPS*.

Yadav, A.; Soriano Marcolino, L.; Rice, E.; Petering, R.; Winetrobe, H.; Rhoades, H.; Tambe, M.; and Carmichael, H. 2015. Preventing HIV spread in homeless populations using PSINET. In *IAAI*.