Resolving von Neumann-Morgenstern Inconsistent Preferences

Niplav

January 5, 2024

Abstract

We consider the problem of resolving preferences that are inconsistent under the von Neumann-Morgenstern axioms into consistent preferences. For preferences over deterministic options, we model inconsistent preferences as directed graphs, and the resolution as selecting acyclic tournaments with the same vertices and minimal graph-edit distance, or Hodge decomposition. For preferences over lotteries, we offer two different methods for modeling inconsistence and methods for resolving them: as edge-weighted weakly connected directed graphs (resolution via Hodge decomposition) and as arbitrary relations over lotteries. None of those two representations prove to be satisfactory. We apply the findings to propose an algorithm for changing a utility function as the underlying set of objects changes.

1 Introduction

In economics, decision theory, game theory and parts of artificial intelligence the standard approach to modeling actors is to assume those actors have a fixed utility function they optimise [Peterson, 2017, ch. 5] [Tadelis, 2013, ch. 2] [Russell, 2010, ch. 16], following the foundations laid by von Neumann and Morgenstern [von Neumann and Morgenstern, 1947, ch. 3]. This model is quite appealing: It assigns a real-numbered value to each possible outcome, and there are several theorems establishing that it can't be money-pumped [Gustafsson, 2022] and is compatible with taking Pareto improvements [Wald, 1947].

However, this model has come under criticism as being non-descriptive of human preferences, which can be experimentally shown to violate one or more of the von Neumann-Morgenstern axioms [Allais, 1953] [El Gamal, 2013]. Furthermore, the capable AI systems humanity has constructed so far usually have no in-built utility functions and appear inconsistent, as they are often programs selected by gradient descent to perform well on a loss or reward function, and it is doubtful that they have internal goal representations that correspond to the their loss or reward function [Hubinger et al., 2019].

This tension between the normative theory of rational agency and the observations we can make about intelligent systems in the real world provides a stark contrast and brings up the question of how one could modify the preferences of intelligent systems to be more consistent.

1.1 Motivation

We claim this work is interesting and important for several different reasons:

- **Learning the preferences of weaker incoherent systems**[Dewey, 2011]: Assuming that one system S_1 wants to learn the preferences of a less coherent system S_2, S_1 might want to "correct" inconsistent preferences learned from S_2 to avoid being exploitable via Dutch books.

- **Managing ontological crises**: If a system defines its preferences using a world model, but this world model changes, those preferences might now be inconsistent. Such situations would benefit from a method for resolving inconsistent preferences [De Blanc, 2011].
- **Creating AI systems with consistent preferences**: Assuming that humans will build capable agentic AI systems, we might want to both describe how those agents might achieve coherence, and prescribe ways for them to reach coherence. There are three reasons why we might expect more capable agents to be more coherent:
  - **Deliberate design**: If e.g. humans create AI systems, they might construct such AI systems to have or develop consistent preferences as to avoid unpredictable behavior.
  - **Competitive pressure**: An agent could modify their preferences in response to competitive pressures that exploit any incoherencies it displays, for example through other agents that are attempting to money-pump it [Gustafsson, 2022].
  - **Self-modification**: Agents might modify their own inconsistent preferences to adhere to the von Neumann-Morgenstern axioms, to avoid wasting resources and making it easier to reason about their own future behavior.

1.2 Structure of the Text

This text starts by explaining the von Neumann-Morgenstern axioms and various theorems relating the axioms to concepts such as Dutch books and Pareto improvements. There is a well-developed literature discussing the relevance of these axioms, and we conclude that these axioms are worth taking as a standard for rational agency. We also observe that humans do not satisfy those axioms.

We then examine the literature on inconsistent preferences, finding investigations from economics on time-inconsistent preferences and some scattered attempts in the non-academic literature, but no satisfactory investigations into the topic that cover all possible violations of the von Neumann-Morgenstern axioms.

We then proceed to analyse the problem of resolving inconsistent preferences in two cases:

- Deterministic case: We propose the set of all directed graphs as a mathematical structure that can represent inconsistent preferences over non-lottery options. We propose three algorithms for resolving inconsistent preferences of this type, prove two of the algorithms as being functionally equivalent, and analyse the algorithms in terms of computational complexity and five other criteria.
- Lottery case: We propose two different mathematical structures for representing potentially inconsistent preferences over lotteries: Edge-weighted weakly connected directed graphs and arbitrary relations over lotteries. We propose Hodge decomposition as an efficient method for resolving inconsistencies in the first case, but find that edge-weighted weakly connected directed graphs are insufficient for

representing common inconsistencies found in reported human preferences. We then note that arbitrary relations over lotteries are able to represent those inconsistencies, but we are unable to find an algorithm for resolving inconsistencies in that format.

We finally speculate about one application of the methods for resolving incoherence: Incorporating changes in the world model into preferences defined over that world model.

2 Related Work

As far as our literature review has revealed, the academic literature has no investigation into the specific question we are attempting to answer.

Modeling Inconsistent Preferences In the economic literature, preferences are usually more restricted than in the von Neumann-Morgenstern setting: It is usually assumed that there is a set of goods B, and utility is linear in the amount of consumption c of those goods: $U_e(b \in B) \propto c(b)$. Additionally, such consumption can take place at different time steps: with a single good b and different quantities $c(b)_1, c(b)_2, \ldots, c(b)_n$ consumed at n timesteps, the time-discounted utility (discount factor δ) of this consumption is $U_e(c(b)_1, c(b)_2, \ldots, c(b)_n) \propto \sum_{i=1}^{n} \delta^i c(b)_i$ (which is equivalent to the use of discount rates in reinforcement learning [Sutton and Barto, 2020, ch. 3]).

A common form of modeling human preferences that are not exponentially time-discounted in this way is hyperbolic discounting, in which the discounting factor is a hyperbolic function with a parameter k instead of an exponential, and $U_h(b_1, \ldots, b_n) \propto \sum_{i=1}^{n} \frac{1}{1+ki} c(b_i)$. This kind of discounting leads to disproportionately preferring small rewards soon over large rewards later, and might lead to preference reversals: For two goods b and b', an agent can have the preference $U_h(c(b)_i) > U_h(c(b')_{i+c})$ at time step i and time step $i + c$, but reverse that preference if it lies at another timestep j so that: $U_h(c(b)_j) < U_h(c(b')_{j+c})$ [Ainslie and Herrnstein, 1981]. Such hyperbolic discounting has been observed in humans [Green et al., 1994] and pigeons [Ainslie and Herrnstein, 1981].

Hyperbolic preferences can be modeled in a a game-theoretic setup, in which subagents in aggregation execute a Pareto-dominated strategy, and via a single agent which follows an unchangeable plan [Caillaud and Jullien, 2000]. They do not attempt to resolve these time-inconsistencies to make them time-consistent. Backus and Zin explore further alternatives to the time-discounted utility setup, though they still work with utility functions that are invariant under positive affine transformation [Backus et al., 2004].

Resolving Inconsistent Preferences In the context of taxonomical data, Sun et al. investigate the problem of recovering hierarchies from noisy data. They represent inconsistent taxonomies with directed acyclic graphs and consistent hierarchical taxonomies using directed graphs. They find that, when measuring the number of edges being removed, a voting

ensemble of several different techniques such as TrueSkill [Herbrich et al., 2007] does well on removing as few edges as possible, and usually outperforms removing greedy approximations of the feedback arc set [Sun et al., 2017].

Outside of the academic literature, [Aird and Shovelain, 2020] represent inconsistent preferences as vector fields on a state space (for example states with more/less security and more/less wealth), where a vector $\mathbf{v}$ at a specific point p in the vector field indicates a preference for a change in the direction of $\mathbf{v}$ at p. However, as they note, such a vector field can have inconsistencies in the form of curl. They then discuss the restrictions on the vector field so that it conforms to the von Neumann-Morgenstern axioms, which they conclude to be potential vector fields, and outline how to use Helmholtz decomposition to decompose inconsistent preference vector fields with three dimensions. Their approach bears a strong resemblance to the Hodge decomposition we use with edge-weighted graphs.

Taking a very different approach, [Kirchner, 2022] investigates how to infer utility functions from non-transitive preferences using a neural network. [Kirchner, 2022] relates inferring such preferences to sorting data in which comparisons sometimes are random, resulting in cycles during comparison. He finds that this approach is able to reconstruct orderings even when 10% of the results of comparisons are noise.

Learning Inconsistent Preferences The problem of *inferring* the preferences of irrational agents has been formally posed in [Armstrong and Mindermann, 2018]: It is in general impossible learn such preferences, as any action is equally compatible both with a preference for that action *and* a systematic bias causing the action. Nevertheless [Evans et al., 2016] find a framework that is experimentally successful at inferring the preferences of an agent with time-inconsistent hyperbolic discounting and incorrect beliefs using Bayesian inference: Their method for inferring preferences of inconsistent software agents gives similar results to estimates made by humans. Their framework does not cover all possible variants of inconsistent preferences, and makes no statement about how to resolve the time-inconsistencies. [Evans et al., 2016] also give no theoretical guarantee about the performance of their method.

2.1 The von Neumann-Morgenstern Axioms

The von Neumann-Morgenstern (vNM) axioms and the framework of utility functions are widely regarded as the standard method of modeling preferences over world-states.

There is an extensive philosophical debate about the reasonableness of the vNM axioms, and a number of proposed alternatives. We have explicitely decided not to contribute to this debate (though some of our findings on the difficulty of establishing vNM-coherence might be interesting to philosophers), and instead assume that preferences conforming to the vNM axioms are a goal to be achieved.

