The effect of policies for selecting the solution of a DisCSP on privacy loss*

Marius-Călin Silaghi and Vaibhav Rajeshirke
Florida Institute of Technology, Melbourne

Abstract

Distributed constraint satisfaction (distributed CSP) addresses the problem of assigning values to a set of variables, to satisfy the secret constraints of several participants. Many have argued so far about how to formalize the privacy requirements on input constraints. However, we notice that not sufficient attention was given to formalizing the description of what solution is desired. Different criteria of selecting a solution have strong consequences on an inherent privacy loss. We conclude proposing a way to minimize privacy loss by specifying expectations about how a solution has to be chosen among possible candidates.

1. Introduction

Distributed problems like meeting-scheduling [3], auctions, timetabling, college admissions, and other *stable marriages* instances are defined by parameters that participant agents often prefer to keep secret. However even the fact of learning a solution leaks by itself some secrets, at least that the solution is acceptable to everybody. There exist known efforts to reduce the amount of secrets that participants learn from the outcome, by changing the definition of the solution to a distributed CSP [4, 1]. Namely, some existing formulations aim to reveal to each agent only an agreed subset of the assignments in the solution. Similarly, aiming to reveal one solution and aiming to reveal all solutions have very different effects on the amount of secrets that are leaked.

Here we show that even when a single solution is required, the way in which the solution is chosen has important consequences on the amount of secret information that is leaked to the participants. The solution can be selected as the first in a known lexicographical order over the tuples in the search space, as implicitly done by most search algorithms. We show that more privacy is achieved by selecting the solution as the lexicographically first candidate given an unknown random order on domains. Even more privacy is offered when the solution is selected according to a uniform distribution over the set of all solutions.

* Thank Debasis Mitra, Richard Wallace, Kamel Recab for feedback.

2. Distributed CSP

In the following we detail a framework for modeling distributed CSPs, where a constraint is public, or is a secret known to an agent, and each participant learns only an agreed part of the solution. An assignment is a pair $\langle x_i, v_k^i \rangle$ meaning that the variable x_i is assigned the value v_k^i. A tuple is an ordered set. The projection of a tuple ϵ of assignments over a tuple of variables X_i is denoted $\epsilon_{|_{X_i}}$.

Definition 1 *A distributed constraint satisfaction problem (DisCSP) is defined by five sets (A, X, D, I, O). X=$\{x_1, ..., x_m\}$ is a set of variables. Each variable x_i can take values from an associated domain D_i=$\{v_1^i, ..., v_{d_i}^i\}$, defined by the set D=$\{D_1, ..., D_n\}$. A=$\{A_1, ..., A_n\}$ is a set of agents. Each agent A_i is willing to enforce a set of constraints, C_i, specified by the set of inputs I=$\{C_0, ..., C_n\}$. The union of all the constraints in I is C= $\cup_{i=0}^{n}$ C_i=$\{\phi_1, ..., \phi_c\}$. Each constraint ϕ_k, $\phi_k \in C_i, i$>0, is defined by a secret predicate on a set of variables X_k, X_k⊆X, predicate known only to A_i. The constraints in C_0 are public. The solution for each agent A_i is a tuple of assignments ϵ_i* for the set of variables O_i, O_i⊆X, defined by the set of outputs O=$\{O_1, ..., O_n\}$. They are such that there exists a tuple of assignments $\epsilon*$ to all variables in X satisfying all constraints in C and $\forall i, \epsilon_i* = \epsilon*_{|O_i}$.*

Note that with this definition we did not specify how the solution will be selected among alternatives, when there exist several solutions. By deciding to select the first solution in the lexicographical order induced on tuples by a known order on variables and values (as often done), we leak to everybody that each tuple lexicographically ordered before the proposed solution is rejected by somebody. This information was not requested and can be exploited in some problems to infer details about some secret constraints.

3. The solution of a DisCSP

A question to be asked is whether a solution computed over a random unknown permutation of variables and domains could help remove the aforementioned leaks.

