Using reasoning patterns to simplify games

Dimitrios Antos and **Avi Pfeffer**
antos@fas.harvard.edu, avi@eecs.harvard.edu
Harvard School of Engineering and Applied Sciences
Cambridge, Massachusetts 02138, tel. 617-381-8123
http://www.people.fas.harvard.edu/~antos/simplify.html

Abstract

In complex strategic situations decision-making agents interact with many other agents and have access to many pieces of information throughout their play. This usually leads to game solving being a very complex, almost intractable procedure. Moreover, algorithms for solving games usually fail to explain how the various equilibria come about and how "plausible" they are. Reasoning patterns try to capture the strategic thinking of agents and formalize the usage of the various information or evidence they obtain during their interactions. Identifying reasoning patterns can lead to a significant refinement over the full range of equilibria, as well as considerable computational savings in solving the game. Here we present a polynomial-time algorithm that simplifies the original game by iteratively identifying non-effective (ignorable) decision nodes and removing redundant information edges. In some cases, this can lead to exponential-time savings in computing an equilibrium, yet some –potentially efficient– equilibria may be lost in the process.

Introduction and previous work

Analyzing the behavior of rational agents in strategic situations is usually performed by constructing a game that describes their interactions, knowledge and payoffs. More recently, graphical means have been developed for representing games, in order to better capture the interaction structure, as well as agents' beliefs at each time through the game. One of the most expressive and powerful such graphical schemes is Multi-Agent Influence Diagrams (MAIDs), first presented by (**?**).

Solving a game usually entails enumerating its equilibria. However, as shown in (**?**), the complexity of computing a Nash equilibrium in general, even in the simplest situation with three agents (or two agents and non-zero-sum payoffs) is super-polynomial. In light of this discouraging result, the importance of being able to simplify a game and compute its equilibria in a smaller-scale version of it becomes extremely high.

In (**?**) the authors have presented an exhaustive list of just four reasoning patterns that may hold for all agent decisions in which their choice "matters." In particular, when the strategy space of the game is restricted in such a way that no agent differentiates his decision rule based on obtained information that has no effect on his/her utility, then for every decision of every agent, if the agent's choice has an effect on his/her utility, then at least one of four well-defined reasoning patterns will always hold. Moreover, they show that such a restriction of the strategy space preserves at least one Nash equilibrium. And whereas identifying the full range of equilibria is useful, sometimes being able to compute a single equilibrium efficiently has been preferred; see for example (**?**).

The four reasoning patterns identified include: *(i)* direct effect, whereby an agent can directly incluence his/her own utility, without the intervention of another agent's decision; *(ii)* manipulation, where an agent A cannot influence his/her own utility directly, but through the actions of another agent B, whose utility A can directly affect; *(iii)* signaling, whereby an agent A has access to information that is of interest to both him/her and another agent B, and that other agent can exert influence over A's utility; and *(iv)* revealing-denying, where an agent A can control whether another agent B gets access to information that is of interest to B, and that agent can influence A's utility. For a detailed, formal analysis of the four reasoning patterns and their derivations, the reader is referred to the original paper of (**?**).

Computational savings emerge by reasoning as follows: Every decision node that does not participate in any reasoning pattern (non-effective node) corresponds to a situation where the agent has no reason to differentiate between his/her choices; thus, the decision node can be ignored in a game-solving algorithm and replaced by a chance node giving equal probability to all available actions; plus, all information arcs incoming to that node can be removed. Furthermore, as (**?**) point out, any other information arc should be removed under a similar reasoning process, unless it satisfies a d-separation criterion. In particular, if an informational parent of a decision node has no effect to that agent's utility, except through the decision itself, then the information arc connecting it to the decision node is ignorable and should be eliminated.

Our work involves a polynomial-time algorithm that iteratively performs elimination of non-effective decision nodes and ignorable information arcs, by identifying reasoning

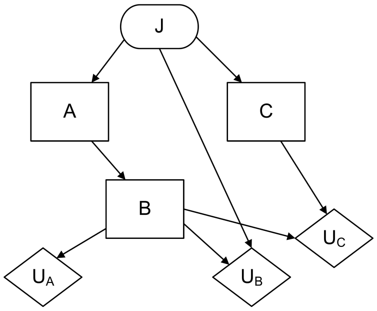

Figure 1: MAID representing the example game

patterns in the graph and applying the d-separation criterion to the remaining decision nodes' incoming information edges. Pseudocode for the main part of the algorithm can be seen below. Subroutines $\mathbf{df}$, $\mathbf{man}$, $\mathbf{sig}$, $\mathbf{rev}$ and $\mathbf{retract\_edges}$, which identify the four reasoning patterns and remove ignorable edges, respectively, are not presented in this restricted version of the algorithm's pseudocode.

