Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2008-029 May 12, 2008

Ignorable Information in Multi-Agent Scenarios

Brian Milch and Daphne Koller

Ignorable Information in Multi-Agent Scenarios*

Brian Milch
Computer Science & Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA 02139-4307, USA
milch@csail.mit.edu

Daphne Koller
Computer Science Dept.
Stanford University
Stanford, CA 94305-9010, USA
koller@cs.stanford.edu

Abstract

In some multi-agent scenarios, identifying observations that an agent can safely ignore reduces exponentially the size of the agent's strategy space and hence the time required to find a Nash equilibrium. We consider games represented using the multi-agent influence diagram (MAID) framework of (Koller and Milch 2001), and analyze the extent to which information edges can be eliminated. We define a notion of a *safe* edge removal transformation, where all equilibria in the reduced model are also equilibria in the original model. We show that existing edge removal algorithms for influence diagrams are safe, but limited, in that they do not detect certain cases where edges can be removed safely. We describe an algorithm that produces the "minimal" safe reduction, which removes as many edges as possible while still preserving safety. Finally, we note that both the existing edge removal algorithms and our new one can eliminate equilibria where agents coordinate their actions by conditioning on irrelevant information. Surprisingly, in some games these "lost" equilibria can be preferred by all agents in the game.

1 Introduction

Consider a scenario where agents make observations and decisions over time. At each decision point, the agent's strategy can depend on everything he has

*This manuscript was written in 2002 when the first author was at the University of California, Berkeley.

observed up to that point. Specifically, a *strategy* specifies a probability distribution over actions for each possible instantiation of all of the variables observed. Thus, the size of the strategy space for an agent grows exponentially with the number of observations the agent makes. Most algorithms for finding good strategies have running time that depends at least linearly on the size of the strategy space, and hence exponentially on the number of observations.

Graphical models, such as Bayesian networks and influence diagrams, explicitly represent conditional independence relations between variables. This information may allow us to conclude efficiently that an agent can safely ignore some observed variables when making a particular decision. For the single-agent case, there are well-known algorithms that identify ignorable information in influence diagrams (IDs) (Shachter, 1990; Zhang and Poole, 1992; Shachter, 1999; Nielsen and Jensen, 1999; Lauritzen and Nilsson, 2001). These algorithms remove ignorable information edges, and are *safe* in the sense that an optimal strategy for the reduced ID is optimal in the original ID as well.

In this paper, we extend this analysis of ignorable information to the multi-agent case. We use the framework of *multi-agent influence diagrams* (MAIDs) of (Koller and Milch, 2001). Our definition of a safe reduction in the multi-agent case is based on the concept of a *Nash equilibrium* (Nash, 1950): an assignment of strategies to agents in which no agent has an incentive to deviate to another strategy, assuming the other agents adhere to their assigned strategies. We say that a reduced version of a given MAID is *safe* if every Nash equilibrium of the reduced MAID is also an equilibrium of the original MAID.

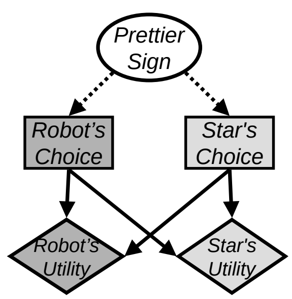

Figure 1: The Robot-Phobic Movie Star example.

Example 1.1 (The Robot-Phobic Movie Star)**.** *A movie star and a robot groupie are both deciding which of two restaurants to dine at tonight. The robot's goal is to eat at the same restaurant as the movie star, but the movie star wants to avoid the robot, whom he finds disconcerting. The agents make their choice simultaneously, but before choosing, they can both observe which restaurant has a prettier sign — a variable that affects neither of their utilities. A MAID for this example is shown in Figure 1. Each agent might choose a strategy that depends on Prettier-Sign (e.g., go to the restaurant with the less pretty sign). However, the unique equilibrium in this example is one where each agent sim-*

ply selects one of the two restaurants at random, with equal probability. Thus, the reduced MAID where we remove the edge from Prettier-Sign to both agents' decisions is safe.

The standard edge removal algorithm for single-agent IDs iteratively removes ignorable edges. We show that this algorithm also yields a safe reduction in multi-agent scenarios, but may not simplify the scenario as much as possible. In Example 1.1, the movie star cannot safely ignore *Prettier-Sign* because the robot might condition on it, and vice versa. Thus, neither of the information edges is ignorable. But removing both edges yields a safe reduction: if one agent ignores *Prettier-Sign*, the other can never gain by conditioning on it. We provide a new algorithm that finds such safe reductions by removing all information edges from the given MAID, and then adding back the ones necessary for safety.

Finally, although both edge removal and edge addition algorithms yield safe reductions of a given MAID, there may be equilibria in the original MAID that are not possible in the reduced MAID. Somewhat surprisingly, we show that both algorithms sometimes eliminate the equilibrium that provides the highest expected utilities to all the agents. Thus, removing ignorable information in a MAID may involve a tradeoff between computational efficiency and quality of the resulting equilibrium.

2 Multi-Agent Influence Diagrams

A *multi-agent influence diagram* (MAID) (Koller and Milch, 2001) consists of a *MAID skeleton* $\mathcal{G}$ and a *parameterization* Pr. The skeleton is a directed acyclic graph whose nodes are partitioned into three sets: a set $\mathcal{X}$ of chance nodes (drawn as ovals), a set $\mathcal{D}$ of decision nodes (drawn as rectangles), and a set $\mathcal{U}$ of utility nodes (drawn as diamonds). We write $Pa_{\mathcal{G}}(X)$ to denote the parents of a node X in $\mathcal{G}$. Utility nodes must be leaf nodes in this graph. Each chance or decision node $X \in \mathcal{X} \cup \mathcal{D}$ has a finite domain $dom(X)$. Associated with $\mathcal{G}$ is a set $\mathcal{A}$ of agents. The decision nodes are partitioned into subsets $\{\mathcal{D}_a\}_{a\in\mathcal{A}}$ that are controlled by the various agents, and the utility nodes are partitioned into subsets $\{\mathcal{U}_a\}_{a\in\mathcal{A}}$ that define the various agents' utility functions.

