WHEN YOU SAY (DCOP) PRIVACY, WHAT DO YOU MEAN?
Categorization of DCOP Privacy and Insights on Internal Constraint Privacy

Tal Grinshpoun
Department of Software Engineering, SCE – Sami Shamoon College of Engineering, Beer-Sheva, Israel

Keywords: Distributed Constraint Optimization, Constraint Privacy.

Abstract: Privacy preservation is a main motivation for using the DCOP model and as such, it has been the subject of comprehensive research. The present paper provides for the first time a categorization of all possible DCOP privacy types. The paper focuses on a specific type, internal constraint privacy, which is highly relevant for models that enable asymmetric payoffs (PEAV-DCOP and ADCOP). An analysis of the run of two algorithms, one for ADCOP and one for PEAV, reveals that both models lose some internal constraint privacy.

1 INTRODUCTION

Constraint optimization (Meseguer and Larrosa, 1995) is a powerful framework for describing optimization problems in terms of constraints. In many real-world problems, such as Meeting Scheduling problems and Mobile Sensor nets, the constraints are enforced by distinct participants (agents). Such problems were termed Distributed Constraint Optimization Problems (DCOPs) (Hirayama and Yokoo, 1997) and in the past decade various algorithms for solving DCOPs were proposed (Modi et al., 2005; Mailler and Lesser, 2004; Petcu and Faltings, 2005; Gershman et al., 2009). While these algorithms compete in terms of reducing the run-time and/or communication overhead, they maintain the inherent distributed structure of the original problem. Nonetheless, up until recently there was no evidence that DCOP algorithms outperform (in terms of run-time) state-of-the-art centralized COP methods. This observation was recently challenged by methods that rely on GDL (Stefanovitch et al., 2011; Vinyals et al., 2011). However, in these new set of algorithms the original distribution of variables among agents is not necessarily preserved. In fact, the "distribution" of variables among agents in these algorithms is actually only for the means of *parallelization* of the problem solving process. Hence, this new group of algorithms can be referred to as Parallel COP algorithms as opposed to Distributed COP algorithms.

Following the subsequent observation, the necessity of "classical" DCOP search remains questionable – if not for performance considerations, then what do we need DCOP for? The answer to that is obviously the will to preserve the original distributed structure of the problem. And why should one try to preserve this structure? The answer to the second question is *privacy*. Indeed, much of DCOP research throughout the years has focused on privacy issues (Maheswaran et al., 2006; Greenstadt et al., 2007; Silaghi and Mitra, 2004). It turns out that the term privacy is too general and almost every work on DCOP privacy considered a slightly different angle of this term.

The first contribution of the present paper is a categorization of the various aspects of DCOP privacy. The given categorization indicates that probably the most interesting privacy aspect in real-world DCOPs is *constraint privacy*, and indeed most previous work on DCOP privacy actually refers to constraint privacy. This aspect is further sub-categorized into *internal* and *external* constraint privacy.

The paper's second contribution regards internal constraint privacy that has not been widely studied. Two models that do consider internal constraint privacy are examined – PEAV-DCOP (Maheswaran et al., 2004) and ADCOP (Grubshtein et al., 2010). Some important insights regarding internal constraint privacy preservation in these models are given.

The rest of the paper is organized as follows. Section 2 formally describes the DCOP model. A categorization of privacy types is given in section 3, and is followed by a review of previous work according to the proposed categories (section 4). Section 5 overviews the existing models that enable asymmetric payoffs, while section 6 exhibits internal constraint privacy loss in these models. The conclusions and ideas for future work are given in section 7.

Grinshpoun T..
WHEN YOU SAY (DCOP) PRIVACY, WHAT DO YOU MEAN? - Categorization of DCOP Privacy and Insights on Internal Constraint Privacy.
DOI: 10.5220/0003752203800386
In *Proceedings of the 4th International Conference on Agents and Artificial Intelligence* (ICAART-2012), pages 380-386
ISBN: 978-989-8425-95-9

2 DCOP DEFINITION

A *DCOP* is a tuple $< \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R} >$. $\mathcal{A}$ is a finite set of agents $A_1, A_2, ..., A_n$. $\mathcal{X}$ is a finite set of variables $X_1, X_2, ..., X_m$. Each variable is held by a single agent (an agent may hold more than one variable). $\mathcal{D}$ is a set of domains $D_1, D_2, ..., D_m$. Each domain D_i contains a finite set of values which can be assigned to variable X_i. $\mathcal{R}$ is a set of relations (constraints). Each constraint $C \in \mathcal{R}$ defines a non-negative *cost* for every possible value combination of a set of variables, and is of the form:

$$C : D_{i_1} \times D_{i_2} \times \ldots \times D_{i_k} \to \mathbb{R}_+ \tag{1}$$