Algorithm 1 Pseudocode for the simplification algorithm

```
Require: MAID G
  for $d$ in $D$ do
    effective($d$) $\leftarrow true$
  end for
  repeat
    $retracted \leftarrow false$; $simplified \leftarrow false$
    for $d$ in $D$ do
      if not ($\mathbf{df}(d)$ or $\mathbf{man}(d)$ or $\mathbf{sig}(d)$ or $\mathbf{rev}(d)$) then
        effective($d$) $\leftarrow false$; $simplified \leftarrow true$
        remove parents of $d$
      end if
    end for
    if $G = \mathbf{retract\_edges}(G)$ then
      $retracted \leftarrow true$
    end if
  until $retracted = simplified = false$
  return $G$
```

An example

Consider the following game, represented as a MAID in fig.1: We have agents A, B and C. Agent A draws a card J, whose value can either be H, M or L. Only agents A and C have knowledge of that card, yet A may communicate its value to agent B, not necessarily truthfully. B gains \$30 by guessing the value of the card correctly. A gains \$10, \$5 or \$1 if B guesses H, M or L, respectively, no matter what the real value of the card is. C, on the other hand, gains \$30 if his choice of H, M or L differs from that of B.

This game has an extensive game form representation with 3^4 leaves. On the other hand, our algorithm can discover the following subtleties in the above scenario: Agent C's decision is not affected by the card value, so the information arc (J, C) can be removed. Then, the decision node for A is non-effective, i.e. A has no reason to act differently upon seeing any of the card values. Knowing that, B will ignore what A tells her and just randomize equally between H, M and L. Thus the large game is reduced to two mini-games of 3^2 leaves each (fig.2). Adding more players to the game, one can easily show that computational savings are exponential in the number of agents, whereas the algorithm that breaks the game down runs in polynomial time.

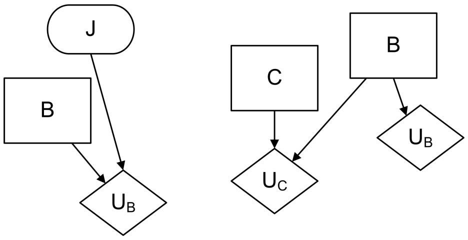

Figure 2: Simplified MAIDs

Of course this process is not without loss. Removing the redundant edges following the algorithm of (**?**) happens to eliminate certain, possibly efficient, equilibria. For example in our little scenario an equilibrium could be as follows: A always communicates H to B, B always believes him and C randomizes equally between M and L. This is clearly a Nash equilibrium: if B always believes him, then A responds best to that by communicating always H to her. Knowing that A will always communicate H and B will believe him and respond with H as well, C will choose either M or L to win. Finally, since the card is H with probability $\frac{1}{3}$ no matter what A tells her, B's strategy to always believe A is equally good as pure randomization. In this equilibrium, the players' expected payoffs are $\$(10, 10, 20)$, whereas in our reduced game equlibirium all players will randomize equally, yielding expected payoffs $\$(\frac{17}{3}, 10, 20)$. One can see that the equilibrium that was "lost" is a Pareto-improvement upon the one that has been retained.

References

Constantinos Daskalakis, Paul W. Goldberg and Christos H. Papadimitriou, *The Complexity of Computing a Nash Equilibrium*, In the 38th ACM Symposium on Theory of Computing, STOC, 2006.

Daphne Koller and Brian Milch, *Multi-Agent Influence Diagrams for Representing and Solving Games*, IJCAI, pp. 1027-1036, 2001.

Daphne Koller and Brian Milch, *Ignorable Information in Multi-Agent Scenarios*, to be published in 2008.

Pierfrancesco La Mura, *Game Networks*, Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, pp. 335-342, 2000.

Avi Pfeffer and Ya'akov Gal, *On the Reasoning Patterns of Agents in Games*, National Conference of Artificial Intelligence, 2007.