A parameterization Pr for $\mathcal{G}$ assigns to each chance node $X \in \mathcal{X}$ a conditional probability distribution (CPD) $\Pr(X \mid Pa_{\mathcal{G}}(X))$. The parameterization also assigns a CPD $\Pr(U \mid Pa_{\mathcal{G}}(U))$ to each utility node $U \in \mathcal{U}$, but this CPD is deterministic: for each instantiation $\mathbf{pa}$ of $Pa_{\mathcal{G}}(U)$, it assigns probability 1 to a single real number.

A MAID does not define CPDs for decision nodes, because the agents get to choose the values of these nodes. Before making a decision D, an agent gets to observe the values of D's parents, so the value of D depends (perhaps stochastically) on some subset of these observations. For this reason, edges into a decision node are called *information edges*; they are drawn as dotted lines. A *decision rule* δ for a decision node D in $\mathcal{G}$ consists of a *consideration set* $\boldsymbol{S} \subset Pa_{\mathcal{G}}(D)$ and a table f that maps each instantiation $\mathbf{s}$ of $\boldsymbol{S}$ to a probability distribution $f(D \mid \mathbf{s})$ over $dom(D)$. We say that $\delta = (\boldsymbol{S}, f)$ *considers* the nodes in $\boldsymbol{S}$, and *ignores* all subsets of $Pa_{\mathcal{G}}(D)$ that do not intersect $\boldsymbol{S}$.

A *strategy* σ_a for agent a assigns a decision rule $\sigma_a(D)$ to each $D \in \mathcal{D}_a$, and a *strategy profile* σ assigns a decision rule $\sigma(D)$ to every decision node D. A decision rule $\delta = (\boldsymbol{S}, f)$ for a node D in $\mathcal{G}$ defines a CPD for D given its parents. For any instantiations $\mathbf{s}$ of $\boldsymbol{S}$ and $\mathbf{r}$ of $Pa_{\mathcal{G}}(D) \setminus \boldsymbol{S}$, the distribution is $\delta(D \mid \mathbf{s}, \mathbf{r}) = f(D \mid \mathbf{s})$. Thus, specifying a decision rule for a decision node D allows us to convert D into a chance node.

Given a MAID $\mathcal{M}$ and a partial strategy profile σ that assigns decision rules to a set $\boldsymbol{D}$ of decision nodes, we can define the *induced MAID* $\mathcal{M}[\sigma]$ that is the same as $\mathcal{M}$, except that each node $D \in \boldsymbol{D}$ has been converted to a chance node with its CPD given by $\sigma(D)$. If σ assigns a decision rule to every decision node, then $\mathcal{M}[\sigma]$ has no decision nodes left, and is simply a Bayesian network (BN). Thus, a MAID defines a set of possible strategy profiles, and maps each strategy profile to a BN.

The *expected utility* of a strategy profile σ for an agent a in $\mathcal{M}$ is: $\mathrm{EU}^a_{\mathcal{M}}(\sigma) = E_{P_{\mathcal{M}[\sigma]}}\left[\sum_{U \in \mathcal{U}_a} U\right]$. A decision rule δ for D is *optimal for the strategy profile* σ in $\mathcal{M} = (\mathcal{G}, \mathrm{Pr})$ if for every alternative decision rule δ' at D, $\mathrm{EU}^a_{\mathcal{M}}(\sigma) \geq \mathrm{EU}^a_{\mathcal{M}}((\sigma_{-D}, \delta'))$, where (σ_{-D}, δ') is the same as σ except that it assigns the decision rule δ' to D. A strategy σ_a for agent a is a *best response* to an assignment σ_{-a} of strategies to the other agents if $\mathrm{EU}^a_{\mathcal{M}}((\sigma_{-a}, \sigma_a)) \geq \mathrm{EU}^a_{\mathcal{M}}((\sigma_{-a}, \sigma'_a))$ for every alternative strategy σ'_a for agent a. Finally, a strategy profile σ is a *Nash equilibrium* in $\mathcal{M}$ if for every agent a, the strategy σ_a that it assigns to a is a best response to the partial strategy profile σ_{-a} that it assigns to the other agents.

3 Reductions and Safety

As we discussed in the introduction, the set of possible decision rules at a decision node grows exponentially with the number of its parents. This blowup is very problematic in domains where agents have access to a large number of observations. In particular, the complexity of most algorithms for finding Nash equilibria, or even for verifying that a given strategy profile σ is an equilibrium, grow exponentially with the dimension of the strategy space. We can therefore gain significant computational savings if we remove "irrelevant" parents of decision nodes, and then find an equilibrium in the resulting reduced MAID.

Definition 3.1. *A MAID skeleton $\mathcal{G}'$ is a* reduction *of a MAID skeleton $\mathcal{G}$ if $\mathcal{G}'$ can be constructed from $\mathcal{G}$ by removing some set of information edges.*

If $\mathcal{G}'$ is a reduction of $\mathcal{G}$, then both $\mathcal{G}$ and $\mathcal{G}'$ allow the same set of possible parameterizations. We say that a MAID $\mathcal{M}' = (\mathcal{G}', \mathrm{Pr})$ is a reduction of $\mathcal{M} = (\mathcal{G}, \mathrm{Pr})$ if $\mathcal{G}'$ is a reduction of $\mathcal{G}$. By removing information edges into a node D, we restrict the set of possible decision rules at D. Therefore the set of possible strategy profiles in a reduction $\mathcal{M}'$ is a subset of the strategy profiles available in the original MAID $\mathcal{M}$. However, the mapping from strategy profiles to expected utilities is unchanged: any strategy profile that is available in both $\mathcal{M}$ and $\mathcal{M}'$ yields the same expected utility for a given agent in both $\mathcal{M}$ and $\mathcal{M}'$.

