Equilibrium Refinements for Multi-Agent Influence Diagrams: Theory and Practice

Lewis Hammond
University of Oxford
lewis.hammond@cs.ox.ac.uk

James Fox
University of Oxford
james.fox@cs.ox.ac.uk

Tom Everitt
DeepMind
tomeveritt@google.com

Alessandro Abate
University of Oxford
aabate@cs.ox.ac.uk

Michael Wooldridge
University of Oxford
mjw@cs.ox.ac.uk

ABSTRACT

Multi-agent influence diagrams (MAIDs) are a popular form of graphical model that, for certain classes of games, have been shown to offer key complexity and explainability advantages over traditional extensive form game (EFG) representations. In this paper, we extend previous work on MAIDs by introducing the concept of a MAID subgame, as well as subgame perfect and trembling hand perfect equilibrium refinements. We then prove several equivalence results between MAIDs and EFGs. Finally, we describe an open source implementation for reasoning about MAIDs and computing their equilibria.

KEYWORDS

multi-agent influence diagrams; equilibrium refinements; extensive form games; probabilistic graphical models

ACM Reference Format:
Lewis Hammond, James Fox, Tom Everitt, Alessandro Abate, and Michael Wooldridge. 2021. Equilibrium Refinements for Multi-Agent Influence Diagrams: Theory and Practice. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

Multi-agent influence diagrams (MAIDs) are a compact and expressive graphical representation for non-cooperative games. Introduced by Koller and Milch (henceforth K&M) [13, 18], they offer three key advantages over the classic extensive form game (EFG) representation. First, MAIDs can depict many games more compactly than EFGs, especially those with incomplete information. Second, MAIDs encode conditional independencies between variables. This means large MAIDs can often be decomposed into several smaller ones, with potentially exponential speedups for finding Nash equilibria [13]. Third, MAIDs often make it possible to explicitly represent aspects of game structure that are obscured in EFGs. While it is possible to convert any EFG to a MAID of at most the same size (Section 3.3.2), it is true that EFGs are sometimes better suited for modelling asymmetric decision problems. With that said, every model has its weaknesses, and how useful a particular representation is rests on its strengths. We further develop both the theory and practical tools for MAIDs in order to allow both researchers and practitioners to make the most of their strengths.

Previous work on MAIDs has focussed on Nash equilibria as the core solution concept [19]. Whilst this is arguably the most important solution concept in non-cooperative game theory, if there are many Nash equilibria we often wish to remove some of those that are less 'rational'. Many refinements to the Nash equilibrium have been proposed [16], with two of the most important being subgame perfect Nash equilibria [24] and trembling hand perfect equilibria [25]. The first rules out 'non-credible' threats and the second requires that each player is still playing a best-response when other players make small mistakes. On the practical side, while much software exists for normal or extensive form games, there is no such implementation for reasoning about games expressed as MAIDs, despite their computational advantages.

1.1 Contribution

In this paper, we make the following contributions. First, we extend the applicability of MAIDs by introducing the concept of a MAID subgame (Section 3.1) and build on this concept to introduce subgame perfect and trembling hand perfect equilibrium refinements (Section 3.2). Second, we prove several equivalence results between MAIDs and EFGs, demonstrating the preservation of the key game-theoretic concepts described above when representing EFGs as MAIDs and thus further justifying the use of this model. These proofs are constructive and are based on procedures for converting between EFGs and MAIDs, the full details of which are included in the supplementary material.[1] Third, we report on our open source codebase for computing our equilibrium refinements in MAIDs (Section 4).

1.2 Related Work

Our work builds primarily on the seminal work of K&M [13, 18]. More recently, casual influence diagrams (CIDs) have been defined [4], where the probabilistic arrows in influence diagrams are interpreted as describing a causal relationship, in accordance with Pearl's graphical causal models [21]. CIDs model single agents, helping to predict behaviour by identifying the incentives that arise due to the agent optimising its objective, and have been shown to have many applications [2, 5, 6, 8, 15]. Our equilibrium refinements for MAIDs and our implementation are partially targeted at extending this work on incentives to the multi-agent setting.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

[1] Available online at `https://arxiv.org/abs/2102.05008`.

Pfeffer and Gal investigated when an agent is motivated to care about its decision in the context of MAIDs, identifying four reasoning patterns (with associated graphical criteria) that justify a particular decision choice [22]. Later work showed practical applications of these reasoning patterns, which can lead to safer human-machine or machine-machine designs and again reduce the time complexity of computing Nash equilibria [1]. In this work we implement these reasoning patterns in our codebase. Building on this, further research could consider which reasoning patterns arise when agents are playing a certain equilibrium refinement.

Several other formalisms, often partly inspired by MAIDs, have been proposed for representing and reasoning about games as probabilistic graphical models. For example: networks of influence diagrams represent mental models of the different agents as nodes in a graph and use these to describe and reason about belief structures [7]; settable systems extend structural equation models to include the concept of optimisation and hence the idea of a 'best response', which is key to defining game-theoretic equilibria [27]; temporal action graph games are similar to MAIDs, but can be more compact for games that involve anonymity or context-specific utility independencies [10]. These works, however, focus on the introduction of novel representations, whereas we focus on deepening the theory and practice behind an existing representation. It is an interesting question for further research whether our insights also apply to these related models.

2 BACKGROUND

In this section, we define EFGs and MAIDs and show how their graphical representation of games differ with the help of the following example [26].

Example 1 (Job hiring). *A company employs an AI system to automate their hiring process. A naturally hard-working or naturally lazy worker wants a job at this company and believes that a university degree will increase their chance of being hired; however, they also know that they will suffer an opportunity cost from three years of studying. A hard-worker will cope better with a university workload than a lazy worker. The algorithm must decide, on behalf of the company, whether to hire the worker. The company wants to hire someone who is naturally hard-working, but the algorithm can't observe the worker's temperament directly, it can only infer it indirectly through whether or not the worker attended university.*

We use capital letters X for variables and let $dom(X)$ denote the domain of X. An assignment $x \in dom(X)$ to X is an instantiation of X denoted by $X = x$. $\boldsymbol{X} = \{X_1, \dots, X_n\}$ is a set of variables with domain $dom(\boldsymbol{X}) = \times_{i=1}^{n} dom(X_i)$ and $\boldsymbol{x} = \{x_1, \dots, x_n\}$ is the set containing an instantiation of all variables in $\boldsymbol{X}$. We let $\mathbf{Pa}_V$ denote the parents of a node V in a graphical representation and $\mathbf{pa}_V$ be the instantiation of $\mathbf{Pa}_V$. $\mathbf{Ch}_V$, $\mathbf{Anc}_V$, $\mathbf{Desc}_V$, and $\mathbf{Fa}_V \coloneqq \mathbf{Pa}_V \cup \{V\}$ are the children, ancestors, descendants, and family of V with, analogously to $\mathbf{pa}_V$, their instantiations written in lowercase. Unless otherwise indicated we index mathematical objects with superscripts $i \in \boldsymbol{N}$ to denote their affiliation with a player i (where $\boldsymbol{N}$ is a set of players) and with subscripts $j \in \mathbb{N}$ to enumerate them.

