A Complete Criterion for Value of Information in Soluble Influence Diagrams

Chris van Merwijk*,[1] **Ryan Carey***,[1] **Tom Everitt**[2]
[1]University of Oxford,
[2]DeepMind
chris.vanmerwijk@philosophy.ox.ac.uk, ryan.carey@jesus.ox.ac.uk, tomeveritt@google.com

Abstract

Influence diagrams have recently been used to analyse the safety and fairness properties of AI systems. A key building block for this analysis is a graphical criterion for value of information (VoI). This paper establishes the first complete graphical criterion for VoI in influence diagrams with multiple decisions. Along the way, we establish two techniques for proving properties of multi-decision influence diagrams: ID homomorphisms are structure-preserving transformations of influence diagrams, while a Tree of Systems is a collection of paths that captures how information and control can flow in an influence diagram.

1 Introduction

One approach to analysing the safety and fairness of AI systems is to represent them using variants of Bayesian networks (Everitt et al. 2019; Kusner et al. 2017). Influence diagrams (IDs) can be viewed an extension of Bayesian networks for representing agents (Howard et al. 2005; Everitt et al. 2021a). This graphical perspective offers a concise view of key relationships, that abstracts away from much of the internal complexity of modern-day AI systems.

Once a decision problem is represented graphically, key aspects can be summarised. One well-studied concept is the *value of information* (VoI) (Howard 1966), which describes how much more utility an agent is able to obtain if it can observe a variable in its environment, compared with if it cannot. Other summary concepts includes "materiality", "value of control", "response incentives".

These concepts have been used to analyse the redirectability (Everitt et al. 2021b; Holtman 2020) of AI systems, fairness (Everitt et al. 2021a; Ashurst et al. 2022), ambitiousness (Cohen, Vellambi, and Hutter 2020), and the safety of reward learning systems (Armstrong et al. 2020; Everitt et al. 2019; Langlois and Everitt 2021; Evans and Kasirzadeh 2021; Farquhar, Carey, and Everitt 2022). Typically, this analysis involves applying *graphical criteria*, that indicate which properties can or cannot occur in a given diagram, based on the graph structure alone. Graphical criteria are useful because they enable qualitative judgements

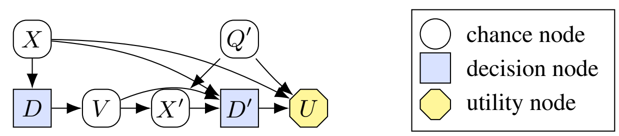


Figure 1: Does X has positive value of information for D?

even when the precise functional relationships between variables are unknown or unspecified.

For the single-decision case, complete criteria have been established for all four of the aforementioned concepts (Everitt et al. 2021a). However, many AI applications such as reinforcement learning involve an agent making multiple decisions. For the multi-decision case, multiple criteria for VoI have been proposed (Nielsen and Jensen 1999; Shachter 1998; Nilsson and Lauritzen 2000), but none proven complete.

This means that for some graphs, it is not known whether a node can have positive VoI. For example, in Fig. 1, it is not known whether it can be valuable for D to observe X. Specifically, the edge $X \to D$ does not meet the criterion of *nonrequisiteness* used by Nilsson and Lauritzen (2000), so we cannot rule out that it contains valuable information. However, the procedure that is used to prove completeness in the single-decision setting (Everitt et al. 2021a) does not establish positive VoI.

We prove that the graphical criterion of Nilsson and Lauritzen (2000) is complete, in that any environmental variable not guaranteed to have zero VoI by their criterion must have positive VoI in some compatible ID. In the course of the proof, we develop several tools for reasoning about soluble IDs. In summary, our main contributions are:

- **ID homomorphisms**. These allow us to transform an ID into another with similar properties, that may be more easily analysed (Section 4).
- **Trees of systems**. A system is a set of paths that make information valuable to a decision. A tree of systems describes how those paths traverse other decisions (Section 5.3).
- **A complete VoI criterion**. We prove the criterion in Section 5. In Section 6 we explain why this criterion may be useful, how it may be used in an AI safety application, and share an open source implementation.

*Equal contribution

2 Setup

Limited memory influence diagrams (also called LIMIDs) are graphical models containing decision and utility nodes, used to model decision-making problems (Howard 1966; Nilsson and Lauritzen 2000).

Definition 1 (Limited memory influence diagram graph; Nilsson and Lauritzen 2000)**.** A *(limited memory) ID graph* is a directed acyclic graph $\mathcal{G} = (\boldsymbol{V}, E)$ where the vertex set $\boldsymbol{V}$ is partitioned into *chance-* ($\boldsymbol{X}$), *decision-* ($\boldsymbol{D}$), and *utility nodes* ($\boldsymbol{U}$). Utility nodes lack children.

Since all of the influence diagram graphs in this paper have limited memory, we will consistently refer to them simply as *influence diagram* (ID) graphs. We denote the parents, descendants, and family of a node $V \in \boldsymbol{V}$ as $\mathbf{Pa}(V), \mathbf{Desc}(V)$, and $\mathbf{Fa}(V) = \mathbf{Pa}(V) \cup \{V\}$. For $Y \in \boldsymbol{V}$, we denote an edge by $V \to Y$, and a directed path by $V \dashrightarrow Y$.

To specify the precise statistical relationships, rather than just their structure, we will use a model that attaches probability distributions to the variables in an ID graph.

Definition 2. An *influence diagram* (ID) is a tuple $\mathcal{M} = (\mathcal{G}, \mathrm{dom}, P)$ where $\mathcal{G}$ is an ID graph, $\mathrm{dom}(X)$ is a finite domain for each node X in $\mathcal{G}$ that is real-valued for utility nodes, and $P(X|\mathbf{Pa}(X))$ is a conditional probability distribution (CPD) for each chance and utility node X in $\mathcal{G}$. We will say that $\mathcal{M}$ is *compatible with* $\mathcal{G}$, or simply that $\mathcal{M}$ is an ID *on* $\mathcal{G}$.

