Experimental analysis of privacy loss in DCOP algorithms

Rachel Greenstadt*
Harvard University
greenie@eecs.harvard.edu

Jonathan P. Pearce, Emma Bowring and Milind Tambe†
University of Southern California
{jppearce,bowring,tambe}@usc.edu

ABSTRACT

Distributed Constraint Optimization (DCOP) is rapidly emerging as a prominent technique for multiagent coordination. Unfortunately, rigorous quantitative evaluations of privacy loss in DCOP algorithms have been lacking despite the fact that agent privacy is a key motivation for applying DCOPs in many applications. Recently, Maheswaran et al. [3, 4] introduced a framework for quantitative evaluations of privacy in DCOP algorithms, showing that early DCOP algorithms lose more privacy than purely centralized approaches and questioning the motivation for applying DCOPs. Do state-of-the art DCOP algorithms suffer from a similar shortcoming? This paper answers that question by investigating the most efficient DCOP algorithms, including both DPOP and ADOPT.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Performance, Security

Keywords

constraint reasoning, DCOP, privacy

1. INTRODUCTION

Promising approaches in distributed constraint optimization (DCOP) [5, 7], enable distributed conflict resolution and coordination while maintaining users' privacy. Indeed, maintaining privacy is a fundamental motivation in DCOP [5, 7, 9]. One approach to privacy in DCOP is to use cryptographic techniques [11] that ensure watertight privacy but require the use of external servers or computationally intensive cryptographic operations. Instead, we focus on an approach in which researchers provide metrics for quantifying the privacy loss in DCOP algorithms [1,3,4,6,10]. If we can bound privacy loss in specific DCOP algorithms, then cryptographic techniques may be avoidable in situations where they are impractical.

There are three key weaknesses in the previous work on privacy loss analysis in DCOP. First, recent cross-algorithm privacy loss analysis focused on a limited number of DCOP algorithms but indicated that these algorithms preserve less privacy than a centralized approach [3], seriously undermining a key motivation for these algorithms. Thus, it is crucial to analyze some of the most used and most recent DCOP algorithms to see whether they are similarly undermined and to measure their cross-algorithm performance. Two notable omissions in previous analysis are ADOPT [7] and DPOP [8], both among the most efficient DCOP algorithms.

This paper overturns the significant negative results from [3, 4] by providing positive privacy results for several DCOP algorithms not considered in [3, 4]. This paper analyzes ADOPT, DPOP and SynchID [7], three recent DCOP algorithms, via a large-scale experimental investigation of privacy loss in DCOP algorithms in the VPS (Valuations of Possible States) analysis framework [3, 4], using several distributed meeting scheduling scenarios.

A predecessor to the recently introduced algroithms above, SynchBB [2] is an early algorithm for DCOP. Previous work has provided a comparison of privacy loss of a centralized approach with SynchBB, suggesting that the centralized approach may lead to lower privacy loss. Hence, this paper focuses on the remaining algorithms above. These algorithms were chosen because they present novel design choices, or occupy a prominent place in the algorithmic space. The following describes key characteristics of these algorithms:

Adopt is an asynchronous complete DCOP algorithm, guaranteed to find the optimal solution. In Adopt, an agent communicates only one value from its domain at a time, or one message indicating the cost of an assignment to a set of variables at a time.

SynchID is an iterative deepening algorithm similar to Adopt, with two primary differences: agents are organized into a linear chain, rather than a tree, and messages are sent synchronously.

DPOP [8] is an synchronous complete DCOP algorithm, using a tree topology. DPOP is a variable elimination algorithm, where all relevant information is sent up the tree in one large message.

SynchBB or synchronous branch-and-bound, was studied in [4]. However, we focus on a slightly modified SynchBB where information irrelevant to the problem is not communicated.

2. EXPERIMENTAL METHODOLOGY

We focus our investigation on privacy loss in the distributed meeting scheduling problem, since this domain presents inherent privacy concerns [1, 4]. However, the results of this work can be generalized to other DCOP settings where privacy matters.

We define a meeting/event scheduling problem based on the for-

*Supported by a U.S. Department of Homeland Security (DHS) Fellowship, a program administered by the Oak Ridge Institute for Science and Education (ORISE). ORISE is managed by Oak Ridge Associated Universities under DOE contract number DE-AC05-00OR22750.
†This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

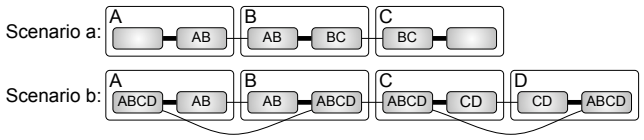

Figure 1: Scenarios: Transparent boxes represent agents and the dark, inner boxes are meeting variables. Thick lines are intra-agent contraints and thin lines are inter-agent constraints.

malism of [5], expressed using the PEAV-DCOP representation [5], which is motivated by privacy considerations.