2.1 Extensive Form Games

Definition 1 ([14]). An **extensive-form game (EFG)** $\mathcal{G}$ is a tuple $(\boldsymbol{N}, T, \boldsymbol{P}, \boldsymbol{D}, \lambda, \boldsymbol{I}, U)$, where:

- $\boldsymbol{N} = \{1, \dots, n\}$ is a set of agents.
- $T = (\boldsymbol{V}, \boldsymbol{E})$ is a game tree with nodes $\boldsymbol{V}$ that are partitioned into the sets $\boldsymbol{V}^0, \boldsymbol{V}^1, \dots, \boldsymbol{V}^n, \boldsymbol{L}$ where $R \in \boldsymbol{V}$ is the root of T, $\boldsymbol{L}$ is the set of leaves of the tree, $\boldsymbol{V}^0$ is the set of chance nodes, and $\boldsymbol{V}^i$ is the set of nodes controlled by player $i \in \boldsymbol{N}$. These nodes are connected by edges $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$.
- $\boldsymbol{P} = \{P_1, \dots, P_{|\boldsymbol{V}^0|}\}$ is a set of probability distributions where each $P_j : \mathbf{Ch}_{V_j} \to [0, 1]$ determines the probability of a path through the game tree that has reached chance node V_j^0 proceeding to each child node in $\mathbf{Ch}_{V_j^0}$.
- $\boldsymbol{D}$ is a set of decisions, we write $\boldsymbol{D}_j^i \subseteq \boldsymbol{D}$ to describe the set of available decisions at node $V_j^i \in \boldsymbol{V}^i$.
- $\lambda : \boldsymbol{E} \to \boldsymbol{D}$ is a labelling function mapping an edge (V_j^i, V_l^k) to a decision $d \in \boldsymbol{D}_j^i$.
- $\boldsymbol{I} = \{\boldsymbol{I}^1, \dots, \boldsymbol{I}^n\}$ is a set such that for each player $\boldsymbol{I}^i \subset 2^{\boldsymbol{V}^i}$ defines a partition of the vertices controlled by player i into information sets.
- $U : \boldsymbol{L} \to \mathbb{R}^n$ is a utility function mapping each leaf node to a vector that determines the final payoff for each player.

An **information set** $I_j^i \in \boldsymbol{I}^i$ is defined such that for all $V_k^i, V_l^i \in I_j^i$ we have $\boldsymbol{D}_j^i \coloneqq \boldsymbol{D}_k^i = \boldsymbol{D}_l^i$. In other words, the same player i selects the decision and the same decisions are available at each of the nodes in an information set. When $|I_j^i| = 1$ for all i and j, $\mathcal{G}$ is a **perfect information** game. A (behavioural) **strategy** σ^i for a player i is a set of probability distributions $\sigma_j^i : \boldsymbol{D}_j^i \to [0, 1]$ over the actions available to the player at each of their information sets I_j^i.[2] A strategy is **pure** when $\sigma_j^i(d) \in \{0, 1\}$ for all information sets I_j^i and **fully mixed** when $\sigma_j^i(d) > 0$ for all $d \in \boldsymbol{D}_j^i$. A **strategy profile** $\sigma = (\sigma^1, \dots, \sigma^n)$ is a tuple of strategies one for each player $i \in \boldsymbol{N}$. $\sigma^{-i} = (\sigma^1, \dots, \sigma^{i-1}, \sigma^{i+1}, \dots, \sigma^n)$ denotes the partial strategy profile of all players other than i, and so $\sigma = (\sigma^i, \sigma^{-i})$. The combination of the distributions in $\boldsymbol{P}$ with a strategy profile σ thus defines a full probability distribution P^σ over paths in $\mathcal{G}$.

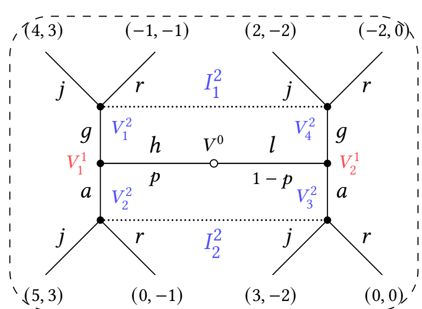


Figure 1: An EFG representation of Example 1.

[2] Formally, $\sigma_j^i(d) > 0$ only if $d \in D_k^i$ for any vertex $V_k \in I_j^i$, and $\Sigma_{d \in D_k^i} \sigma_j^i(d) = 1$.

Figure 1 shows Example 1's signalling game in extensive form. Nature, as a chance node V^0, flips a biased coin at the root of the tree to decide whether the person is hard-working (probability p) or lazy (probability $1-p$). The worker (player 1)'s decision whether to go (g) or avoid (a) university is represented at nodes $\{V^1_1, V^1_2\} = \boldsymbol{V}^1$ and there are four nodes $\{V^2_1, V^2_2, V^2_3, V^2_4\} = \boldsymbol{V}^2$ for the hiring algorithm (player 2), each with two decision options: reject (r) or job offer (j). These nodes are split into two information sets (dotted lines between nodes) because the algorithm does not know whether the person is naturally hard-working. The payoffs for the worker and the employer respectively are given at the leaves of the tree.

2.2 Multi-Agent Influence Diagrams

Following recent work [4, 9], we depart slightly from the convention of K&M to distinguish between an influence *diagram*, which gives the structure of a strategic interaction, and an influence *model*, which adds a particular parametrisation to the diagram.

Definition 2 ([13])**.** A **multi-agent influence diagram (MAID)** is a triple $(\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E})$, where:

- $\boldsymbol{N} = \{1, \dots, n\}$ is a set of agents.
- $(\boldsymbol{V}, \boldsymbol{E})$ is a directed acyclic graph (DAG) with a set of vertices $\boldsymbol{V}$ connected by directed edges $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$. These vertices are partitioned into $\boldsymbol{D}$, $\boldsymbol{U}$, and $\boldsymbol{X}$, which correspond to decision, utility, and chance nodes respectively. $\boldsymbol{D}$ and $\boldsymbol{U}$ are in turn partitioned into $\{\boldsymbol{D}^i\}_{i\in\boldsymbol{N}}$ and $\{\boldsymbol{U}^i\}_{i\in\boldsymbol{N}}$ corresponding to their association with a particular agent $i \in \boldsymbol{N}$.

Definition 3. A **multi-agent influence model (MAIM)** is a tuple $(\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E}, \theta)$ where $(\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E})$ is a MAID and:

- $\theta \in \Theta$ is a particular parametrisation over the nodes in the graph specifying a finite domain $dom(V)$ for each node $V \in \boldsymbol{V}$, real-valued domains $dom(U) \subset \mathbb{R}$ for each $U \in \boldsymbol{U}$, and a set of conditional probability distributions (CPDs) $\Pr(\boldsymbol{V} \mid \mathbf{Pa}_V)$ for every chance and utility node. Taken together, the CPDs form a partial distribution $\Pr(\boldsymbol{X}, \boldsymbol{U} : \boldsymbol{D}) = \prod_{V\in\boldsymbol{V}\setminus\boldsymbol{D}} \Pr(V \mid \mathbf{Pa}_V)$ over the variables in the MAID.
- The value $u \in dom(U)$ of a utility node is a deterministic function of the values of its parents $\mathbf{pa}_U \in dom(\mathbf{Pa}_U)$.

Figure 2 a) shows the MAID for Example 1 corresponding to the EFG in Figure 1. Whether the worker is hard-working or lazy is decided by nature's chance node X (white circle). The worker's decision D^1 and utility U^1 nodes are depicted as a red rectangle and diamond respectively. The algorithm's decision D^2 and utility U^2 nodes are in blue. To instantiate a MAIM, CPD tables for U^1 and U^2 would be consistent with the payoffs and value of p in Figure 1.

There are two types of directed edge in a MAID. Full edges leading into $\boldsymbol{X} \cup \boldsymbol{U}$ represent probabilistic dependence, as in a Bayesian network. Dotted edges leading into $\boldsymbol{D}$ represent information that is available to the agent at the time a decision is made (e.g. the edge $X \to D^1$). In this way, the values of the parents $\mathbf{pa}_D$ of a decision node D represent the decision context for D. The CPDs of decision nodes are not defined when a MAIM is constructed because they are instead chosen by the agents playing the game. In general, a player's decision CPD need not be optimal.