The decision-making task is to maximize the sum of expected utilities by selecting a CPD $\pi^D(D|\mathbf{Pa}(D))$, called a *decision rule*, for each decision $D \in \boldsymbol{D}$. A *policy* $\pi = \{\pi^D\}_{D \in \boldsymbol{D}}$ consists of one decision rule for each decision. Once the policy is specified, this induces joint probability distribution $P_\pi^{\mathcal{M}}$ over all the variables. We denote expectations by $\mathbb{E}_\pi^{\mathcal{M}}$ and omit the superscript when clear from context. A policy π is called *optimal* if it maximises $\mathbb{E}_\pi[\mathcal{U}]$, where $\mathcal{U} := \sum_{U \in \boldsymbol{U}} U$. Throughout this paper, we use subscripts for policies, and superscripts for indexing. A lowercase $v \in \mathrm{dom}(V)$ denotes an outcome of V.

Some past work has assumed "no-forgetting", meaning that every decision d is allowed to depend on the value v of any past decision D' or its observations $\mathbf{Pa}(D')$, even when that variable $V \in \mathbf{Fa}(D')$ is not a parent of the current decision ($V \notin \mathbf{Pa}(D)$) (Shachter 1986). In contrast, we follow the more flexible convention of limited memory IDs (Nilsson and Lauritzen 2000), by explicitly indicating whether a decision d can depend on the value of an observation or decision v by the presence (or absence) of an edge $V \to D$, just as we would do with any variable that is not associated with a past decision.

Within the space of limited memory IDs, this paper focuses on *soluble* IDs (Nilsson and Lauritzen 2000), also known as IDs with "sufficient recall" (Milch and Koller 2008). The solubility assumption requires that it is always possible to choose an optimal decision rule without knowing what decision rules were followed by past decisions. The formal definition uses d-separation.

Definition 3 (d-separation; Verma and Pearl (1988))**.** A path p is *blocked* by a set of nodes $\boldsymbol{Z}$ if p contains a collider $X \to W \leftarrow Y$, such that neither W nor any of its descendants are in $\boldsymbol{Z}$, or p contains a chain $X \to W \to Y$ or fork $X \leftarrow W \to Y$ where W is in $\boldsymbol{Z}$. If p is not blocked, then it is *active*. For disjoint sets $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, the set $\boldsymbol{Z}$ is said to *d-separate* $\boldsymbol{X}$ from $\boldsymbol{Y}$, $(\boldsymbol{X} \perp \boldsymbol{Y} \mid \boldsymbol{Z})$ if $\boldsymbol{Z}$ blocks every path from a node in $\boldsymbol{X}$ to a node in $\boldsymbol{Y}$. Sets that are not d-separated are called *d-connected*.

Definition 4 (Solubility; Nilsson and Lauritzen (2000))**.** For an ID graph $\mathcal{G}$ let the *mapping extension* $\mathcal{G}'$ be a modified version of $\mathcal{G}$ where a chance node parent Π^i is added to each decision D^i. Then $\mathcal{G}$ is *soluble* if there exists an ordering $D^1, \ldots, D^n$ over the decisions, such that in the mapping extension $\mathcal{G}'$, for all i:

$$\Pi^{<i} \perp \boldsymbol{U}(D^i) \mid \mathbf{Fa}(D^i)$$

where $\Pi^{<i} := \{\Pi^j \mid j < i\}$ and $\boldsymbol{U}(D^i) := \boldsymbol{U} \cap \mathbf{Desc}(D^i)$.

We will subsequently only consider ID graphs that are soluble. Solubility is entailed by the popular more restrictive "no forgetting" assumption, where the decision-maker remembers previous decisions and observations (Shachter 1986, 2016): in no forgetting, the family $\mathbf{Fa}(D^i)$ includes $\mathbf{Fa}(D^j)$ for $j < i$, so every policy node Π^j is d-separated from $\boldsymbol{V} \setminus \mathbf{Fa}(D^j) \supseteq \boldsymbol{U} \cap \mathbf{Desc}^{D^j}$. However, solubility is more general, for example Fig. 1 is soluble, even though past decisions are forgotten.

3 Value of Information

The VoI of a variable indicates how much the attainable expected utility increases when a variable is observed compared to when it is not:

Definition 5 (Value of Information; Howard (1966))**.** For an ID $\mathcal{M}$ and $X \notin \mathbf{Desc}_D$, let $\mathcal{M}_{X \not\to D}$ and $\mathcal{M}_{X \to D}$ be $\mathcal{M}$ modified by respectively removing and adding the edge $X \to D$. Then, the *value of information* of X for D is:

$$\max_\pi \mathbb{E}_\pi^{\mathcal{M}_{X \to D}}[\mathcal{U}] - \max_\pi \mathbb{E}_\pi^{\mathcal{M}_{X \not\to D}}[\mathcal{U}].$$

This is closely related to the concept of *materiality*; an observation $X \in \mathbf{Pa}(D)$ is called material if its VoI is positive.

The graphical criterion for VoI that we will use iteratively removes information links that cannot contain useful information, based on a condition called *nonrequisiteness*. If $X \perp \boldsymbol{U}(D^i) \mid \mathbf{Fa}(D^i) \setminus \{X\}$, then both X and the information link $X \to D^i$ are called *nonrequisite*, otherwise, they are *requisite*. Intuitively, nonrequisite links contain no information about influencable utility nodes, so the attainable expected utility is not decreased by their removal. Removing one nonrequisite observation link can make a previously requisite information link nonrequisite, so the criterion involves iterative removal of nonrequisite links. The criterion was first proposed by Nilsson and Lauritzen (2000), who also proved that it is sound. Formally, it is captured by what we calll a d-reduction:

Definition 6 (d-reduction)**.** The ID graph $\mathcal{G}'$ is a *d-reduction* of $\mathcal{G}$ if $\mathcal{G}'$ can be obtained from $\mathcal{G}$ via a sequence $\mathcal{G} = \mathcal{G}^1, ..., \mathcal{G}^k = \mathcal{G}'$ where each $\mathcal{G}^i, i > 1$ differs from its predecessor $\mathcal{G}^{i-1}$ by the removal of one nonrequisite information link. A d-reduction is called *minimal* if it lacks any nonrequisite information links.

