Analysis of Privacy Loss in Distributed Constraint Optimization

Rachel Greenstadt[*]
Harvard University
(greenie@eecs.harvard.edu)

Jonathan P. Pearce and **Milind Tambe**[†]
University of Southern California
{jppearce,tambe}@usc.edu

Abstract

Distributed Constraint Optimization (DCOP) is rapidly emerging as a prominent technique for multiagent coordination. However, despite agent privacy being a key motivation for applying DCOPs in many applications, rigorous quantitative evaluations of privacy loss in DCOP algorithms have been lacking. Recently, [Maheswaran *et al.*2005] introduced a framework for quantitative evaluations of privacy in DCOP algorithms, showing that some DCOP algorithms lose more privacy than purely centralized approaches and questioning the motivation for applying DCOPs. This paper addresses the question of whether state-of-the art DCOP algorithms suffer from a similar shortcoming by investigating several of the most efficient DCOP algorithms, including both DPOP and ADOPT. Furthermore, while previous work investigated the impact on efficiency of distributed contraint reasoning design decisions (e.g. constraint-graph topology, asynchrony, message-contents), this paper examines the privacy aspect of such decisions, providing an improved understanding of privacy-efficiency tradeoffs.

Introduction

Understanding agents' privacy loss in multiagent coordination and conflict resolution is emerging as a critical issue in many applications. For example, personal assistant agents deployed to facilitate collaboration in businesses, office environments and research organizations [Modi & Veloso2005, Hassine, Defago, & Ho2004, Maheswaran *et al.*2004] must possess potentially private information about their users, e.g. salary, capabilities, and preference information about meetings and schedules. In the course of negotiations and conflict resolutions, the exchange of some private information is necessary to achieve a good team outcome. For humans to entrust their personal assistant agents with private information, they need assurance that their privacy will be protected. Such privacy loss considerations are important during resource allocation negotiations even in highly cooperative domains such as disaster rescue where several government, corporate and nonprofit groups may come together. Although these organizations wish to contribute resources that will lead to the optimal end, they are mutually distrustful in most of their other endeavors and do not want other entities to know the details of their individual constraints. A lack of understanding about privacy loss could undermine collaboration in these settings.

Maintaining privacy is a fundamental motivation for work in distributed constraint optimization (DCOP) [Maheswaran *et al.*2004,Modi *et al.*2005,Silaghi & Faltings2002]. Several recent approaches to DCOP [Modi *et al.*2005, Maheswaran *et al.*2004], attempt to enable distributed conflict resolution and coordination while maintaining users' privacy. However, privacy loss analysis indicates that these algorithms preserve less privacy than a centralized approach [Maheswaran *et al.*2005, Maheswaran *et al.*2006]. One approach to privacy in DCOP is to use cryptographic techniques [Yokoo, Suzuki, & Hirayama2002]. These techniques ensure watertight privacy but require the use of external servers or computationally intensive secure function evaluation techniques that may not always be available or justifiable for their benefits.

In fact, it is useful to understand that the DCOP problem has three axes of tradeoffs: efficiency, privacy and optimality. Rather than requiring both optimality and watertight privacy (as above) at the expense of efficiency, this paper pursues an alternative approach to this tradeoff, where we maintain optimality but are willing to engage in some privacy-efficiency tradeoffs. To understand the nature of this tradeoff, the first step is to understand the privacy loss in existing DCOP algorithms using quantitative metrics [Franzin *et al.*2004, Silaghi2004, Meisels & Lavee2004, Maheswaran *et al.*2005]. If we can bound privacy loss in specific DCOP algorithms, then cryptographic techniques may be avoidable in situations where they are impractical.

There are two key weaknesses in the previous work. First, it does not provide insight on *how* various design choices impact privacy in these algorithms. Such design decisions as constraint-graph topology, asynchrony, and message content have been shown to affect efficiency, but their impact on pri-

[*]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.

vacy has not been studied. Second, recent cross-algorithm privacy loss analysis focused on a limited number of DCOP algorithms [Maheswaran *et al.*2005]. This limitation led to the false conclusion that DCOP algorithms provide less privacy than a centralized approach.