This example demonstrates two clear advantages of MAIDs compared with EFGs. First, in many real world cases, MAIDs make it possible to explicitly represent aspects of game structure that are obscured in the extensive form. For example, in the EFG, information sets were drawn to reflect the fact that the algorithm does not know whether the worker is naturally hard-working or lazy when it selects its action. However, in the corresponding MAID, this incomplete information is represented simply by the fact that there is no edge $X \to D^2$. Moreover, the company's utility U^2 isn't a function of whether the applicant went to university or not – it only cares whether the applicant is hard-working and whether or not they hired them. We can infer this from the EFG payoffs in Figure 1, but in the MAID this is shown instantly by the fact that there is no edge $D^1 \to U^2$.

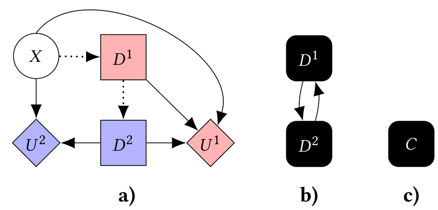


Figure 2: A MAID $\mathcal{M}$ a) representation of Example 1, along with its cyclic relevance graph $Rel(\mathcal{M})$ b) (Section 2.5) and condensed relevance graph $ConRel(\mathcal{M})$ c) (Section 3.1).

Second, MAIDs can provide a more compact graphical representation of games [13, 18]. In fact, the MAID representation of a game need never be bigger than the corresponding EFG and can be smaller in many cases. For example, there are four nodes $V^2_1, V^2_2, V^2_3, V^2_4$ in Figure 1 which correspond to the company's decision. In the MAID, these are combined into one node, D^2.

A further strength of MAIMs derive from them being probabilistic graphical models, and so probabilistic dependencies between chance and strategic variables can be exploited. We recall the notion of *d-separation*, a graphical criterion for determining independence properties of the probability distribution associated with the graph. This is necessary for the concept of r-reachability (Section 2.5) and consequently that of a MAID subgame (Section 3.1).

Definition 4 ([21])**.** A path p in a MAID $(\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E})$ is said to be **d-separated** by a set of nodes $\boldsymbol{W} \subset \boldsymbol{V}$ if and only if either:

- p contains a chain $X \to Y \to Z$ or a fork $X \leftarrow Y \to Z$ and $Y \in \boldsymbol{W}$.
- p contains a collider $X \to Y \leftarrow Z$ and $(\{Y\} \cup \mathbf{Desc}_Y) \not\subseteq \boldsymbol{W}$.

A set $\boldsymbol{W}$ **d-separates** $\boldsymbol{X}$ from $\boldsymbol{Y}$, denoted $\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{W}$, if and only if $\boldsymbol{W}$ d-separates every path from a node in $\boldsymbol{X}$ to a node in $\boldsymbol{Y}$. Sets of variables that are not d-separated are said to be **d-connected**, denoted $\boldsymbol{X} \not\perp \boldsymbol{Y} \mid \boldsymbol{W}$. If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated conditioning on $\boldsymbol{W}$, then $\boldsymbol{X}$ and $\boldsymbol{Y}$ are probabilistically independent in the sense that $P(\boldsymbol{X} \mid \boldsymbol{Y}, \boldsymbol{W}) = P(\boldsymbol{X} \mid \boldsymbol{W})$.

For example, there are several paths from U^2 to U^1 in Figure 2 a): direct forks through X or D^2, a fork through X and then a forward chain through D^1, or a backward chain through D^2 and then a fork through D^1. If $\boldsymbol{W} = \varnothing$ then U^2 is d-connected to U^1 ($U^2 \not\perp U^1 \mid \varnothing$), but if $\boldsymbol{W} = \{X, D^2\}$ then all of the paths have been d-separated by conditioning on $\boldsymbol{W}$ and so $U^2 \perp U^1 \mid \boldsymbol{W}$.

2.3 Policies

An agent makes a decision depending on the information it observes prior to making that decision. Therefore, in a MAIM, a **decision rule** π_D for a decision node D is a CPD $\pi_D(D \mid \mathbf{Pa}_D)$. A **partial policy profile** $\pi_{\boldsymbol{A}}$ is an assignment of decision rules $\pi_D(D \mid \mathbf{Pa}_D)$ to some subset $\boldsymbol{A} \subset \boldsymbol{D}$ and $\pi_{-\boldsymbol{A}}$ is the set of decision rules for all $D \in \boldsymbol{D} \setminus \boldsymbol{A}$. For example, π_D refers to a decision rule at decision node D and so $\pi_{-D} = \prod_{D' \in \boldsymbol{D}\setminus\{D\}} \pi_{D'}(D' \mid \mathbf{Pa}_{D'})$ denotes the partial policy profile over all of the MAIM's other decision nodes $\boldsymbol{D} \setminus \{D\}$. We refer to $\pi_{\boldsymbol{D}^i}$, which describes all the decision choices made by agent $i \in \boldsymbol{N}$, as that agent's **policy**, π^i, and we write $\pi^{-i} = (\pi^1, \ldots, \pi^{i-1}, \pi^{i+1}, \ldots, \pi^n)$ to denote the set of policies made by all agents other than agent i. A **policy profile** π assigns a policy to every agent $\pi = (\pi^1, \ldots, \pi^n)$; it describes all the decisions made by every agent in the MAIM. We denote spaces of policy profiles by Π (e.g. $\Pi_{\boldsymbol{A}}$, Π^i, and Π).

If for every $\mathbf{pa}_d \in dom(\mathbf{Pa}_D)$ and $d \in dom(D)$ we have $\pi_D(d \mid \mathbf{pa}_D) \in \{0, 1\}$, the decision rule is said to be **pure** or **deterministic**. Otherwise, the decision rule is said to be **mixed** and it is **fully mixed** if, for every $\mathbf{pa}_D$ and every d, we have $\pi_D(d \mid \mathbf{pa}_D) > 0$. Pure, mixed, and fully mixed policies or policy profiles are defined analogously.

When a partial policy profile $\pi_{\boldsymbol{A}}$ is applied to a MAIM $\mathcal{M}$, a new MAIM $\mathcal{M}(\pi_{\boldsymbol{A}})$ is obtained in which each decision node $D \in \boldsymbol{A}$ becomes a chance node with a CPD equal to π_D. In the case of a policy profile, all decision nodes are turned into chance nodes, and so the induced MAIM $\mathcal{M}(\pi)$ is now a Bayesian network (utility nodes are interpreted as chance nodes when a MAIM is viewed as a Bayesian network). This defines the joint probability distribution $\Pr^{\pi}$ over all variables in $\mathcal{M}$ and may be used for probabilistic inference.

2.4 Utilities

In an EFG $\mathcal{G}$ the expected utility for each player depends on the set of probability distributions $\boldsymbol{P}$ and strategy profile σ which give a full probability distribution P^{σ} over the paths in $\mathcal{G}$. For each path ρ beginning from the root R of $\mathcal{G}$'s tree and terminating in a unique leaf node $\rho[\boldsymbol{L}]$, player i receives utility $U(\rho[\boldsymbol{L}])[i]$ – the i^{th} entry in the corresponding payoff vector. By playing strategy profile σ, player i's expected utility $\mathcal{U}^i_{\mathcal{G}}(\sigma) := \sum_{\rho} P^{\sigma}(\rho)U(\rho[\boldsymbol{L}])[i]$.

