An Overview of Privacy Improvements to k-Optimal DCOP Algorithms

(Extended Abstract)

Rachel Greenstadt
Dept of Computer Science
Drexel University
Philadelphia, PA
greenie@cs.drexel.edu

ABSTRACT

For agents to be trusted with sensitive data, they must have mechanisms to protect their users' privacy. This paper explores the privacy properties of k-optimal algorithms: those algorithms that produce locally optimal solutions that cannot be improved by changing the assignments of k or fewer agents. While these algorithms are subject to large amounts of privacy loss, they can be modified to reduce this privacy loss by an order of magnitude. The greatest improvements are achieved by replacing the centralized local search with a distributed algorithm, such as DPOP.

Categories and Subject Descriptors

I.2.m [**Computing Methodologies**]: Artificial Intelligence—*Miscellaneous*

General Terms

Security,Algorithms

Keywords

Distributed Constraint Optimization,Security and Privacy

1. INTRODUCTION

Many multi-agent systems require solutions to constraint optimization problems, such as resource allocation, supply chain negotiation and meeting scheduling. For these problems, the constraints are often extremely sensitive, representing personal or proprietary information of the agents. As a result, measuring and improving the privacy of distributed constraint optimization (DCOP) algorithms has been the subject of much recent research [5, 3, 2, 8]. When the coordination problem is so large and/or dynamic that complete algorithms are infeasible, a local approximation is often the best option [6, 4]. Local algorithms are most useful in situations with large numbers of agents coordinating large amounts of data, in which some agents and constraints might be dynamic over time. In such situations, all participants in the computation cannot be assumed to be trustworthy and discreet.

This paper analyzes the privacy properties of currently proposed k-optimal algorithms, showing experimentally that they often distribute private constraint information to far more than k agents. This paper discusses two approaches to improving privacy in k-optimal algorithms: (1) To run a k-optimal algorithm, groups of k agents must selected to deviate from the current solution. We propose a method of choosing these groups so as to avoid unnecessary information sharing. (2) These groups of k agents must run a complete constraint optimization algorithm to solve their local problem. In current algorithms, this is a centralized computation, however, increased privacy can be achieved by replacing it with a distributed algorithm. With these modifications, k-optimal algorithms can provide better privacy than many complete algorithms, enabling designers to consider a three-way tradeoff between privacy, efficiency, and optimality.

2. PRIVACY IN LOCAL ALGORITHMS

1**-optimal Algorithms.** In these algorithms [9, 1], each agent sends its current value to its neighbors[1], then uses the values received to determine if it can modify its value and increase global utility. 1-optimal algorithms have excellent privacy properties. They are a good choice for situations where privacy and efficiency are more important than solution quality. Because each agent acts alone, no agent needs to learn of the existence of any constraints other than its own, nor of their values.

k**-optimal Algorithms.** These algorithms have been extended to allow groups of k agents to coordinate to find a local solution, not just individual agents. The MGM (Maximum Gain Message) algorithm extends DBA and SCA (Stochastic Coordination Algorithm) extends DSA to coordinate local groups of two or three agents [6]. These algorithms go through a similar set of three phases as the 1-optimal algorithms. In the first phase, agents are organized into groups of k agents and agents send their constraint information to a mediator. In the second phase, the mediator finds the optimal solution available by varying the values for the group members and considering their neighbors' values as fixed in the current context. In the third phase, the agents decide whether to act.

Extending to general k requires a process for gathering constraint information from agents that may not be neighbors of the mediator. The KOPT algorithm [4] accomplishes this by augmenting the initial phase with an information gathering section where information is gathered from agents up to $\lfloor \frac{k}{2} \rfloor$ links away from the mediating agent in the graph. A better solution only iterates the information gathering algorithm if the mediating agent has the information of fewer than k agents.

Figure 1 shows the effect of implementing this change. The results for $k = 4$ now mostly track with $k = 2$, only neighbors'

[1] Agents who share constraints.

Cite as: An Overview of Privacy Improvements to k-Optimal DCOP Algorithms, (Extended Abstract), Rachel Greenstadt, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1279–1280

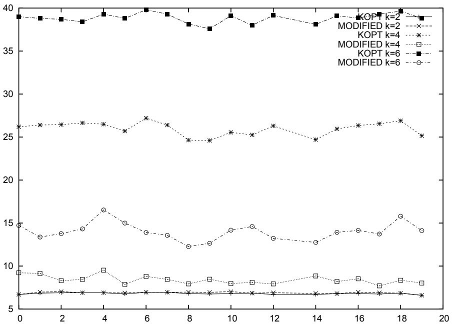

Figure 1: Privacy loss in KOPT vs. MODIFIED algorithm: X-axis—randomized 40 variable, 120 constraint scenarios. Y-axis—average number of agents whose constraint information is learned by the average agent.