The work reported in this paper addresses both of these weaknesses. It analyzes the privacy impact of DCOP design decisions, including constraint-graph topology, asychrony and message-contents, and analyzes ADOPT [Modi *et al.*2005], DPOP [Petcu & Faltings2005] and SynchID [Modi *et al.*2005], three recent DCOP algorithms which are heavily used for their efficiency. We overturn the significant negative results from [Maheswaran *et al.*2005] by providing positive privacy results for the above DCOP algorithms not considered in that work. These contributions are obtained by a large-scale experimental investigation of privacy loss in DCOP algorithms in the VPS (Valuations of Possible States) analysis framework [Maheswaran *et al.*2005], using several distributed meeting scheduling scenarios with each data point averaged over 25 runs. Overall, while our results are more promising than [Maheswaran *et al.*2005], we also investigated upper bounds on privacy loss in DCOP algorithms which indicate the need for further attention to privacy preservation.

Background

A DCOP consists of a set of variables assigned to agents who control their values. The agents must coordinate their local choices of variable values so that a global objective function, modeled as a set of distributed valued constraints, is optimized. DCOPs are often represented as graphs, where nodes are variables and edges join variables involved in a constraint. We can then define a cost function over each constraint. The objective is to find an assignment of variables such that the total cost is minimized.

A predecessor to the recently introduced algorithms mentioned above, SynchBB [Hirayama & Yokoo1997] 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 DPOP, ADOPT, and SynchId. These algorithms were chosen because they present novel design choices, or occupy a prominent place in the algorithmic space.

ADOPT is an asynchronous complete DCOP algorithm, guaranteed to find the optimal solution. In ADOPT, an agent communicates only 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 main differences: agents are organized into a chain, not a tree, and messages are sent synchronously. We initially believed that asynchrony would be harmful to privacy since more messages are sent.

DPOP 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 [Maheswaran *et al.*2005]. However, we focus on a slightly modified SynchBB where information irrelevant to the problem is not communicated.

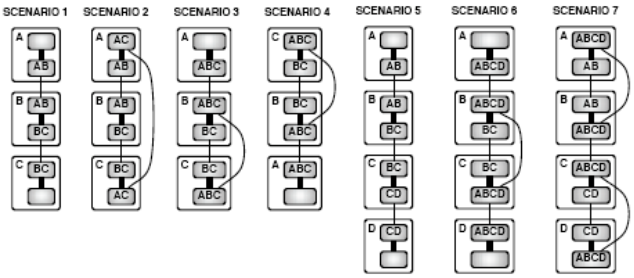

Figure 1: Scenarios: Transparent boxes represent agents and the dark, inner boxes are meeting variables. Thick lines are intra-agent contraints and thin lines are inter-agent constraints.

Experimental Methodology

We focus our investigation on privacy loss in the distributed meeting scheduling problem, since this domain presents inherent privacy concerns [Maheswaran *et al.*2005, Franzin *et al.*2004]. 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 formalism of [Maheswaran *et al.*2004], expressed using the PEAV-DCOP representation, which is motivated by privacy considerations.

- $\mathcal{R} := \{R_1, \ldots, R_N\}$ is a set of N agents.
- $\mathcal{E} := \{E^1, \ldots, E^K\}$ is a set of K events.
- $\mathcal{T} := \{1, \ldots, T\}$ is the set of available timeslots.
- $E^k := (A^k, V^k)$ is the k^{th} event, where $A^k \subset \mathcal{R}$ are the required attendees and $V^k := \{V_1^k, \ldots, V_N^k\}$ is a value vector, where V_n^k represents the value to the n^{th} person for attending event k.
- $V_n^0(t) : \mathcal{T} \rightarrow \mathbb{V}$ denotes the n^{th} person's valuation for keeping time slot t free, due to a preference to keep that time open or the value of an already scheduled event, where $\mathbb{V}$ is a discrete set.

The goal is to schedule meetings maximizing the SUM $\Sigma(V_n^k - V_n^0(t))$, where the $V_n^k - V_n^0(t)$ is an agent's utility scheduling event k at time t.