Scenarios in PEAV-DCOP: The majority of scheduling instances in a functional personal assistant agent system will consist of a small number of meetings that need to be negotiated simultaneously. While larger-scale problems may present themselves, if privacy is a critical factor, the coordination protocols must be effective for these small-scale instances. We consider two scenarios of three (A, B, C) or four (A, B, C, D) agents. The PEAV-DCOP graphs in Figure 1 show the events, labeled by their attendees, and decomposed into variables and constraints.

2.1 VPS: Measuring Privacy Loss

The Valuation of Possible States (VPS) framework [4] was proposed to quantitatively evaluate privacy loss among a group of agents $R_1 \cdots R_n$. Quantification of privacy loss in VPS is based on a valuation on the other agents' estimates about (i.e. a probability distribution over) an agent's possible states. There are three key elements in VPS: (i) agent R_n's private information, modeled as a state $s_n \in S_n$, where S_n is a set of possible states that R_n may occupy; (ii) other agents' estimates about agent R_n's possible states, expressed as a probability distribution $\mathbb{P}_n((S_n)^{N-1})$, (iii) the utility that agent R_n derives from the distribution of other agents' beliefs about R_n's states, yielding value function $\mathbb{V}_n(\mathbb{P}_n((S_n)^{N-1}))$. Note that $\mathbb{P}_n((S_n)^{N-1}) = [\mathbb{P}_n^1(S_n)\ \mathbb{P}_n^2(S_n)\ \cdots \mathbb{P}_n^N(S_n)]$, where $\mathbb{P}_n^j(S_n)$ provides agent R_j's probability distribution over states of agent R_n.

Before negotiation, each agent knows only that the other agents exist in one of $|\mathcal{V}|^T$ possible states. Afterwards, each agent will be modeled by all other agents whose estimate of the observed agent is captured by $\mathbb{P}_n((S_n)^{N-1})$. In this analysis, we focus on the information-theoretic analysis introduced in [1] or the EntropyTS metric [3]. Privacy loss between two agents is studied on a per-timeslot basis, averaged over all timeslots.

We can scale this function and average the privacy loss between all pairs of agents such that the valuations span [0 1] with zero is no loss and one is complete loss of privacy.

2.2 Inference Algorithms

Based on the VPS framework, we define a process by which agents can infer information about other agents while running various DCOP algorithms, in order to measure the likely privacy loss between agents in a DCOP. All inference experiments for all algorithms (including the centralized method) start with the same initial assumptions. We assume that the constraint graph and the valuation of each *meeting* is known to all agents, but the valuations of *time slots* are private. These assumptions are exactly as in [3, 4], allowing comparison of the results; in addition, for the scenarios with few meetings, it is reasonable to assume that the valuations for meetings are public knowledge. Based on these assumptions, we developed the following methods for agent inference for SynchID, Adopt and DPOP.

Centralized: In a centralized algorithm, the agents all send their valuation information to one agent, who computes the result and returns. In every case the centralized agent can "infer" the valuations perfectly [4]. Since we express our results as an average of each agent's privacy loss, the privacy loss of the centralized algorithm is $\frac{1}{N}$ where N is the total number of agents.

SynchID: SynchID is a synchronous algorithm in which agents are ordered in a chain, and messages are passed up and down the chain. An upward message from agent R_n contains a number m_n, which is equal to the best currently known total reward for the subchain of agents under and including R_n. For PEAV, the total reward for the chain is equal to the sum of differences between the valuation of a scheduled meeting and the valuation of the time slot it occupies for every scheduled meeting for every person. We henceforth use $\Delta_{R_n}^{E_k}(t) = V_n^k - V_n^0(t)$ to denote the change in utility to the n^{th} agent for scheduling the k^{th} event at time t. When agent R_n receives an upward message it knows that m_n = a sum of Δ terms lower in the chain from R_n.