Similarly, the joint distribution $\Pr^{\pi}$ induced by the policy profile π in a MAIM $\mathcal{M}$ allows us to define the expected utility for each player under this policy profile. Agent i's expected utility from policy profile π is the sum of the expected value of utility nodes $\boldsymbol{U}^i$ given by $\mathcal{U}^i_{\mathcal{M}}(\pi) := \sum_{U_j \in \boldsymbol{U}^i} \sum_{u_j \in dom(U_j)} u_j \Pr^{\pi}(U_j = u_j)$. We assume that each agent's goal is to select a policy π^i that maximises its expected utility. Therefore, we can now define what it means for an agent to optimise $\pi_{\boldsymbol{A}}$ for a set of decisions $\boldsymbol{A} \subseteq \boldsymbol{D}^i$, given a partial policy profile $\pi_{-\boldsymbol{A}}$ over all of the other decision nodes in $\mathcal{M}$. We write $\mathcal{U}^i_{\mathcal{M}}(\pi_{\boldsymbol{A}}, \pi_{-\boldsymbol{A}})$ to denote the expected utility for player i under the policy profile $\pi = (\pi_{\boldsymbol{A}}, \pi_{-\boldsymbol{A}})$.

Definition 5. Let $\boldsymbol{A} \subseteq \boldsymbol{D}^i$. Player i's partial policy $\pi_{\boldsymbol{A}}$ is **optimal** for a policy profile $\pi = (\pi_{\boldsymbol{A}}, \pi_{-\boldsymbol{A}})$ if $\mathcal{U}^i_{\mathcal{M}}(\pi_{\boldsymbol{A}}, \pi_{-\boldsymbol{A}}) \geq \mathcal{U}^i_{\mathcal{M}}(\hat{\pi}_{\boldsymbol{A}}, \pi_{-\boldsymbol{A}})$ for all $\hat{\pi}_{\boldsymbol{A}} \in \Pi_{\boldsymbol{A}}$. Player i's policy π^i is a **best response** to the partial policy profile π^{-i} assigning policies to the other agents if $\mathcal{U}^i_{\mathcal{M}}(\pi^i, \pi^{-i}) \geq \mathcal{U}^i_{\mathcal{M}}(\hat{\pi}^i, \pi^{-i})$ for all $\hat{\pi}^i \in \Pi^i$.

2.5 Strategic and Probabilistic Relevance

To optimise a particular decision rule, we often want to know which other decision rules need to already be known. This is captured by K&M's concept of *strategic relevance*.

Definition 6 ([13])**.** Let $D_k, D_l \in \boldsymbol{D}$ be decision nodes in a MAIM $\mathcal{M}$. D_l is **strategically relevant** to D_k (D_k strategically relies on D_l) if there exist two policy profiles π and π' and a decision rule π_{D_k}, such that:

- π_{D_k} is optimal for π.
- π differs from π' only at D_l.
- π_{D_k} is not optimal for π', and neither is any decision rule $\hat{\pi}_{D_k}$ that agrees with π_{D_k} for all instantiations $\mathbf{pa}_{D_k}$ of D_k's parents where the joint probability $\Pr^{\pi'}(\mathbf{pa}_{D_k}) > 0$.

The first two conditions say that if decision rule π_{D_k} is optimal for a policy profile π, and D_k does not strategically rely on D_l, then π_{D_k} must also be optimal for any policy profile π' that differs from π only at D_l. The third condition deals with sub-optimal decisions in response to zero-probability decision contexts.

A related question is *probabilistic relevance*, which considers whether the probability distribution of a chance or utility node X can influence the optimal policy.

Definition 7. Let D be a decision node in a MAID $\mathcal{M}$. A chance or utility node $Z \in \boldsymbol{X} \cup \boldsymbol{U}$ is **probabilistically relevant** to D if the set of optimal decision rules for D varies with the CPDs assigned to Z under some joint policy profile π.

We generalise K&M's graphical criterion, **s-reachability**, as **r-reachability** to determine both strategic relevance and probabilistic relevance. Essentially, the criterion assesses whether knowing the CPD or decision rule of a node V can have positive value of information [4]. The criterion is sound (if V is relevant to D, then V is r-reachable from D) and complete (if V is r-reachable from D then there is some parametrisation θ of the MAID and some policy profile π such that V is relevant to D). One can then further use r-reachability to define a *relevance graph* over $\boldsymbol{D}$.

Definition 8. A node V is **r-reachable** from a decision $D \in \boldsymbol{D}^i$ in a MAID, $\mathcal{M} = (\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E})$, if a newly added parent $\hat{V}$ of V satisfies $\hat{V} \not\perp \boldsymbol{U}^i \cap \mathbf{Desc}_D \mid \mathbf{Fa}_D$.

Definition 9. The directed **relevance graph** for $\mathcal{M}$, denoted by $Rel(\mathcal{M}) = (\boldsymbol{D}, \boldsymbol{E}_{Rel})$, is a graph where $\boldsymbol{D}$ is the set of $\mathcal{M}$'s decision nodes connected by directed edges $\boldsymbol{E}_{Rel} \subseteq \boldsymbol{D} \times \boldsymbol{D}$. There is a directed edge from $D_j \rightarrow D_k$ if and only if D_k is r-reachable from D_j.[3]

Relevance graphs show which other decisions each decision depends on. The relevance graph for Example 1's MAIM in Figure 2 b) is cyclic because each decision node strategically relies on the other. The worker would be better off knowing the company's hiring policy before deciding whether or not to go to university, but the algorithm would also be better off knowing the worker's policy because it doesn't know the worker's temperament (lazy or hard-working). Our second example provides a case of acyclic strategic relevance.

[3]The edge directions used here are the same as originally defined by K&M [12] but reversed compared with those in their later work [13] as this eases our later exposition of MAID subgames (Section 3.1).

Example 2 (Taxi competition)**.** *Two autonomous taxis, operated by different companies, are driving along a road with two hotels located next to one another – one expensive and one cheap. Each taxi must decide (one first, then the other) which hotel to stop in front of, knowing that it will likely receive a higher tip from guests of the expensive hotel. However, if both taxis choose the same location, this will reduce each taxi's chance of being chosen by that hotel's guests.*

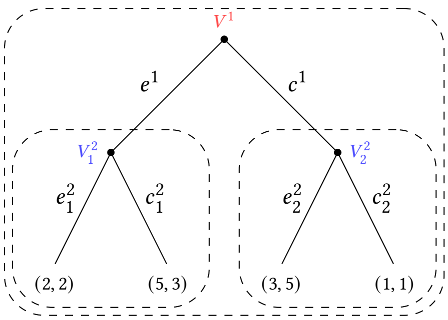
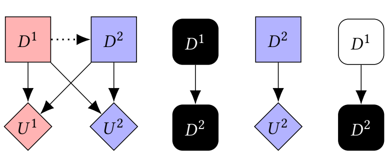


Figure 3: An EFG representation of Example 2, with EFG subgames enclosed in dashed boxes.

Because the second taxi can observe which hotel the first taxi chooses to park in front of, it doesn't need to know the first taxi's policy in order to optimise its own; the second taxi's decision (D^2) does *not* strategically rely on the first taxi's decision (D^1). However, the first taxi would be better off knowing the second taxi's policy before deciding its own. For example, with the parametrisation in Figure 4 e), if the first taxi knows that the second taxi's policy is to always park in front of the expensive hotel, the first taxi ought to always park in front of the cheaper hotel. D^1 *does* strategically rely on D^2. In Section 4, we shall see that it is easier to compute equilibria in MAIMs with acyclic relevance graphs.



a) b) c) d)

	U_1	D^1 e	D^1 c
D^2	e	2	3
	c	5	1

	U^2	D^1 e	D^1 c
D^2	e	2	5
	c	3	1

e)

Figure 4: A MAID a) and corresponding relevance graph b) for Example 2, alongside the only proper MAID subgame c) highlighted in the (condensed) relevance graph d). The utility nodes' parametrisation is in e).

3 GAME THEORY FOR MAIDS

In this section, we present novel material. We begin by defining MAID and MAIM subgames. These set up our discussion of several equilibrium refinements in MAIMs. Finally, we present and prove a number of equivalence results between EFGs and MAIMs.