Scenarios: 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 [Franzin *et al.*2004, Modi *et al.*2005]. We consider seven scenarios of three ($R = \{A, B, C\}$) or four ($R = \{A, B, C, D\}$) agents. The PEAV-DCOP graphs in Figure 1 show the events, labeled by their attendees, and decomposed into variables and constraints.

Measuring Privacy Loss: In order to measure the privacy loss in a system, we first consider how to measure from one agent R_i to another agent R_j. Then, we combine these measures to determine system-wide privacy loss.

Measuring Privacy Loss Between Two Agents: We make use of the Valuation of Possible States (VPS) framework [Maheswaran *et al.*2005] to quantitatively evaluate privacy loss between a pair of agents: from R_i to R_j. Privacy loss in VPS is based on a valuation on R_j's estimates about (i.e. a probability distribution over) R_i's possible states. In VPS, agent R_i's private information is modeled as a state $s_n \in S_n$, where S_n is a set of possible states that R_i may occupy. R_j estimates about agent R_i's possible states are expressed as a probability distribution $\mathbb{P}_n((S_n)^j)$. The utility that agent R_i derives from R_j's beliefs about R_i's states yields value function $\mathbb{V}_i(\mathbb{P}_i((S_n)^j))$.

Now if we apply VPS to the meeting scheduling problem, R_j knows only that R_i exists in one of $|\mathbb{V}|^{\mathcal{T}}$ possible states after running the DCOP algorithm. R_i is modeled by R_j whose estimate of R_i is captured by $\mathbb{P}_i((S_n)^j)$. Different possibilities for how these values may be assigned are captured in six metrics introduced in [Maheswaran *et al.*2005] that define the privacy of an agent R_i with respect to R_j.

Due to the nature of messaging in DCOPs, the typical form of information gathered is the elimination of a possible state. We define the total number of states as S and the states remaining at the end of the inference as s_r. We scale privacy loss between 0 and 1. The six metrics (from [Maheswaran *et al.*2005] we use are as follows: *LinearS* gives the number of states not eliminated by other agents: $\frac{S-s_r}{S-1}$. *GuessS* gives the probability that R_j will be able to guess the state of R_i accurately from the among the eliminated states: $\frac{1}{S-1} * \frac{S}{s_r}$. *EntropyS* was introduced in [Franzin *et al.*2004] and considers privacy loss from an information-theoretic perspective: $1 - \frac{lg(s_r)}{lg(S)}$. In meeting scheduling, one way to view R_i's state is the vector of possible valuations it has for each of its timeslots. So, in a scenario with 3 timeslots and 4 valuations, S would be $4^3 = 64$. We could also consider the states in a per timeslot way, in which case $S = 4$, and then average the results over all possible timeslots. When doing this, we refer to the metrics as *LinearTS*, *GuessTS* and *EntropyTS*.

Measuring System-wide Privacy Loss: Once we have measures of privacy loss between all pairs of agents, we must aggregate them into a measure of the privacy loss of the whole system. One way to do this is to average the privacy loss between all pairs of agents R_1 to R_N in the system. This is the approach taken by [Maheswaran *et al.*2005].

In this AVERAGE method, a centralized algorithm has a privacy of $\frac{1}{N}$, and the privacy approaches infinity as the number of agents increases. The effect of one agent learning more than others, and gaining an asymmetric advantage over them, is not considered. To address this issue, we devised the MAX aggregation method. In MAX, we consider only the total privacy loss to the single agent that learns the most information about other agents, rather than the mean of all pairs of agents. This method is also relevant when there is concern that an agent might reveal information outside of the group of collaborators.

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. Previous work in constraint satisfaction has considered private information to be whether an agent can attend a meeting [Wallace & Freuder2005]. This makes sense in a CSP framework with hard constraints. In DCOP we are optimizing soft constraints, so the private information is expressed in weighted valuations of each timeslot.

The assumption that agents know the existence and importance of all meetings to be scheduled comes from common situations in universities and research labs where some fixed project meetings and/or student- advisor meetings are well known. (Already scheduled meetings, such as people's personal meetings would be represented as timeslot valuations, which are considered private.) If this assumption is removed, our framework still applies, but the results would show less privacy loss, thus strengthening our conclusions.