Let Ω be a set of n distinct outcomes, and let $\Delta(\Omega)$ be the set of all probability distributions on Ω, which in [von Neumann and Morgenstern, 1947] are called "lotteries". For a given $\omega_1, \omega_2 \in \Omega$, a lottery in which ω_1

and ω_2 both have 50% probability is written as $[0.5 : \omega_1, 0.5 : \omega_2]$. Let $\preceq$ be a preference relation on all lotteries on Ω, that is $\preceq \subseteq \Delta(\Omega) \times \Delta(\Omega)$. Let $l_1, l_2, l_3 \in \Delta(\Omega)$. If it holds both that $l_1 \preceq l_2$ and $l_2 \preceq l_1$, then we write $l_1 \sim l_2$, and we denote the probability l assigns to $\omega \in \Omega$ as $p_l(\omega)$. Then the four von Neumann-Morgenstern axioms are constraints on $\preceq$:

1. **Completeness**: For any l_1, l_2 it holds that $l_1 \preceq l_2$ or $l_2 \preceq l_1$.
2. **Transitivity**: For any l_1, l_2, l_3, if $l_1 \preceq l_2$ and $l_2 \preceq l_3$, then it must also hold that $l_1 \preceq l_3$.
3. **Continuity**: Given l_1, l_2, l_3, if it holds that $l_1 \preceq l_2 \preceq l_3$, then there must be a probability $p \in [0; 1]$ so that $l_2 \sim [p : l_1, (1-p) : l_3]$.
4. **Independence**: Given l_1, l_2, l_3 it holds that $l_1 \preceq l_2$ if and only if for any $p \in [0; 1]$ it holds that $[p : l_1, (1-p) : l_3] \preceq [p : l_2, (1-p) : l_3]$.

It is possible to create a utility function $U : \Delta(\Omega) \to [0; 1]$ from $\preceq$ if and only if $\preceq$ fulfills these four axioms. Let us as a shorthand write ω for the lottery that assigns probability 1 to ω, and probability 0 to all other options (we call such a lottery a "deterministic option"). U has the property that for any lottery l from $\Delta(\Omega)$, the value $U(l)$ is simply the *expected* value of l, that is the mean of the utilities weighted by the probabilities:

$$U(l) = \sum_{\omega \in \Omega} U(\omega) \cdot p_l(\omega)$$

2.1.1 Assuming Asymmetry

In some parts of the text, we will assume that $\preceq$ is strict over deterministic options, in which case we will write it as $\prec$. That means that we demand that for any $\omega_1, \omega_2 \in \Omega$, it must hold that either $\omega_1 \prec \omega_2$ or $\omega_2 \prec \omega_1$.

The reason for this assumption is that one of the algorithms we investigate (namely `EGEDmin`) produces a total order over Ω.

This restriction does not change the fundamental structure of the vNM axioms; specifically, it does not affect the continuity axiom (as even with strict preferences over deterministic options, there can still be non-strict preferences over lotteries).

3 Inconsistent Preferences over Deterministic Options

A consistent preference over Ω that fulfills **completeness**, **transitivity**, **asymmetry** and **irreflexivity** can be represented by an acyclic tournament $G = (\Omega, E)$[1], with $E \subseteq \Omega \times \Omega$. That is, G itself is **complete**, **transitive**, **asymmetric** and **irreflexive**. We call such a G a **consistent** graph (or consistent directed graph, or acyclic tournament).

The set of possible preferences over Ω (including inconsistent preferences), $\mathfrak{P}_\Omega$, may be represented as the set of all directed graphs with

[1] Unless further specified, in this text it will always be the case that the nodes of G are called Ω and its edges are called E.

vertices Ω. We will use $\mathfrak{P}_n$ to denote the set of all directed graphs with $n \in \mathbb{N} = |\Omega|$ vertices, allowing for reflexive edges (that is edges of the form (ω_1, ω_1)). The set $\mathfrak{P}_\Omega$ can be constructed by enumerating the set of adjacency matrices (elements of $\{0,1\}^{n \times n}$) and then, for each adjacency matrix, constructing the corresponding graph.

There are 2^{n^2} possible preferences in $\mathfrak{P}_\Omega$.

For a directed graph $G \in \mathfrak{P}_\Omega$, one can interpret the presence of an edge $(\omega_1, \omega_2) \in E_G$, with $\omega_1, \omega_2 \in \Omega$, as "$\omega_1$ is preferred over ω_2", written $\omega_1 \succ \omega_2$ or $\omega_1 \to \omega_2$.

Let $\mathfrak{C}_\Omega$ be the set of **consistent graphs** over Ω, with $\mathfrak{C}_\Omega \subset \mathfrak{P}_\Omega$, can be constructed by enumerating the set of permutations of Ω, constructing a strict total order out of each permutation, and taking the transitive closure of that strict total order.

There are $n!$ elements in $\mathfrak{C}_\Omega$.

We take the set of **inconsistent graphs** $\mathfrak{I}_\Omega \subset \mathfrak{P}_\Omega$ to be all graphs that are not consistent, that is $\mathfrak{I}_\Omega = \mathfrak{P}_\Omega \backslash \mathfrak{C}_\Omega$.

Let $\mathfrak{W}_\Omega$ be the set of **weakly consistent** graphs over Ω, which may be represented as the set of all graphs that directed graphs that are equivalent to some weak ordering. It can be constructed by taking all weak orderings on Ω, for each weak ordering $\preceq$ creating an edge from ω_1 to ω_2 if and only if $\omega_1 \preceq \omega_2$, and then taking the transitive closure of that graph.

The weak orderings are counted by the ordered Bell numbers.

3.1 Violating the von Neumann-Morgenstern Axioms

In the deterministic case there are only two vNM axioms that can be violated: completeness and transitivity, since continuity and independence rely on the underlying objecs of the preference relation being lotteries.

Directed graphs are well able to represent all violations of these vNM axioms.

Incompleteness Incompleteness is distinct from indifference: indifference between ω_1 and ω_2 exists if both $\omega_1 \preceq \omega_2$ and $\omega_2 \preceq \omega_1$, incompleteness (or incomparability) is the case if neither $\omega_2 \preceq \omega_1$ or $\omega_1 \preceq \omega_2$.

The presence of an incomplete preference in an agent is difficult to operationalize, [Gustafsson, 2022] treats incomparable options as interchangeable, but setups in which an agent takes a default choice or randomizes when presented with incomparable options are also possible (however, as Gustaffson notes, the randomization offers an adversary the option to (in expectation) perform money-pumps).

In a graph-theoretic setting, incomparability between options ω_1, ω_2 is represented by the absence of any edge between ω_1 and ω_2 in the graph G representing the preference.

Intransitivity Intransitivity is quite easy to represent in a graph G: If there is an edge $\omega_1 \to \omega_2 \in E$ and an edge $\omega_2 \to \omega_3 \in E$, but no edge $\omega_1 \to \omega_3 \in E$, then one has represented an intransitive preference $\omega_1 \prec \omega_2, \omega_2 \prec \omega_3, \omega_1 \not\prec \omega_3$.

Symmetry A symmetric (or indifferent) preference between ω_1, ω_2 (written as $\omega_1 \sim \omega_2$) can also easily be represented by a directed graph by having the edges $\omega_1 \to \omega_2, \omega_2 \to \omega_1 \in E$.

3.2 Algorithms for Resolving Inconsistencies

Any method for resolving inconsistent graphs is a function f that maps any inconsistent graph to a set of consistent graphs $f : \mathfrak{P}_\Omega \to \mathcal{P}(\mathfrak{C}_\Omega)$, which might contain more than one element since the inconsistent graph might not fully determine its consistent counterpart.

3.2.1 Finding Consistent Graphs with the Smallest Graph-Edit Distance

One potential class of such functions would be ones that minimize the "distance" $d : \mathfrak{G}_\Omega \times \mathfrak{C}_\Omega \to \mathbb{R}$ from the (possibly inconsistent) graph and its consistent counterparts.

This function f_m would then return

$$f_d(G) = \operatorname*{argmin}_{C \in C_\Omega} d(C, G)$$

We propose a candidate for f_d, which minimizes the edge-graph-edit distance between any $G \in \mathfrak{P}_\Omega$ and the set of consistent versions $\mathbf{C} \subseteq \mathfrak{C}_\Omega$ of G.

Formally:

$$f_{\mathrm{EGED}}(G) = \operatorname*{argmin}_{C \in C_\Omega} \mathrm{EGED}(C, G)$$

where $\mathrm{EGED}(X, Y)$ is the smallest number of edges that need to be added or removed from X to create Y. The addition or removal of vertices is not allowed, since the elements of Ω can be distinguished from one another.

This function is is intuitively appealing: Let $G \in \mathfrak{P}_\Omega$ be a (possibly inconsistent) preference over Ω. Then let $\omega_1, \omega_2 \in \Omega$ be two possible outcomes. the existence of an edge $(\omega_1, \omega_2) \in V_P$ represents that ω_1 is preferred over ω_2.

Then, given G, if one desired a consistent version of G, one would want to give up *as few as possible* of such rankings of two options. One must sometimes give up some of those rankings to achieve von Neumann-Morgenstern consistent preferences (for example to break cycles), but a high number of deletions or additions of rankings is undesirable.

Proposition 1. For two directed graphs on the same set of vertices, $G_1 = (\Omega, E_1), G_2 = (\Omega, E_2)$ the edge-graph-edit distance is the same as the size of the symmetrical difference of the sets of edges, that is $\mathrm{EGED}(G_1, G_2) = |E_1 \Delta E_2|$.

Proof. $\mathrm{EGED}(G_1, G_2) \leq |E_1 \Delta E_2|$: To generate G_2 from G_1 it is necessary to remove edges from G_1 not in G_2, and then add edges from G_2 not in G_1. These comprise the set $(E_1 \backslash E_2) \cup (E_2 \backslash E_1)$. So the graph-edit distance is upper-bounded by the size of the symmetric difference.