Let $\mathcal{M}'$ be some reduction of a MAID $\mathcal{M}$. Suppose that a strategy profile σ is a Nash equilibrium in $\mathcal{M}$. If σ is a legal strategy in $\mathcal{M}'$, it must also be a Nash equilibrium in $\mathcal{M}'$, since the set of alternative strategies for each agent is smaller. On the other hand, suppose σ is a Nash equilibrium in $\mathcal{M}'$. Then no agent has an incentive to deviate to an alternative strategy that is available in $\mathcal{M}'$, but some strategy available in $\mathcal{M}$ might be more attractive. So σ is not necessarily an equilibrium in $\mathcal{M}$. Thus, if we want to find an equilibrium for $\mathcal{M}$ by finding an equilibrium in a reduction, we must choose the reduction with care. Specifically, we need a reduction that is *safe* in the following sense:

Definition 3.2. *A reduction $\mathcal{M}'$ of a MAID $\mathcal{M}$ is a* safe reduction *of $\mathcal{M}$ if every strategy profile that is a Nash equilibrium in $\mathcal{M}'$ is also a Nash equilibrium in $\mathcal{M}$. A MAID skeleton $\mathcal{G}'$ is a safe reduction of $\mathcal{G}$ if for every parameterization* $\Pr$ *of $\mathcal{G}$, $(\mathcal{G}', \Pr)$ is a safe reduction of $(\mathcal{G}, \Pr)$.*

In this paper, we concentrate on safe reductions of MAID skeletons, rather than safe reductions of particular MAIDs. As we show, at the skeleton level, we can construct safe reductions that consider only graphical criteria, which can be checked in polynomial time.

In the single agent case, Definition 3.2 reduces to a requirement that any optimal strategy in the reduced ID is also an optimal strategy in the original ID. Several papers have proposed graph-based algorithms for finding safe reductions for skeletons of single-agent IDs (Shachter, 1990; Zhang and Poole, 1992; Shachter, 1999; Nielsen and Jensen, 1999). These algorithms are based on the insight that some parents of a decision node may be *ignorable*: the agent has no incentive to consider them, because he has an optimal decision rule that ignores them.

Definition 3.3. *A set $\boldsymbol{X} \subset Pa_{\mathcal{G}}(D)$ is* ignorable *at D in a MAID skeleton $\mathcal{G}$ if, for any parameterization* $\Pr$ *of $\mathcal{G}$ and any strategy profile σ for $\mathcal{G}$, there is a decision rule for D that ignores $\boldsymbol{X}$ and is optimal for σ in $(\mathcal{G}, \Pr)$.*

This definition of ignorable information is *local* in that it only considers the usefulness of information for making a single decision, and does not make any assumptions about the decision rules adopted at other nodes—even nodes controlled by the same agent. It is easy to test for ignorability using d-separation. If $D \in \mathcal{D}_a$, then $\mathit{RelUtils}_{\mathcal{G}}(D)$ denotes the utility nodes that matter to a when making decision D: that is, the elements of $\mathcal{U}_a$ that are descendants of D in $\mathcal{G}$.

Theorem 3.4. *A set $\boldsymbol{X} \subset Pa_{\mathcal{G}}(D)$ is ignorable at D in $\mathcal{G}$ if and only if $d\text{-}sep_{\mathcal{G}}(\boldsymbol{X}, \mathit{RelUtils}_{\mathcal{G}}(D) \mid (Pa(D) \cup \{D\}) \setminus \boldsymbol{X})$. Furthermore, there is a unique subset of $Pa_{\mathcal{G}}(D)$, which we will call $\mathit{Ignorable}_{\mathcal{G}}(D)$, such that $\boldsymbol{X}$ is ignorable at D in $\mathcal{G}$ if and only if $\boldsymbol{X} \subset \mathit{Ignorable}_{\mathcal{G}}(D)$.*

This d-separation criterion allows us to determine efficiently which parents are ignorable at D. We make a single pass over $\mathcal{G}$ with a breadth-first search algorithm that follows active paths (Geiger *et al.*, 1990; Shachter, 1998). We start the breadth-first traversal at $RelUtils_{\mathcal{G}}(D)$; if it does not find an active path to a node $X \in Pa_{\mathcal{G}}(D)$ given $Pa(D) \cup \{D\}$, then X is ignorable at D.

4 Algorithms

In this section, we describe algorithms that use the d-separation criterion of Theorem 3.4 to find safe reductions of MAIDs. We begin with the edge removal algorithm, proposed for the single-agent case; we show in Section 5 that it also applies to the multi-agent case. We then discuss why this algorithm can be overly conservative, and introduce a new algorithm based on the notion of an edge addition fixpoint.

4.1 Stepwise Edge Removal

The most obvious approach to removing edges from a MAID skeleton $\mathcal{G}$ is to simply remove all the ignorable parents from some decision node D. More precisely, we define REMOVE-EDGES$(\mathcal{G}, D)$ to be the skeleton resulting from removing the edges between $Ignorable_{\mathcal{G}}(D)$ and D. If we apply this operator enough times to all the decision nodes in a MAID, we eventually reach an *edge removal fixpoint*:

Definition 4.1. *A MAID skeleton $\mathcal{G}$ is an* edge removal fixpoint *if, for every $D \in \mathcal{D}$,* REMOVE-EDGES$(\mathcal{G}, D) = \mathcal{G}$.