A *binary constraint* refers to exactly two variables and is of the form $C_{ij} : D_i \times D_j \to \mathbb{R}^+$. A *binary DCOP* is a DCOP in which all constraints are binary. An *assignment* is a pair including a variable, and a value from that variable's domain. A *partial assignment* (PA) is a set of assignments, in which each variable appears at most once. A *solution* is a full assignment of aggregated minimal cost.

3 PRIVACY CATEGORIZATION

In order to categorize the different pieces of data that one may wish to keep private during DCOP solving, we begin with a distinction between two types of private data. The first type is data regarding the original problem to be solved. This type refers to the definition of a DCOP as given in section 2. The second type is data about the search process itself.

When considering the first type of data we refer to the tuple $< \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R} >$. We look at privacy from the point of view of a variable. In most cases this also coincides with the perspective of an agent, as it is a common assumption in DCOP research that every agent holds a single variable. We will refer to cases in which this assumption does not hold in following sections. Thus, we remain with two elements of the tuple that a variable/agent may wish to preserve their privacy – $\mathcal{D}$ and $\mathcal{R}$:

- **Domain Privacy** – agents may desire to keep their domains private. The standard DCOP model usually assumes that the domains of all the agents are fully known, leading in most cases to an a-priory loss of any domain privacy. Nevertheless, agents may decide to abandon this assumption and still use the standard DCOP model. Contrary to that, in Open Constraint Optimization Problems (OCOP) (Faltings and Macho-Gonzalez, 2005) the domains are inherently gradually revealed in an attempt to preserve some domain privacy.
- **Constraint Privacy** – agents may desire to keep the information contained in the constraints private. We distinguish between preserving the constraint information from agents participating in the constraint (*internal constraint privacy*) and preserving the constraint information from other agents (*external constraint privacy*). Various DCOP algorithms differ in their level of external constraint privacy. For instance, in the OptAPO algorithm (Mailler and Lesser, 2004; Grinshpoun and Meisels, 2008) agents disclose all their constraint information to mediator agents, which obviously leads to poor external constraint privacy preservation. The story with internal constraint privacy is more complicated, since the standard DCOP model assumes that a constraint is *totally known* by the agents participating in it. This obviously leads to absolutely no internal constraint privacy preservation. Two variations of the DCOP model – PEAV-DCOP (Maheswaran et al., 2004) and ADCOP (Grubshtein et al., 2010) – allow for preservation of internal constraint privacy. An overview of these models is given in section 5.

The second type of data includes information regarding the current state (assignment) of a variable or the behavior of the agent:

- **Assignment Privacy** – agents may desire to keep the assigned values to their variables private. To achieve this, agents must avoid sending their assigned values to other agents. This category refers to the dynamic changes of assigned values throughout the search process, as well as to the assigned values in the final solution.
- **Algorithmic Privacy** – It is a common assumption in DCOP research that all participating agents/variables run the same algorithm – otherwise, there seems to be no sense for the search process. However, an agent may attempt to "tweak" an algorithm for some personal benefits. For example, consider a scenario in which each change of assignment applies some minor cost to the agent holding the variable. The agents may decide together to run the DSA algorithm (Zhang et al., 2005) with a certain value of the p parameter that controls the likelihood of updating its value if the agent attempts to do so. In this example, an agent may decide to use a lower value of p than initially decided, thus reducing its personal cost during the search process. Naturally, this agent would like to keep the knowledge of its selfish behavior private.

4 RELATED WORK

In light of the above categorization, we go over previous DCOP privacy studies in order to put them in context of the proposed categorization.

A previous study has characterized different types of DCOP privacy loss (Greenstadt et al., 2007). The types are – *initial* vulnerability, *intersection* vulnerability, *domain* vulnerability and *solution* vulnerability. While some of these types may seem similar to our proposed categories, they actually all belong to the same category – constraint privacy. In fact, these all represent different vulnerabilities of DCOP search that may lead to external constraint privacy loss. For example, solution vulnerability refers to the external constraint information that is revealed by the solution itself – in some cases there may be several solutions, and the specific solution that was chosen may reveal some constraint information.