$\text{EGED}(G_1, G_2) \geq |E_1 \Delta E_2|$: Assume that $|E_1 \Delta E_2| < \text{EGED}(G_1, G_2)$. Removing $E^- = E_1 \backslash E_2$ from G_1 and adding the edges $E^+ = E_2 \backslash E_1$ results in G_2. But then $E^- \uplus E^+$ is already a graph edit that creates G_2 from G_1, so $\text{EGED}(G_1, G_2)$ can't be a minimal edge-graph-edit distance between G_1 and G_2. □

Algorithm 1 A naive algorithm for computing `EGEDmin`.

```
function EGEDMIN(G)
    m ← ∞, R ← ∅
    for L ∈ 𝔊_Ω do    ▷ L is a consistent graph with vertices Ω and edges E_L
        d ← |EΔE_L|
        if d < m then
            R ← {L}, m ← d
        else if d = m then
            R ← R ∪ {L}
        end if
    end for
    return R
end function
```

3.2.2 Establishing Consistency Stepwise

An alternative approach to resolve a graph G to a set $\mathbf{C}$ of consistent graphs is to proceed by establishing the desired properties stepwise. One such algorithm (which we call "`stepwise`") is to execute the following steps:

- **Remove minimum feedback arc sets**. [Sun et al., 2017] use a greedy approximation algorithm to find and remove the minimum feedback arc set from a "noisy hierarchy" and create a directed acyclic graph. `stepwise` takes a similar approach by computing all minimum feedback arc sets for G and then removing them to ensure the graph is acyclic (so that later establishing transitivity does not violate asymmetry). The result is a set of directed acyclic graphs $\mathbf{A}$, one for each minimum feedback arc set removed from G. For this, one can use an algorithm for finding the minimum feedback arc set from [Baharev et al., 2021], called `mfas` in **Algorithm** 2.
- **Generate all compatible topological sortings**. The elements of $\mathbf{A}$ are now to be converted into acyclic tournaments. We achieve this by computing all topological sortings for each element $A \in \mathbf{A}$ with a recursive algorithm based on Kahn's algorithm [Kahn, 1962] that appends nodes with in-degree 0 in front of a strict order C. The result is a set of acyclic tournaments $\mathbf{C}$ on Ω.

We can now prove that `stepwise` has the same output as `EGEDmin`. First we prove that all outputs of `stepwise` have the same edge-graph-edit distance from G.

Algorithm 2 Computing `stepwise`.

```
function STEPWISE(G)
    if G then is consistent
        return {G}
    end if
    Remove reflexive edges from G
    A ← ∅, R ← ∅
    for fas ∈ MFAS(G) do
        A ← A ∪ {G\fas}
    end for
    for A ∈ A do
        R ← R∪ TOPOLOGICAL_SORTS(A)
    end for
    return R
end function
function TOPOLOGICAL_SORTS(G)
    if |Ω| = 0 then
        return G
    end if
    R ← ∅
    for ω ∈ Ω so that ω has in-degree 0 in G do
        M ← G with ω removed
        T ← TOPOLOGICAL_SORTS(M)
        for T ∈ T do
            R ← R ∪ {T*}                    ▷ T* is the transitive closure of T
        end for
    end for
    return R
end function
```

Lemma 1. For a given $G = (\Omega, E_G)$, all results of `stepwise` have the same edge-graph-edit distance from G.

Proof. Let $\mathbf{S} = \texttt{stepwise}(G)$, and $S = (\Omega, E_S) \in \mathbf{S}$. Since all S are transitive, complete and irreflexive, all S have the same number of edges, namely the triangular number $|E_S| = \frac{|\Omega|(|\Omega|+1)}{2}$. We also know that $\mathrm{EGED}(G, S) = |E_G \Delta E_S|$, and $E_G \Delta E_S = E_G \backslash E_S \cup E_S \backslash E_G$ (the edges we remove from E_G and the edges we add to E_S). The edges removed from E_G are the minimal feedback arc sets, so they all have the same size $m = |E_G \backslash E_S|$. It now suffices to show that $i = E_S \backslash E_G$, the size of the edges added, is constant. It holds that $|E_G| - m + i = |E_S|$, and then $i = |E_S| - |E_G| + m$, which must be constant. So $\mathrm{EGED}(S, G) = m + i$ is also constant for a given $G, S \in \mathbf{S}$. □

We then show that the edges removed by `EGEDmin` are always a minimum feedback arc set.

Lemma 2. Given a directed graph G, let $T = (\Omega, E_T) \in \texttt{EGEDmin}(G)$. Let $E_T^- = E \backslash E_T$ (the edges removed from G to achieve T) and $E_T^+ = E_T \backslash E$ (the edges added to G to create T). Then E_T^- is a minimum feedback arc set of G.

Proof. E_T^- is a feedback arc set: Assume that E_T^- was not a feedback arc set. Then G would need to contain a cycle of directed edges $E_c = \omega_1 \to \omega_2 \to \cdots \to \omega_{k-1} \to \omega_k \to \omega_1$ so that the cycle was still present after removing E_T^-, that is $E_c \subseteq E \backslash E_T^-$. We know that then $E_T = (E \backslash E_T^-) \cup E_T^+$, but adding edges can't remove a subset, so $E_c \subseteq E \backslash E_T^- \Rightarrow E_c \subseteq (E \backslash E_T^-) \cup E_T^+$.

But then T can't be transitive, asymmetric and complete: If it was transitive and complete, then there would need to be an edge $\omega_1 \to \omega_3$ (created through $\omega_1 \to \omega_2 \to \omega_3$), an edge $\omega_1 \to \omega_4$ (created through $\omega_1 \to \omega_3 \to \omega_4$), and so on. Then E_T would also contain the edge $\omega_1 \to \omega_{k-1}$, and thereby also the edge $\omega_k \to \omega_{k-1}$ (through the transitivity of $\omega_k \to \omega_1 \to \omega_{k-1}$). But since both $\omega_k \to \omega_{k-1} \in E_T$ and $\omega_{k-1} \to \omega_k \in E_T$, it can't be asymmetric.

E_T^- is minimal: Assume E_T^- was a feedback arc set, but not minimal. Then there would need to be another feedback arc set $E_T^{-'}$ so that $|E_T^{-'}| < |E_T^-|$. Then one can create $T' = (\Omega, E_T')$ from G by removing $E_T^{-'}$ from E and then completing the resulting directed acyclic graph to a consistent graph.

We know that $|E_T| = |E_T'| = \frac{|\Omega|(|\Omega|+1)}{2}$, since both T and T' are acyclic tournaments.

Then it is the case that $\mathrm{EGED}(G, T) > \mathrm{EGED}(G, T')$:

$$
\begin{aligned}
&\mathrm{EGED}(G,T) > \mathrm{EGED}(G,T') \\
&\Leftrightarrow |E \Delta E_T| > |E \Delta E'_T| \\
&\Leftrightarrow |E_T^- \uplus E_T^+| > |E_T^{-'} \uplus E_T^{+'}| \\
&\Leftrightarrow |E_T^-| + |E_T| - (|E| - |E_T^-|) > |E_T^{-'}| + |E'_T| - (|E| - |E_T^{-'}|) \\
&\Leftrightarrow |E_T^-| - |E| + |E_T^-| > |E_T^{-'}| - |E| + |E_T^{-'}| \\
&\Leftrightarrow 2 \cdot |E_T^-| > 2 \cdot |E_T^{-'}|
\end{aligned}
$$

So E_T^- must be minimal, since otherwise it is not a set of edges removed by `EGEDmin`. □

Using the fact that E_T^- is a minimum feedback arc set, and that all outputs of stepwise have the same edge-edit distance from the input, we can prove that all outputs of `stepwise` are contained in `EGEDmin`.

Lemma 3. $\forall G \in \mathfrak{P} : \texttt{stepwise}(G) \subseteq \texttt{EGEDmin}(G)$.

Proof. Let $S = (\Omega, E_S) \in \texttt{stepwise}(G)$ for any G, and let $T = (\Omega, E_T) \in \texttt{EGEDmin}(G)$. Let $E_S^- = E \backslash E_S$ be the minimum feedback arc set we remove from S to create G, and $E_S^+ = E_S \backslash E$ the edges we add to make G complete. We similarly define $E_T^- = E \backslash E_T$ and $E_T^+ = E_T \backslash E_T$.

We can now show that $\mathrm{EGED}(S,G) \leq \mathrm{EGED}(T,G)$: Assume that $\mathrm{EGED}(S,G) > \mathrm{EGED}(T,G)$. Per **Lemma** 2 E_T^- is a minimum feedback arc set, and so $|E_T^-| = |E_S^-|$. Furthermore, $|E_S| = |E_T|$, since they are both acyclic tournaments on Ω.

Then

$$
\begin{aligned}
&\mathrm{EGED}(G,S) = |E \Delta E_S| \\
&= |(E \backslash E_S^-) \uplus E_S^+| \\
&= (|E| - |E_S^-|) + |E_S^+| \\
&= (|E| - |E_S^-|) + |E_S| - (|E| - |E_S^-|) \\
&= (|E| - |E_T^-|) + |E_T| - (|E| - |E_T^-|) \\
&= (|E| - |E_T^-|) + |E_T^+| \\
&= |(E \backslash E_T^-) \uplus E_T^+| \\
&= |E \Delta E_T| = \mathrm{EGED}(G,T)
\end{aligned}
$$

So it can't be the case that $\mathrm{EGED}(S,G) > \mathrm{EGED}(T,G)$.

We can also show that $\mathrm{EGED}(S,G) \geq \mathrm{EGED}(T,G)$: Assume that $\mathrm{EGED}(S,G) < \mathrm{EGED}(T,G)$. Since both $S, T \in \mathfrak{C}_\Omega$, this contradicts the assumption that the output of `EGEDmin` has minimal distance. □

We now show that all outputs of `EGEDmin` are also outputs of `stepwise`.

Lemma 4. $\forall G \in \mathfrak{P} : \texttt{EGEDmin}(G) \subseteq \texttt{stepwise}(G)$.

Proof. Assume there exists a $G \in \mathfrak{P}_\Omega$ so that there exists a $T = (\Omega, E_T) \in \texttt{EGEDmin}(G)$ so that $T \notin \texttt{stepwise}(G)$.