For any ID graph $\mathcal{G}$, there is only one minimal d-reduction (Nilsson and Lauritzen 2000), i.e. the minimal d reduction is independent of the order in which edges are removed. We can therefore denote *the minimal* d*-reduction* of $\mathcal{G}$ as $\mathcal{G}^*$. Thus, Nilsson and Lauritzen (2000, Theorem 3) states that *if* an ID graph $\mathcal{G}$ contains $X \to D$ but $\mathcal{G}^*$ does not, then X has zero VoI in every ID compatible with $\mathcal{G}$. Our completeness result replaces this with an *if and only if* statement.

Theorem 7 (VoI Criterion)**.** *Let $\mathcal{G}$ be a soluble ID graph containing an edge $X \to D$ from chance node $X \in \boldsymbol{X}$ to decision $D \in \boldsymbol{D}$. There exists an ID $\mathcal{M}$ compatible with $\mathcal{G}$ such that X has strictly positive VoI for D if and only if the minimal d-reduction contains $X \to D$.*

The VoI criterion is posed in terms of a graph $\mathcal{G}$ that contains $X \to D$. To analyse a graph that does not, one can simply add the edge $X \to D$ then apply the same criterion as long as the new ID graph is soluble (Shachter 2016).

The proof will be given in Section 5, with details in Appendices C and D. We note that this excludes the case of remembering a past decision $X \in \boldsymbol{D}$, because Nilsson's criterion is incomplete for this case. For example, the simple ID graph with the edges $D \to D' \to U$ and $D \to U$, D satisfies the graphical criterion of being requisite for D', but D' has zero VoI because it is possible for the decision D to be deterministically assigned some optimal value. This means that there is no need for D' to observe D.

4 ID Homomorphisms

To make the analysis easier, we will often want to transform an original ID graph into a more structured one. Before describing the structure we will be aiming for, we consider the general question of when a modified ID graph retains important properties of the original. To this end, we will define the concept of an *ID homomorphism*, which we then use to define a class of property-preserving ID transformations. (Proofs are supplied in Appendix B.)

Definition 8 (ID homomorphism)**.** For ID graphs $\mathcal{G}=(\boldsymbol{V}, E)$ and $\mathcal{G}'=(\boldsymbol{V}', E')$, a map $h: \boldsymbol{V}' \to \boldsymbol{V}$ is an *ID homomorphism* from $\mathcal{G}'$ to $\mathcal{G}$ iff:

(a) (Preserves node types) h maps each chance-, decision-, or utility-node to a node of the same type;

(b) (Preserves links) For every $A \to B$ in $\mathcal{G}'$ either $h(A) \to h(B)$ is in $\mathcal{G}$, or $h(A) = h(B)$;

(c) (Covers all information links) If $h(N) \to h(D)$ is in $\mathcal{G}$ for $D \in \boldsymbol{D}$, then $N \to D$ is in $\mathcal{G}'$; and

(d) (Combines only linked decisions) If $h(D_1) = h(D_2)$ for decisions $D^1 \neq D^2$ in $\mathcal{G}'$ then $\mathcal{G}'$ contains $D^1 \to D^2$ or $D^2 \to D^1$.

An ID homomorphism is analogous to the notion of graph homomorphism from graph theory, which essentially requires that edges are preserved along the map. An ID homomorphism additionally requires that decisions in the two graphs have equivalent parents (c), and that split decisions are connected (d). This requirement maintains a direct correspondence between policies on the two graphs, so that, as we will see, ID homomorphisms preserve VoI. Examples of ID homorphisms are given in Fig. 2.

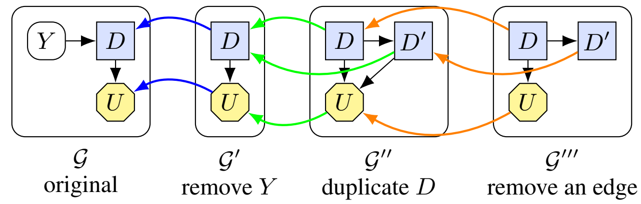


Figure 2: A sequence of homorphic transformations showing how $\mathcal{G}$ can be homorphically transformed into $\mathcal{G}'''$ by composition of Lemmas 13 and 14. In the first step from $\mathcal{G}$ to $\mathcal{G}'$, Y is removed; in the step from $\mathcal{G}'$ to $\mathcal{G}''$ a decision is duplicated; and in the final step from $\mathcal{G}''$ to $\mathcal{G}'''$, a link is removed. Since the mapping at each step (blue, green, and orange respectively) meets the definition of an ID homomorphism, $\mathcal{G}'''$ must be an ID homorphism of $\mathcal{G}$ (Lemma 15).

The following three lemmas establish properties that are preserved under ID homomorphisms.

Lemma 9 (Preserves Solubility)**.** *Let $\mathcal{G} = (\boldsymbol{V}, E)$ and $\mathcal{G}' = (\boldsymbol{V}', E')$ be ID graphs. If $\mathcal{G}$ is soluble, and there exists a homomorphism $h: \boldsymbol{V}' \to \boldsymbol{V}$, then $\mathcal{G}'$ is also soluble.*

Given a homomorphism h from $\mathcal{G}'$ to $\mathcal{G}$, we can define a notion of equivalence between IDs (and policies) on each graph. Roughly, two IDs are equivalent if the domain of every node is a cartesian product of the domains of the nodes in its pre-image (or the sum, in the case of a utility node). Formally:

Definition 10 (Equivalence)**.** $\mathcal{M}_\pi$ on $\mathcal{G}$ and $\mathcal{M}'_{\pi'}$ on $\mathcal{G}_{\pi'}$ are *equivalent* if each non-utility node N in $\mathcal{G}$ has $\mathrm{dom}(N) := \bigtimes_{N^i \in h^{-1}(N)} \mathrm{dom}(N^i)$, and $P^{\mathcal{M}}_{\pi}(N = (n^1, ..., n^k)) = P^{\mathcal{M}'}_{\pi'}(N^1 = n^1, ..., N^k = n^k)$, and each utility node has $P^{\mathcal{M}}_{\pi}(U = u) = P^{\mathcal{M}'}_{\pi'}(\sum_{U^i \in h^{-1}(U)} U^i = u)$.