The above study is a very good representative of previous work – most of the studies on DCOP privacy actually focus on external constraint privacy. The same study introduces an algorithm that reduces the privacy loss of all the types except for solution vulnerability (Greenstadt et al., 2007). Solution vulnerability is addressed by another study (Silaghi and Mitra, 2004). Both studies selectively use encryption techniques in strategic stages of the problem solving.

In early research of DCSP and DCOP privacy, every work had a slightly different notion of what constraint privacy is, which resulted in the use of various metrics for privacy loss (Silaghi and Faltings, 2002; Franzin et al., 2004; Silaghi and Mitra, 2004). These metrics were all unified with the introduction of the *Valuation of Possible States* (*VPS*) framework (Maheswaran et al., 2006). This general quantitative framework enables the expression, analysis, and comparison of various constraint-privacy-loss metrics. The VPS framework was later used for a comprehensive privacy-loss analysis of different DCOP algorithms (Greenstadt et al., 2006). Additional research on DCOP external constraint privacy includes using cryptographic techniques for *Vehicle Routing Problems* (Léauté and Faltings, 2011), investigating the privacy-loss in k-optimal algorithms (Greenstadt, 2009), and integrating privacy-loss into the utility function (Doshi et al., 2008).

In cases where the agents sharing a constraint have equal payoffs, they naturally want to prevent *other* agents from learning about their mutual constraints. However, when this constraints symmetry does not exist, the agents are naturally more concerned about their private information not reaching their peers (internal privacy) than not reaching some third-party agents that are not involved in the constraint (external privacy). Consequently, models that enable internal private constraint information are in the center of this paper, and are thoroughly discussed in the following sections. An interesting work on DCSPs has considered internal constraint privacy (Brito et al., 2009). In this work entropy was used as a measure of privacy loss. The same paper also proposes algorithms that account for assignment privacy as well as internal constraint privacy.

The centralized Open Constraint Optimization Problems (OCOP) model (Faltings and Macho-Gonzalez, 2005) assumes that domains may be very large and even infinite, and are thus not revealed a-priorly. In DCOP this is not the case and domains are usually assumed to be known in advance. However, except for values of an agent that are constrained with values of another agent, there is no necessity to reveal all the domains to all the participants before the search process begins. Therefore, in sparse and loose problems most domain information is expected to remain private when using Branch & Bound algorithms. Moreover, the ODPOP algorithm (Petcu and Faltings, 2006) reveals values in a best-first manner and is expected to effectively preserve domain privacy.

To the best of our knowledge there has been no research regarding algorithmic privacy. This is probably due to the fact that most DCOP studies assume that the agents are cooperative (though privacy-preserving). However, a recently proposed model, Quantified DCOP, assumes the existence of adversaries (Matsui et al., 2010). This research direction may potentially lead to algorithmic privacy issues.

5 DCOP WITH ASYMMETRIC PAYOFFS

The standard DCOP model assumes equal payoffs for constrained agents. When such a strong assumption is applied, there is no meaning at all to internal constraint privacy, and the agents holding a constraint only worry about the data of this constraint leaking to other agents (external constraint privacy). However, in many real-life situations constrained agents value differently the results of decisions on constraint issues even if they consider the same constraints. In fact, this is the natural scenario in a typical *Multi-Agent System* (*MAS*). In the meeting scheduling problem for example, agents which attend the same meeting may derive different utilities from it. Moreover, their preferences and constraints regarding its time and location are expected to be different. Several alternatives for representing asymmetric constraints by DCOPs are given next.

5.1 Disclosure of Constraint Payoffs

The simplest way to solve MAS problems with asymmetric payoffs by DCOPs is through the disclosure of constraint payoffs. That is, aggregate values of all agents taking a joint action. However, constraint disclosure a-priorly reveals all internal constraint information, and is thus an inadequate solution.

5.2 Private Events as Variables (PEAV)

The PEAV model (Maheswaran et al., 2004) successfully captures asymmetric payoffs within standard DCOPs. The incurred cost of the PEAV model on an agent involves one mirror variable per each of its neighbors in the constraints graph. Consistency with the neighbors' original variables is imposed by a set of hard equality constraints. One way to represent hard constraints is to assign a cost that is calculated for each specific problem (Maheswaran et al., 2004).