Then, by **Lemma** 2, $E_T^- = E \backslash E_T$ is a minimum feedback arc set. Therefore, removing E_T^- from E results in a directed acyclic graph G_A which is an element of the intermediate set $\mathbf{A}$ of directed acyclic graphs in `stepwise`.

Let $E_T^+ = E_T \backslash E$. Assume E_T^+ was not a set of edges added to G_A in a topological sort.

Then let $\omega \in \Omega$ be the node in T that has no incoming edges. ω must also have had no incoming edges in G_A, since we only add edges to G_A to achieve T, and therefore has in-degree 0 in G_A, which means that ω must have been added first to some topological sort in $\mathbf{T}$ by `topological_sorts`.

One can now create T' and G'_A by removing ω and all edges from ω from T and G_A. Let the node in T' with no incoming edges be called ω'. Then in G_A the node ω' either had no incoming edges or one incoming edge from ω, since one can create T' from G_A by adding E_T^+ and then (potentially) removing the edge $\omega \to \omega'$. So in the graph G'_A with ω and all its outgoing edges removed from G_A, the node ω' has in-degree zero, and is therefore also selected as the first element in some topological sort of G'_A, to which ω is prepended after recursion. In the base case of a $T^\$$ with one element $\omega^\$$, this element $\omega^\$$ is the only element of $G_A^\$$ and also the only element of the topological sort of $G_A^\$$.

Therefore, by induction, given an acyclic tournament T and a set of edges $E_T^+ = E_T \backslash E$, this set E_T^+ must be the edges added by some topological sort of $G_A = (\Omega, E \backslash E_T^-)$.

□

This concludes the proof that both algorithms always have the same output.

Theorem 5. $\forall G \in \mathfrak{P} : \texttt{stepwise}(G) = \texttt{EGEDmin}(G)$.

Proof. By **Lemma** 3 $\texttt{stepwise}(G) \subseteq \texttt{EGEDmin}(G)$ and by **Lemma** 4 $\texttt{stepwise}(G) \supseteq \texttt{EGEDmin}(G)$, so the sets must be equal. □

3.2.3 Applying `HodgeRank`

Another option to resolve inconsistent preferences over deterministic options into consistent preferences is to apply the `HodgeRank` algorithm from Jiang et al. to an unweighted graph G [Jiang et al., 2011].

`HodgeRank` is described in further detail in section 4.3.1.

To apply `HodgeRank` to unweighted graphs one simply sets both weights of each edge to 1 ($w(e \in E) = 1$, $l(e \in E) = 1$).

Then, for a directed graph G, we can define an algorithm `HodgeResolve` that applies `HodgeRank` to G, and then converts the potential function p on Ω into an acyclic tournament. Here $\omega_1 \to \omega_2$ if and only if $p_{\omega_1} > p_{\omega_2}$.

One issue with `HodgeRank` is that the potentials of two options are sometimes equal to each other, which violates the criterion of asymmetry. There are two ways of dealing with this symmetry:

1. Keep the symmetric edges and accept that the output is a weak ordering, and modify the criteria to be applicable.

2. Resolve ties in the ordering by returning all topological sorts as a result. This has the disadvantage of potentially returning a set of results that is factorial in the size of Ω.

We decide to take the first option, to preserve the polynomial runtime of `HodgeRank`.

Algorithm 3 Computing `HodgeResolve`.

function HODGERESOLVE(G)
  For all $e \in E$, $w(e) \leftarrow 1, l(e) \leftarrow 1$
  $G_h \leftarrow (\Omega, E, w, l)$
  $p \leftarrow$ HODGERANK(G_h) $\triangleright$ p_ω is the potential that `HodgeRank` assigns to ω
  $E_r \leftarrow \emptyset$
  for $\omega_1, \omega_2 \in \Omega \times \Omega$ **do**
    if $p_{\omega_1} \geq p_{\omega_2}$ **then**
      $E_r \leftarrow E_r \cup \{(\omega_1, \omega_2)\}$
    end if
  end for
  $G_r \leftarrow (\Omega, E_r)$
  return G_r
end function

3.3 Criteria

Given the algorithms outlined above, one might want to compare them according to different criteria, similar to the method of evaluating voting methods in social choice theory by some criteria [Austen-Smith and Banks, 2000, ch. 2], such as the Condorcet criterion [McLean, 1990] or manipulability [Gibbard, 1973]. For this purpose, we examine the algorithms with regards to the computational complexity, size of output, and two additional criteria.

3.3.1 Surjectivity and Identity

A fairly intuitive criterion is that for a given f, and for every $C \in \mathfrak{C}_\Omega$, there should be a $G \in \mathfrak{P}_\Omega$ so that $C \in f(G)$ (**Surjectivity**). This condition is implied by the stronger condition of f being the identity function for already consistent graphs: $\forall C \in \mathfrak{C}_\Omega : f(C) = \{C\}$ (**Identity**).

Minimizing Graph-Edit Distance `EGEDmin` fulfills both conditions: C trivially has the smallest graph-edit distance to itself (namely zero), and is unique in that regard.

Applying `HodgeRank` [Jiang et al., 2011] state that for complete graphs, computing the potential function of a graph G via `HodgeRank` on the nodes is equivalent to minimizing the squared distance between the edge weights of G and the edge-weights induced by the potential function. If G already

is consistent, the resulting potential function simply re-creates G, since their distance is 0. So `HodgeResolve` maps every consistent graph to itself, and therefore fulfills **Identity** and therefore also **Surjectivity**.

3.3.2 Polynomial Time Complexity

Ideally, a method for resolving inconsistent graphs into consistent graphs would be efficiently computable.

Minimizing Graph-Edit Distance However, the method that attempts to find consistent graphs by minimizing edge-graph-edit distance fails this criterion.

Finding all acyclic tournaments with the smallest edit-distance to a given directed graph is NP-hard. This can be shown by a reduction to Slater's problem. Slater's problem is the problem of, given any tournament T, finding a linear order T_L (an acyclic tournament, also called a *Slater order*) that has the smallest distance to T, where the distance between two tournaments T_1, T_2 is the number of edges that have to be flipped in T_1 to create T_2. Slater's problem (and a number of related problems, such as finding *all* acyclic tournaments with the smallest distance to a given tournament) is known to be NP-hard [Hudry, 2010].

Theorem 6. Finding the set of acyclic tournaments with smallest edge-graph-edit distance to a given graph G is NP-hard.

Proof. Reduction to finding all Slater orders with the smallest distance to a given tournament T.

Assume we know an algorithm `A` to compute $f_{\mathrm{EGED}}(G)$ efficiently, that is, to compute the set of all acyclic tournaments with the minimal graph-edit distance to a given directed graph G.

Then one could solve Slater's problem efficiently: For any given tournament T, `A` would compute a set $\mathbf{C}_T$ of acyclic tournaments which have the same minimal graph-edit distance $2k$ to T, and the distance is divisible by two because by editing a tournament T into a tournament T', edges can only be flipped, which engenders two edge operations (removing an edge and then adding a new one). Then that set would also be the set of Slater orders of T (with distance k), a solution to (P_3) from [Hudry, 2010], which is known to be NP-hard. □

Similarly, finding only *one* element from $f_{\mathrm{EGED}}(G)$ is also NP-hard, by reducing it to P_2 from [Hudry, 2010] ("PROBLEM P_2. Given a tournament T, compute a Slater order $O^*(T)$ of T.")

Applying `HodgeRank` [Jiang et al., 2011] state that computing the potential function of a graph $G = (\Omega, E)$ is equivalent to solving a $n \times n$ least-squares problem ($n = |\Omega|$), which requires $\mathcal{O}(n^3)$ time. `HodgeResolve` executes `HodgeRank` and then iterates through all possible edges of G, which takes at most $\mathcal{O}(n^2)$ time, so the time complexity of `HodgeResolve` is also $\mathcal{O}(n^3)$.

3.3.3 Uniqueness

It would be desirable if one could guarantee that the function f that resolves inconsistent graphs returns a single consistent graph for each inconsistent graph, that is $\forall G \in \mathfrak{P}_\Omega : |f(G)| = 1$.

Minimizing Graph-Edit Distance However, `EGEDmin` does not fulfill this criterion.

Theorem 7. For a graph G_e with no edges and n vertices Ω, every acyclic tournament with the same set vertices has the same graph-edit distance to G_e. Therefore, $|\texttt{EGEDmin}(G_e)| = n!$, which is not unique.

Proof. Let T be any acyclic tournament with vertices Ω. Then T has $\binom{n}{2}$ edges. Since G_e has no edges, one can edit G_e to be T simply by adding all edges of T to G_e. This is sufficient and necessary for turning G_e into T. Since this holds for any tournament T, the graph-edit distance from G_e to any acyclic tournament is the same, namely $\binom{n}{2}$. So $|\texttt{EGEDmin}(G_e)| = |\mathfrak{C}_\Omega| = n!$. □

Applying `HodgeRank` If one allows for the output of `HodgeResolve` to be a weak ordering, then `HodgeResolve` has a unique output, since assigning each vertex a real-valued potential $p : \Omega \to \mathbb{R}$ and then ordering vertices by that potential creates a weak ordering W.

However, if one demands that the output of `HodgeResolve` be a total order then the output is dependent on the method of achieving that total order. If one generates the total orders by generating all acyclic tournaments with vertices Ω that are subgraphs of W, the output is no longer unique: In the worst case $G = (\Omega, \emptyset)$, which results in `HodgeRank` assigning a potential of 0 to every node, and `HodgeResolve` putting every vertex in the same equivalence class in the weak ordering. As a graph this is the complete directed graph on Ω, which contains all acyclic tournaments on Ω as subgraphs. Then there are $|\Omega|!$ acyclic tournaments generated from this weak ordering, since all acyclic tournaments are equally compatible with the weak ordering.

Further Considerations Violating **Uniqueness** appears to have consequences for decision-making: If we want to use the output of f for prioritising which actions to take to achieve high-ranking options, having more than one result leaves it unclear which options to prioritize (since there will be two $\omega_1, \omega_2 \in \Omega$ that are ranked differently by different elements of the set of results).