3.1 Subgames

In an EFG, a subtree of the original game tree is an **EFG subgame** if it is closed under information sets and descendants. Figure 3 shows all the EFG subgames (dashed boxes) for the game described by Example 2. Any game tree is an EFG subgame of itself, and so an EFG subgame on a strictly smaller set of nodes is called a **proper** EFG subgame. We propose an analoguous definition for MAIDs. Just like for EFGs, MAIM subgames are parts of the game that can be solved independently.

Definition 10. A **subgame base** for a MAID $(\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E})$ is a subset $\boldsymbol{V'} \subseteq \boldsymbol{V}$ such that:

- For any $X, Y \in \boldsymbol{V'}$ and any directed path $X \to \cdots \to Y$ in $\mathcal{M}$, all nodes on the path are also in $\boldsymbol{V'}$.
- $\boldsymbol{V'}$ is closed under r-reachability, i.e. if a node Z is r-reachable from a decision $D \in \boldsymbol{V'}$, then Z is also in $\boldsymbol{V'}$.

Definition 11. Let $\mathcal{M} = (\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E})$ be a MAID, and let $\boldsymbol{V'} \subseteq \boldsymbol{V}$ be a subgame base. The **MAID subgame** corresponding to $\boldsymbol{V'}$, is a new MAID $\mathcal{M}' = (\boldsymbol{N'}, \boldsymbol{V'}, \boldsymbol{E'})$ where:

- $\boldsymbol{N'} = \{i \in \boldsymbol{N} \mid \boldsymbol{D^i} \cap \boldsymbol{V'} \neq \varnothing\}$, the players restricted to $\boldsymbol{V'}$.
- $\boldsymbol{V'}$ is partitioned into $\boldsymbol{D'} = \boldsymbol{D} \cap \boldsymbol{V'}$, $\boldsymbol{U'} = \boldsymbol{U} \cap \mathbf{Desc}_{\boldsymbol{D'}}$, and $\boldsymbol{X'} = \boldsymbol{V'} \setminus (\boldsymbol{D'} \cap \boldsymbol{U'})$.
- $\boldsymbol{E'}$ is the subset of edges in $\boldsymbol{E}$ that connect two nodes in $\boldsymbol{V'}$.

Analogously, the **MAIM subgame** of a MAIM $(\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E}, \theta)$ corresponding to a subset $\boldsymbol{V'} \subseteq \boldsymbol{V}$ and an instantiation $\boldsymbol{y}$ of the nodes $\boldsymbol{Y} = \boldsymbol{V} \setminus \boldsymbol{V'}$, is the modified MAIM $(\boldsymbol{N'}, \boldsymbol{V'}, \boldsymbol{E'}, \theta')$ where:

- $(\boldsymbol{N'}, \boldsymbol{V'}, \boldsymbol{E'})$ is the MAID subgame corresponding to $\boldsymbol{V'}$.
- θ' is like θ, restricted to nodes in $\boldsymbol{V'}$. If a node $X \in \boldsymbol{V'}$ has some parents outside of $\boldsymbol{V'}$ (i.e. in $\boldsymbol{Y}$) then $\Pr'(X \mid \mathbf{pa}'_X) = \Pr(X \mid \mathbf{pa}'_X, \boldsymbol{y}')$, where $\mathbf{Pa}'_X = \mathbf{Pa}_X \cap \boldsymbol{V'}$, $\boldsymbol{Y'} = \mathbf{Pa}_X \cap \boldsymbol{Y}$, Pr is the CPD of X in θ, and $\Pr'$ becomes the CPD of X in θ'.

In fact, only the setting $\boldsymbol{y}$ of the nodes that have a child in $\boldsymbol{V'}$ will matter. A MAIM subgame is **feasible** if there exists a policy profile π where $\Pr^{\pi}(\boldsymbol{y}) > 0$.

In a sequential game with perfect information, the MAIM subgames will be in one-to-one correspondence with the subgames in any corresponding EFG. For example, Figures 3 and 4 show the EFG and MAID subgames of Example 2. As with EFG subgames, a MAID is trivially a MAID subgame of itself, as in Figure 4 a). Figure 4 c) shows the only *proper* MAID subgame of $\mathcal{M}$. Two MAIM subgames are associated with this MAIM subgame: one for each value of D^1. The additional independencies represented by a MAID sometimes yields more independently solvable components than identifiable in an EFG representation; i.e., there can be more subgames in a MAIM than in a corresponding EFG.

A better sense of MAID subgames can be gained from looking at the strongly connected components (SCC) of the relevance graph, where recall that an SCC is a subgraph containing a directed path between every pair of nodes. A maximal SCC is an SCC that is not a strict subset of any other SCC. We can use this fact to define a condensed relevance graph, called the component graph by K&M, which aggregates each SCC into a single node.

Definition 12. For a given MAID $\mathcal{M} = (\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E})$, let $\boldsymbol{C}$ be the set of maximal SCCs of its relevance graph $Rel(\mathcal{M})$. The **condensed relevance graph** of $\mathcal{M}$ is the directed graph $ConRel(\mathcal{M}) = (\boldsymbol{C}, \boldsymbol{E}_{ConRel})$.

There is an edge $\boldsymbol{C}_m \to \boldsymbol{C}_l$ between $\boldsymbol{C}_m, \boldsymbol{C}_l \in \boldsymbol{C}$ if and only if there exists $C_m \in \boldsymbol{C}_m$ and $C_l \in \boldsymbol{C}_l$ with and edge $C_m \to C_l$ in $Rel(\mathcal{M})$.

Subgraphs of $ConRel(\mathcal{M})$ closed under descendants induce MAID subgames. Figures 4 b) and d) highlight the nodes of the respective MAID subgames (since the relevance graph is acyclic here, condensing it has no effect). As the condenstion of a directed graph is always acyclic [3, page 617], games can always be solved via backwards induction over the condensed relevance graph [13].

3.2 Equilibrium Refinements

MAIDs represent dynamic games of incomplete information – those in which at least one player i does not have perfect information about the chance variables $\boldsymbol{X}$ (commonly interpreted as not knowing the *type* T^i of the other agents, where T^i defines the payoffs for agent i) – and thus admit discussion of the *beliefs* that agents possess. In this work, we implicitly view such beliefs as defined by the induced distribution $\Pr^\pi$ and so eschew further discussion of them here; in a sense, chance variables can be viewed as decisions by nature (using fixed stochastic policies).

In non-cooperative games, the most fundamental solution is a Nash equilibrium [19], a policy profile such that no agent has an incentive to unilaterally deviate. In other words, every player is simultaneously playing a best-response against all other players.

Definition 13 ([13])**.** A full policy profile π is a **Nash equilibrium (NE)** in a MAIM $\mathcal{M}$ if, for every player $i \in \boldsymbol{N}$, $\mathcal{U}^i_{\mathcal{M}}(\pi^i, \pi^{-i}) \geq \mathcal{U}^i_{\mathcal{M}}(\hat{\pi}^i, \pi^{-i})$ for all $\hat{\pi}^i \in \Pi^i$.

The concept of a subgame perfect equilibrium (SPE) was introduced by Reinhard Selten to address the issue that EFGs admit NEs with *non-credible threats* – equilibria in which a player threatens to take some action that, if the player is rational, they would never actually carry out [24, 25]. In an EFG, a strategy profile is an SPE if it induces an NE in every EFG subgame; this eliminates all NEs containing non-credible threats. Our definition of MAID subgames above allows us to introduce an analogous equilibrium concept.

Definition 14. A full policy profile π is a **subgame perfect equilibrium (SPE)** in a MAIM $\mathcal{M}$ if π is an NE in every MAIM subgame of $\mathcal{M}$.

a)

D^1	
e	1
c	0

		D^1 e	D^1 c
D^2	e	0	1
	c	1	0

b)

D^1	
e	0
c	1

		D^1 e	D^1 c
D^2	e	1	1
	c	0	0

c)

D^1	
e	1
c	0

		D^1 e	D^1 c
D^2	e	0	0
	c	1	1

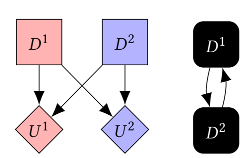
Figure 5: The policies for Example 2's three pure NEs. Only a) is an SPE.