Theorem 1 *For any CSP whose search space has size Θ, and for any j, 0≤j<Θ, there exists a shuffling of the val-*

ues in its domains such that a solution with any initial lexicographic position i in its search space is mapped into the position j of the obtained problem.

Proof. This is proven easily by constructing the shuffling. □

Corollary 1.1 *For any CSP and a given solution, there exists a shuffling of the values in its domains mapping that solution into the lexicographically first tuple.*

As follows from the previous corollary, one cannot extract with certitude any secret by an inference based on the identity of the solution of the problem shuffled with unknown permutations of the domains (except that the solution is accepted by everybody). However, statistical information may be leaked as seen further.

3.1. Shuffling Domains and Variables

A random variable generates the solution over the set of tuples ϵ that satisfy the constraints. Let us analyze this random variable where values (and eventually variables) are permuted randomly according to a uniform distribution over the set of all possible permutations.

Theorem 2 *Shuffling randomly the domains for a CSP does not guarantee that the first solution in the obtained lexicographic order is selected according to a uniform distribution over the set of all solutions.*

Proof. Can be checked on any CSP with 2 variables, 2 values/variable, and 3 solutions. It can be noticed that the frequency with which the solution is drawn is inverse proportional to the frequency of its assignments among the other solutions. □

Theorem 3 *Shuffling variables and domains for a CSP does not guarantee that the first solution in the obtained lexicographic order is selected according to a uniform distribution over the set of all solutions.*

Proof. Can be checked on the same problem as in the previous proof. The lack of uniformity is slightly less accentuated than for the case where only domains are reordered. □

The frequency with which the solution is drawn is inverse proportional to the frequency of its assignments among the other solutions. Therefore, if an agent participates with the same constraints in several computations, statistical information can be extracted. Namely, a solution that occurs very often indicates that some of its assignments are rare.

3.2. Uniform distribution over all solution

Now we define an abstract method that will be proven to select a solution of a constraint satisfaction problem according to a uniform distribution over the set of all solutions.

Theorem 4 *If the following process is applied to a CSP:*

- *Create a vector S with a value, $p(\epsilon)$ — 1 for solutions, 0 otherwise—, for each tuple ϵ, in lexicographic order.*
- *Shuffle S according to a permutation π picked with a uniform distribution over the possible permutations.*
- *Pick the first value of S having $p(\epsilon) = 1$. Choose ϵ as the solution to be returned.*

the tuple returned by the three steps above is chosen according to a uniform distribution over all solutions.

Proof. By symmetry, each ϵ with $p(\epsilon) = 1$ will be placed with equal probability before all the other solutions. □

4. Conclusions

With DisCSPs and known solving algorithms we detect a privacy loss inherent to how the solution is selected among alternatives. We compare the differences between the outcomes when a solution is picked as the first in a known lexicographic order, or the lexicographically first in an unknown random permutation of the problem description. Three existing types of unknown permutations of problem descriptions were studied:

- an unknown permutation of the values in domains,
- an unknown permutation of domains and variables,
- an unknown permutation of all the tuples, chosen according to a uniform distribution over the set of all possible permutations.

It was shown that they offer an increasing degree of privacy. The last type of unknown permutation guarantees that the solution is picked according to a uniform distribution over the set of all solutions, minimizing the loss of statistical data about secrets.

Here we explain the importance of specifying with the definition of a DisCSP, the way in which the solution must be chosen. Techniques for solving problems with different such specifications are outside the scope of this article and we propose some in [2].

References

[1] B. Faltings. Incentive compatible open constraint optimization. In *Electronic Commerce*, 2003.

[2] M. Silaghi. A suite of secure multi-party computation algorithms for solving distributed CSPs. Technical Report CS-2004-05, FIT, 2004. www.cs.fit.edu/~tr/tr2004.html.

[3] R. Wallace and M. Silaghi. Using privacy loss to guide decisions in distributed CSP search. In *FLAIRS'04*, 2004.

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