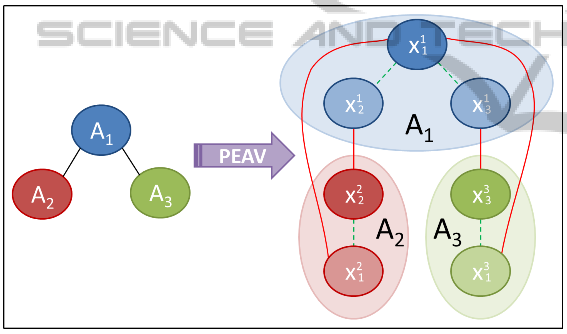

Figure 1: Example of a PEAV-DCOP.

As an example, consider the DCOP on the left-hand side of Figure 1. The presented example includes three agents (each holding a single variable that coincides with the agent's name). When representing this problem as a PEAV-DCOP, each agent needs to produce a mirror variable for each of its neighbors in the original constraints graph. The resulting problem is illustrated in the right-hand side of Figure 1. Here the subscript in each new variable refers to the original variable the new one represents, whereas the superscript refers to the agent in which this new variable resides in. So x_2^1 is the mirror variable of x_2 in agent A_1, and x_1^2 is the mirror variable of x_1 in agent A_2. The solid lines represent the hard equality constraints, while the dashed lines represent the internal constraint information.

Since the PEAV model uses the standard DCOP model (with multiple variables per agent), any DCOP algorithm can be used to solve the resulting PEAV-DCOP problem. Note that most DCOP algorithms assume a single-variable-per-agent in their original description. Some methods for dealing with multiple variables per agent in DCSP have been suggested (Yokoo, 2001). One solution is to convert a constraint reasoning problem involving multiple variables into a problem with only one variable by defining a new variable whose domain is the cross product of the domains of each of the original variables. Such complex variables imply extremely large domains, and therefore this solution is seldom used. A more applicable method is to create multiple virtual agents within a single real agent and assign one local variable to each virtual agent.

5.3 Asymmetric DCOPs (ADCOPs)

ADCOPs (Grubshtein et al., 2009; Grubshtein et al., 2010) generalize DCOPs in the following manner: instead of assuming equal payoffs for constrained agents, the ADCOP constraint explicitly defines the exact payoff of each participant. That is, domain values are mapped to a tuple of costs, one for each constrained agent.

More formally, an ADCOP is defined by the tuple $< \mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{R} >$, where $\mathcal{A}$, $\mathcal{X}$ and $\mathcal{D}$ are defined exactly as in regular DCOPs. Each constraint $C \in \mathcal{R}$ of an asymmetric DCOP defines a set of non-negative *costs* for every possible value combination of a set of variables, and takes the following form:

$$C : D_{i_1} \times D_{i_2} \times \cdots D_{i_k} \rightarrow \mathbb{R}_+^k \qquad (2)$$

This definition of an asymmetric constraint is natural to general MAS problems, and requires little manipulation when formulating these as ADCOPs.

ADCOPs require that the solving process will always take into account both sides of a constraint, rendering existing DCOP algorithm inapplicable. Along with the ADCOP model, several designated ADCOP algorithms have been proposed (Grubshtein et al., 2009). We briefly present the simplest one.

The SyncABB algorithm is an asymmetric version of SyncBB (Hirayama and Yokoo, 1997). After each step of the algorithm, when an agent adds an assignment to the *Current Partial Assignment* (CPA) and updates one direction of the bound, the CPA is sent back to the assigned agents to update the bound by the costs of all backwards directed constraints (backchecking). When the bound on the CPA is exceeded, a new value must be assigned, and the state of the CPA must be restored. The state of the CPA includes assignments of all higher priority agents and the cost (both backward and forward) associated with them. As the assignment does not change when backchecking, only the cost must be restored. More details can be found in (Grubshtein et al., 2009).

6 LOSS OF INTERNAL CONSTRAINT PRIVACY

The ADCOP and PEAV-DCOP models were both designed to not a-priorly reveal all internal constraint data. Nevertheless, while performing search, personal costs of agents must be revealed, aggregated and compared against the value of the objective. Thus, information may be gathered by different agents about the nature of their peers' private valuations (costs). We illustrate this fact by choosing two very simple algorithms, one for each model – SyncABB (ADCOP) and SyncBB (PEAV).