We assume that agents do not deviate from the protocols specified by the algorithms since (1) deviating from the protocol may lead to suboptimality and involve other messaging costs; (2) if agents are discovered, they may face sanctions. Investigating situations where the agents deviate from the protocol is a worthwhile area for future work.

These assumptions are exactly as in [Maheswaran *et al.*2005], allowing comparison of the results. Based on these assumptions, we developed methods for agent inference for SynchID, ADOPT and DPOP.

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} meeting at time t. When agent R_n receives an upward message it knows that m_n = a sum of Δ terms from agents 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 1. 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 lower in the chain 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 Δ terms 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:

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

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 [Maheswaran *et al.*2005].

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 [Maheswaran *et al.*2005].

Experimental Results

In this section, we present experimental results from the seven scenarios. We begin with comparisons of privacy loss in the algorithms according to the EntropyTS metric, then examine the algorithms using all metrics from [Maheswaran *et al.*2005]. We introduce a new aggregation method (MAX) to highlight privacy benefits of all studied DCOP algorithms over centralized approaches. We then explore the privacy impact of more sophisticated inference techniques and diverse topologies.

For the three-agent scenarios, we varied $|\mathbb{V}|$ from 3 to 7 while holding the number of timeslots $T = 3$. For the four-agent scenarios, for reasons of computational complexity, we varied $|\mathbb{V}|$ from 3 to 5 while holding $T = 3$. For each $(T, |\mathcal{V}|)$ pair, we performed 25 runs for each of the following algorithms: SynchID, ADOPT, SynchBB and DPOP. The privacy loss for each pair of agents was measured using all six metrics, assuming the agents were using the inference algorithms given in Section 3. We aggregate systemwide privacy loss using the AVERAGE and MAX methods.

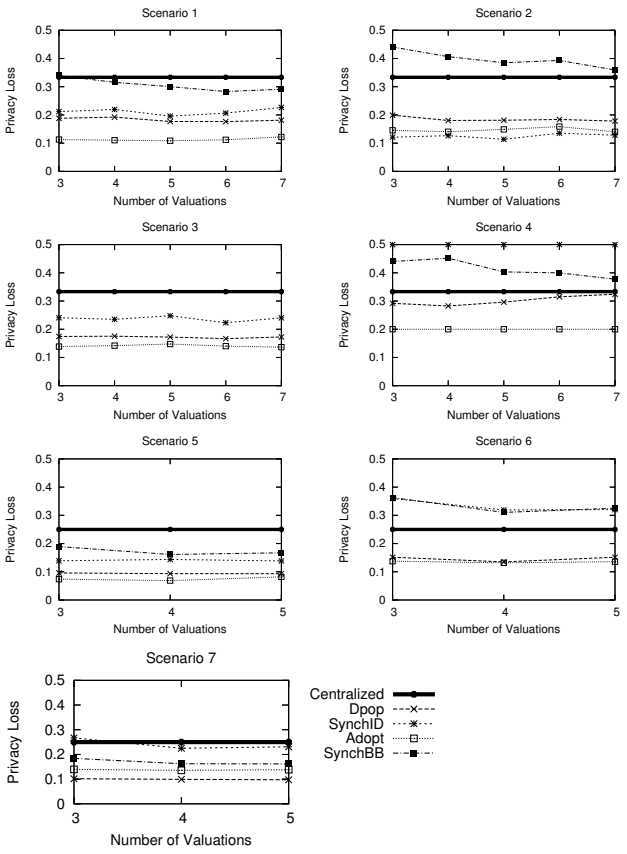

Figure 2: Privacy loss comparisons between algorithms.

Space limitations preclude us from presenting all our results (additional results are available at http://teamcore.usc.edu/dcop/aaai06). Thus, in some cases, we present results from only some of the seven scenarios. Each data point is an average of 25 runs, and we provide statistical significance results to support our main conclusions. When directly comparing algorithms, we use a chain topology (since not all algorithms could use a tree); we investigate the impact of graph topology on privacy in a separate experiment.