However, results from two different fields apply to this case.

- **Social Choice Theory**: Since all elements of $\mathbf{C}_G = f(G)$ are complete, transitive, and asymmetric, one can apply the large toolbox of methods and results from social choice theory, outlined in e.g. [Austen-Smith and Banks, 2000] to elements from $\mathbf{C}_G$ by treating them as individual preferences in a preference profile by applying a social welfare function in sense of Arrow to it [Gaertner, 2009, ch.2]. Some impossibility results such as Arrow's theorem [Arrow,

1950] still apply, but at least results about tactical voting (such as the Gibbard-Satterthwaite theorem [Gibbard, 1973]) are irrelevant in this case, since the inconsistent preference does not "control" outputs of f, and there are no reasons for manipulation.

- **Moral Uncertainty**: [MacAskill et al., 2020, ch. 2] outline how to make decisions given multiple ethical theories and credences on those ethical theories, using the so-called Maximum Expected Choiceworthiness rule. In the case of ordinal preferences, they use the Borda count [McLean, 1990] for establishing cardinal values for options.

3.3.4 Resolution to Polynomially Many Preferences

If **uniqueness** can't be fulfilled (perhaps because the given graph G is under-determined), a weaker criterion is that the number of consistent graphs corresponding to G is polynomial in the size of Ω ($\forall G \in \mathfrak{P}_\Omega : |f(G)| \leq p(|\Omega|)$, where $p(n)$ is some polynomial in n).

Minimizing Graph-Edit Distance However, as proven in **Theorem** 7 above, this criterion is not fulfilled for `EGEDmin`, instead in the worst case the number is factorial in the size of Ω.

We decided to also investigate the number of results for `EGEDmin` for small graphs. For this purpose, we generated all directed graphs with five nodes or less and computed `EGEDmin`(G).

Definition 3.1. Let G be any directed graph. Then the **confusion** of G is the number of acyclic tournaments with the smallest edge-graph-edit distance to G, that is the confusion $c : \mathfrak{P} \to \mathbb{N}^+$ of G is $c(G) = |\texttt{EGEDmin}(G)|$. The set of graphs with n vertices and confusion c shall be denoted $\mathbf{G}_{n,c}$.

The term "confusion" was chosen to emphasize that graphs with a lower such number have fewer consistent versions. An acyclic tournament has minimal confusion (namely 1, where the output of `EGEDmin` is simply itself). G_e from **Theorem** 7 has maximal confusion, namely $n!$.

A natural question to ask is whether, with bigger graphs, the average confusion converges to a certain value or diverges, or shows no clear behavior. We generated all directed graphs with up to 5 vertices and computed their confusion.

$|\mathbf{G}_{n,1}|$ is the number of all graphs with n vertices and confusion 1, $|\mathbf{G}_{n,1}|/n!$ is the same number but up to isomorphism of the graphs. $|\mathbf{G}_{n,n!}|$ is the number of of graphs with n vertices and maximal confusion.

For some given set of directed graphs $\mathfrak{P}_n$, not all numbers between 1 and $n!$ can be confusions. There are, for example, no graphs of size 3 with confusion 4 (or 5).

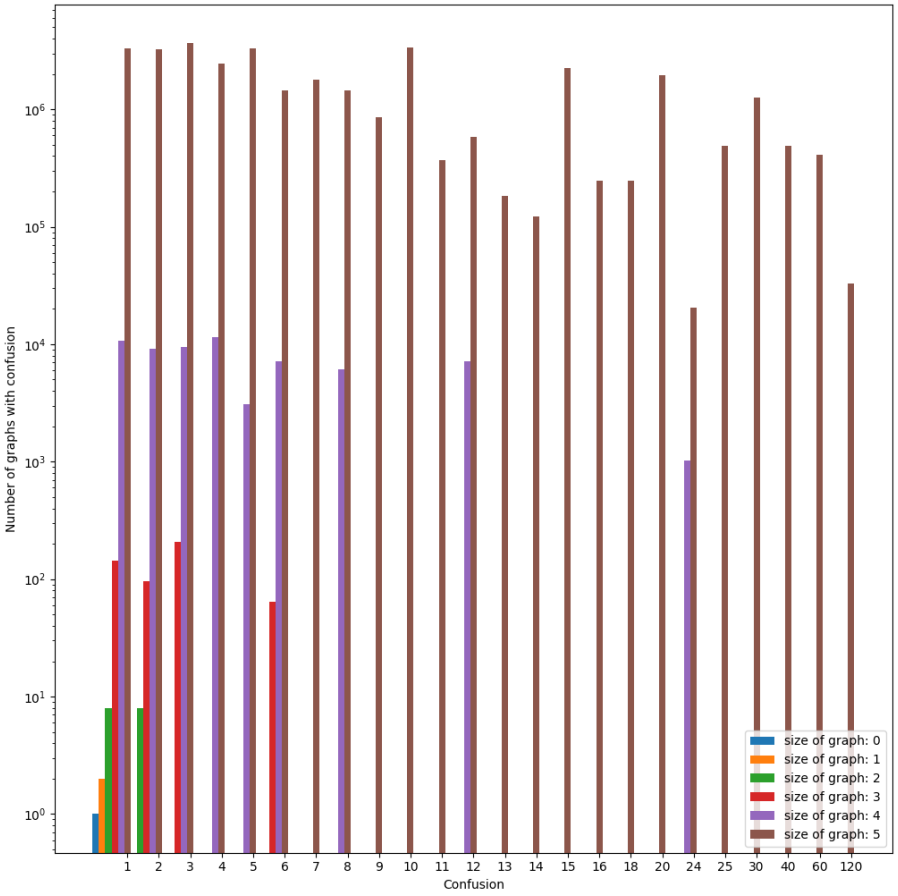


Figure 1: Number of graphs by number of nodes and confusion.

n	Samples	Average confusion	$\lvert\mathbf{G}_{n,1}\rvert$	$\lvert\mathbf{G}_{n,1}\rvert/n!$	$\lvert\mathbf{G}_{n,n!}\rvert$
0	1	1	100% (1)	1	2^0
1	2	1	100% (2)	2	2^1
2	16	1.5	50% (8)	4	2^3
3	512	2.625	28.125% (144)	24	2^6
4	65536	≈ 4.91	$\approx 16.4\%$ (10752)	448	2^{10}
5	33554432	≈ 9.43	$\approx 9.853\%$ (3306240)	27552	2^{15}
6	90927	≈ 18.138	$\approx 6.225\%$ (5660)	?[a]	?
7	1580	≈ 36.412	$\approx 3.608\%$ (57)	?	?

[a] Sample size too small.

Interestingly, neither $\lvert\mathbf{G}_{n,1}\rvert$ nor $\lvert\mathbf{G}_{n,1}\rvert/n!$ are known integer sequences: a search on the OEIS and via SuperSeeker [Sloane et al., 2003] yield no matching results.

Conjecture 1. The average confusion of all directed graphs with size n diverges to infinity:

$$\lim_{n\to\infty} \frac{1}{2^{n^2}} \sum_{i=1}^{n!} |\mathbf{G}_{n,i}| \cdot i = \infty$$

Proposition 2. $|\mathbf{G}_{n,1}|$ is always divisible by 2^n.

Proof. This is an artifact of including graphs with reflexive edges in the set of graphs tested. Let G be a graph with confusion k and no reflexive edges.

Let now $\mathbf{G}^\circ$ be the set of all graphs that are G with all possible reflexive edges added. Every element in $\mathbf{G}^\circ$ also has confusion k: all reflexive edges must be removed to create a consistent preference, yielding G, and there are k unique acyclic tournaments that has the smallest edge-graph-edit distance to G.

$|\mathbf{G}^\circ \cup \{G\}| = 2^n$: for each node, the presence of a reflexive edge on that node can be described by one bit of information. □

Dividing $\mathbf{G}_{n,1}$ by both $n!$ and 2^n yields the sequence $1, 1, 1, 3, 28, 861$, which also doesn't occur in the OEIS, and also can't be found using SuperSeeker.

Applying `HodgeRank` As seen in the case of **Uniqueness**, this depends on whether one demands the output of `HodgeResolve` to be a total order: If a weak ordering is allowed, the output of `HodgeResolve` is always a single graph, so the output size is polynomial, but if we demand a total order as an output the output size can be factorial in the number of nodes.

3.3.5 Preservation of Consistent Subgraphs

Definition 3.2. For a given $G = (\Omega, E_P) \in \mathfrak{P}_\Omega$, a subgraph $S_G = (\Xi, E)$ of G (with $\Xi \subseteq \Omega$, and the set of edges E of S_G being a subset of E_P) is an **inclusion-maximal consistent subgraph** of G if and only if:

- S_G is a consistent graph (equivalently an acyclic tournament)[2].
- S_G inherits all available edges from G, that is if there are two $\xi_1, \xi_2 \in \Xi$ and $(\xi_1, \xi_2) \in E_G$ then $(\xi_1, \xi_2) \in E$ as well.
- S_G is inclusion-maximal, that is, there exists no $\omega \in \Omega \backslash \Xi$ so that adding ω and its edges adjacent to all $\xi \in \Xi$ to S_G is still a consistent graph.

Definition 3.3. Let $\mathcal{S}_G$ be the set of all inclusion-maximal consistent subgraphs of G and let $f : \mathfrak{P} \to \mathcal{P}(\mathfrak{C})$ be a function that turns any G into a set $\mathbf{C}_G = f(G)$ of consistent graphs. Then f fulfills **Preservation of Consistent Subgraphs** if and only if every element of $\mathcal{S}_G$ is a subgraph of at least one $\mathbf{C}_G$, that is

$$\forall S \in \mathcal{S}_G : \exists C \in \mathbf{C}_G : V_S \subseteq V_C \wedge E_S \subseteq E_C$$

[2]Without reflexive edges $(\xi, \xi) \in E$.