Following common practice, we evaluate privacy loss in terms of entropy (Greenstadt et al., 2006; Brito et al., 2009). Each constraint of an agent is represented by a matrix. Each cell in the matrix represents the agent's valuation of the constrained tuple. The entropy represents the uncertainty of the agent regarding the content of the matrices held by other agents. The general scheme for computing the entropy decrement is based on knowledge gain from a received CPA.

6.1 Privacy Loss in ADCOP

In ADCOP algorithms the data is revealed in *returning* CPAs. We require a returning CPA because only after the initial assignment is made by the agent can others asses its impact on them. An agent receiving a CPA message and a cost must first reduce the cost incurred by itself and the initial value the CPA had when it was last received by the agent. The remaining cost value k represents the aggregated cost which can be due to conflicts with its assignment or other inter-agent conflicts. Assuming a binary ADCOP, the agent may calculate the number of all the possible conflicts that may have triggered the cost k and were not part of the conflicts relevant to the previous version of the CPA. This number of conflicts will be denoted by n in the following analysis.

In our analysis, we use Asymmetric Max-DCSPs. Max-DCSP is a subclass of DCOP in which all constraint costs are equal to one (Modi et al., 2005). We use Asymmetric Max-DCSPs to simplify the privacy loss computation. Since Asymmetric Max-DCSPs are more restrictive than general DCOPs, their privacy loss can actually serve as an upper bound for general ADCOPs.

Throughout the SyncABB search process an agent may collect samples whenever the CPA reaches it in order to learn about the costs of its peers. When a CPA is moving backwards for back-checking it includes one new value assignment which was not included when it was last moved backwards for this purpose. Thus, an agent which receives the CPA can identify the new assignment and the difference in cost between the current lower bound and the lower bound which was on the CPA when it last held it. However, the agent cannot tell which of the agents, through which the CPA has passed, has contributed to this change of cost. This amount of uncertainty is dependent upon the number of agents which are already assigned on the CPA. The longer the partial assignment is the less an agent can learn from a CPA sent back for back-checking.

For each relevant sample the agent must calculate the appropriate k and n values. When agent A_i receives a returning CPA it can obtain data about A_j, the initiator of the back-checking (one of its succeeding peers). When receiving a returning CPA from its successor ($A_j = A_{i+1}$) the agent can also obtain data about its preceding peers ($\forall A_l | l < i$). In all samples, k is equal to the difference between the current CPA's cost and the value the CPA had when it was previously received by the agent (excluding the cost incurred by the agent itself). The value of n is determined by j, the priority (position in the order) of the initiator of the back-checking:

$$n = 2 \cdot (j-1) - 1 \tag{3}$$

Since internal constraint private data can be obtained only when back-checking, the last agent (A_n) cannot learn anything about its peers.

The scenario depicted in Figure 2 demonstrates the information leak in SyncABB when running on some asymmetric Max-DCSP example with 4 agents. We join the example after the back-checking initiated by A_2 is finished and A_1 sends the CPA to the next assigning agent, which is A_3 (stage a). The cost incurred by the CPA at this point is 2. A_3 remembers this cost (stage b) until the next time it receives the CPA. During the back-checking (stages c and d) the cost increases to 4. A_1 sends the CPA to the next assigning agent, which is A_4. After A_4 assigns its value the cost of the CPA increases to 6 (stage e). Following that, the CPA returns to A_3 for back-checking (stage f). A_3 remembers the cost from the previous time that he held the CPA (which was 2). The difference between the new cost and the old one is $k = 4$. The 5 thin arrows in Figure 2(f) represent the possible conflicts which triggered the cost k and were not part of the conflicts relevant to the previous version of the CPA. Since the initiator of this back checking is A_4, one can see that the value of n in the given example coincides with Equation 3:

$$n = 2 \cdot (j-1) - 1 = 2 \cdot (4-1) - 1 = 5$$

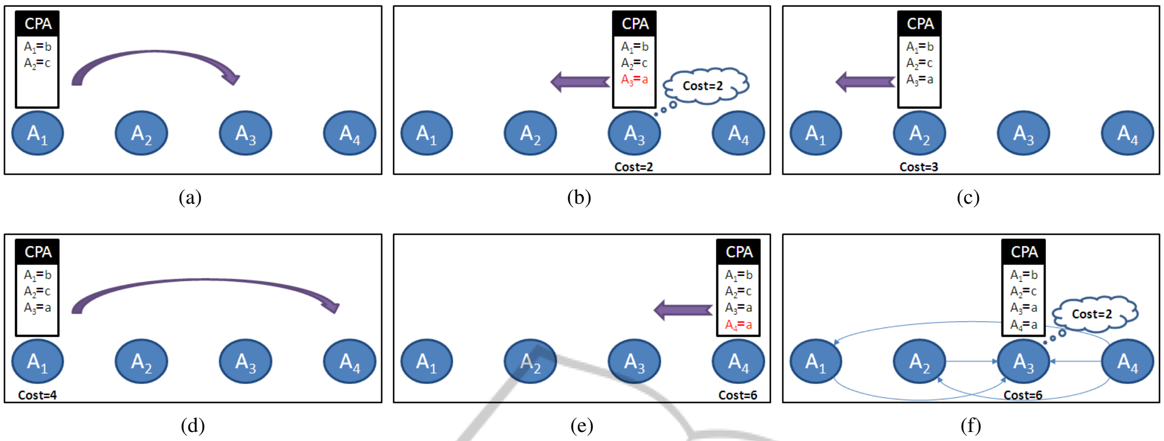

Figure 2: Information leak in SyncABB running on an asymmetric Max-DCSP example.

6.2 Privacy Loss in PEAV

Contrary to the ADCOP case, in PEAV-DCOPs the private information is revealed when the CPA advances forward. In fact, in most standard DCOP algorithms there is no such thing as a returning CPA (except after backtracking).

We will demonstrate the leakage of internal constraint information in PEAV through the simplest algorithm – SyncBB (Hirayama and Yokoo, 1997). The SyncBB algorithm requires an ordering of all participating variables. A natural example of such an ordering to the PEAV-DCOP that was presented in Figure 1 is given in Figure 3.

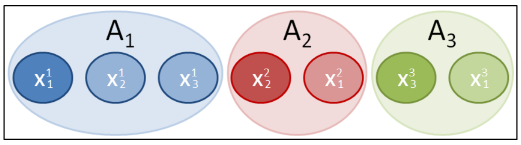

Figure 3: A variable ordering for the example in Figure 1.

In the beginning of the search process, agent A_1 assigns a value to variable x_1^1. It then goes on and assigns values to the variables x_2^1 and x_3^1, which are the next variables in the ordering. At this point the CPA holds the sum of costs of the constraints between variable x_1 and variables x_2 and x_3 from the point-of-view of agent A_1. Let us now look at the fourth variable in the ordering, i.e. variable x_2^2. When the CPA reaches it, some internal constraint information (of agent A_1) reaches agent A_2. To evaluate the privacy-loss we use the abbreviations k and n from the previous subsection. In this example we have $n = 2$, whereas the value of k is exactly the current cost of the CPA. In case k is equal to 0 or 2 (considering this is a Max-DCSP) the internal constraint information of agent A_1 is completely revealed to agent A_2. This example resembles the initial vulnerability that was described for external constraint privacy (Greenstadt et al., 2007). A conclusive formula such as the one in Equation 3 cannot be given for SyncBB, since it depends on the chosen variable ordering.

7 CONCLUSIONS

Privacy preservation is probably the main motivation for using the DCOP model. This explains the comprehensive research on DCOP privacy throughout the years. The present paper examines all the data that may be accumulated during DCOP search. This examination leads to categorization of all possible DCOP privacy types, including types that have not been studied in the past. According to the proposed categorization, most DCOP privacy studies in the past have targeted *constraint privacy*. The paper distinguishes between the constraint information of agents participating in the constraint (*internal constraint privacy*) and the constraint information of other agents (*external constraint privacy*). While most DCOP privacy research has focused on external constraint privacy, two models, PEAV and ADCOP, inherently relate to internal constraint privacy.

An analysis of the run of two algorithms, one for ADCOP and one for PEAV, reveals that both models lose some amount of internal constraint privacy. A question remains regarding which of the two models is better when considering both privacy preservation and performance. Previous studies have shown that PEAV is inapplicable for Local Search (Grubshtein et al., 2010). However, when it comes to complete search both models are fitting. We leave the investigation of the privacy/performance tradeoff in complete search for future work.

REFERENCES

Brito, I., Meisels, A., Meseguer, P., and Zivan, R. (2009). Distributed constraint satisfaction with partially known constraints. *Constraints*, 14(2):199–234.

