Stochastic Game Modelling for Distributed Constraint Reasoning with Privacy

Julien Savaux, Julien Vion, Sylvain Piechowiak, René Mandiau
Université de Valenciennes, LAMIH UMR CNRS 8201 (France)

Toshihiro Matsui, Nagoya Tech (Japan)
Katsutoshi Hirayama, Kobe University (Japan)
Makoto Yokoo Kyushu University (Japan)

Shakre Elmane, Marius Silaghi
Florida Institute of Technology (USA)

Abstract

Privacy has been a major concern for agents in distributed constraint reasoning. In this work, we approach this issue in distributed constraint reasoning by letting agents compromise solution quality for preserving privacy, using utility theory. We formalize privacy in the context of distributed constraint reasoning, detail its different aspects, and present model and solvers, as well as their properties. We then show how distributed constraint reasoning with privacy requirements can be modelled as a planning problem, and more specifically as a stochastic game.

Introduction

In Distributed Constraint Reasoning (DCR), agents have to find values to a set of shared variables while respecting given constraints (frequently assumed to have unspecified privacy implications). To find such assignments, agents exchange messages until a solution is found or until some agents detect that there is no solution to the problem. Thus, commonly agents reveal information during the solution search process, causing privacy to be a major concern in DCR (Yokoo *et al.* 1998a).

Here we approach the problem by assuming that privacy has a utility that can be aggregated with the utility value of solving the problem. The availability of a value from the domain of a variable of the DisCSP in the presence of the constraints of an agent, is the kind of information that the agents want to keep private. The artificial intelligence assumption is that utility-based agents are able to associate each state with a utility value (Wooldridge and Jennings 1995). As such each action is associated with the difference between initial and final utilities. In our extended algorithms, agents may stop their participation if the utility of the privacy expected to be lost overcomes the reward for finding a solution of the problem. Extensions to basic solvers are investigated to preserve privacy. Then, we recast the problem as a reiterated stochastic game, and show how approaches from other fields can be used to let agents preserve privacy in DCR.

Next section discusses previous research concerning privacy for distributed constraint reasoning. Further we formally define the concepts involved in Utilitarian Distributed Constraint Reasoning (UDCR). Next section introduces some extensions to common DisCSP solvers that let agents preserve privacy. After a presentation of the properties of our extensions, we show how UDCR can be modelled as repeated stochastic games. Last section concludes this paper.

Background on Privacy

Privacy has been an important aspect for DCR. Privacy is the concern of agents to not reveal their personal information. In this work, we define privacy as follows:

Definition 1. *Privacy is the utility that agents benefit from conserving the secrecy of their personal information.*

Contrary to the standard rewards in DCR, privacy costs are proper to each individual agent. Therefore, the computation is now performed by utility-based and self-interested agents, whose decisions aim at maximizing a utility function. The objective is then to define a policy associating an expected utility maximizing action (communication act or computation) to each state, where the state includes the belief about the global state). In existing works, several approaches have been developed to deal with privacy in DCR (Greenstadt *et al.* 2006; Léauté and Faltings 2013).

Privacy Categorization

We choose to deal with privacy by embedding it into agents' decision-making. Other approaches use various metrics and frameworks to quantify privacy loss. According to previous works (Grinshpoun 2012), agents privacy may concern the four following aspects:

- Domain privacy: Agents want to keep the domain of their variable private. The common benchmarks and some algorithms assume that all the domains are public, which leads to a complete loss of domain privacy. In the original DCR approach, a form of domain privacy is implicit.
- Constraint privacy: Agents want to keep the information related to their constraints private (Silaghi *et al.* 2000). If variables involved in constraints are considered to belong to only one agent, we can distinguish the revelation of information to agents that participate in the constraint (internal constraint privacy) and the one to other agents (external constraint privacy).

- Assignment privacy: Agents want to keep the assigned values to their variables private. The revelation of assigned values concerns the assignment of the final solution, as well as the ones proposed during search (Silaghi *et al.* 2005).
- Algorithmic privacy: Even though it is commonly assumed that all agents run the same algorithm during the solving, agents may modify the value of some parameters guiding the search process for some personal benefit (*e.g.*, the likelihood of updating its value). This can be achieved by keeping the message structure and contracts of certain existing solvers to be used as communication *protocols* rather than *algorithms*, as introduced in (Silaghi and Faltings 2002), where protocols obtained in such ways are compared with respect to the flexibility offered for agents to hide their secrets.