To illustrate how possible states can be eliminated in SynchID, we outline the inferences that one can make from messages received in Scenario a. In SynchID, upward messages to agent R_n contained information of the form:

$$m_n = \sum \Delta_{R_n}^{E_k}(t_{E_k}) + \sum \Delta_{R_n}^{E_k}(\tilde{t}_{E_k}), \quad (1)$$

where the summations include events downstream from R_n. t_{E_k} is the time of an event E_k when that time is known to R_n (because R_n is a participant in event E_k), and $\tilde{t}_{E_k}$ is the time of an event E_k when that time is not known to R_n. For example, since B knows when meeting BC is scheduled, as well as the value of meeting BC, a message from C to B (m_B) allows B to know $V_C(t_{BC})$ (the valuation vector component of C at the time at which meeting BC is scheduled). Similarly, a message from B to A (m_A) allows A to know $v_B(t_{AB}) + v_B(\tilde{t}_{BC}) + v_C(\tilde{t}_{BC})$, where $\tilde{t}_{BC}$ is some time not equal to t_{AB}, but otherwise unknown to A. Each of these relations allows the observing agent to reduce the number of possible states the other agents could be in. We obtain the privacy loss for SynchID by allowing each agent to collect these relations, iterate over them, and test each relation against a list of possible states for the other agents, discarding states that conflict with any of the relations.

Adopt: Adopt contains the same type of upward messages as in SynchID, but, due to its asynchrony, it may be impossible for agents to tell how many Δs are contained in the reward component of each message. When a message is received, we know it contains rewards for at least one agent more than the previous message it sent. However, due to asynchrony, our agent might have included more descendants in the message. So, for our inference, we use a $\leq$ sign. The inference equation is:

$$m_n \leq \sum \Delta_{R_n}^{E_k}(t_{E_k}) + \sum \Delta_{R_n}^{E_k}(\tilde{t}_{E_k}), \quad (2)$$

This relation changes to an equality in the special case when only one agent is downstream from agent R_n.

DPOP: In the DPOP algorithm, each agent sends exactly one cost message to its parent. This message is a table of all possible assignments of constrained upstream events and the aggregate costs of those assignments to the agents downstream of R_n. Each entry in the table is used to create inference rules as in equation 1. The events in the entry are the $\Delta_{R_n}^{E_k} t_{E_k}$ terms and other events with participating agents downstream of R_n are the $\Delta_{R_n}^{E_k} \tilde{t}_{E_k}$ terms.

SynchBB: Inference rules for SynchBB are as described in [4].

3. EXPERIMENTAL RESULTS

In this section, we present experimental results from two scenarios. We compare privacy loss in the studied algorithms according to the EntropyTS metric and we introduce a new metric to highlight privacy benefits of distribution over centralized approaches.

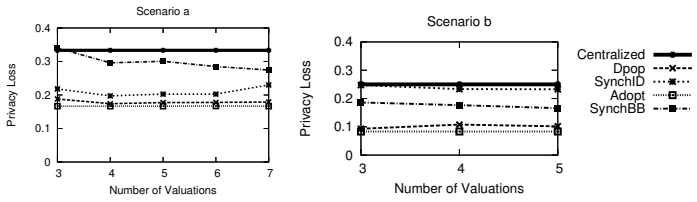

Figure 2: Privacy loss comparisons of the different algorithms.

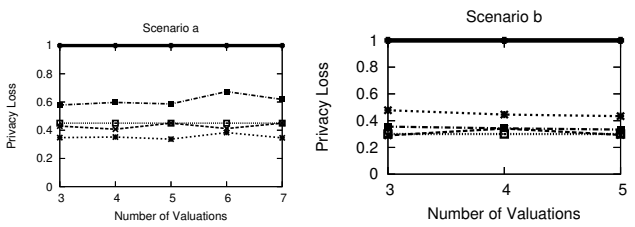

Figure 3: Algorithms compared using the MAX metric.

For the three agent scenario, we varied V, the number of valuations, from 3 to 7 while holding the number of timeslots = 3. For the four agent scenario, for reasons of computational complexity, we varied $|\mathcal{V}|$ from 3 to 5 while holding $T = 3$. For each $(T, |\mathcal{V}|)$ pair, we performed 10 runs per algorithm. For each run, the privacy loss for each agent was measured using the inference algorithms given in Section 3. The systemwide privacy loss was expressed as the arithmetic mean of each agent's privacy loss: $\sum_N \mathbb{V}_n/N$.

In each of our graphs, each data point is an average of 10 runs, and we provide statistical significance results to support our main conclusions. We use a chain topology for all algorithms, to allow fair comparison, since not all algorithms can use a DFS tree.

Cross-algorithm comparison: Figure 2 shows the comparison of privacy loss for the four algorithms mentioned above, for two scenarios, as well as providing a comparison of privacy loss with the centralized approach. The *x*-axis plots the different number of valuations (with number of time-slots fixed at 3) and the *y*-axis plots privacy loss. The thick horizontal line shows the centralized approach, for scenario a (three agents), its privacy loss is 0.33, but for scenario b (four agents) it is 0.25. We use the EntropyTS metric as the metric for privacy loss in this result. We obtained results for other metrics, and mostly these metrics agree with the conclusions drawn using the EntropyTS metric.