Lemma 11 (Equivalence)**.** *If there is an ID homomorphism h from $\mathcal{G}'$ to $\mathcal{G}$, then for any policy π' in any ID $\mathcal{M}'$ on $\mathcal{G}'$ there is a policy π in a ID $\mathcal{M}$ on $\mathcal{G}$ such that $\mathcal{M}_\pi$ and $\mathcal{M}'_{\pi'}$ are equivalent.*

In this case, we will call $\mathcal{M}$ and π the *ID and policy transported along the homomorphism h*. In the appendix, we show that this correspondence between policies on $\mathcal{M}'$ and $\mathcal{M}$ is a bijection. Intuitively, if there is an ID homomorphism $\mathcal{G}' \to \mathcal{G}$, this means we have a particular way to *fit an ID on $\mathcal{G}'$ into $\mathcal{G}$*, while preserving the information that the decisions can access. The basis of this proof is that properties (c,d) of ID homomorphisms (Definition 8) require decisions to have precisely the same information in $\mathcal{M}$ as in $\mathcal{M}'$.

For our proof of Theorem 7, we will require that VoI is preserved under homomorphism.

Lemma 12 (Preserves VoI)**.** *Let $h: \mathcal{G}' \to \mathcal{G}$ be an ID homomorphism. If X' has positive VoI for D' in an ID $\mathcal{M}'$ on $\mathcal{G}'$, then $X = h(X)$ has positive VoI for $D = h(D')$ in the transported ID $\mathcal{M} = h(\mathcal{M}')$.*

The proof builds heavily on there being a precise correspondence between policies on $\mathcal{M}$ and on $\mathcal{M}'$. Since these two IDs are equivalent (Lemma 11), if obtaining certain information in $\mathcal{M}'$ has value, so does obtaining that information in $\mathcal{M}$. The formal details are left to Appendix B.

We next present two transformation rules with which to modify any ID graph, which are illustrated in Fig. 2. The first transformation obtains a new graph $\mathcal{G}'$ by deleting or duplicating nodes, while preserving all links. Under this transformation, the function that maps a node in $\mathcal{G}'$ to its 'originating node' in $\mathcal{G}$ is an ID homomorphism:

Lemma 13 (Deletion & Link-Preserving Copying)**.** *Let $\mathcal{G}=(\boldsymbol{V}, E)$ be an ID graph and $\mathcal{G}'=(\bigcup_{N\in\boldsymbol{V}} \text{Copies}(N), E')$ an ID graph where* Copies *maps nodes in $\mathcal{G}$ to disjoint sets in $\mathcal{G}'$, and where E' is a minimal set of edges such that for any edge $A \to B$ in E and $A^i \in \text{Copies}(A)$ and $B^i \in \text{Copies}(B)$ there is an edge $A^i \to B^i$, and if $A^i, A^j \in \text{Copies}(A)$ are non-utility nodes then either $A^i \to A^j$ or $A^i \leftarrow A^j$. Then the function h that maps each $V \in \text{Copies}(N)$ to N is an ID homomorphism.*

Edges that are not information links can also be removed, while having a homomorphism back to the original:

Lemma 14 (Link Pruning)**.** *Let $\mathcal{G} = (\boldsymbol{V}, E)$ and $\mathcal{G}' = (\boldsymbol{V}, E')$ be ID graphs, where $E' \subseteq E$ and where for each decision node D in $\boldsymbol{V}$, every incoming edge $N \to D$ in E is in E'. Then the identity function $h(N) = N$ on $\boldsymbol{V}$ is a homomorphism from $\mathcal{G}'$ to $\mathcal{G}$.*

Finally, we can chain together a sequence of such graph transformation steps, and still maintain a homomorphism to the original. The justification for this is that a composition of ID homomorphisms is again an ID homomorphism:

Lemma 15 (Composition)**.** *If $h\colon \mathcal{G}' \to \mathcal{G}$ and $h'\colon \mathcal{G}'' \to \mathcal{G}'$ are ID homomorphisms then the composition $h \circ h'\colon \mathcal{G}'' \to \mathcal{G}$ is an ID homomorphism.*

5 Completeness of the VoI Criterion

We will now prove that the *value of information* (VoI) criterion of Nilsson and Lauritzen (2000) is complete for chance nodes (details are deferred to Appendix C and **??**).

5.1 Parameterising One System

To prove that the criterion from Theorem 7 is complete we must show that for any graph where $X \to D$ is in the minimal d-reduction, X has positive VoI for D. For example, consider the graph in Fig. 3, which is its own d-reduction, and contains $X \to D$. In this graph, we can choose for X to be Bernoulli distributed, for D to have the boolean domain $\{0,1\}$, and for U to be equal to 1 if and only if X and D match. Clearly, the policy $d = x$ will obtain $\mathbb{E}[U] = 1$. In contrast, if X were not observed (no link $X \to D$), then no policy could achieve expected utility more than 0.5; so the VoI of X in this ID is 0.5.

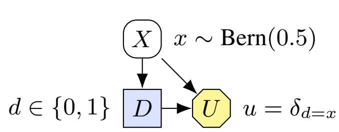


Figure 3: The observation X has positive VoI for D.

A general procedure for parameterising any single-decision ID graph meeting the Theorem 7 criterion to exhibit positive VoI has been established by Everitt et al. (2021a) and Lee and Bareinboim (2020). This procedure consists of two steps: first, establish the existence of some paths, then choose CPDs for the nodes on those paths. We call the paths found in the first step a *system*, which will be a building block for our analysis of IDs with multiple decisions. A fully-general illustration of a system is shown in Fig. 4.

Definition 16 (System)**.** A *system* s in an ID graph $\mathcal{G}$ is a tuple $(\text{control}^s, \text{info}^s, \text{obs}^s)$ where:

- The *control path*, control^s, is a directed path $D^s \dashrightarrow U^s$ where $D^s \in \boldsymbol{D}$ and $U^s \in \boldsymbol{U}$,
- The *info path*, info^s, is a path $\mathbf{Pa}(D^s) \ni X^s \text{ --- } U^s$, active given $\mathbf{Fa}(D^s) \setminus \{X^s\}$,
- obs^s maps each collider C^i in info^s to an *obs path*, a *minimal-length* directed path $C^i \dashrightarrow D^s$.

We denote the *information link of* s, $X^s \to D^s$, by infolink^s and the union of nodes in *all* paths of s by $\boldsymbol{V}^s$.