In an edge removal fixpoint, no information edges are ignorable. We define the algorithm REMOVE-UNTIL-FIXPOINT to be the algorithm that iterates over the nodes in some order until no more edges can be removed. The order in which we examine the decision nodes does not matter for correctness, but does matter for efficiency; we address this issue in Section 4.3.

Example 4.2 (The Food-Loving Movie Star)**.** *Consider again our robot and movie star scenario, but now suppose the movie star has overcome his discomfort with robots, and just wants to eat at the restaurant with the best food. The robot knows which restaurant has better food, but its only goal is to end up in the same restaurant as the star. The movie star does not know which restaurant's food is better, but he can observe where the robot goes before making his decision. The MAID for this example is shown in Figure 2.*

REMOVE-UNTIL-FIXPOINT iterates over the decision nodes in some order; suppose it processes *Robot's-Choice* before *Star's-Choice*. *Better-Food* is ignorable at *Robot's-Choice*: conditioning on *Better-Food* cannot help the robot catch the movie star, because the movie star cannot observe it. So the edge *Better-Food* → *Robot's-Choice* is removed. However, *Prettier-Sign* is not ignorable at this point because the movie star might condition on it. Moving on to *Star's-Choice*, the algorithm finds that both *Robot's-Choice* and *Prettier-Sign*

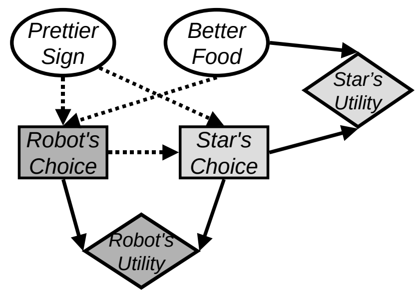

Figure 2: The Food-Loving Movie Star example.

are ignorable, because *Robot's-Choice* is no longer an indicator of *Better-Food*. So the two edges into *Star's-Choice* are removed. Then in the second iteration, the algorithm returns to *Robot's-Choice* and finds that *Prettier-Sign* is now ignorable.

However, there are other scenarios where REMOVE-UNTIL-FIXPOINT is unable to remove any edges, even though a smaller safe reduction exists. One example is our scenario of the robot-phobic movie star, where neither of the edges from *Prettier-Sign* to the two decision nodes can be removed.

4.2 Finding an Edge Addition Fixpoint

Consider the process by which we convince ourselves that removing both information edges in Example 1.1 is safe. The argument is that in the reduction where these information edges are removed, if we offered one agent the opportunity to condition on *Prettier-Sign*, there would be no incentive to take it. In other words, if we added one information edge without the other, that edge would be ignorable in the resulting MAID.

We can define an operator, which we call ADD-EDGES, which starts out from an already reduced MAID $\mathcal{G}'$, and considers reintroducing to a decision D the parents $\boldsymbol{X} = Pa_{\mathcal{G}}(D) \setminus Pa_{\mathcal{G}'}(D)$ that existed in the original MAID $\mathcal{G}$. However, it then eliminates those parents that are now ignorable. More precisely, we define $\boldsymbol{X}_{\text{ignore}} = \mathit{Ignorable}_{\mathcal{G}(\boldsymbol{x} \to D)}(D)$, and only add the parents in $\boldsymbol{X} \setminus \boldsymbol{X}_{\text{ignore}}$.

An edge addition fixpoint is defined as follows:

Definition 4.3. *If $\mathcal{G}'$ is a reduction of $\mathcal{G}$, then $\mathcal{G}'$ is a $\mathcal{G}$-*edge addition fixpoint *if for every $D \in \mathcal{D}$,* ADD-EDGES$(\mathcal{G}, \mathcal{G}', D) = \mathcal{G}'$*.*

Our argument above leads to the intuition that, if $\mathcal{G}'$ is a $\mathcal{G}$-edge addition fixpoint, then $\mathcal{G}'$ is a safe reduction of $\mathcal{G}$. We prove this property in Section 5.1.

We can define a function ADD-UNTIL-FIXPOINT$(\mathcal{G}, \mathcal{G}')$ that starts with any reduction $\mathcal{G}'$ of $\mathcal{G}$, and applies ADD-EDGES until it reaches a fixpoint. To get the smallest possible fixpoint, we can run this algorithm on $\mathcal{G}^{\varnothing}$, which is the reduction of $\mathcal{G}$ that has no information edges.

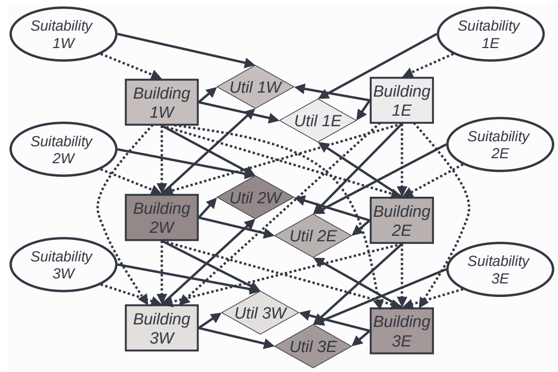

Figure 3: The extended Road example with $n = 6$.

It is easy to see that Add-Until-Fixpoint($\mathcal{G}, \mathcal{G}^{\varnothing}$) gives the desired result in Example 1.1. To provide a more complex scenario, we consider an extension of the Road example presented in (Koller and Milch, 2001).