Our Approach

An agent revealing an assignment to another agent, incurs a cost. While some previously described frameworks do model the details of our example, it has until now been an open question as to how they can be dynamically used by algorithms in the solution search process. We propose to define a framework which, while potentially being equivalent in expressing power to existing DisCSP extensions, would nevertheless explicitly specify the elements of the corresponding family of planning problems. We introduce the Utilitarian Distributed Constraint Reasoning (UDCR). Unlike previous DCR frameworks, besides constraint satisfaction, we are also interested in the solution process. A policy is a function that associates each state of an agent with an action that it should perform (Russell and Norvig 2010).

We define an *agreement* as a set of assignments for all the variables with values from their domain, such that all the constraints are satisfied.

Definition 2. *A UDisCSP is formally defined as a tuple $\langle A, V, D, C, U, R\rangle$ where, A,V,D,C are standard DisCSP parameters, and U,R are:*

- $U = \{u_{1,1}, ..., u_{n,d}\}$ *is a matrix of costs where* $u_{i,j}$ *is the cost of agent* A_i *for revealing whether* $j \in D_i$.
- $R = \langle r_1, ..., r_n\rangle$ *is a vector of rewards, where* r_i *is the reward agent* A_i *receives if an agreement is found.*

The state *of agent* A_i *includes the subset of* D_i *that it has revealed, as well as the achievement of an agreement. The problem is to define a set of communication actions and a policy for each agent such that their utility is maximized.*

Example 1. *Suppose a meeting scheduling problem between three participants, a professor and two students (called* A_1*,* A_2 *and* A_3*). They all consider to agree on a time slot to meet on a given day, having to choose between 8 am, 10 am and 2 pm. Professor* A_1 *is unavailable at 2 pm, Student* A_2 *is unavailable at 10 am, and Student* A_3 *is unavailable at 8 am.*

There can exist various reasons for privacy. For example, A_2 *does not want to reveal the fact that it is busy at 10 am. The value that* A_2 *associates with not revealing the 10 am unavailability is the salary from a second job ($ 2,000). The utility of finding an agreement is for each student the stipend for their studies ($5,000). For* A_1*, the utility is a fraction of the value of its project ($4,000). This is an example of privacy for absent values or constraint tuples. Further* A_3 *had recently boasted to* A_2 *that at 8 am it interviews for a job, and it would rather pay $1,000 than to reveal that it is not. This is an example of privacy for feasible values of constraint tuples.*

The DisCSP is extended to UDisCSP by specifying the additional parameters U, R*:*
$A = \{A_1, A_2, A_3\}$
$V = \{x_1, x_2, x_3\}$
$D = \{\{1, 2, 3\}, \{1, 2, 3\}, \{1, 2, 3\}\}$
$C = \{x_1 = x_2 = x_3, x_1 \neq 3, x_2 \neq 2, x_3 \neq 1\}$
$U = \{u_{1,1} = 1, u_{1,2} = 2, u_{1,3} = 4,$
$u_{2,1} = 1, u_{2,2} = 2, u_{2,3} = 4,$
$u_{3,1} = 1, u_{3,2} = 2, u_{3,3} = 4\}.$
$R = \langle 5, 4, 4\rangle.$

Illustration on Standard Solvers

Now we discuss how the basic ABT and SyncBT algorithms are adjusted to UDCR (Yokoo *et al.* 1992; 1998b; Zivan and Meisels 2003). The state of an agent includes the `agent view`. After each state change, each agent computes the estimated utility of the state reached by each possible action, and selects randomly one of the actions leading to the state with maximal expected utility.

In our algorithms, an information used by agents in their estimation of expected utilities is the risk of one of their assignments being rejected. This risk can be re-evaluated at any moment based on data recorded during previous runs on problems of similar parameters (e.g, problem density). It calculates the risk for a solution to not lead to the termination of the algorithm, called $agreementProb$.

When `ok?` messages are sent, the agent has the choice of which assignment to propose. When a `nogood` message is scheduled to be sent, agents also have choices of how to express them. Before each `ok?` or `nogood` message, the agents check which available action leads to the highest expected utility. If the highest expected utility is lower than the current one, the agent announces failure. The result is used to decide the assignment, nogood, or failure to perform.