Cross-algorithm comparison: Figure 2 shows the comparison of privacy loss for the four algorithms mentioned above, for each of the seven scenarios, as well as the centralized approach. The x-axis plots the different number of valuations and the y-axis plots privacy loss. The thick horizontal line shows the centralized approach, for scenarios 1-4 (three agents), its privacy loss is 0.33, but for scenarios 5-7 (four agents) it is 0.25. The privacy loss in the centralized case is the same no matter which of the six metrics is used to measure it. We use the EntropyTS metric as the metric for privacy loss in this result; as seen later, EntropyTS provides results that are in the mid-range among all metrics. Results are aggregated using the AVERAGE method.

We conclude the following from Figure 2: (1) Except for SynchBB, the remaining algorithms have a privacy loss that is lower than the centralized approach. In contrast with the negative results presented in [Maheswaran *et al.*2005],

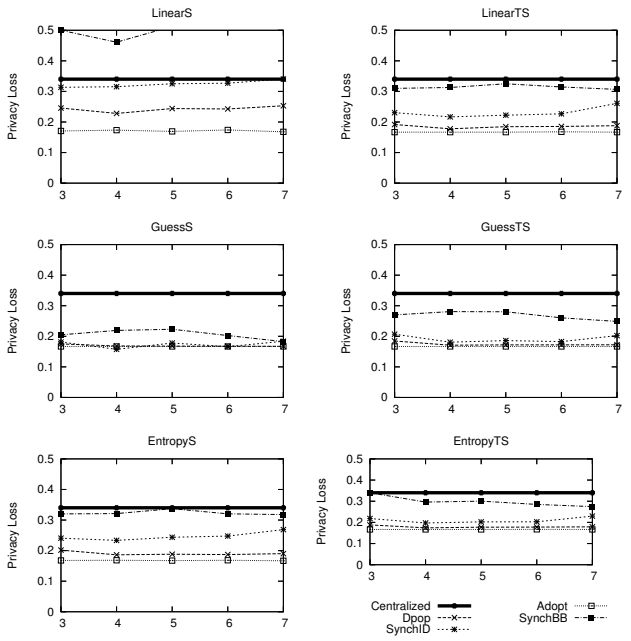

Figure 3: Cross-metric comparisons, scenario 1

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 except scenario 4 (significance level of 5%). (2) DPOP and ADOPT had very similar privacy loss, despite their vastly different approaches. In particular, 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 (see in particular Scenario 4), but not to the level anticipated, at least in these scenarios. (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—we removed unnecessary context messages—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.

Cross-metric comparison: Figure 3 shows results comparing the algorithms' privacy loss for Scenario 1 according to each of the metrics from section 3. We conclude the following from Figure 3: (1) Even if we examine other metrics beyond EntropyTS, DCOP algorithms do not suffer from privacy loss to the extent seen in the earlier investigation [Maheswaran *et al.*2005], further confirming the positive results seen earlier. (2) The choice of metric affects how the algorithms compare to the centralized approach (with LinearS suggesting the most privacy loss), but seems to preserve the qualitative ranking of the algorithms.

MAX method: Figure 4 shows the results for all the algorithms aggregated by the MAX method for all seven scenarios, with $T = 3$. The number of valuations is plotted on the *x*-axis and the privacy loss is plotted on the *y*-axis. The MAX method shows that there is always a privacy benefit obtained by using DCOP algorithms, even those that, by the AVERAGE methode (and any of the six metrics) perform worse than the centralized approach, if the major privacy concern is one agent accumulating too much knowledge. Under the MAX method, DPOP and ADOPT continued to outperform SynchBB, while SynchID varied widely.