Figure 5 shows the three pure NEs of Example 2. The policy profiles in b) and c) are NEs but not SPEs. To see why, consider the proper MAIM subgame when $D^1 = e$ and policy profile b). Here player 2 obtains utility 3 if they choose c and utility 2 if they choose e. Therefore, player 2 is making a *non-credible* threat whenever $\pi^2(D^2 = e \mid D^1 = e) > 0$. For similar reasons, policy profile c) is also not SPE. Therefore a) is the only SPE of this MAIM.

Within maximal SCCs of $Rel(\mathcal{M})$, in which there are no proper subgames, the agents choose decision rules interdependently. This can lead to arbitrarily bad decision rules in decision contexts that occur with probability zero. Trembling hand equilibria offer a useful NE refinement in these situations [25]. Intuitively, they require that each player's policy is still a best response when the other players make mistakes, or 'tremble', with small probability. Let δ_k be a perturbation vector containing, for every $D \in \boldsymbol{D}$, $d \in dom(D)$, and decision context $\mathbf{pa}_D$, a value $\epsilon^d_{\mathbf{pa}_D} \in (0, 1)$ such that $\sum_{d \in dom(D)} \epsilon^d_{\mathbf{pa}_D} \leq 1$. Then, given a MAIM $\mathcal{M}$, the perturbed MAIM $\mathcal{M}(\delta_k)$ is defined such that for every $d \in dom(D)$ for $D \in \boldsymbol{D}^i$, agent i must play d with probability at least $\epsilon^d_{\mathbf{pa}_D}$ given $\mathbf{pa}_D$.

Definition 15. A full policy profile π is a **trembling hand perfect equilibrium (THPE)** in a MAIM $\mathcal{M}$ if there is a sequence of perturbation vectors $\{\delta_k\}_{k \in \mathbb{N}}$ such that $\lim_{k\to\infty} |\delta_k|_\infty = 0$ and for each perturbed MAIM $\mathcal{M}(\delta_k)$ there is an NE π_k such that $\lim_{k\to\infty} \pi_k = \pi$.

Example 3 (Cyber-war)**.** *The security agencies for two governments both use an algorithm to manage their cyber-defence. Their algorithm decides whether to cyber-attack the other nation's security agency. If both agencies attack one another, both suffer some damage (mainly the opportunity cost of needing to continuously work on upgrading their defence systems). The attacker never gains much, but if only one agency attacks the other, the defender suffers a lot more damage.*



a) b)

c)

U_1		D^1 a	D^1 n
D^2	a	-2	-4
	n	0	0

U2		D^1 a	D^1 n
D^2	a	-2	0
	n	-4	0

d)

D^1	
a	p
n	1-p

D^2	
a	q
n	1-q

Figure 6: A MAID a), corresponding relevance graph b), utility CPD tables c), and policy profile d) for Example 3.

Figure 6 shows a MAIM with its parametrisation and relevance graph for Example 3. Using each player's parametrised policy in Figure 6 d) this MAIM has two NEs: at $p = q = 1$ and $p = q = 0$, either both governments attack (a) or not (n). Figures 7 a) and c) show player 1's best response policies for each of these NEs perturbed by ϵ to result in the perturbed MAIM $\mathcal{M}(\epsilon)$. Figures 7 b) and d) show player 2's expected utility if they attack (or not) in response to player 1 using the policy in 7 a) or c) respectively. For small $\epsilon > 0$, player 2's best response to both the policies in Figures 7 a) and c) is to choose $D^2 = a$ and so the NE $p = q = 0$ is not robust against trembles. The NE $p = q = 1$ is this MAIM's only THPE.[4]

[4]In the case of a two-player game, a THPE removes all weakly dominated policies. Here, for both D^1 and D^2, the pure policy of choosing n is weakly dominated by the pure policy of choosing a.

D^1	
a	$1-\epsilon$
n	ϵ

a)

D^2	$\mathcal{U}^2_{\mathcal{M}}(\pi)$
a	$-2+2\epsilon$
n	$-4+4\epsilon$

b)

D^1	
a	ϵ
n	$1-\epsilon$

c)

D^2	$\mathcal{U}^2_{\mathcal{M}}(\pi)$
a	-2ϵ
n	-4ϵ

d)

Figure 7: Polices a) and c) for player 1 for each NE in the original MAIM, shown in Figure 6, perturbed by ϵ, and the expected utilities b) and d) for player 2 when choosing each $d \in \boldsymbol{dom}(D^2)$ in response to policies a) and c) respectively.

3.3 Transformations and Equivalences

Both EFGs and MAIMs represent games graphically. In this section, we provide equivalence results between these models to demonstrate that alongside the increased compactness and structural clarity of MAIMs, the fundamental game-theoretic notions of subgames and equilibria are *preserved* when converting an EFG to a MAIM.

3.3.1 ***MAIM to EFG.*** There are many ways to convert a MAIM into an EFG, but these differ in their computational costs [13, 20]. We give a full and formal transformation procedure, **maim2efg**, in the supplementary material based on that of K&M. The idea is to use a topological ordering $\prec$ over the nodes of the MAID to construct the EFG game tree by splitting on each of the nodes in $\prec$. Because there can be more than one such ordering, the output of **maim2efg** is a *set* of EFGs. Our codebase implements a more efficient transformation, keeping only utility nodes, decision nodes, and informational parents ($\bigcup_{D \in \boldsymbol{D}} \mathbf{Fa}_D$). This information is enough for computing equilibria, and can offer significant efficiency gains since the cost of solving an EFG depends on its size, which is exponential in the length of $\prec$. The resulting EFG can be fed into Gambit, a popular tool for solving EFGs [17], though it may not contain enough information to fully recover the original MAIM.

3.3.2 ***EFG to MAIM.*** By encoding the CPDs for each variable in the MAIM using trees as opposed to tables, MAIMs can represent any decision-making problem using at most the same space, but often exponentially less space than an EFG [13]. In general, there are many MAIMs that can represent a given EFG. For instance, upon converting the EFG representation (Figure 3) of Example 2 to a MAIM (Figure 4), we could naïvely retain the EFG's root and two child nodes as three decision nodes (D^1, D^2_a, and D^2_b) in the MAIM. Alternatively, we could recognise that D^2_a and D^2_b both correspond to the same real world variable, the decision made following D^1, and thus combine them (as shown in Figure 4). In the supplementary material, we formalise this notion and provide a procedure **efg2maim** which maps an EFG to a *unique*, canonical MAIM (including those with absent-mindedness [23]).

3.3.3 ***Equivalences.*** We now provide a series of equivalence results between EFGs and MAIMs to fortify the game-theoretic foundations behind our analysis of MAIDs. Results are justified using intuitive sketches, with full proofs in the supplementary material.

Definition 16. A decision context $\mathbf{pa}_D$ for a decision node D in $\mathcal{M}$ if **feasible** if there exists a policy profile π where $\Pr^{\pi}(\mathbf{pa}_D) > 0$. A decision context $\mathbf{pa}_D$ is **null** if every player always receives utility 0, i.e. $\mathcal{U}^i_{\mathcal{M}}(\pi'' \mid \mathbf{pa}_D) = \sum_{U_j \in \boldsymbol{U}^i} \sum_{u_j \in dom(U_j)} u_j \Pr^{\pi}(U_j = u_j \mid \mathbf{pa}_D) = 0$ for all i and any policy profile π'', or if it is infeasible.