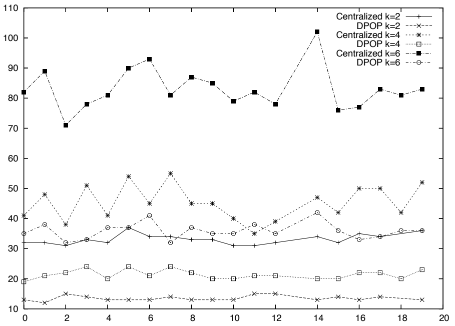

Figure 2: Privacy loss using a centralized algorithm vs. DPOP: X-axis—40 variables, 120 constraint scenarios. Y-axis: average number of constraints learned by the average agent.

information is learned, except for a few agents which require otherwise. In the $k = 6$ results, we see dramatic improvements since sometimes one set of neighbors is needed and sometimes a second iteration, but never three. Using this algorithm for identifying potential group members, it is not until $k = 14$ that a third iteration is routinely needed, and even then, it is only needed for a few agents who have few neighbors. Using this algorithm causes the privacy loss to track much closer to k.

Replacing the Search Algorithm. Current k-optimal algorithms use a centralized search to do the local optimizing. However, it is a simple matter to replace this search with a distributed algorithm [4]. As previous work has shown that some distributed algorithms can provide better privacy to agents than centralized algorithms [3].

We replaced the search algorithm with DPOP [7]. DPOP has fairly low privacy loss [3], which is concentrated at the bottom of the tree or chain. This privacy loss occurs because the bottom agent sends its utility information unaggregated with that of any other agents. In a k-optimal calculation, however, this vulnerability is greatly lessened because the the constraint information is aggregated with information for other constraints that are outside the group and whose existence and values are unknown to the other agents. This effect is similar to that of uncertainty about the constraint graph in studies of privacy in complete DCOPs [5].

The existence of constraints can be learned by an adverarial agent when the agent below it, B, is the bottom agent in the chain and some of B's neighbors are in the group. These neighbors will then appear in the utility message sent up by B. Higher agents in the chain will not learn about the existence of constraints because agents in utility messages sent to them could be from any agent below them in the chain. Figure 2 shows the average number of constraints learned by the average agent using DPOP compared to the number of constraints learned by agents in the centralized case. As Figure 2 shows, using DPOP reduces the average number of constraints learned by approximately half. For DPOP, these numbers refer to agents learning only of the existence of these constraints, not their associated utilities, whereas for the centralized case all constraint information is learned. Further reductions were found when utility information was considered: DPOP improves upon a centralized algorithm by roughly an order of magnitude.

3. CONCLUSION

By paying attention to the way k is chosen and the internal search algorithm used, we can reduce the privacy loss in local algorithms by an order of magnitude. Further improvements might be obtained by using secure multiparty computation (SMC) as the internal search algorithm for a k-optimal solution.

We show that distributing the internal algorithms is an effective means of improving the privacy of the constraint graph and the constraints themselves (the utilities that agents achieve for a particular assignment). The cost of this change is in a linear increase in the number of messages with k and its performance impact will be determined by the network latency of the environment in which the algorithm is run. The UTIL messages are of size exponential in k, and the impact of their size will be primarily determined by the throughput of the network environment. These results enable designers of multi-agent systems to consider a three-way tradeoff between privacy, efficiency, and optimality when choosing coordination algorithms for agents in their systems.

Acknowledgements. Supported by the Harvard Center for Research in Computation and Society and Drexel University.

4. REFERENCES

[1] S. Fitzpatrick and L. Meertens. Distributed coordination through anarchic optimization. In *Distributed Sensor Networks: A Multiagent Perspective*. Kluwer Academic Publishers, 2003.

[2] R. Greenstadt, B. Grosz, and M. Smith. Ssdpop: Improving the privacy of dcop with secret sharing. In *DCR*, 2007.

[3] R. Greenstadt, J. Pearce, and M. Tambe. An analysis of privacy loss in distributed constraint optimization. In *AAAI*, 2006.

[4] H. Katagishi and J. Pearce. Kopt: Distributed dcop algorithm for arbitrary k-optima with monotonically increasing utility. In *DCR*, 2007.

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

[6] R. Maheswaran, J. Pearce, and M. Tambe. Distributed algorithms for dcop: A graphical-game-based approach. In *PDCS*, 2004.

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

[8] M. Silaghi and D. Mitra. Distributed constraint satisfaction and optimization with privacy enforcement. In *IAT*, 2004.

[9] M. Yokoo and K. Hirayama. Distributed breakout algorithm for solving distributed constraint satisfaction problems. In *ICMAS*, 1996.