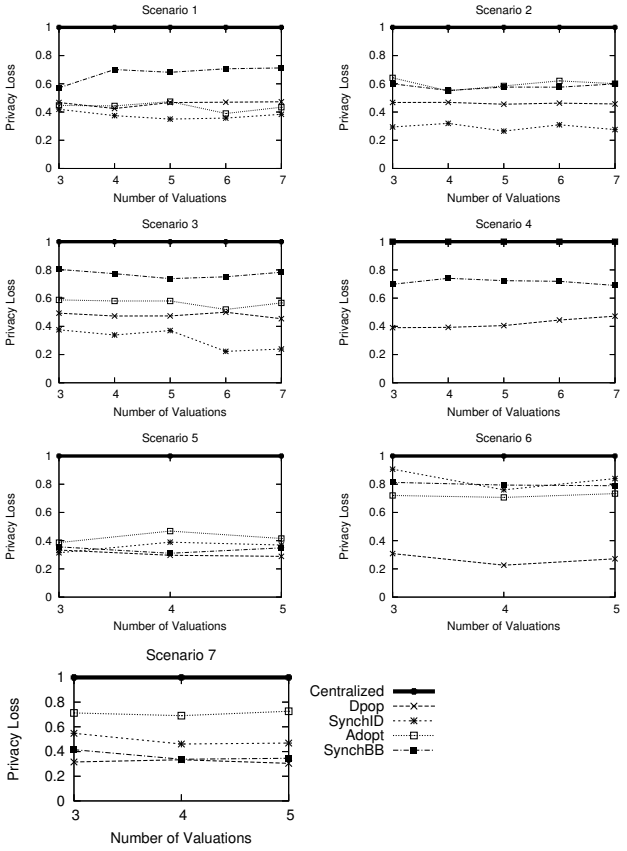

Figure 4: Algorithms compared using the MAX metric: Centralized has a privacy loss of 1

Upper bounds: The results so far all used the inference algorithms described earlier, and provided a lower bound on privacy loss, since they only considered each message in isolation. Although lower bounds are sufficient to demonstrate a negative result, they must be augmented to demonstrate a positive one. While there is no theoretical limit to the quantity of domain knowledge an inference algorithm may possess (making a tight upper bound impossible to calculate),

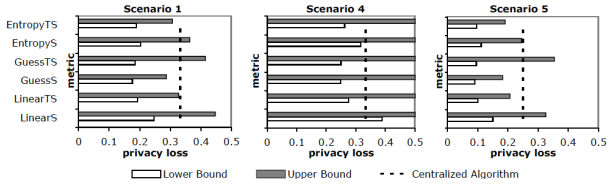

Figure 5: DPOP: upper bound results compared to lower bound and centralized

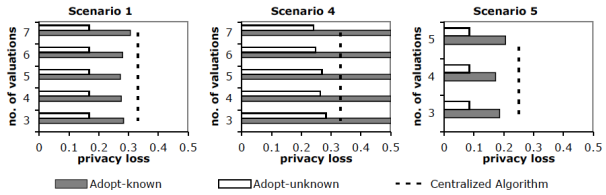

Figure 6: ADOPT, assuming cost message participants are known, compared with the unknown case and centralized

we can calculate upper bounds assuming agents know only the message contents and graph structure.

We calculated upper bounds on privacy loss for one of the most promising DCOP algorithms: DPOP. We used a brute force approach which generated all possible combinations of input valuations, ran DPOP on them to generate a trace of the messages each combination would produce and then for each agent matched these up to the messages that were actually received. We performed simulations of this type for DPOP, which took several days to run, compared to the several hours taken by our primary inference algorithms. Results for upper and lower bound inference for DPOP using all six metrics are shown in Figure 5. Scenarios 1, 4, and 5 are shown for three timeslots and three valuations, with the metric plotted on the *y*-axis and the privacy loss plotted on the *x*-axis. Due to asynchrony and the randomness in a variable's initial choice of value, it is not possible to analyze ADOPT with this approach.

For each scenario, the lower bound showed DPOP outperforming the centralized approach (except on LinearS for Scenario 4) while the upper bound was comparable or worse than the centralized approach. We conclude that while privacy results on recent DCOP algorithms are encouraging, there is still a need for improvement.