This criterion is quite strong, as we will show. Its intuitive appeal can be explained as follows: Assume one has overall inconsistent preferences, but there is some subset of objects one has consistent preferences over, e.g. an agent has consistent preferences over all fruit and consistent preferences over dairy products, but inconsistent preferences over food in general. Then a method for resolving those inconsistent preferences into consistent ones should "preserve" those consistent preferences over subsets of options a non-zero amount — after becoming consistent the agent still has the same preferences over fruit as before.

Furthermore, one can show that there are graphs with an exponential number of inclusion-maximal consistent subgraphs in the number of nodes.

Lemma 8. Let $G \in \mathfrak{P}_n$ be an arbitrary directed graph with n nodes, and let $\mathcal{S}_G$ be the set of inclusion-maximal consistent subgraphs of G. Then there exists no polynomial p so so that $\forall G \in \mathfrak{P}_n : |\mathcal{S}_G| \leq p(n)$.

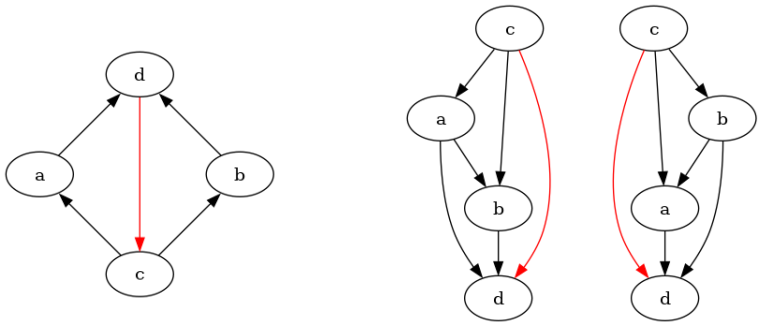
Proof. Moon and Moser describe how to construct an undirected graph $G_n = (V_G, E_G)$ with n vertices and $3^{\frac{n}{3}}$ inclusion-maximal cliques [Moon and Moser, 1965]. Then one can construct a directed graph $P_n = (V_P, E_P)$ with $3^{\frac{n}{3}} \approx 1.4422^n$ inclusion-maximal consistent subgraphs from G_n, which grows faster than any polynomial. First, P_n receives the same vertices as G_n. Then, every $v \in V$ is assigned a unique number $j(v) : V \to \mathbb{N}$, and for each $\{u, v\} \in E_G$, E_P contains (u, v) iff $j(u) > j(v)$, and (v, u) iff $j(v) > j(u)$. Now, if a subgraph S_G of G_n with vertices V_S is a maximal clique, then a subgraph S_P of P_n with vertices V_S is an inclusion-maximal consistent subgraph in P_n:

1. S_P is complete, because for every $\{u, v\}$ in S_G, either (u, v) or (v, u) exists in S_P.
2. S_P is transitive. For any three vertices $\{u, v, w\}$ in S_G, S_G contains the edges $\{\{u, v\}, \{v, w\}, \{u, w\}\}$ (since it is a clique). Then, without loss of generality, assume that $j(u) > j(v) > j(w)$. Then $(u, w) \in E_P$. Therefore S_P contains the edges $\{(u, v), (v, w), (u, w)\}$.
3. S_P is asymmetric, because for any edge $\{u, v\}$ in S_G $j(u) > j(v)$ and $j(v) > j(u)$ can't be true at the same time (since j assigns each vertex a unique natural number). So S_P can only contain either (u, v) or (v, u).
4. S_P is inclusion-maximal. If S_P were not inclusion-maximal, there'd exist a vertex u so that every vertex v of S_P had an edge with u. But since the procedure of constructing P_n above did not add any edges, that would mean that S_G was not a maximal clique.

□

Minimizing Graph-Edit Distance `EGEDmin` violates this criterion, which can be easily shown by a counterexample in Figure 2:

Example 1. The graph G_c above contains a subgraph $S_{cd} = (\{c, d\}, \{(c, d)\})$ that is also an inclusion-maximal acyclic tournament in G_c. The two acyclic tournaments with the lowest graph-edit distance (namely 3: reversing the edge $d \to c$ (2 operations) and adding an edge between a and



(a) Counterexample

(b) Counterexample resolved versions

Figure 2: G_c on the left above is resolved into two acyclic tournaments, none of which contain the edge $d \to c$.

b) to G_c are shown in Figure 2. Note that none of them contain S_{cd} as a subgraph.

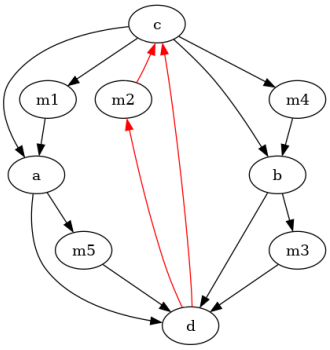


Figure 3: The counter-example for $n = 3$.

This counter-example can be generalized so that inclusion-maximal consistent subgraphs with an arbitrary number of nodes n get reversed: Each edge $\omega_1 \to \omega_2$ of G_c gets replaced by an acyclic tournament $T_i = (\Xi_i, E_i)$ with $n-2$ vertices, so that there is an edge from ω_1 to every $\xi_i \in \Xi_i$ and an edge from every $\xi_i \in \Xi_i$ to ω_2. The graph on the left has confusion 40, and the subgraph emphasized in red is preserved in none of the outputs of `EGEDmin`.

We also investigated the number of inclusion-maximal consistent subgraphs preserved by `EGEDmin`. We again did this by analyzing the outputs of `EGEDmin` for all graphs with five nodes or less, and some graphs with six or seven nodes.

Let IMCS : $\mathfrak{P}_n \to \mathfrak{P}_{1..n}$ be a function that returns the **i**nclusion-**m**aximal **c**onsistent **s**ubgraphs for a given graph.

Given a directed graph G, let $\mathcal{S}$ be the set of inclusion-maximal consistent subgraphs of G. One can now ask: For a given inclusion-maximal consistent subgraph, how often did that subgraph occur in the set of outputs $\texttt{EGEDmin}(G)$? Let $\mathrm{RSP}(S \in \mathcal{S}, G)$ be the **r**atio of **s**ubgraph **p**reservation:

$$\mathrm{RSP}_{\texttt{EGEDmin}}(S \in \mathcal{S}, G) = \frac{|\{R \in \texttt{EGEDmin}(G) | S \text{ subgraph of } R\}|}{|\texttt{EGEDmin}(G)|}$$

As we saw above, there are graphs with inclusion-maximal consistent subgraphs S so that $\mathrm{RSP}(S) = 0$.

One can then use RSP to define a metric that tells us, for a given graph, how often inclusion-maximal consistent subgraphs were preserved one average (**a**verage **m**aximal **s**ubgraph **p**reservation):

$$\mathrm{AMSP}(G) = \frac{1}{|\mathrm{IMCS}(G)|} \sum_{S \in \mathrm{IMCS}(G)} \mathrm{RSP}_{\texttt{EGEDmin}}(S)$$

A higher number for AMSP is better: It mean that more inclusion-maximal consistent subgraphs get preserved more often by the method for resolving inconsistent preferences.

n	Samples	Avg $\lvert\mathrm{IMCS}(G)\rvert$	Avg $\mathrm{AMSP}(G)$	Min $\mathrm{AMSP}(G)$	Graphs with $\mathrm{AMSP}(G) = 1$
0	1	1	1	1	1 (100%)
1	2	1	1	1	2 (100%)
2	16	1.125	1	1	16 (100%)
3	512	≈ 1.32	≈ 0.995	2/3	496 ($\approx 98.4\%$)
4	65536	≈ 1.568	≈ 0.984	0	57728 ($\approx 94.4\%$)
5	33554432	≈ 1.864	≈ 0.969	0	7803263 ($\approx 80.1\%$)
6	90927	≈ 2.207	≈ 0.95	0	72209 ($\approx 79.4\%$)
7	1580	≈ 2.618	≈ 0.932	0	1095 ($\approx 69.3\%$)

One can see that the average number of inclusion-maximal consistent subgraphs increases, albeit initially slowly. The number of times that maximal consistent subgraphs are preserved (Avg $\mathrm{AMSP}(G)$) starts dropping, though the shrinking behavior isn't clear from the limited amount of data. The number of graphs in which all inclusion-maximal consistent subgraphs are preserved by `EGEDmin` shrinks even more quickly, indicating that preserving all consistent subgraphs is a property that is difficult to fulfill.

Only for small graphs (up to 3 vertices) it is guaranteed that at least one inclusion-maximal consistent subgraph occurs in the output of `EGEDmin`.

So we can pose some conjectures indicated by the datapoints observed above:

Conjecture 2. In the limit of graph size, on average `EGEDmin` preserves almost none of the inclusion-maximal consistent subgraphs:

$$\lim_{n \to \infty} \frac{1}{|\mathfrak{P}_n|} \sum_{G \in \mathfrak{P}_n} \mathrm{AMSP}(G) = 0$$

Conjecture 3. For graphs with > 7 nodes it remains the case that there are graphs for which the smallest number of inclusion-maximal consistent subgraphs preserved by `EGEDmin` is zero:

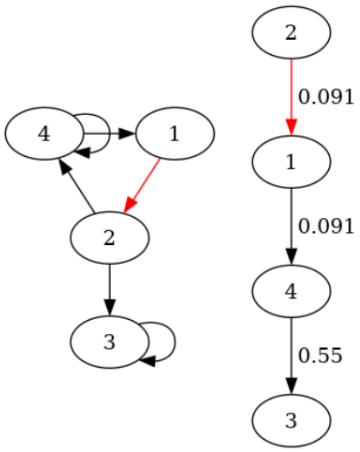
$$\lim_{n \to \infty} \min_{G \in \mathfrak{P}_n} \mathrm{AMSP}(G) = 0$$

Conjecture 4. In the limit of number of nodes in a graph, for almost no graphs does `EGEDmin` preserve all inclusion-maximal consistent subgraphs.