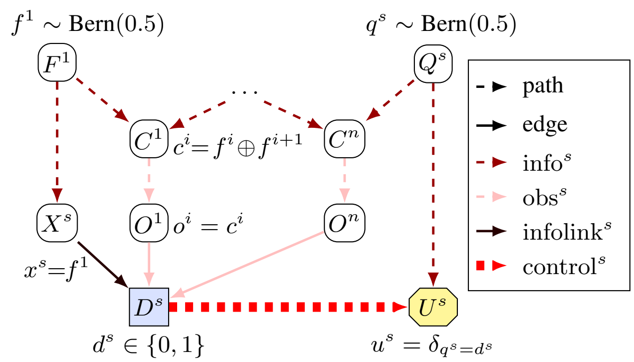


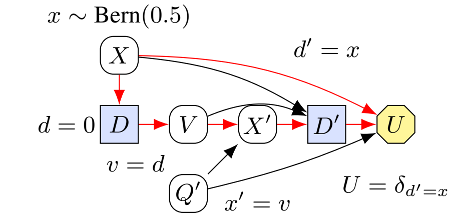
Figure 4: A system, annotated with a parameterization that has positive VoI in the single-decision case. Dashed arrows can zero or more nodes.

The existence of these paths follow from the graphical criterion of Theorem 7. In particular, since $X \to D$ is in the minimal d-reduction of $\mathcal{G}$, there must exist a path from X to some utility node $U \in \boldsymbol{U} \cap \mathbf{Desc}^{D^s}$, active given $\mathbf{Fa}(D^s) \setminus \{X^s\}$ (the "info path" in Definition 16).

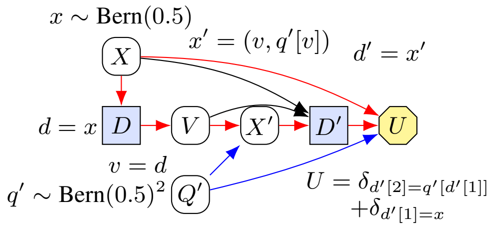
The second step is to choose CPDs for the nodes $\boldsymbol{V}^s$ in the system s, as also illustrated in Fig. 4. The idea is to require the decision D^s to match the value of Q^s, by letting the utility U^s equal 1 if and only if its parents along the control and information paths are equal. If X^s is observed, the decision $D^s = X^s \oplus O^1 ... \oplus O^n = Q^s$ yields $\mathbb{E}[U^s] = 1$, where $\oplus$ denotes *exclusive or* (XOR). Otherwise, the observations $O^1, ..., O^n$ are insufficient to decrypt Q^s, giving $\mathbb{E}[U^s] < 1$. So X^s has positive VoI. The intuitive idea is that U^s tests whether D^s knows Q^s, based on the value d^s transmitted along control^s.

5.2 Parameterising Two Systems

When we have two decisions, however, it becomes insufficient to parameterise just one system. For example, suppose that we try to apply the same scheme as in the previous subsection to the graph of Fig. 5a. Then, we would generate a random bit at X and stipulate that the utility is $U = 1$ if the par-



(a) The variable X has zero VoI for D.



(b) The variable X has positive VoI for D.

Figure 5: In (a), a parameterisation of nodes in a single (red) system fails to exhibit that X has positive VoI for D, whereas in (b), positive VoI is exhibited by parameterising two (red and blue) systems.

ents X and D' on the red paths are equal. One might hope that this would give D an incentive to observe X, so that $d = x$ is copied through D' to obtain $\mathbb{E}[U] = 1$. And that is indeed one way to obtain optimal expected utility. However, the presence of a second decision D' means that maximal utility of $U = 1$ may also be obtained using the policy $d = 0, d' = x$, which does not require X to be observed by D.

To achieve positive VoI, it is necessary to parameterise two systems as shown in Fig. 5b. We first parameterise the second (blue) system to ensure that x' is transmitted to U, and then parameterise the initial (red) system.

To check that X has positive VoI for D, we now solve the combined model. Due to the solubility assumption, we know that the optimal decision rule at D' does not depend on the decision rule taken at D. So let us consider D' first. D' chooses a pair (i, j) where i is interpreted as an index of the bits generated at Q', and j is interpreted as a claim about the i^{th} bit of Q'. The first term of the utility U is equal to 1 if and only if the "claim" made by D' is correct, i.e. if the i^{th} bit generated by Q' really is j. X' contains (only) the v^{th} digit of Q'. Hence D' can only ensure its "claim" is correct if it chooses $d' = x' = (v, q'[v])$, where $q'[v]$ denotes the v^{th} bit of q'. Having figured out the optimal policy for D', we next turn our attention to D. Intuitively, the task of D is to match X, as in Fig. 3. The parameterization encodes this task, by letting D determine V, which in turn influences which bit of Q' is revealed to D'. This allows U to check the output of D via the index outputted by D', and thereby check whether D matched X. This means the second term of U is 1 if and only if $D = X$ so $d = x$ the optimal policy for D, with expected utility $\mathbb{E}[U] = 2$.

In contrast, if X were unobserved by D, then it would no-longer be possible to achieve a perfect score on both terms of U, so $\mathbb{E}[U] < 2$. This shows that X has positive VoI for D.

5.3 A Tree of Systems

In order to generalise this approach to arbitrary number of decisions, we need a structure that specifies a system for each decision, and indicates what downstream decisions that system may depend on. These relationships may be represented by a tree.

Definition 17 (Tree of systems)**.** A *tree of systems* on an ID graph $\mathcal{G}$ is a tuple $T = (\mathcal{S}, \mathrm{pred})$ where:

- $\mathcal{S} = (s^0, ..., s^k)$ is a list of systems (which may include duplicates).
- pred maps each s^i to a pair (s^j, p), where $s^j \in (\mathcal{S} \setminus \{s^i\})$ is a system, p is one of the paths of s^j (info, control, or obs), and infolink$^{s^i}$ is in the path p, except there is a unique "root system" s^{root} that is mapped to $(s^{\text{root}}, \text{"None"})$.

Moreover, a *full tree of systems* is one where for each information link $X' \to D'$ in each path p in each system s, there is precisely one system s' whose information link equals $X' \to D'$ and with $\mathrm{pred}(s') = (s, p)$.