Example 4.4. *Suppose a road is being built from north to south through undeveloped land, and n agents have purchased plots of land along the road. As the road reaches each agent's plot, the agent needs to choose what to build on his land: a factory, a shopping mall, or some houses. His utility depends on what he builds, on some private information about which type of building his land is suitable for, and on what is built north, south, and across the road from his land. The agent can observe what has already been built on all plots north of his land (on both sides of the road), but he cannot observe what will be built across from his land or south of it. A MAID for this example with $n = 6$ is shown in Figure 3.*

In the version of the Road example used by (Koller and Milch, 2001), agents cannot observe what was built more than one plot north of them. That restriction is necessary to keep the strategy spaces from growing exponentially: in our new version, an agent in row k can observe $2(k-1)$ buildings to the north, plus a suitability node, yielding $3^{(2k-1)}$ parent instantiations. The resulting strategy space grows exponentially in the number of agents, and finding equilibria for this scenario is therefore infeasible in any but the smaller scenarios.

Remove-Until-Fixpoint does not remove any edges: each agent can observe the same buildings as his neighbor across the road, and he wants to predict his neighbor's decision, so he cannot ignore those buildings unless his neighbor ignores them as well. Fortunately, Add-Until-Fixpoint($\mathcal{G}, \mathcal{G}^{\varnothing}$) yields a safe reduction where information edges between non-adjacent rows are removed.

Suppose the procedure iterates over the decision nodes from north to south and from west to east. At every decision node, it adds an edge from the corresponding suitability node. At *Building-2W*, it also adds an edge from the northern neighbor *Building-1W*. However, it does not yet add an edge from

Building-1E, because in the absence of an edge from *Building-1E* to *Building-2E*, *Building-1E* does not affect anything agent *2W* cares about. Moving on to *Building-2E*, the algorithm adds an edge from *Building-1W* (as well as from *Building-1E*) because *Building-1W* may be useful to agent *2E* in predicting agent *2W*'s action. At *Building-3W*, the algorithm adds an edge from *Building-2W*. It does not add an edge from *Building-1W*, because once agent *3W* knows what was built immediately north of him, knowing what was built two rows north is not helpful. This situation illustrates why ADD-EDGES does not consider adding each element of $Pa_{\mathcal{G}}(D)$ individually: indeed, *Building-1W* would be not be ignorable if it were the only information agent *3W* had.

ADD-UNTIL-FIXPOINT continues in this way until it completes one iteration. In the second iteration, it adds an edge *Building-kE* $\rightarrow$ *Building-(k+1)W* for each k, since *Building-kE* may now be useful for predicting agent $(k+1)E$'s decision. The third iteration adds no edges, and the algorithm terminates. Edges between non-consecutive rows are never added.

In the resulting reduced MAID, each decision has at most three parents: its two neighbors to the north, and its own suitability node. The resulting strategy space is significantly smaller (only $3^3 = 27$ information sets). As shown in (Koller and Milch, 2001), the resulting game can be decomposed and solved in time linear in the number of agents.

4.3 Finding Fixpoints Efficiently

Both REMOVE-UNTIL-FIXPOINT and ADD-UNTIL-FIXPOINT are guaranteed to terminate, because they can only remove or add as many information edges as exist in the original MAID, and they terminate if they make an iteration over $\mathcal{D}$ without removing or adding edges. However, in the worst case, they will only remove or add one edge per iteration. Thus, the total number of calls to REMOVE-EDGES or ADD-EDGES may be $|Pa_{\mathcal{G}}(\mathcal{D})| \cdot |\mathcal{D}|$, where $Pa_{\mathcal{G}}(\mathcal{D})$ is the sum of $|Pa_{\mathcal{G}}(D)|$ for all $D \in \mathcal{D}$.

It is well known that for edge removal in single-agent IDs, it suffices to iterate over the decision nodes in reverse chronological order, and process each decision node only once (Nielsen and Jensen, 1999). Can we do something similar in MAIDs, for both edge removal and edge addition? It turns out that we can, by exploiting the *relevance graph* introduced by (Koller and Milch, 2001).

A decision node D' is *strategically relevant* to a decision node D if the choice of decision rule at D' can affect the optimality of a decision rule at D; see (Koller and Milch, 2003) for a precise definition. Intuitively, D' is relevant to D if the CPD of D' can affect the utility nodes that are relevant to D. Using the criterion of (Geiger *et al.*, 1990), Koller and Milch define a node D' to be *s-reachable* from D in $\mathcal{G}$ if there is an active path from a new parent $\widehat{D'}$ of D' to $RelUtils_{\mathcal{G}}(D)$ given $Pa_{\mathcal{G}}(D) \cup \{D\}$. They show that s-reachability is a sound and complete criterion for determining strategic relevance at the skeleton level.

As strategic relevance is a binary relation, it an be represented as a directed graph.

Definition 4.5. *The* relevance graph *for a MAID skeleton $\mathcal{G}$, denoted Rel($\mathcal{G}$), is the directed graph whose nodes are the decision nodes of $\mathcal{G}$, and which contains an edge from D' to D if and only if D strategically relies on D.*[1]

The relevance graph for the Road example is shown in Figure 4.

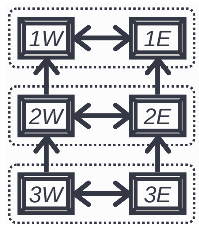

Figure 4: The relevance graph for the Road MAID ($n = 6$), with strongly connected components outlined.