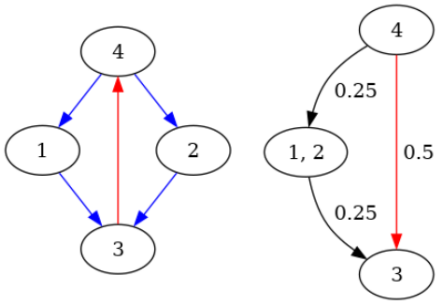
$$\lim_{n\to\infty} \frac{1}{|\mathfrak{P}_n|} |\{G \in \mathfrak{P}_n | \mathrm{AMSP}(G) = 1\}| = 0$$

Applying `HodgeRank` If the output of `HodgeResolve` is allowed to be a weak ordering, then the original definition of **Preservation of Consistent Subgraphs** does not apply, as it presumes a mapping f from $\mathfrak{P}$ to $\mathfrak{C}$. However, the definition can easily be transferred by defining f as a function from directed graphs to weakly consistent graphs, that is $f : \mathfrak{P}_\Omega \to \mathfrak{W}_\Omega$. The definition of **Preservation of Consistent Subgraphs** stays otherwise unchanged[3].

`HodgeResolve` does not fulfill **Preservation of Consistent Subgraphs**. Figure 4 shows two graphs (both on the left in their respective subfigures). For the graph in the left subfigure no inclusion-maximal consistent subgraphs are preserved, for the right one all but one inclusion-maximal consistent subgraphs are preserved.



(a) $1 \to 2$ is the only consistent subgraph, but it gets reversed.



(b) Each edge is an inclusion-maximal consistent subgraph, and only the edge $3 \to 4$ gets reversed. 1 and 2 in the result have the same potential.

Figure 4: On the left side a graph with 1 inclusion-maximal consistent subgraph and its resolution through `HodgeResolve`, and on the right side a graph with several inclusion-maximal consistent subgraphs and and its resolution through `HodgeResolve`. The labels at the edges are the gradients that `HodgeRank` has computed.

[3]This definition allows for for there to be graph $G = (\Omega, V)$, a consistent subgraph S_G of G and resolved weakly consistent graph $W = (\Omega, E_W) \in f(G)$ such that there exist nodes $\omega_1, \omega_2 \in \Omega$ in S_G which are not *strictly* ordered in W, that is both $\omega_1 \to \omega_2 \in E_W$ and $\omega_2 \to \omega_1 \in E_W$. It is possible to define a stronger criterion, **Strict Preservation of Consistent Subgraphs**, which requires that for such ω_1, ω_2 *only* the edge $\omega_1 \to \omega_2$ being present in E_W, but we will not work with that definition here.

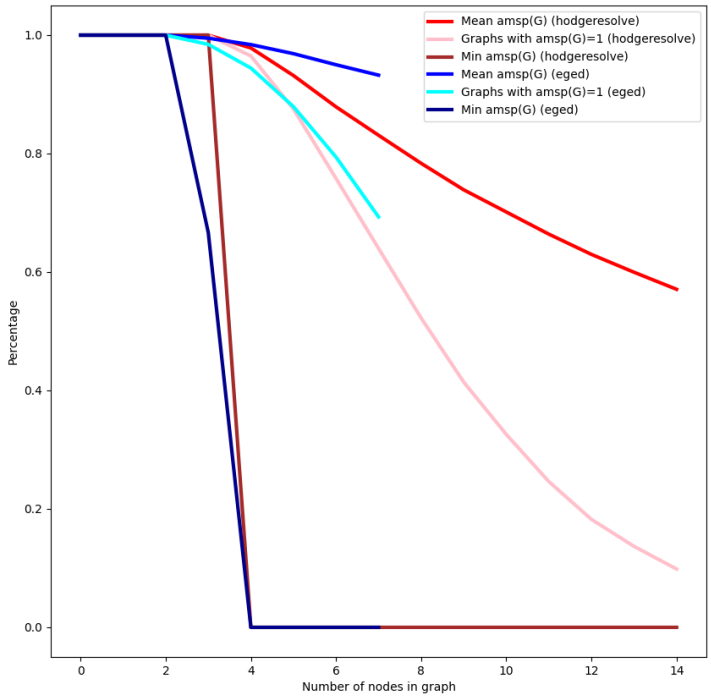


Figure 5: Comparing `EGEDmin` and `HodgeResolve` at how well they perform on various metrics of preserving inclusion-maximal consistent subgraphs.

n	Samplesize	Avg $\lvert\text{IMCS}(G)\rvert$	Avg AMSP(G)	Min AMSP(G)	Graphs with AMSP(G) $= 1$
0	1	1	1	1	1 (100%)
1	2	1	1	1	2 (100%)
2	16	1.125	1	1	16 (100%)
3	512	≈ 1.32	≈ 1	1	512 (100%)
4	65536	≈ 1.568	≈ 0.978	0	63232 ($\approx 96.5\%$)
5	33554432	≈ 1.864	≈ 0.932	0	29373632 ($\approx 87.5\%$)
6	65536	≈ 2.209	≈ 0.879	0	49680 ($\approx 75.8\%$)
7	65536	≈ 2.612	≈ 0.831	0	41926 ($\approx 63.9\%$)
8	65536	≈ 3.064	≈ 0.783	0	34227 ($\approx 52.2\%$)
9	65536	≈ 3.567	≈ 0.738	0	27138 ($\approx 41.4\%$)
10	65536	≈ 4.13	≈ 0.701	0	21349 ($\approx 32.6\%$)

With this data, Figure 5 plots how well `EGEDmin` and `HodgeResolve`

perform at preserving inclusion-maximal consistent subgraphs.

One can see that on average, `EGEDmin` preserves inclusion-maximal consistent subgraphs more often, and may also retain all inclusion-maximal consistent subgraphs more often (although the low sample sizes for graphs with six and seven nodes makes this difficult to conclude without doubt).

3.3.6 Preservation of Completely Dominating and Dominated Set

Inclusion-maximal consistent subgraphs are a way of formalizing what it means for a preference to be *locally* consistent: there is some subset of Ω so that the preferences are not "confused" about this subset. One can also try to find a corresponding condition that would make a statement about *global* consistency. Voting theory offers some inspiration here: the **minimal undominated set** (also **Condorcet set**) [Miller, 1977] is defined for every tournament $T = (V_T, E_T)$ as a set of vertices $V^* \subseteq V_T$ so that (1) there is no edge from $V_T \backslash V^*$ to V^* and (2) there is no proper subset of V^* that meets (1).

One can create a related (but weaker) definition for directed graphs:

Definition 3.4. For a given $G = (\Omega, E)$, let Σ_1, Σ_2 be non-empty sets of vertices of G such that $\Sigma_1 \uplus \Sigma_2 = \Omega$. Then Σ_1 is a **completely dominating set** and Σ_2 is a **completely dominated set** if and only if $\forall \sigma_1 \in \Sigma_1, \sigma_2 \in \Sigma_2 : (\sigma_1, \sigma_2) \in E \wedge (\sigma_2, \sigma_1) \notin E$.

This means that all elements in a completely dominating set are strictly preferred to all elements in a completely dominated set—there is a subset of options that are clearly better than all other options.

A change from the Condorcet set is that we don't demand the completely dominating set to be minimal (which would always make the empty set the completely dominating set). Additionally, the completely dominating set is not unique: In an acyclic tournament, the i greatest elements form a dominating set for $1 \leq i \leq |\Omega|$.

A completely dominating set then represents a global consistency in the preference: within Σ_1 and Σ_2 we are unsure about our preference, but we know that any element of Σ_1 is better than any element of Σ_2.

Definition 3.5. A function $f : \mathfrak{P} \to \mathcal{P}(\mathfrak{C})$ fulfills **Preservation of Complete Domination** if and only if for any directed graph G with a completely dominating set Σ_1 and a completely dominated set Σ_2 it holds that $\forall C \in f(G)$ Σ_1 is a completely dominating set of Σ_2 in C.

Proposition 3. Let f be a function that fulfills **Preservation of Complete Domination**. If for a graph $G = (\Omega, E)$ there are n sets of vertices $\Sigma_1, \ldots, \Sigma_n$ so that $\biguplus_{i=1}^{n} \Sigma_i = \Omega$ and

$$\forall c \in 1 \ldots n : \biguplus_{i=1}^{c} \Sigma_i \text{ completely dominates } \biguplus_{j=c+1}^{n} \Sigma_j$$

then for any $C = (\Omega, E_C) \in f(G)$ it holds that $\forall 1 < j < k < n$: $\forall \sigma_j \in \Sigma_j, \sigma_k \in \Sigma_k : (\sigma_j, \sigma_k) \in E_C \wedge (\sigma_k, \sigma_j) \notin E_C$ (or, less formally, every element from a subset of a completely dominating set is strictly preferred over any element from a subset of a completely dominated set).

Proof. Fix $1 < j < k < n$. Let $\Sigma_l = \biguplus_{i=1}^{k-1} \Sigma_i$ and $\Sigma_r = \biguplus_{i=k}^{n} \Sigma_i$. Then Σ_l dominates Σ_r in G, and by assumption also in $C \in f(G)$. Since $\Sigma_j \subsetneq \Sigma_l$, $\Sigma_k \subsetneq \Sigma_r$, it holds that $\forall \sigma_j \in \Sigma_j, \sigma_k \in \Sigma_k : \sigma_j \to \sigma_k \in E_C \wedge \sigma_k \to \sigma_j \notin E_C$. So Σ_j now completely dominates Σ_k in C. □

Remark 1. Sets of such $\Sigma_1, \ldots, \Sigma_n$ such that there is a relationship of complete domination between any two of them are quite similar to graph quotients, but is somewhat stricter (demanding that each $\sigma_i \in \Sigma_i$ be preferred to each other $\sigma_j \in \Sigma_j$).