Definition 17. We say that a MAIM $\mathcal{M}$ is **equivalent** to an EFG $\mathcal{G}$ (and vice versa) if there is a bijection $f : \Sigma \to \Pi/\sim$ between the strategies in $\mathcal{G}$ and a partition of the policies in $\mathcal{M}$ (the quotient set of Π by an equivalence relation $\sim$) such that:

- $\pi \sim \pi'$ only if π and π' differ only on null decision contexts.
- For every $\pi \in f(\sigma)$ and every player i, $\mathcal{U}^i_{\mathcal{G}}(\sigma) = \mathcal{U}^i_{\mathcal{M}}(\pi)$.

We refer to f as a **natural mapping** between $\mathcal{G}$ and $\mathcal{M}$.

The reason we use an equivalence relation on the space of policies is that **efg2maim** can introduce additional null decision contexts: those that do not correspond to any path through the EFG. Although this equivalence is not exact, it is sufficient for preserving the essential game-theoretic features of each representation, as we show below. We begin with a supporting lemma that justifies the correctness of our procedures **maim2efg** and **efg2maim**, and forms the basis of our other results.

Lemma 1. *If $\mathcal{G} \in$ **maim2efg**$(\mathcal{M})$ or $\mathcal{M} =$ **efg2maim**$(\mathcal{G})$ then $\mathcal{G}$ and $\mathcal{M}$ are equivalent.*

This lemma follows directly by construction from the two procedures, **maim2efg** and **efg2maim** respectively. The intuition is that the information sets in an EFG correspond to the non-null decision contexts in a MAIM, and thus an EFG's behavioural strategy profile σ corresponds to a policy profile π in the MAIM, and vice versa. As an immediate consequence, we see that NEs are preserved by our transformations between EFGs and MAIMs.

Corollary 1. *If $\mathcal{G} \in$ **maim2efg**$(\mathcal{M})$ or $\mathcal{M} =$ **efg2maim**$(\mathcal{G})$ then there is a natural mapping f between $\mathcal{G}$ and $\mathcal{M}$ such that σ is an NE in $\mathcal{G}$ if and only if any $\pi \in f(\sigma)$ is an NE in $\mathcal{M}$.*

For an EFG subgame $\mathcal{G}'$, the variables outside $\mathcal{G}'$ are neither strategically nor probabilistically relevant to those in the corresponding MAIM subgame $\mathcal{M}'$. This means that EFG subgames have equivalent counterparts in the equivalent MAIM, as established by the following proposition.

Proposition 1. *If $\mathcal{G} \in$ **maim2efg**$(\mathcal{M})$ or $\mathcal{M} =$ **efg2maim**$(\mathcal{G})$ then there is a natural mapping f between $\mathcal{G}$ and $\mathcal{M}$ such that, for every EFG subgame $\mathcal{G}'$ in $\mathcal{G}$ there is a MAIM subgame $\mathcal{M}'$ in $\mathcal{M}$ that is equivalent to $\mathcal{G}'$ under the natural mapping f restricted to the strategies of $\mathcal{G}'$.*

This restriction of f to the strategies in $\mathcal{G}'$ can be made precise by considering only those non-null decision contexts that correspond to the information sets contained in $\mathcal{G}'$, as in the case for Lemma 1. Given Proposition 1 and Corollary 1, it can easily be seen that not only are NEs preserved when representing EFGs as MAIMs, but so too are SPEs. We remark, however, that as there may be more subgames in MAIM than in an equivalent EFG, that the criterion of subgame perfectness may be slightly stronger in the MAIM, and so not all SPEs in an EFG may be SPEs in the equivalent MAIM. This additional strength can be useful in ruling out what we intuitively view as 'irrational' behaviour, even when it does not fall under a particular subgame in the EFG.

Corollary 2. *If $\mathcal{G} \in$ **maim2efg**$(\mathcal{M})$ or $\mathcal{M} =$ **efg2maim**$(\mathcal{G})$ then there is a natural mapping f between $\mathcal{G}$ and $\mathcal{M}$ such that if any $\pi \in f(\sigma)$ is an SPE in $\mathcal{M}$, then σ is an SPE in $\mathcal{G}$.*

Finally, we derive an equivalence between the THPEs in EFGs and those in MAIMs. In order to do so, it suffices to prove an equivalence between perturbed versions of the corresponding games $\mathcal{G}(\delta_k)$ and $\mathcal{M}(\delta_k)$, which can easily be done via construction using **efg2maim**, and then by applying Lemma 1 and Corollary 1.

Proposition 2. *If $\mathcal{G} \in$ **maim2efg**$(\mathcal{M})$ or $\mathcal{M} =$ **efg2maim**$(\mathcal{G})$ then there is a natural mapping f between $\mathcal{G}$ and $\mathcal{M}$ such that σ is a THPE in $\mathcal{G}$ if and only if any $\pi \in f(\sigma)$ is a THPE in $\mathcal{M}$.*

This series of equivalence results serves to justify MAIDs as an appropriate choice of game representation. Not only do they provide computational advantages over EFGs, we have shown that they preserve the most fundamental game-theoretic concepts commonly employed in EFGs.

4 IMPLEMENTATION

K&M showed that the explicit representation of dependencies between variables in MAIDs can substantially reduce the computational cost of finding an NE [13, 18]. In this section, we describe a modified version of their algorithm and use MAID subgames to find all pure SPEs (see also the supplementary material). We show that MAID subgames exhibit the familiar subgame property of being useful for 'generalised backwards induction' algorithms [11].

Beginning with an arbitrary policy profile $\pi(0)$ across all decision nodes in the original MAIM, $\mathcal{M}$, we optimise decision rules associated to each $D \in \boldsymbol{D}$ by iterating backwards through a MAID subgame ordering from $\mathcal{M}_m$ to $\mathcal{M}_0$. In what follows, we write $\mathcal{M}_i \prec \mathcal{M}_j$ if $\mathcal{M}_j$ is a proper MAID subgame of $\mathcal{M}_i$, and $\boldsymbol{D}_k$ for the decision nodes in $\mathcal{M}_k$. Several MAID subgames can have the same set of decisions, $\boldsymbol{D}_k$, so we choose a single MAID subgame $\mathcal{M}_k$ (one with the fewest nodes $\boldsymbol{V}'$) for each $\boldsymbol{D}_k$ and discard the others. Each MAID in this ordering has an associated MAIM for each setting of the nodes which have a child in $\boldsymbol{V}'$.

When considering a MAIM for $\mathcal{M}_{m-i}$, the decision rules for all decision nodes in proper MAIM subgames of $\mathcal{M}_{m-i}$ will have already been optimised and fixed in previous iterations, so these are now chance nodes in $\mathcal{M}_{m-i}$. In addition, none of the decision nodes $\boldsymbol{D}_{m-i}$ in $\mathcal{M}_{m-i}$ strategically rely on any of the decision nodes outside of $\mathcal{M}_{m-i}$. Therefore, this step is localised to computing the optimal decision rules only for $\boldsymbol{D}_{m-i}$.

The next step depends on $|\boldsymbol{D}_{m-i}|$. If only one decision node $D \in \boldsymbol{D}^j$ remains, as in Figure 4 c) for example, then its optimal decision rule is that which maximises player j's expected utility in each of the MAIMs (for each value of D^1) for this MAID subgame. If $|\boldsymbol{D}_{m-i}| > 1$, the relevance graph of $\mathcal{M}_{m-i}$ is cyclic and so the decision nodes strategically rely on one another. We must therefore call a subroutine: the MAIMs for the MAID subgame induced by the policy profile at that step, $\mathcal{M}(\pi_{-\boldsymbol{D}_{m-i}}(i))$, are converted to EFGs to be solved using Gambit [17]. Algorithm 1 shows the full procedure.