Koller and Milch showed that if two fully mixed strategy profiles[2] σ and σ' differ only at nodes that D does not rely on, then the set of decision rules for D that are optimal for σ is the same as the set optimal for σ'. We can derive a similar result relating strategic relevance and ignorability. Roughly, if two skeletons $\mathcal{G}$ and $\mathcal{G}'$ differ (i.e., assign different parents) only at nodes that D does not rely on, then $\mathit{Ignorable}_{\mathcal{G}}(D) = \mathit{Ignorable}_{\mathcal{G}'}(D)$. The following lemma (which generalizes Lemma 10 in (Lauritzen and Nilsson, 2001)) makes this statement more precise:

Lemma 4.6. *Let $\mathcal{G}'$ be a reduction of $\mathcal{G}$. If $\mathit{Pa}_{\mathcal{G}}(D) = \mathit{Pa}_{\mathcal{G}'}(D)$, and $\mathit{Pa}_{\mathcal{G}}(D') = \mathit{Pa}_{\mathcal{G}'}(D')$ for all D' that D relies on in $\mathcal{G}$, then $\mathit{Ignorable}_{\mathcal{G}}(D) = \mathit{Ignorable}_{\mathcal{G}'}(D)$.*

Note that, in order to apply this lemma, we must check strategic relevance in the original skeleton $\mathcal{G}$. It is not sufficient to consider strategic relevance in $\mathcal{G}'$.

Given Lemma 4.6, it is fairly obvious that if *Rel*($\mathcal{G}$) is acyclic, we can obtain an edge removal fixpoint by taking a topological ordering of *Rel*($\mathcal{G}$) and processing each decision node once in this order. Removing a parent from a node D' cannot make something ignorable at a node D earlier in the ordering, because D does not rely on D'. We can obtain an edge addition fixpoint using this same ordering as well.

When *Rel*($\mathcal{G}$) is cyclic, we cannot just find a topological ordering. However, we can use an algorithm similar to the divide-and-conquer algorithm of (Koller

[1] The edges in this definition are the reverse of those in (Koller and Milch, 2001); the definition was changed in (Koller and Milch, 2003) to make the parent relationship more analogous to the parent relationship in BNs.

[2] A strategy profile is fully mixed if all actions have non-zero probability at all information sets.

and Milch, 2003), which breaks the relevance graph into *strongly connected components.*

Definition 4.7. *A set $\boldsymbol{C}$ of nodes in a directed graph is a* strongly connected component *(SCC) if for every pair of nodes $D \neq D' \in \boldsymbol{C}$, there exists a directed path from D to D'. An SCC is a* maximal *SCC if it is not a strict subset of any other SCC.*

Note that if $Rel(\mathcal{G})$ is acyclic, then each SCC of $Rel(\mathcal{G})$ consists of a single node.

We can thus break our relevance graph up into maximal SCCs, which can in turn be sorted in some topological ordering. Within an SCC, removing or adding parents at one node may change the ignorable set at another, so we may need to loop over the nodes in the SCC several times. However, if we process SCCs in toplogical order, the removal or addition of edges for one SCC cannot affect the ignorable sets in the preceding SCCs. We can use this ordering for our algorithms Remove-Until-Fixpoint and Add-Until-Fixpoint For example, on the Road MAID, if we apply Add-Until-Fixpoint in order of the SCCs, the procedure first loops over $\{Building\text{-}3W, Building\text{-}3E\}$ until it cannot add any more edges; then it loops over $\{Building\text{-}2W, Building\text{-}2E\}$, and finally over $\{Building\text{-}1W, Building\text{-}1E\}$.

This ordering is considerably more efficient than an arbitrary ordering. In the worst case, a call to these algorithms loops over the elements of $\boldsymbol{C}_j$ once for each information edge into decision nodes in $\boldsymbol{C}_j$, removing one edge each time, and then terminates after one more cycle. Hence, the maximum number of times any decision node is processed is $\max_j(|Pa_{\mathcal{G}}(\boldsymbol{C}_j)| + 1)$, which we will call K. The total number of calls to Remove-Edges or Add-Edges is therefore bounded by $K \cdot |\mathcal{D}|$, as compared to $(|Pa_{\mathcal{G}}(\mathcal{D})|+1) \cdot |\mathcal{D}|$ for an arbitrary ordering.

5 Properties of the Algorithms

5.1 Safety

We would like to prove that if $\mathcal{G}'$ is a $\mathcal{G}$-edge addition fixpoint, then $\mathcal{G}'$ is a safe reduction of $\mathcal{G}$. However, we need to include a caveat related to imperfect recall.

Example 5.1 (Forgetful Movie Star)**.** *Consider Example 1.1, but where the robot-phobic movie star must choose a restaurant again a week later. His main goal is still to avoid the robot the first night, but he also receives some utility for choosing the same restaurant both times. The difficulty is that on the second night, he does not remember where he went the first night. However, he has some information that the robot does not have: he knows which restaurant sponsored the winning youth baseball team this year. Assume both restaurants are equally likely to sponsor the winning team. A MAID for this scenario is shown in Figure 5.*

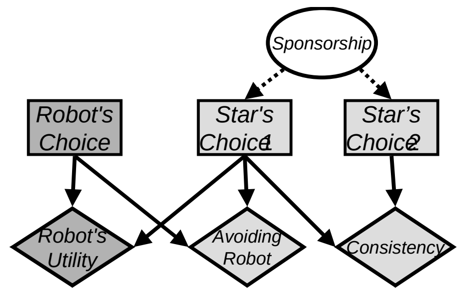

Figure 5: The Forgetful Movie Star example.

The reduction where both of the edges from Sponsorship to the movie star's decisions are removed is an edge addition fixpoint: the movie star cannot gain by conditioning on Sponsorship at one decision and not the other. The only equilibrium in this reduction is where the movie star picks a restaurant uniformly at random each night, and the robot picks uniformly at random too. However, this is not an equilibrium in the original MAID: the movie star can gain by deviating at both his decision nodes, and (for example) always going to the restaurant that sponsored the winning team. Then the robot still cannot predict the star's decision, and the star's restaurant choices are always consistent.

Such phenomena cannot arise if all agents have perfect recall in the original MAID. In fact, we can define a weaker criterion of *sufficient recall*. Let $Rel_a(\mathcal{G})$ denote the graph whose nodes are the decision nodes of agent a in $\mathcal{G}$, with an edge from D' to D if and only if D strategically relies on D.