To calculate the estimated utility of pursuing an agreement (revealing an alternative) assignment, the agent considers all different possible scenarios of the subsets of values that might have to be revealed in the future based on possible rejections received, together with their probability (see Algorithm 1). The algorithm assumes as parameters: (i) the computed agreementProb, (ii) the possible values D, and, (iii) the probability of having to select from D. The algorithm then recursively calculates the utility of the next possible states, and whether the revelation of the current value v leads to the termination of the algorithm, values stored in variables $costRound$ and $costNonTerminal$. The algorithm returns the estimated cost of privacy loss for the future possible states currently, called $estimatedCost$. Solving this problem with ABTU is depicted in Figure 1, respectively.

Algorithm 1: estimateCostDisCSP

Input: $agreementProb$, $\mathscr{D}'_{self}$, $probD$
Output: $estimatedCost$
$valueId = j \mid (\mathscr{D}_{self}[j] = \mathscr{D}'_{self}[1])$;
if $(|\mathscr{D}'_{self}| = 1)$ **then**
  return $(\sum_{j=1}^{j=valueId} u_{d(self,j)}) \times probD$;
else
  $v \leftarrow \mathscr{D}'_{self}[1]$;
  $costRound \leftarrow$ estimateCostDisCSP ($agreementProb$, $\{v\}$, $agreementProb \times probD$);
  $costTemp \leftarrow$ estimateCostDisCSP ($agreementProb$, $\mathscr{D}'_{self} \setminus \{v\}$, $(1 - agreementProb) \times probD$);
  $estimatedCost \leftarrow costRound + costTemp$;
  return $estimatedCost$;

Example 2. *Continuing with Example 1 (whose a possible trace is illustrated by Figure 1), at the beginning of the solving, Agent A_1 has to decide for a first action to perform. We suppose the $agreementProb$ learned from previous solvings is 0.5. To decide whether it should propose an available value or not, it calculates the corresponding $estimatedCost$ by calling Algorithm 1 with parameters: the learned $agreementProb = 0.5$, the set of possible solutions ($\mathscr{D}'_1 = \{1, 2, 3\}$) and $probD = 1$.*

For each possible value, this algorithm recursively sums the cost for the two scenarios corresponding to whether the action leads immediately to termination, or not. Given privacy costs, the availability of three possible subsets of $\mathscr{D}'_1$ may be revealed in this problem: $\{1\}, \{1, 2\}$, and $\{1, 2, 3\}$. Each set of size S consists of S first elements of the list solution based on this initial order.

The $estimatedCost$ returned is the sum of the costs for all possible sets, weighted by the probability of their feasibility being revealed if an agreement is pursued. At the function call: $costRound = u_{d(1,1)} \times 0.5 = 1 \times 0.5 = 0.5$. At the next recursion: $costRound = (u_{d(1,1)} + u_{d(1,2)}) \times 0.25 = (1 + 2) \times 0.25 = 0.75$. At the last recursion: $costRound = (u_{d(1,1)} + u_{d(1,2)} + u_{d(1,3)}) \times 0.25 = (1 + 2 + 4) \times 0.25 = 1.75$. The algorithm returns the sum of these three values: estimatedCost$= 0.5 + 0.75 + 1.75 = 3$.
The expected utility of pursuing a solution being positive ($reward$−estimatedCost$= 4 - 3 = 1$), the first value is proposed.

Other works evaluate this utilitarian approach on state of the art problems and compared obtained privacy loss with the one occurring during standard DisCSP solving (Savaux *et al.* 2016; Mandiau *et al.* 2014). Next section present properties of UDCR.

Utilitarian Approach Properties

Generalities

Proposition 1. *UDCR is sound, terminates, is correct.*

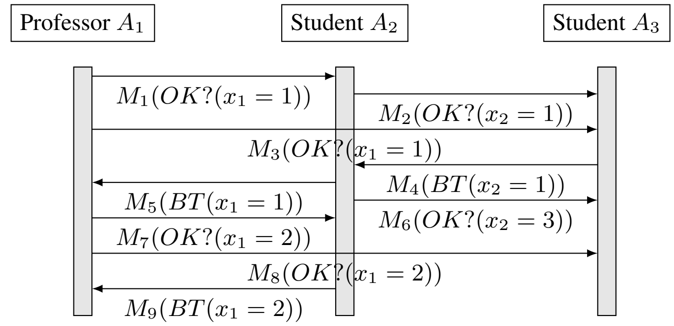

Figure 1: Interactions between agents during ABT

Proof. Note that UDCR guarantees soundness, termination, and correction on condition that the DCR solver used does as well.