We conclude the following from Figure 2: (1) All algorithms but SynchBB have a privacy loss that is lower than the centralized approach. In contrast with the negative results presented in [4], which illustrated DCOP algorithms as having worse privacy loss than a centralized approach, this is a significant positive result. Indeed, the privacy loss in Adopt and DPOP is less than half that of the centralized approach. Furthermore, statistical tests show that Adopt performs better than centralized in all scenarios and DPOP performs better than centralized in all but one (significance level of 5%). (2) DPOP and Adopt had similar privacy loss, despite their vastly different approaches. Despite DPOP's one-shot communication of all information, it performed surprisingly well in terms of privacy loss. Adopt does perform slightly better than DPOP for privacy loss, but not to the level anticipated. (3) Adopt significantly outperformed SynchID in terms of privacy protection. The asynchrony in Adopt was expected to be significantly detrimental to privacy due to the increased numbers of messages. Instead, we found that the uncertainty introduced by asynchrony as to which agents participate in each cost message provides significant privacy gains compared to synchronous algorithms such as SynchID. (4) Despite modifications to improve privacy, SynchBB still performed the worst in terms of its privacy loss; often worse than centralized. The key reason for SynchBB's low performance is its bi-directional messaging of cost information. Thus, it is important to avoid bi-directional cost propagation in DCOP algorithms when privacy is a goal.

MAX metric: In Figure 2, we measured the loss of privacy between pairs of agents, averaged to find the systemwide privacy loss. The effect of one agent learning more than others, and gaining an asymmetric advantage over them, is not considered. The MAX metric addresses this effect by considering only the single agent that learns the most information about other agents (by EntropyTS), rather than the mean of the individual privacy loss figures. Thus, a centralized algorithm will have a value of 1, since the MAX agent learns all the information. Figure 3 shows the results for all the algorithms according to the MAX metric. The number of valuations is plotted on the *x*-axis and the privacy loss is plotted on the *y*-axis.

The MAX metric suggests that there is always a privacy benefit obtained by using DCOP algorithms, even those that perform worse than centralized by our other metrics, when the major privacy concern is one agent accumulating excessive knowledge. In the MAX metric, DPOP and Adopt tended to outperform SynchBB, while SynchID varied widely from scenario to scenario.

4. CONCLUSION

Distributed Constraint Optimization (DCOP) is rapidly emerging as a tool for multiagent coordination. Previous work [4] showed a negative result on privacy loss in early DCOP algorithms, casting doubt on the efficacy of DCOP in privacy requiring domains. This paper presents an investigation of several leading algorithms, including ADOPT and DPOP, and overturns earlier negative results. Our metrics show DCOP algorithms outperforming a centralized approach, with DPOP, SynchID and ADOPT performing best.

5. REFERENCES

[1] M. S. Franzin, F.Rossi, E. C. Freuder, and R. Wallace. Multi-agent meeting scheduling with preferences: efficiency, privacy loss, and solution quality. *Computational Intelligence*, 20(2), 2004.

[2] K. Hirayama and M. Yokoo. Distributed partial constraint satisfaction problem. In G. Smolka, editor, *Principles and Practice of Constraint Programming*, 1997.

[3] R. T. Maheswaran, J. P. Pearce, E. Bowring, P. Varakantham, and M. Tambe. Privacy loss in distributed constraint reasoning: A quantitative framework for analysis and its applications. *JAAMAS*, 2006.

[4] R. T. Maheswaran, J. P. Pearce, P. Varakantham, E. Bowring, and M. Tambe. Valuations of possible states (VPS):a quantitative framework for analysis of privacy loss among collaborative personal assistant agents. In *AAMAS*, 2005.

[5] R. T. Maheswaran, M. Tambe, E. Bowring, J. P. Pearce, and P. Varakantham. Taking DCOP to the real world: efficient complete solutions for distributed multi-event scheduling. In *AAMAS*, 2004.

[6] A. Meisels and O. Lavee. Using additional information in DisCSPs search. In *DCR*, 2004.

[7] P. J. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence Journal*, 161:149–180, 2005.

[8] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *IJCAI*, 2005.

[9] M. C. Silaghi and B. Faltings. A comparison of distributed constraint satisfaction approaches with respect to privacy. In *DCR*, 2002.

[10] M.C. Silaghi. Meeting scheduling guaranteeing n/2-privacy and resistant to statistical analysis (applicable to any discsp). In *3rd IC on Web Intelligence*, 2004.

[11] M. Yokoo, K. Suzuki, and K. Hirayama. Secure distributed constraint satisfaction: Reaching agreement without revealing private information. In *CP 2002*, Berlin, 2002.