Asynchrony: The privacy loss of an asynchronous algorithm such as ADOPT is difficult to analyze. Due to its asynchrony, it may be difficult for agents to ascertain which (or even how many) other agents' valuations are part of any particular cost message. However, implementation artifacts can make this information easier to infer. For instance, one technique to implement the meeting scheduling problem as a DCOP solvable by ADOPT requires all rewards to be converted into costs by subtracting all rewards from a large offset number. If this number is high enough, agents can determine the number of valuations in a cost message by rounding the cost in the message to the nearest multiple of this number. The results for ADOPT assuming the participants in each cost message are known are presented in Figure 6. Results for scenarios 1, 4 and 5 are shown with $T = 3$, the number of valuations plotted on the *y*-axis and the privacy loss according to the EntropyTS metric on the *x*-axis.

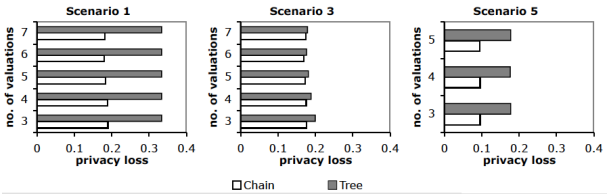

Figure 7: DPOP: tree vs. chain

Figure 6 shows that the privacy loss in ADOPT is much higher if the agents involved in a cost message are known to the inferring agent. Thus, if privacy is a goal, care must be taken with the implementation of distributed algorithms such as ADOPT to ensure that the privacy benefits of asynchrony are realized. In fact, we observe that SynchID generally falls between ADOPT with message participants revealed and without this information revealed (Figure 6).

Topology: While we held the constraint graph topology fixed as a chain in our experiments so far, this experiment investigates the impact of graph topology on privacy loss. Indeed, DPOP and ADOPT were designed to be run on trees, not chains, and gain much of their efficiency from that distinction. Figure 7 shows the results of running DPOP on a tree topology, as compared to the chain. The tree was built by chosing the most constrained agent as the root, then adding other agents lexicographically. Results for scenarios 1, 3 and 5 are shown with $T = 3$, the number of valuations plotted on the *y*-axis and the privacy loss according to the EntropyTS metric on the *x*-axis. In all cases, privacy loss using a tree was higher than that of a chain. This occurs because agents at the top of a tree will receive information aggregated from smaller groups of agents, due to the parallelism of the tree. This result shows the privacy-efficiency tradeoff in the design of DCOP algorithms. While trees provide improved efficiency, they led to more privacy loss than chains in the scenarios tested.

Conclusion

DCOP is rapidly emerging as a tool for multiagent coordination. Previous work [Maheswaran et al.2006] 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 a large-scale investigation of several leading algorithms, including ADOPT and DPOP, and overturns earlier negative results. Furthermore, we investigated the privacy side of the privacy/efficiency tradeoff in DCOP design decisions and concluded in addition: (i) Asynchrony in ADOPT improves privacy by obscuring the identities of agents involved in a message and by making sophisticated inference difficult. This is offset to a degree by its use of more messages. (ii) Topology has significant impact on system-wide privacy loss. (iii) Measures of information centralization (MAX) show DCOP algorithms outperforming a centralized approach, with DPOP, SynchID and ADOPT performing best. Finally, sophisticated inferences (e.g. our upper bounds) indicate that there is still work to be done in reducing privacy loss.

References

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

Hassine, A.; Defago, X.; and Ho, T. 2004. Agent-based approach to dynamic meeting scheduling problems. In *AAMAS*.

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

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

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

Maheswaran, R. T.; Pearce, J. P.; Bowring, E.; Varakantham, P.; and Tambe, M. 2006. Privacy loss in distributed constraint reasoning: A quantitative framework for analysis and its applications. *Journal of Agents and Multiagent Systems (to appear)*.

Meisels, A., and Lavee, O. 2004. Using additional information in DisCSPs search. In *Workshop on Distributed Constraint Reasoning*.

Modi, P. J., and Veloso, M. 2005. Bumping strategies for the multiagent agreement problem. In *AAMAS*.

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

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

Silaghi, M. C., and Faltings, B. 2002. A comparison of distributed constraint satisfaction approaches with respect to privacy. In *Workshop on Distributed Constraint Reasoning*.

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

Wallace, R., and Freuder, E. 2005. Constraint-based reasoning and privacy/efficiency tradeoffs in multi-agent problem solving. *AIJ* 161(1-2):209–227.

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