- UDCR is sound, as any solution to a problem with privacy requirements it computes is also a solution to the same problem modelled using standard DCR models.
- Moreover, UDCR guarantees the termination of the solving, like standard DCR. Any detection of unsolvability in DCR leading to termination is also detected by UDCR.
- Finally, UDCR is correct, as we can show that each solution to UDCR is also a DCR solution. More specifically, proposed extensions do not modify the generation of a solution.

□

Proposition 2. *UDCR does not guarantee completion.*

Proof. As privacy requirements modify the nature of the problem, some of the solutions that can be found with DCR may not be found with UDCR. Note that some DCR solvers do not guarantee completion as well. Still, with certain parameters (all privacy costs being null), UDCR solving will be similar to DCR one, and completion will be guaranteed. □

Complexity

Proposition 3. *Compared to DisCSP solving, UDisCSP extensions increase computation of a time linear with domain size.*

Proof. Agents compute estimated utilities by building a binary tree corresponding to the succession of solution proposal, with the two different outcomes (solution acceptance or rejection). DCR are already NP-hard problems, and UDCR are therefore NP-hard also. □

Proposition 4. *UDCR increase space complexity with a cost linear with standard DCR data size.*

Proof. Agents have to store the costs of privacy for revealing the different information of their local problem, and keep track of the already proposed solutions. More specifically, for a problem with n variables, d values per domain, and

c constraints, each agent has to store $(2d + c)$ values for revealing domain values , assignments and constraints. □

Proposition 5. *In the worst case, the number of messages exchanged between agents during UDCR solving cannot exceed that of DCR solving.*

Proof. If privacy requirements are too constraining, agents do not investigate other solutions and do not exchange messages with each other. The number of exchanged messages may be zero. Agents then focus on solving their local problem, which is relevant with multi-variable problems (*i.e.*,, timetabling problems). On the opposite in worst case, privacy requirements are insignificant compared to standard DisCSP solving. In this case, UDCR solving then requires an equivalent number of exchanged messages as does DCR solving . □

Privacy Properties

Proposition 6. *UDCR guarantees a correct privacy measure.*

Proof. UDCR guarantees to keep track and to measure all revelation of information. UDCR assigns a cost to each revelation of information, as defined by the content of the communication protocols defined by the solvers. Also, UDCR keeps track of all communication and measures the corresponding privacy loss. In UDCR, agents only perceive information through communication with each other. Therefore, privacy loss measurement is correct, and no privacy loss can occur without being measured. □

Proposition 7. *UDCR preserves data (domain, constraint, assignment) privacy to the same extent as DCR does.*

Proof. In standard DisCSP as well as in UDCR, data privacy is degraded each time a solution is proposed. However, UDCR does not degrade domain privacy more than DisCSP does. Indeed, privacy requirements do not modify constraints between variables, nor the domains of variables. □

Proposition 8. *UDCR preserves algorithmic privacy for different agents.*

Proof. Agents do not share with each other any information about their utility function used to define which decision to take at each step, as well as the resulting valuation for each action. Moreover, information observed and recorded used to guide search are kept private. Thus we conclude that UDCR preserves algorithmic privacy. □

Stochastic Game Modelling

Earlier works already proposed to recast UDCR with privacy requirements as Partially Observable Markov Decision Processes (POMDPs) (Savaux *et al.* 2016). Hybrid approaches using both DCOP and Dec-POMDP have also been investigated to deal with multi-agent teamwork (Zhang and Lesser 2011). As Partially Observable Stochastic Games (POSG) can be seen as generalizations of POMDPs, we propose in this section to recast UDCR as POSG. Indeed, we can note that commmunication protocols in UDCR can be seen as the cooperating action in game theory, while refusing to communicate in order to preserve privacy can be seen as game theory deceiving actions. Relation between UDCR and PSOG elements is depicted in Figure 2, for each agent, where CSP refers to basic parameters of DCR, *i.e.*, variables, domains, and constraints.