Definition 5.2. *A MAID skeleton $\mathcal{G}$ has* sufficient recall *if for every agent a, the graph $Rel_a(\mathcal{G})$ is acyclic.*

In Example 5.1, the movie star's two decisions rely on each other, so the MAID does not have sufficient recall. Under the assumption of sufficient recall, we can now state the desired theorem: for any MAID skeleton $\mathcal{G}$ with sufficient recall, ADD-UNTIL-FIXPOINT$(\mathcal{G}, \mathcal{G}^{\varnothing})$ returns a safe reduction of $\mathcal{G}$.

Theorem 5.3. *If $\mathcal{G}$ has sufficient recall and $\mathcal{G}'$ is a $\mathcal{G}$-edge addition fixpoint, then $\mathcal{G}'$ is a safe reduction of $\mathcal{G}$.*

Proof. Assume for contradiction that a strategy profile σ is an equilibrium in $\mathcal{M}' = (\mathcal{G}', \mathrm{Pr})$, but that in $\mathcal{M} = (\mathcal{G}, \mathrm{Pr})$, some agent a wants to deviate to a strategy σ'_a. Because $Rel_a(\mathcal{G})$ is acyclic, we can use a backward induction algorithm to construct a strategy σ^*_a that is a best response to σ_{-a} in $\mathcal{M}$. In particular, σ^*_a is at least as good for a as σ'_a. Furthermore, by the definition of an edge addition fixpoint, at each node $D \in \mathcal{D}_a$ we can choose an optimal decision rule that ignores $Pa_{\mathcal{G}}(D) \setminus Pa_{\mathcal{G}'}(D)$. So we can construct σ^*_a so it is a possible strategy in $\mathcal{M}'$. But then a must want to deviate to σ^*_a in $\mathcal{M}'$ as well, contradicting the assumption that σ is an equilibrium in $\mathcal{M}'$. $\square$

We can use the safety result for edge addition to prove safety for the edge removal algorithm. We first show that edge removal is conservative relative to edge addition: if we remove edges using edge removal, we will not want to add any edges back.

Lemma 5.4. *If $\mathcal{G}'$ is a $\mathcal{G}$-edge addition fixpoint and D is any decision node in $\mathcal{G}'$, then the output of* Remove-Edges$(\mathcal{G}', D)$ *is a $\mathcal{G}$-edge addition fixpoint.*

Since $\mathcal{G}$ is trivially a $\mathcal{G}$-edge addition fixpoint, it follows from Theorem 5.3 that Remove-Until-Fixpoint$(\mathcal{G})$ returns a safe reduction of $\mathcal{G}$. This result generalizes a known result for the single-agent case (Shachter, 1990).

Another important property of stepwise edge removal in the single-agent case is that it does not add edges to the relevance graph (Lauritzen and Nilsson, 2001). We can generalize this result to all edge addition fixpoints, and hence to the output of our two multi-agent algorithms:

Theorem 5.5. *If $\mathcal{G}'$ is a $\mathcal{G}$-edge addition fixpoint, then every edge in Rel$(\mathcal{G}')$ is also an edge in Rel$(\mathcal{G})$.*

This theorem is useful for two reasons. First, the complexity of finding an equilibrium in a MAID using an algorithm like the divide-and-conquer algorithm of (Koller and Milch, 2001) depends on the size of the largest SCC in the relevance graph. Theorem 5.5 implies that the SCCs in the reduction generated by Add-Until-Fixpoint$(\mathcal{G}, \mathcal{G}^{\varnothing})$ are no larger than those in the original MAID.

Also, although every game has a Nash equilibrium (Nash, 1950), it may be that all the equilibria require an agent's moves at different decision nodes to be correlated. In a MAID with imperfect recall, there may not be a *behavior strategy profile* — an assignment of decision rules to individual decision nodes — that corresponds to such an equilibrium (Kuhn, 1953). However, we can show that in a MAID with sufficient recall, there is always a behavior strategy equilibrium. Theorem 5.5 guarantees that when we move from $\mathcal{G}$ to a $\mathcal{G}$-edge addition fixoint, we maintain sufficient recall, and so there is still a behavior strategy equilibrium.

5.2 Maximality and minimality

By definition, Remove-Until-Fixpoint$(\mathcal{G})$ yields an edge removal fixpoint, and we have seen that its output is also a $\mathcal{G}$-edge addition fixpoint. We will refer to such a double fixpoint as an *edge-stable reduction* of $\mathcal{G}$. It turns out that Add-Until-Fixpoint$(\mathcal{G}, \mathcal{G}^{\varnothing})$ also yields an edge-stable reduction, as we can show using the following lemma:

Lemma 5.6. *If $\mathcal{G}'$ is an edge removal fixpoint and D is any decision node in $\mathcal{G}'$, then* Add-Edges$(\mathcal{G}, \mathcal{G}', D)$ *returns an edge removal fixpoint.*

$\mathcal{G}$ may have many other edge-stable reductions besides the ones our algorithms return. For example, in the Road MAID, there is an edge-stable reduction where the agents in row 3 ignore *Building-1W*, but not *Building-1E*. We can define a partial order over edge-stable reductions, where we write $\mathcal{G}' \sqsubset \mathcal{G}$ if $\mathcal{G}'$ is a reduction of $\mathcal{G}$. Then we can prove the following:

Proposition 5.7. *For any skeleton $\mathcal{G}$, the reduction $\mathcal{G}'$ returned by* REMOVE-UNTIL-FIXPOINT($\mathcal{G}$) *is the* maximum edge-stable reduction *of $\mathcal{G}$, in the sense that $\mathcal{G}'' \sqsubset \mathcal{G}'$ for every edge-stable reduction $\mathcal{G}''$ of $\mathcal{G}$.*