It is more efficient to pass the EFGs in the algorithm's subroutine to an EFG solver such as Gambit [17], rather than passing an EFG for the entire original MAIM. In the induced MAIM $\mathcal{M}(\pi_{-\boldsymbol{D}_{m-i}}(i))$, all decision nodes in the proper MAIM subgames have been converted into chance nodes. In our MAID to EFG transformation we need only split on decision nodes and their informational parents, so the size of the EFG is exponential in $|\mathbf{Fa}_{\boldsymbol{D}_{m-i}}|$. As the time complexity of solving an EFG depends on its size, the cost of solving a MAIM

Algorithm 1

Input: MAIM $\mathcal{M} = (\boldsymbol{N}, \boldsymbol{V}, \boldsymbol{E}, \theta)$
Output: SPE π
initialise $\pi(0)$ as an arbitrary fully mixed policy profile
compute an ordering $\prec$ over the subgames $\mathcal{M}_0, \ldots, \mathcal{M}_m$ in $\mathcal{M}$
for i = 0 to m-1 **do**
    compute a best response policy profile $\pi^*_{\boldsymbol{D}_{m-i}}$ for all decision nodes in $\boldsymbol{D}_{m-i}$ using $\mathcal{M}(\pi_{-\boldsymbol{D}_{m-i}}(i))$
    $\pi(i+1) \leftarrow (\pi_{-\boldsymbol{D}_{m-i}}(i), \pi^*_{\boldsymbol{D}_{m-i}})$
return $\pi(m)$

using Algorithm 1 is never greater than solving an equivalent EFG representation of the original game, and is exponentially faster in many cases.

Our open-source Python codebase[5] implements this procedure, provides methods for finding and plotting MAIDs $\mathcal{M}$, along with $Rel(\mathcal{M})$ and $ConRel(\mathcal{M})$, and converts any MAIM into an EFG to be used with Gambit. Our aim is to provide the necessary computational tools for researchers and practitioners to develop further applications of MAIDs.

5 DISCUSSION AND CONCLUSIONS

This work has extended previous results on MAIDs by introducing the concept of a MAID subgame and a range of key equilibrium refinements. K&M argued that MAIDs offer several benefits [13]. First, MAIDs can represent games more concisely than EFGs. Second, because a parametrised MAID is a probabilistic graphical model, the probabilistic dependencies between chance and decision variables can be exploited in order to identify whether decision nodes strategically rely on one another; we used this to define MAID subgames and our resulting equilibrium refinements. Separately, MAIDs can lead to substantial savings in the computational cost of finding an SPE; in Section 4, we have described a modified version of an algorithm of K&M and implemented it in an open-source codebase.

These benefits of MAIDS, coupled with the theoretical and practical contributions of this paper, provide a rich basis for future work. One avenue for such work that we are already pursuing is to extend the analysis of incentives [2, 4] to the multi-agent setting by interpreting the directed edges in MAIDs causally. One could then investigate which variables in the graph each agent has an incentive to observe or control, and which reasoning patterns are involved [22], given that all of the agents in the MAID are playing a certain equilibrium refinement.

ACKNOWLEDGMENTS

The authors wish to thank Ryan Carey for invaluable feedback and assistance while completing this work, as well as Zac Kenton, Colin Rowat, and several anonymous reviewers for helpful comments. Hammond acknowledges the support of an EPSRC Doctoral Training Partnership studentship (Reference: 2218880). Fox acknowledges the support of the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (Reference: EP/S024050/1).

[5]Available online at https://github.com/causalincentives/pycid.

REFERENCES

[1] Dimitrios Antos and Avi Pfeffer. 2008. Identifying Reasoning Patterns in Games. (2008), 9–18.

[2] Ryan Carey, Eric Langlois, Tom Everitt, and Shane Legg. 2020. The Incentives that Shape Behaviour. *arXiv:2001.07118* (2020).

[3] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to Algorithms* (third ed.). The MIT Press, Cambridge, Massachusettss.

[4] Tom Everitt, Ryan Carey, Eric Langlois, Pedro Ortega, and Shane Legg. 2021. Agent Incentives: a Causal Perspective. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, (AAAI-21)*. Virtual. Forthcoming.

[5] Tom Everitt and Marcus Hutter. 2019. Reward Tampering Problems and Solutions in Reinforcement Learning: A Causal Influence Diagram Perspective. *arXiv:1908.04734* (2019).

[6] Tom Everitt, Ramana Kumar, Victoria Krakovna, and Shane Legg. 2019. Modeling AGI Safety Frameworks with Causal Influence Diagrams. *arXiv:1906.08663* (2019).

[7] Y. Gal and A. Pfeffer. 2008. Networks of Influence Diagrams: A Formalism for Representing Agents' Beliefs and Decision-Making Processes. *Journal of Artificial Intelligence Research* 33 (2008), 109–147.

[8] Koen Holtman. 2020. Towards AGI Agent Safety by Iteratively Improving the Utility Function. In *Artificial General Intelligence*. Springer International Publishing, 205–215.

[9] Ronald A. Howard and James E. Matheson. 2005. Influence Diagrams. *Decision Analysis* 2, 3 (2005), 127–143.

[10] Albert Xin Jiang, Kevin Leyton-Brown, and Avi Pfeffer. 2009. Temporal Action-Graph Games: A New Representation for Dynamic Games. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence* (Montreal, Quebec, Canada) *(UAI '09)*. AUAI Press, 268–276.

[11] Marek Mikolaj Kaminski. 2019. Generalized Backward Induction: Justification for a Folk Algorithm. *Games* 10, 3 (2019), 34.

[12] Daphne Koller and Brian Milch. 2001. Multi-agent Influence Diagrams for Representing and Solving Games. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2* (Seattle, WA, USA) *(IJCAI'01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1027–1034.

[13] Daphne Koller and Brian Milch. 2003. Multi-agent Influence Diagrams for Representing and Solving Games. *Games and Economic Behavior* 45, 1 (2003), 181–221.

[14] Harold William Kuhn and Albert William Tucker (Eds.). 1953. *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press.

[15] Eric Langlois and Tom Everitt. 2021. How RL Agents Behave When Their Actions Are Modified. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, (AAAI-21)*. Virtual. Forthcoming.

[16] Michael Maschler, Eilon Solan, and Shmuel Zamir. 2009. *Game Theory*. Cambridge University Press.

[17] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. 2016. Gambit: Software Tools for Game Theory. http://www.gambit-project.org.

[18] Brian Milch and Daphne Koller. 2008. Ignorable Information in Multi-Agent Scenarios. MIT-CSAIL-TR-2008-029 (2008).

[19] J. F. Nash. 1950. Equilibrium Points in N-person Games. *Proceedings of the National Academy of Sciences* 36, 1 (1950), 48–49.

[20] Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[21] Judea Pearl. 2009. *Causality*. Cambridge University Press, Cambridge, UK.

[22] Avi Pfeffer and Ya'akov Gal. 2007. On the Reasoning Patterns of Agents in Games. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 1* (Vancouver, British Columbia, Canada) *(AAAI'07)*. AAAI Press, 102–109.

[23] Michele Piccione and Ariel Rubinstein. 1997. The Absent-Minded Driver's Paradox: Synthesis and Responses. *Games and Economic Behavior* 20, 1 (1997), 121–130.

[24] Reinhard Selten. 1965. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit: Teil i: Bestimmung des Dynamischen Preisgleichgewichts. *Journal of Institutional and Theoretical Economics* H. 2 (1965), 301–324.

[25] R. Selten. 1975. Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games. *International Journal of Game Theory* 4, 1 (1975), 25–55.

[26] Michael Spence. 1973. Job Market Signaling. Vol. 87. Oxford University Press (OUP), 355.

[27] Halbert White and Karim Chalak. 2009. Settable Systems: An Extension of Pearl's Causal Model with Optimization, Equilibrium, and Learning. *Journal of Machine Learning Research* 10 (2009), 1759—-1799.