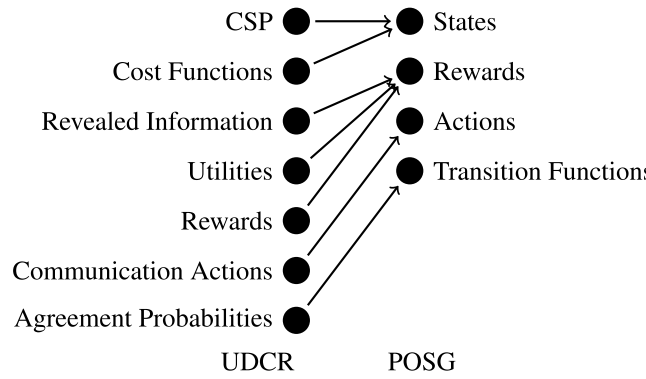

Figure 2: Relation between UDCR and POSG elements

Then, during UDCR solving, each solution proposal can be interpreted as a game, when proposing a solution incurs privacy loss, and may imply a reward if the agent receiving the proposal accepts to communicate and accepts the solution. Complete UDCR solving can be therefore recast as a reiterated stochastic game.

Example 3. *Suppose a Distributed Meeting Scheduling problem with two participants, Alice and Bob, who have to choose a location where to meet among two possibilities, London and Madrid. We refer to Alice and Bob as agents A and B, respectively. We refer to the locations London and Madrid as their identifier 1 and 2, respectively. Note $r_{x,y}$, the reward for agent x to attend the meeting at location y. Note also that $p_{x,y}$ is the privacy loss for agent x to propose location y. $r_{x,y}$ and $p_{x,y}$ are dictated by the problem. t, the probability for a solution proposal to be accepted. By default, $t = 0.5$. U_x is the utility for agent x after the first step. U'_x is the utility for agent x after the second step. MU_x is the marginal utility for agent x to perform the considered action.*

At the **first** step, suppose A chooses solution 2, as this solutions maximizes its expected marginal utility (EMU). A gets the reward r, weighted by the probability for the solution to be accepted t, diminished by the cost for revealing information p. Indeed:

$$\begin{cases} EMU_{A,2} = t \times r_{A,2} - p_{A,2}, EMU_{A,1} = t \times r_{A,1} - p_{A,1} \\ EMU_{A,2} \geqslant EMU_{A,1} \\ MU_A = -p_{A,2} \end{cases}$$

Similarly, B chooses solution 1, as:
$EMU_{B,1} > EMU_{B,2}$. $MU_B = -p_{B,1}$.
The reached state is $\{A = 2, B = 1\}$. However, agents have proposed different solutions and have not reached an agreement. Therefore, they have revealed information but have not gained any reward. Their utilities are:

$$\begin{cases} U'_A = -p_{A,2} \\ U'_B = -p_{B,1} \end{cases}$$

At the **second** step, agents have to decide which action to make. Each one can participate and can change their value, or not participate (deceive) and keep their current value. No matter its action, an agent does not know beforehand what action the other agent will make. Four scenarios are then possible:

- $\langle$deceive, deceive$\rangle$: no agent changes its value
- $\langle$participate, deceive$\rangle$: A changes its value
- $\langle$deceive, participate$\rangle$: B changes its value
- $\langle$participate, participate$\rangle$: A and B change their value

Let us study them in more details. $EMU_{X,Y}$ represents the estimated marginal utility Agent X gets from doing action Y, where Y can be null when no action is done.

deceive, deceive: no more action is performed. Therefore there is no more reward or privacy loss for any agent. However, an agent does not know the behavior of the other, so a reward may be expected.

$$\begin{cases} EMU_{A,null} = t \times r_{A,2}, EMU_{B,null} = t \times r_{B,1} \\ MU_A = 0 \ \ (a), MU_B = 0 \ \ (b) \\ U'_A = MU_A - p_{A,2}, U'_B = MU_B - p_{B,1} \end{cases}$$

participate, deceive: A changes its value for the one proposed by B. Both agents get the corresponding reward but A has to reveal a second solution. Since B proposed Solution 1, reward for revealing the solution 1 is guaranteed for A.

$$\begin{cases} EMU_{A,1} = t \times r_{A,1} - p_{A,1}, EMU_{B,null} = t \times r_{B,1} \\ MU_A = r_{A,1} - p_{A,1} \ \ (c), MU_B = r_{B,1} \ \ (d) \\ U'_A = r_{A,1} - (p_{A,1} + p_{A,2}), U'_B = r_{B,1} - p_{B,1} \end{cases}$$

deceive, participate: B changes its value for the one proposed by A. Both agents get the corresponding reward, but B has to reveal a second solution. Since A proposed Solution 2, reward for revealing the solution 2 is guaranteed for B.