The idea of a tree of systems is that if a decision $D^{s'}$ lies on a path in the system s of some decision D^s, then s is a predecessor of s'. We will use this tree to parameterise the ID graph, and then we will also use it to supply an ordering over the decisions (from leaf to root) in which the model can be solved by backward induction.

In order to generalise the approach taken to parameterising two systems, we need to reason about the systems independently, in reverse order. If the systems overlap, however, this makes it harder to reason about them independently. Thus it is useful to define a notion of systems called *normal form* that are well-behaved.

Definition 18 (Normal form tree)**.** A tree T on $\mathcal{G}$ is in *normal form* if all of the following hold:

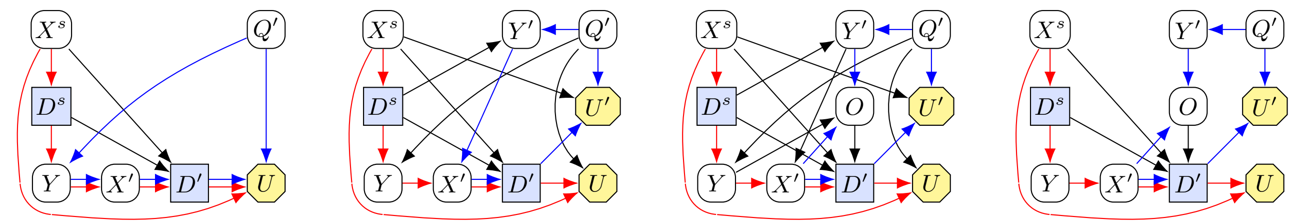
(a) (position-in-tree-uniqueness) A node N in T can only be in multiple paths $p^1, ..., p^k$ of systems in the tree, if splitting N into $\{N, N'\}$ via Lemma 13 and obtaining T' from T by replacing N with N' in one of those paths would make T' no longer a tree of systems.

(b) (no-backdoor-infopaths) Every system s in T has an info path that starts with an outgoing link from X^s.

(c) (no-redundant-links) If $N \to N'$ is an edge to a non-decision N', where one of N and N' is in a path in a system of T, not including the nodes of the root information link, then $N \to N'$ is in a path of a system of T.

An arbitrarily chosen tree will not generally be in normal form. For example, Fig. 6a contains two systems (a red root system for $X^s \to D^s$ and a blue child system for $X' \to D'$) that constitute a tree, but this tree fails all three requirements for being in normal form. However, by a series of homomorphic transformations, it is possible to obtain a new graph with a tree of systems that is in normal form (as in Fig. 6d).

Lemma 19 (Normal Form Existence)**.** *Let $\mathcal{G}$ be a soluble ID graph whose minimal d-reduction $\mathcal{G}^*$ contains $X \to D$. Then there is a normal form tree T' on a soluble ID graph $\mathcal{G}'$, with a homomorphism h from $\mathcal{G}'$ to $\mathcal{G}$ where the information*



(a) Y and U occur in both s (red) and in s' (blue)

(b) Copying Y and U ensures position-in-tree-uniqueness

(c) Making a copy O of X', ensures no-backdoor-infopaths.

(d) Finally, links are removed, ensuring no-redundant-links

Figure 6: An ID graph (a) is homomorphically transformed via graphs (b) and (c) into a graph (d) whose tree is in normal form.

link $X' \to D'$ of the root system of T', has $h(X') = X$, $h(D') = D$, and every node in $\mathcal{G}$ is also in $\mathcal{G}'$ but the only nodes in $\mathcal{G}$ that are in T' are X and D.

Essentially, the procedure for obtaining a normal form tree proceeds in four steps:

1. Construct a tree of systems on $X \to D$: First, pick any system for $X \to D$. Then, pick any system for every other information link $X' \to D'$ in the existing system. Iterate until every link in the tree has a system.
2. Make a copy (lemma 12) of each node for each position (basically, each path) that node has in the tree. This ensures position-in-tree-uniqueness.
3. For systems whose infopath starts with an incoming link $X \leftarrow Y$, copy X (lemma 12), to obtain $X \to O \leftarrow Y$. This ensures no-backdoor-infopaths.
4. Prune the graph (using lemma 13), by removing any (non-information) links outside the tree of systems. This ensures no-redundant-links.

For example, in Figs. 6a to 6d, three transformations are performed, each of which makes the tree meet one additional requirement, ultimately yielding a normal form tree (Fig. 6d) with a homomorphism to the original.

5.4 Proving Positive VoI Given a Normal Form Tree

The reason for using normal form trees is that they enable each system to be parameterized and solved independently. In particular, we know that the optimal policy for one system involves reproducing information from ancestor nodes such as Q^s. As optimal policies can be found with backwards induction in soluble graphs, our approach involves finding optimal policies in reverse order. It will therefore suffice to prove that non-descendant systems cannot provide information about ancestor nodes within the system. For example, in Fig. 6a, when solving for $\pi^{D'}$, we would like to know that D^s cannot provide information about Q'.

Lemma 20 (Subtree Independence)**.** *Let s be a system in a normal form tree $\mathcal{T}$ on a soluble ID graph $\mathcal{G}$. Let $\mathbf{Pa}^{-s} = \mathbf{Pa}(D^s) \setminus \boldsymbol{V}^s$ be D^s's out-of-system parents, $\mathbf{Pa}^s = \mathbf{Pa}(D^s) \cap \boldsymbol{V}^s$ be the within-system parents of D^s, $\mathbf{ObsDesc^s}$ be the observation nodes in descendant systems of s, and let $\mathbf{Back^s} = \mathbf{V^s} \cup (\boldsymbol{Anc}(\mathbf{D^s}) \setminus \mathbf{Fa}(\mathbf{D^s}))$. Then $\mathbf{Back^s} \perp \mathbf{Pa^{-s}} \setminus \mathbf{ObsDesc^s} \mid \mathbf{Pa^s} \cup \mathbf{ObsDesc^s}$.*

For example, Fig. 6d, has a normal form tree, which implies the assurance that X' cannot use information from the red system to tell it about Q'; formally, $Q' \perp (Y \cup X^s) \mid X'$. Given that each decision D^s in the tree cannot use information from ancestor systems, we can then prove that D^s cannot know enough about X^s and Q^s to perform optimally, without observing X^s. More formally:

Lemma 21 (VoI Given Normal Form Tree)**.** *Let $\mathcal{G}$ be a soluble ID graph with a normal form tree with root info link $X \to D$. Then there exists an ID compatible with $\mathcal{G}$ for which X has positive VoI for D.*

The formal proof is given in Appendix D.3. Informally, in order to show that the decision of each system is forced to behave as intended despite there now being a tree of systems full of other decisions, we use Lemma 20 to show that the utility that a decision obtains in system s only depends on the information it obtains from within system s. This rules out that ancestor decisions can observe and pass along relevant information via a path outside the system. Moreover, we know by the solubility assumption that the optimal decision rule at a later decision cannot depend on the decision rule followed by earlier decisions. The argument then proceeds by backward induction. The final decision D^{s^n} must copy the value of X^{s_n}. Given that it does so, the penultimate decision $D^{s^{n-1}}$ must do the same. And so on, until we find that D must copy X, and cannot do so in any way other than by observing it, meaning that X has positive VoI for D.

Finally, we can prove our main result, that there exists an ID on $\mathcal{G}$ where X has positive VoI.

Proof of Theorem 7 (completeness direction). We know that the d-reduction $\mathcal{G}^*$ of $\mathcal{G}$ contains $X \to D$. By Lemma 19, there exists an ID graph $\mathcal{G}'$ with normal form tree rooted at a link $X' \to D'$, with an ID homomorphism from $\mathcal{G}'$ to $\mathcal{G}$ that has $h(X') = X$ and $h(D') = D$. By Lemma 21, since $\mathcal{G}'$ has a normal form tree rooted at $X' \to D'$, there exists an ID on $\mathcal{G}'$ in which X' has positive VoI for D'. By Lemma 12, the presence of the ID homomorphism h from $\mathcal{G}'$ to $\mathcal{G}$ means that there also exists an ID $\mathcal{M}$ on $\mathcal{G}$ such that $h(X') = X$ has positive VoI for $h(D') = D$, showing the result. □

6 Applications & Implementation

Graphical criteria can help with modeling agents' incentives in a wide range of settings including (factored) Partially Observed Markov Decision Processes (POMDPs) and

Modified-action Markov Decision Processes (Langlois and Everitt 2021). For concreteness, we show how our contributions can aid in analysing a supervision POMDP (Milli et al. 2017). In a supervision POMDP, an AI interacts with its environment, given suggested actions from a human player. We will assume that the human's policy has already been selected, in order to focus on the incentives of the AI system.

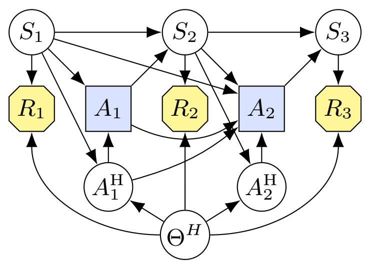


Figure 7: A supervision POMDP with the human considered part of the environment; we show 3 timesteps and 2 actions.

Given the graph in Fig. 7, we can apply the VoI criterion to each A_i^H, the sole parent of A^i. The minimal d-reduction is identical to the original graph, so since $A_i^H \not\perp R^{i+1} \mid \emptyset$, the observation A_i^H can have positive VoI. This formalises the claim of Milli et al. (2017) that in a supervision POMDP, the agent "can learn about reward through [the human's] orders". We can say the same about Cooperative Inverse Reinforcement Learning (CIRL). CIRL differs from supervision POMDPs only in that each human action A_i^H directly affects the state S_{i+1}. If $\mathcal{G}$ is modified by adding edges $A_i^H \to S_{i+1}$, and the VoI criterion is applied at A_i^H once again, we find that A_i^H may have positive VoI for A^i, thereby formalising the claim that the robot is "incentivised to learn" (Hadfield-Menell et al. 2016, Remark 1).

To facilitate convenient use of the graphical criterion, we have implemented it in the open source ID library *pycid* (Fox et al. 2021), whereas the previous implementation was limited to single-decision IDs.[1]

7 Related Work

Value of information The concept of value of information dates back to the earliest papers on influence diagrams (Howard 1966; Matheson 1968). For a review of recent advances, see Borgonovo and Plischke (2016).

Previous results have shown how to identify observations with zero VoI or equivalent properties in various settings. In the no forgetting setting, Fagiuoli and Zaffalon (1998) and Nielsen and Jensen (1999) identified "structurally redundant" and "required nodes" respectively. In soluble IDs, Nilsson and Lauritzen (2000) proved that optimal decisions need not rely on nonrequisite nodes. Completeness proofs in a setting of one decision have been discovered for VoI and its analogues by Zhang, Kumor, and Bareinboim (2020); Lee and Bareinboim (2020); Everitt et al. (2021a). Finally, in insoluble IDs, Lee and Bareinboim (2020) proved that certain nodes are "redundant under optimality". Of these works, only Nielsen and Jensen (1999) attempts a completeness result for the multi-decision setting. However, as pointed out by Everitt et al. (2021a), it falls short in two respects: Firstly, the criterion $X \not\perp \boldsymbol{U}^D \mid \mathbf{Pa}(D)$ is proposed, which differs from nonrequisiteness in the conditioning set. Secondly, and more importantly, the proof is incomplete because it assumes that positive VoI follows from d-connectedness.

[1] Code is available at www.github.com/causalincentives/pycid.

Submodel-trees Trees of systems are loosely related to the "submodel-trees" of Lee, Marinescu, and Dechter (2021). In both cases, the tree encodes an ordering in which the ID can be solved, so the edges in a tree of systems are analogous to those in a submodel-tree. The nodes, however, (i.e. systems and submodels) differ. Whereas a submodel-tree aids with solving IDs, a tree of systems helps with parameterising an ID graph. As a result, a submodel contains all nodes relevant for D, whereas a system consists just one set of info-/control-/obs-paths. Relatedly, in a submodel, downstream decisions may be solved and replaced with a value node, whereas in a tree of systems, they are not.

8 Discussion and Conclusion

This paper has described techniques for analyzing soluble influence diagrams. In particular, we introduced ID homomorphisms, a method for transforming ID graphs while preserving key properties, and showed how these can be used to establish equivalent ID graphs with conveniently parameterizable "trees of systems". These techniques enabled us to derive the first completeness result for a graphical criterion for value of information in the multi-decision setting.