Doshi, P., Matsui, T., Silaghi, M., Yokoo, M., and Zanker, M. (2008). Distributed private constraint optimization. In *IAT*, pages 277–281.

Faltings, B. and Macho-Gonzalez, S. (2005). Open constraint programming. *Artificial Intelligence*, 161:1-2:181–208.

Franzin, M. S., Rossi, F., Freuder, E. C., and Wallace, R. J. (2004). Multi-agent constraint systems with preferences: Efficiency, solution quality, and privacy loss. *Computational Intelligence*, 20(2):264–286.

Gershman, A., Meisels, A., and Zivan, R. (2009). Asynchronous forward bounding. *JAIR*, 34:25–46.

Greenstadt, R. (2009). An overview of privacy improvements to k-optimal dcop algorithms. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1279–1280, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Greenstadt, R., Grosz, B., and Smith, M. D. (2007). Ssdpop: improving the privacy of dcop with secret sharing. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, pages 171:1–171:3, New York, NY, USA. ACM.

Greenstadt, R., Pearce, J., and Tambe, M. (2006). Analysis of privacy loss in distributed constraint optimization. In *Proc. AAAI-06*, pages 647–653, Boston, MA.

Grinshpoun, T. and Meisels, A. (2008). Completeness and performance of the apo algorithm. *J. of Artificial Intelligence Research*, 33:223–258.

Grubshtein, A., Grinshpoun, T., Meisels, A., and Zivan, R. (2009). Asymmetric distributed constraint optimization. In *IJCAI-2009 Workshop on Distributed Constraints Reasoning (DCR-09)*, pages 60–74, Pasadena, CA.

Grubshtein, A., Zivan, R., Grinshpoun, T., and Meisels, A. (2010). Local search for distributed asymmetric optimization. In *Proc. 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 1015–1022, Toronto, Canada.

Hirayama, K. and Yokoo, M. (1997). Distributed partial constraint satisfaction problem. In *Proc. CP-97*, pages 222–236.

Léauté, T. and Faltings, B. (2011). Coordinating logistics operations with privacy guarantees. In *IJCAI*, pages 2482–2487.

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. *Autonomous Agents and Multi-Agent Systems*, 13(1):27–60.

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 *Proc. AAMAS-04*, pages 310–317, New York, NY.

Mailler, R. and Lesser, V. R. (2004). Solving distributed constraint optimization problems using cooperative mediation. In *Proc. AAMAS-04*, pages 438–445.

Matsui, T., Matsuo, H., Silaghi, M. C., Hirayama, K., Yokoo, M., and Baba, S. (2010). A quantified distributed constraint optimization problem. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, pages 1023–1030, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Meseguer, P. and Larrosa, J. (1995). Constraint satisfaction as global optimization. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, pages 579–584, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Modi, P. J., Shen, W., Tambe, M., and Yokoo, M. (2005). ADOPT: asynchronous distributed constraints optimizationwith quality guarantees. *Artificial Intelligence*, 161:149–180.

Petcu, A. and Faltings, B. (2005). A scalable method for multiagent constraint optimization. In *Proc. IJCAI-05*, pages 266–271, Edinburgh, Scotland, UK.

Petcu, A. and Faltings, B. (2006). ODPOP: An algorithm for open/distributed constraint optimization. In *Proc. AAAI-06*, pages 703–708, Boston.

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. C. and Mitra, D. (2004). Distributed constraint satisfaction and optimization with privacy enforcement. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, IAT '04, pages 531–535, Washington, DC, USA. IEEE Computer Society.

Stefanovitch, N., Farinelli, A., Rogers, A., and Jennings, N. R. (2011). Resource-aware junction trees for efficient multi-agent coordination. In *The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 363–370.

Vinyals, M., Rodriguez-Aguilar, J. A., and Cerquides, J. (2011). Constructing a unifying theory of dynamic programming dcop algorithms via the generalized distributive law. *Autonomous Agents and Multi-Agent Systems*, 22:439–464.

Yokoo, M. (2001). *Distributed constraint satisfaction: foundations of cooperation in multi-agent systems*. Springer-Verlag, London, UK.

Zhang, W., Xing, Z., Wang, G., and Wittenburg, L. (2005). Distributed stochastic search and distributed breakout: properties, comparishon and applications to constraints optimization problems in sensor networks. *Artificial Intelligence*, 161:1-2:55–88.