Proposition 5.8. *For any skeleton $\mathcal{G}$, the reduction $\mathcal{G}'$ returned by* ADD-UNTIL-FIXPOINT($\mathcal{G}, \mathcal{G}^{\varnothing}$) *is the* minimum edge-stable reduction *of $\mathcal{G}$, in the sense that $\mathcal{G}' \sqsubset \mathcal{G}''$ for every edge-stable reduction $\mathcal{G}''$ of $\mathcal{G}$.*

Thus, we see that REMOVE-UNTIL-FIXPOINT($\mathcal{G}$) and ADD-UNTIL-FIXPOINT($\mathcal{G}, \mathcal{G}^{\varnothing}$) provide two extremes on a spectrum of possible edge-stable reductions, the first resulting in the largest number of remaining information edges, and the second in the smallest number. Interestingly, when $\mathcal{G}$ has an acyclic relevance graph, the two algorithms yield the same output, so $\mathcal{G}$ has exactly one edge-stable reduction. This implies that in single-agent IDs (with sufficient recall), the edge addition fixpoint algorithm provides no benefit over stepwise edge removal.

6 Discussion and Conclusions

Detecting ignorable information allows us to reduce the dimensionality of the strategy space in a decision problem, sometimes exponentially, allowing us to find strategies in scenarios that would otherwise be intractable. In single-agent scenarios, we can obtain a safe reduction of an influence diagram by stepwise removal of ignorable edges. We show that this reduction is still safe in the multi-agent case, but does not obtain the minimal safe reduction in many cases. We show that it is sufficient for a reduction $\mathcal{G}'$ to be a $\mathcal{G}$-edge addition fixpoint, and, based on this result, provide an algorithm that removes all information edges from $\mathcal{G}$ and then adds them back as necessary. Our addition fixpoint algorithm can reduce the strategy space exponentially in scenarios such as the Road example, where stepwise edge removal is not helpful.

Both the stepwise edge removal and the edge addition fixpoint algorithm are safe, in that they guarantee that we can find an equilibrium in the reduced MAID which is also an equilibrium in the original MAID. However, there may be other equilibria in $\mathcal{M}$ that are lost in the transformation to $\mathcal{M}'$. Surprisingly, even the conservative stepwise edge removal algorithm can eliminate the equilibrium that provides the highest expected utilities to all the agents. Recall that in Example 4.2, the robot knows which restaurant has better food, but its only goal is to end up at the same restaurant as the movie star. The movie star's only goal is to get the best meal. The equilibrium preferred by both agents is where the robot always chooses the restaurant with the best food, and the movie star always follows the robot. However, we saw that a single application of REMOVE-EDGES to *Robot's-Choice* removes the incoming edge from *Better-Food*. This eliminates the preferred equilibrium.

How do such results arise from our apparently conservative criterion for removing edges? Recall that $\boldsymbol{X}$ is ignorable at a decision node $D \in \mathcal{D}_a$ if for any strategy profile σ, there is a decision rule δ for D that ignores $\boldsymbol{X}$ and is optimal (in terms of a's utility) for σ. Suppose σ is the preferred equilibrium that we just described for Example 4.2, and D is *Robot's-Choice*. Given that the movie star will follow the robot, the robot is indifferent between always choosing the better restaurant and choosing a restaurant uniformly at random. So indeed there is a decision rule δ that ignores *Better-Food* and is still (weakly) optimal for σ. But the definition does not consider the fact that deviating to δ would disturb the equilibrium.

Thus, removing ignorable information in a MAID may involve a tradeoff between computational resources and quality of the solution. Removing edges can make the solution of a game computationally tractable, but may result in a suboptimal equilibrium. To avoid removing "good" equilibria, one might want to define an ignorability criterion that effectively asked, "Would anyone care if agent a ignored $\boldsymbol{X}$ at D?" Providing a formal definition for such a criterion, and for the notion of safety to which it corresponds, is a topic for future work.

Acknowledgements. This work was supported by the DoD MURI program under ONR Grant N00014-00-1-0637, and by Air Force contract F30602-00-2-0598 under DARPA's TASK program. The first author was also supported by a University of California Microelectronics Fellowship.

References

(Geiger *et al.*, 1990) D. Geiger, T. Verma, and J. Pearl. Identifying independence in Bayesian networks. *Networks*, 20:507–534, 1990.

(Koller and Milch, 2001) D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. In *Proc. 17th IJCAI*, pages 1027–1034, 2001.

(Koller and Milch, 2003) D. Koller and B. Milch. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45(1):181–221, 2003.

(Kuhn, 1953) H. Kuhn. Extensive games and the problem of information. *Annals of Mathematics Studies*, 28, 1953.

(Lauritzen and Nilsson, 2001) S. L. Lauritzen and D. Nilsson. Representing and solving decision problems with limited information. *Management Science*, 47(9):1235–51, 2001.

(Nash, 1950) J. Nash. Equilibrium points in n-person games. *PNAS*, 36:48–49, 1950.

(Nielsen and Jensen, 1999) T.D. Nielsen and F.V. Jensen. Welldefined decision scenarios. In *Proc. 15th UAI*, pages 502–511, 1999.

(Shachter, 1990) R. D. Shachter. An ordered examination of influence diagrams. *Networks*, 20:535–563, 1990.

(Shachter, 1998) R. D. Shachter. Bayes-ball: The rational pastime. In *Proc. 14th UAI*, pages 480–487, 1998.

(Shachter, 1999) R. D. Shachter. Efficient value of information computation. In *Proc. 15th UAI*, pages 594–601, 1999.

(Zhang and Poole, 1992) N. L. Zhang and D. Poole. Stepwise-decomposable influence diagrams. In *Proc. 3rd KR*, pages 141–152, 1992.