$$\begin{cases} EMU_{A,null} = t \times r_{A,2}, EMU_{B,2} = t \times r_{B,2} - p_{B,2} \\ MU_A = r_{A,2} \ \ (e), MU_B = r_{B,2} - p_{B,2} \ \ (f) \\ U'_A = r_{A,2} - p_{A,2}, U'_B = r_{B,2} - (p_{B,1} + p_{B,2}) \end{cases}$$

participate, participate: both agents change their values. Then, the two solutions have been proposed by each agent. Each one is available as the final one. If Solution 1 is chosen:

$$\begin{cases} EMU_{A,1} = t \times r_{A,1} - p_{A,1}, EMU_{B,1} = t \times r_{B,1} - p_{B,2} \\ MU'_A = r_{A,1} - p_{A,1} \ \ (g), MU'_B = r_{B,1} - p_{B,2} \ \ (h) \\ U'_A = r_{A,1} - (p_{A,1} + p_{A,2}), U'_B = r_{B,1} - (p_{B,1} + p_{B,2}) \end{cases}$$

If Solution 2 is chosen:

$$\begin{cases} EMU_{A,2} = t \times r_{A,2} - p_{A,1}, EMU_{B,2} = t \times r_{B,2} - p_{B,2} \\ MU'_A = r_{A,2} - p_{A,1} \ \ (i), MU'_B = r_{B,2} - p_{B,2} \ \ (j) \\ U'_A = r_{A,2} - (p_{A,1} + p_{A,2}), U'_B = r_{B,2} - (p_{B,1} + p_{B,2}) \end{cases}$$

Each agent has two possibilities: participate or deceive. Participating or deceiving actions correspond to different communication protocols. The solver used dictates what action is expected if they participate. The marginal utilities of the four possible scenarios are depicted in Table 1 (with payoffs values). Such a scenario is verified when:

If the solver chooses Solution 2:
$(t \times r_{A,2}) > t \times (r_{A,1} - p_{A,1}) + (1-t) \times r_{A,1}$ and
$(t \times r_{B,1}) > t \times (r_{B,1} - p_{B,2}) + (1-t) \times (r_{B,2} - p_{B,2})$

If the solver chooses Solution 2:
$(t \times r_{A,2}) > t \times (r_{A,2} - p_{A,1}) + (1-t) \times r_{A,1}$ and
$(t \times r_{B,1}) > t \times (r_{B,2} - p_{B,2}) + (1-t) \times (r_{B,2} - p_{B,2})$

Table 1: Payoffs matrix

	participate	deceive
participate	1:$g = r_{A,1} - p_{A,1}$ $h = r_{B,1} - p_{B,2}$ 2:$i = r_{A,2} - p_{A,1}$ $j = r_{B,2} - p_{B,2}$	$e = r_{A,2}$ $f = r_{B,2} - p_{B,2}$
deceive	$c = r_{A,1} - p_{A,1}$ $d = r_{B,1}$	$a = 0$ $b = 0$

In the situations considered here, p is always positive. Indeed, agents have no interest in revealing their information. Each revelation has a cost, but cannot give a reward to the agents. p is also always lower than r. Indeed, only solutions whose cost of revelation is lower than the reward are considered. Therefore, UDCOP solving may contain steps that can be represented as a chicken game (Rapoport and Chammah 1966).

However, when the values v respect the following property, it corresponds to a prisoners' dilemma:

$$v_{(d,p)} > v_{(p,p)} > v_{(d,d)} > v_{(p,d)} \tag{1}$$

In the context of a distributed constrained problem, presented in Table 1, this corresponds to the following equation:
$r_{A,2} > r_{A,1} - p_{A,1} > 0 > r_{A,1} - p_{A,1}$ (with Solution 1)
$r_{A,2} > r_{A,2} - p_{A,1} > 0 > r_{A,1} - p_{A,1}$ (with Solution 2)

This equation is never verified with the hypothesis requiring p to be lower than r. However, in future works, we can suppose than revealing information does not only have a cost (related to privacy loss) but can also have a reward. For example, malicious agents can reveal fake information in order to make other agents waste computation time, or to mislead them (Marsh 1994). In such cases, p can be greater than r, and UDCOP solving can be interpreted as a prisoners' dilemma.