Given the promise of reinforcement learning methods, it is essential that we obtain a formal understanding of how multi-decision behavior is shaped. The graphical perspective taken in this paper has both advantages and disadvantages. On the one hand, some properties cannot be distinguished from a graphical perspective alone. On the other hand, it means our results are applicable even when the precise relationships are unspecified or unknown. There are a range of ways that this work could be beneficial. For example, analogous results for the single-decision setting have contributed to safety and fairness analyses (Armstrong et al. 2020; Cohen, Vellambi, and Hutter 2020; Everitt et al. 2021b, 2019; Langlois and Everitt 2021; Everitt et al. 2021a).

Future work could include applying the tools developed in this paper to other incentive concepts such as value of control (Shachter 1986), instrumental control incentives, and response incentives (Everitt et al. 2021a), to further analyse the value of remembering past decisions (Shachter 2016; Lee and Bareinboim 2020), and to generalize the analysis to multi-agent influence diagrams (Hammond et al. 2021; Koller and Milch 2003).

Acknowledgments

This work was supported in-part by the Leverhulme Centre for the Future of Intelligence, Leverhulme Trust, under Grant RC2015-067.

References

Armstrong, S.; Orseau, L.; Leike, J.; and Legg, S. 2020. Pitfalls in learning a reward function online. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Ashurst, C.; Carey, R.; Chiappa, S.; and Everitt, T. 2022. Why Fair Labels Can Yield Unfair Predictions: Graphical Conditions for Introduced Unfairness. In *AAAI*.

Borgonovo, E.; and Plischke, E. 2016. Sensitivity analysis: a review of recent advances. *European Journal of Operational Research*.

Cohen, M. K.; Vellambi, B. N.; and Hutter, M. 2020. Asymptotically Unambitious Artificial General Intelligence. In *AAAI Conference on Artificial Intelligence*.

Evans, C.; and Kasirzadeh, A. 2021. User Tampering in Reinforcement Learning Recommender Systems. In *FAccTRec Workshop on Responsible Recommendation*.

Everitt, T.; Carey, R.; Langlois, E.; Ortega, P. A.; and Legg, S. 2021a. Agent Incentives: A Causal Perspective. In *AAAI*.

Everitt, T.; Hutter, M.; Kumar, R.; and Krakovna, V. 2021b. Reward Tampering Problems and Solutions in Reinforcement Learning: A Causal Influence Diagram Perspective. *Synthese*.

Everitt, T.; Kumar, R.; Krakovna, V.; and Legg, S. 2019. Modeling AGI Safety Frameworks with Causal Influence Diagrams. In *IJCAI Workshop on AI Safety*.

Fagiuoli, E.; and Zaffalon, M. 1998. A note about redundancy in influence diagrams. *International Journal of Approximate Reasoning*.

Farquhar, S.; Carey, R.; and Everitt, T. 2022. Path-Specific Objectives for Safer Agent Incentives. In *AAAI*.

Fox, J.; Everitt, T.; Carey, R.; Langlois, E.; Abate, A.; and Wooldridge, M. 2021. PyCID: A Python Library for Causal Influence Diagrams. In *Scientific Computing with Python Conference (SciPy)*.

Hadfield-Menell, D.; Dragan, A.; Abbeel, P.; and Russell, S. J. 2016. Cooperative Inverse Reinforcement Learning. In *Advances in Neural Information Processing Systems (Neurips)*.

Hammond, L.; Fox, J.; Everitt, T.; Abate, A.; and Wooldridge, M. 2021. Equilibrium Refinements for Multi-Agent Influence Diagrams: Theory and Practice. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Holtman, K. 2020. AGI Agent Safety by Iteratively Improving the Utility Function. *International Conference on Artificial General Intelligence*.

Howard, R. A. 1966. Information Value Theory. *IEEE Transactions on Systems Science and Cybernetics*.

Howard, R. A.; Matheson, J. E.; Howard, R. A.; and Matheson, J. E. 2005. Influence Diagram Retrospective. *Decision Analysis*.

Koller, D.; and Milch, B. 2003. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*.

Kusner, M. J.; Loftus, J. R.; Russell, C.; and Silva, R. 2017. Counterfactual Fairness. In *Advances in Neural Information Processing Systems (Neurips)*.

Langlois, E.; and Everitt, T. 2021. How RL Agents Behave when their Actions are Modified. In *AAAI*.

Lee, J.; Marinescu, R.; and Dechter, R. 2021. Submodel Decomposition Bounds for Influence Diagrams. In *AAAI*.

Lee, S.; and Bareinboim, E. 2020. Characterizing optimal mixed policies: Where to intervene and what to observe. *Advances in Neural Information Processing Systems (Neurips)*.

Matheson, J. E. 1968. The economic value of analysis and computation. *IEEE Transactions on Systems Science and Cybernetics*.

Milch, B.; and Koller, D. 2008. Ignorable Information in Multi-Agent Scenarios. Technical report, Massachusetts Insitute of Technology (MIT).

Milli, S.; Hadfield-Menell, D.; Dragan, A.; and Russell, S. J. 2017. Should robots be obedient? In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Nielsen, T. D.; and Jensen, F. V. 1999. Welldefined decision scenarios. In *Uncertainty in Artificial Intelligence (UAI)*.

Nilsson, D.; and Lauritzen, S. L. 2000. Evaluating influence diagrams using LIMIDs. *Uncertainty in Artificial Intelligence (UAI)*.

Shachter, R. D. 1986. Evaluating influence diagrams. *Operations research*.

Shachter, R. D. 1998. Bayes-Ball: The Rational Pastime (for Determining Irrelevance and Requisite Information in Belief Networks and Influence Diagrams). *Uncertainty in Artificial Intelligence (UAI)*.

Shachter, R. D. 2016. Decisions and Dependence in Influence Diagrams. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*.

Verma, T.; and Pearl, J. 1988. Causal Networks: Semantics and Expressiveness. In *Uncertainty in Artificial Intelligence (UAI)*.

Zhang, J.; Kumor, D.; and Bareinboim, E. 2020. Causal imitation learning with unobserved confounders. *Advances in Neural Information Processing Systems (Neurips)*.