Now that UDCR are recast as POSG, for future works we plan to use standard game theory strategies, (*e.g.*,, tit for tat) in order to let agents decide if they should participate in the solving of if they should not communicate and preserve their privacy. Comparing this approach with standard DCR approaches can lead to interesting results, without having to develop complex models or solvers.

Conclusions

In this work, we propose an approach to deal with privacy in Distributed Constraint Reasoning, using utilities, Utilitarian Distributed Constraint Reasoning (UDCR). We formalized privacy in this context, and present works extending standard DCR models and solvers considering privacy. We then proove theretical properties of our approach, and show that interactions and communication protocols in UDCR has similarities with game theory concepts. We then recast UDCR as Partially Observable Stochastic Games and show how our approach can lead to significant privacy preservation.

References

Rachel Greenstadt, Jonathan P. Pearce, Emma Bowring, and Milind Tambe. Experimental analysis of privacy loss in dcop algorithms. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 1424–1426, New York, NY, USA, 2006. ACM.

Tal Grinshpoun. When you say (DCOP) privacy, what do you mean? - categorization of DCOP privacy and insights on internal constraint privacy. In Joaquim Filipe and Ana L. N. Fred, editors, *ICAART 2012 - Proceedings of the 4th International Conference on Agents and Artificial Intelligence, Volume 1 - Artificial Intelligence, Vilamoura, Algarve, Portugal, 6-8 February, 2012*, pages 380–386. SciTePress, 2012.

Thomas Léauté and Boi Faltings. Protecting privacy through distributed computation in multi-agent decision making. *J. Artif. Intell. Res.*, 47:649–695, 2013.

René Mandiau, Julien Vion, Sylvain Piechowiak, and Pierre Monier. Multi-variable distributed backtracking with sessions. *Appl. Intell.*, 41(3):736–758, 2014.

Stephen Marsh. *Trust in distributed artificial intelligence*, pages 94–112. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

Anatol Rapoport and Albert M Chammah. The game of chicken. *American Behavioral Scientist*, 10(3):10–28, 1966.

Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2010.

Julien Savaux, Julien Vion, Sylvain Piechowiak, René Mandiau, Toshihiro Matsui, Katsutoshi Hirayama, Makoto Yokoo, Shakre Elmane, and Marius-Calin Silaghi. Discsps with privacy recast as planning problems for self-interested agents. In *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*, pages 359–366. IEEE, 2016.

Marius-Calin Silaghi and Boi Faltings. A comparison of distributed constraint satisfaction techniques with respect to privacy. In *Proceedings of the 3rd workshop on distributed constraints reasoning (AAMAS - DCR-02). Bologna, Italy*, 2002.

Marius-Calin Silaghi, Djamila Sam-Haroud, and Boi Faltings. Distributed asynchronous search with private constraints. In Jeffrey S. Rosenschein Carles Sierra, Maria Gini, editor, *AGENTS 2000 Proceedings of the fourth international conference on Autonomous agents, Barcelona, Spain, June 3-7, 2000*, pages 177–178. ACM, 2000.

Marius-Calin Silaghi, Amit Abhyankar, Markus Zanker, and Roman Barták. Desk-mates (stable matching) with privacy of preferences, and a new distributed CSP framework. In *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference, Clearwater Beach, Florida, USA*, pages 671–677, 2005.

Michael Wooldridge and Nicholas R. Jennings. *Agent theories, architectures, and languages: A survey*. 1995.

Makoto Yokoo, Toru Ishida, Edmund H Durfee, and Kazuhiro Kuwabara. Distributed constraint satisfaction for formalizing distributed problem solving. In *Proceedings of the 12th IEEE International Conference on Distributed Computing Systems, June 9-12, 1992, Yokohoma, Japan*, pages 614–621. IEEE, 1992.

Makoto Yokoo, Edmund H Durfee, Toru Ishida, and Kazuhiro Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 10(5):673–685, 1998.

Makoto Yokoo, Edmund H Durfee, Toru Ishida, and Kazuhiro Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Transactions on knowledge and data engineering*, 10(5):673–685, 1998.

Chongjie Zhang and Victor R. Lesser. Coordinated multi-agent reinforcement learning in networked distributed pomdps. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011.

Roie Zivan and Amnon Meisels. Synchronous vs asynchronous search on discsps. In *Proceedings of the First European Workshop on Multi-Agent Systems (EUMAS), December 18-19, Oxford, England*, 2003.