Remark 2. Preservation of complete dominance implies some other criteria: If there is a consistent subgraph which is a completely dominating set, then it will comprise the "greatest" subgraph in the resolved preference, with the greatest element in G also being the greatest element in C. The same holds for the a completely dominated consistent subgraph, which stays at the bottom.

Minimizing Graph-Edit Distance

Theorem 9. `EGEDmin` fulfills **Preservation of Complete Domination**.

Proof. Let $C = (\Omega, E_C) \in \texttt{EGEDmin}(G)$ be a consistent graph for a directed graph G, where G has a completely dominating set Σ_1 and a completely dominated set Σ_2. Assume C does not have the completely dominating set Σ_1, and let $n = \texttt{EGED}(G, C)$. Then there must be a "highest" or "largest" $\sigma_2 \in \Sigma_2$ in C (one for which there is no other $\sigma_2' \in \Sigma_2$ so that $\sigma_2' \to \sigma_2$ is an edge in C). There must also be a "highest" or "largest" $\sigma_1^* \in \Sigma_1$ so that $\sigma_2 \to \sigma_1^*$ is an edge in C.

Let there be $m \geq 0$ elements of Σ_1 "between" σ_2 and σ_1^*, that is $|\Sigma_2^* = \{\sigma_2^* | \sigma_2 \to \sigma_2^* \in E_C \ \wedge \sigma_2^* \to \sigma_1^* \in E_C\}| = m$. One can now create a C' from C so that $\texttt{EGED}(G, C') = n - 2(m+1)$ by moving σ_1^* into the position directly above σ_2 by reversing the edges $\sigma_2 \to \sigma_1^*$ and $\sigma_2^* \to \sigma_1^*$ for all $\sigma_2^* \in \Sigma_2^*$. C' now contains some edges from G that need to be reversed to create C: $\sigma_1^* \to \sigma_2$ and $\{\sigma_1^* \to \sigma_2^* | \sigma_2^* \in \Sigma_2^*\}$ are already edges in G, and because edge reversals have weight 2 (deleting and then adding one edge), this saves $2(m+1)$ edge operations. Furthermore all other edge operations to minimally achieve C from G can be held constant to create C', so that the graph-edit distance is not changed otherwise. C' is now an output with a smaller edge-graph-edit distance from G than C. Thus all other outputs $\mathbf{C} = \texttt{EGEDmin}(G)$ must also have a smaller edge-graph-edit distance than C has to G.

If C' does not have the same completely dominating set Σ_1 that G has, one can create a new graph C'' by finding a new "highest" σ_2 and corresponding σ_1^* and switching them. This C'' again has shorter edge-graph-edit distance.

This process can be repeated as long as Σ_1 is not a completely dominating set in the consistent graph, monotonically decreasing the edge-graph-edit distance, until no furtehr such modifications can be found.

The final consistent graph resulting from this process contains Σ_1 as a completely dominating set: Every $\sigma_1 \in \Sigma_1$ has a unidirectional edge to every $\sigma_2 \in \Sigma_2$. □

Applying `HodgeRank`

Conjecture 5. `HodgeResolve`(G) fulfills **Preservation of Complete Domination** for every $G \in \mathfrak{P}$.

This conjecture holds for all directed graphs with 5 nodes or less, by computational experiment, and for random samples of graphs (2^{16} graphs generated for each number of nodes, using the Erdős-Rényi model [Erdős et al., 1960] with the probability $\frac{1}{2}$ of edge creation) with up to 13 nodes.

3.4 Summary

We can now summarize how well the two algorithms fulfill the different criteria:

Criterion	`EGEDmin`	`HodgeResolve`
Surjectivity	✓	✓
Identity	✓	✓
Worst-case computational complexity	NP-hard	$\mathcal{O}(n^3)$
Uniqueness	✗	$\sim$ [d]
Polynomial output size	✗	$\sim$ [d]
Preservation of consistent subgraphs	✗	✗
Preservation of complete domination	✓	?

[d] Only if the output is allowed to be a weak ordering.

3.5 Impossibilities

Some of the criteria listed in Section 3.3 are incompatible with each other.

3.5.1 Resolution to Polynomially Many Preferences and Preservation of Consistent Subgraphs are Incompatible

It is not possible to have an algorithm that retains every maximal consistent subgraph at least once in the set of outputs and has only polynomially many outputs.

Theorem 10. Let $f : \mathfrak{P} \to \mathcal{P}(\mathfrak{C})$ be a function for resolving inconsistent graphs that fulfills **Preservation of Consistent Subgraphs** for all graphs $\mathfrak{P}$. Then there exists no polynomial p so that for all directed graphs $\mathfrak{P}_n$ of size n it holds that $\forall P_n \in \mathfrak{P}_n : |f(P_n)| \leq p(n)$.

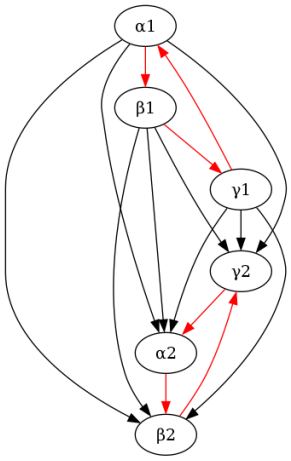


Figure 6: The graph E_2.

We show this with a counterexample.

Definition 3.6. Let V denote a directed graph with with three vertices α, β, γ and three edges $\alpha \to \beta, \beta \to \gamma, \gamma \to \alpha$. Let now denote E_n be a graph that is constructed out of n copies of V, "stacked" on top of each other. More formally, let the vertices of E_n be the set $\{\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \gamma_1, \ldots, \gamma_n\}$ so that $\alpha_i, \beta_i, \gamma_i$ are the vertices of the graph V_i, and the edges of E_n are the edges of each V_i and the edges $\{(u_i, v_j) | i > j \wedge u, v \in \{\alpha, \beta, \gamma\}\}$.

Lemma 11. Every inclusion-maximal consistent subgraph V of E_n contains exactly one edge from each $V_i \in \{V_1, \ldots, V_n\}$.

Proof. Assume S is a subgraph of E_n, and there exists (without loss of generality) a V_i so that $S \cap V_i$ has two edges $\alpha_i \to \beta_i$ and $\beta_i \to \gamma_i$. Since S is stipulated to be consistent, due to the transitivity requirement it must also contain the edge $\alpha_i \to \gamma_i$. But then S would no longer be a subgraph of E_n, since $\alpha_i \to \gamma_i$ is not an edge in V_i. If $S \cap V_i$ has three edges, S must be inconsistent, since transivity or asymmetry are violated. Assume now there is a subgraph V_i of E_n so that $S \cap V_i$ has no edges. Then one can add any one edge from V_i to S while retaining consistency: If one adds (without loss of generality) $\alpha_i \to \beta_i$, this preserves consistency, since

- **Completeness** is preserved (α_i, β_i are connected to all ω_h, ω_j ($h < i < j$)).
- **Transitivity** is preserved ($\omega_h \to \alpha_i, \alpha_i \to \beta_i$ also means that $\omega_h \to \beta_i$ since $h < i$, and similar for $\alpha_i \to \beta_i, \beta_i \to \omega_j$).
- **Asymmetry** is preserved because we add no reversed edges where there were edges in S before.

□

Lemma 12. Let $\mathcal{S}$ be a set of inclusion-maximal consistent subgraphs of E_n, and $|\mathcal{S}| = 2^n + 1$. Then there exists no consistent graph C on the vertices of E_n so that $\forall S \in \mathcal{S} : S \subseteq C$.

Proof. We showed that each $S \in \mathcal{S}$ contains exactly one edge from each V_i. If two S_1, S_2 for a given V_i share the same edge (i.e. $S_1 \cap V_i = S_2 \cap V_i$), S_1 and S_2 can be subgraphs of the same consistent graph C. If two $S_1, S_2 \in \mathcal{S}$, for a given V_i, *don't* share the same edge ($S_1 \cap V_i \neq S_2 \cap V_i$), they can be nevertheless still be subgraphs of the same consistent C: If (without loss of generality) $S_1 \cap V_i = \alpha_i \to \beta_i$ and $S_2 \cap V_i = \beta_i \to \gamma_i$, C can contain those edges as well as $\alpha_i \to \gamma_i$. If, though, there are

three $S_1, S_2, S_3 \in \mathcal{S}$ that each don't share an edge on a given V_i, they can't be subgraphs of any consistent C: Such a C would need to contain $\{\alpha_i \to \beta_i, \beta_i \to \gamma_i, \gamma_i \to \alpha_i\}$, but this would violate either asymmetry (if one added $\alpha_i \to \gamma_i$ as well) or transitivity (through the absence of of $\alpha_i \to \gamma_i$). Therefore, for each V_i, only two edges from V_i can occur in in any element of $\mathcal{S}$. Then an $S \in \mathcal{S}$ can be uniquely identified by which edge from each V_i it contains, since there are two edges for each V_i and there are n such "levels" V_i, and no two edges from different V_i, V_j are mutually exclusive. Therefore, $|\mathcal{S}| \leq 2^n$. But introducing an additional distinct S_{2^n+1} to $\mathcal{S}$ must add a a third edge from at least one V_i, thus 2^n is the maximal size of $\mathcal{S}$. □

Lemma 13. The set of consistent graphs $\mathbf{C}$ on the vertices of E_n that includes all inclusion-maximal consistent subgraphs of E_n has size at least $(\frac{3}{2})^n$.

Proof. Assume that one can partition the set $\mathbf{C}$ of inclusion-maximal consistent subgraphs of E_n into a set $\mathbf{P}$ of disjoint sets of size $\leq 2^n$ ($\forall \mathcal{C}_i \in \mathbf{P} : |\mathcal{C}_i| = 2^n|$) such that there exists a consistent graph C that contains all $\mathcal{C}_i$. Then the number of such partitions would be the number of consistent graphs required to "cover" all elements in $\mathbf{C}$, since by **Lemma** 12 the sets of compatible graphs have at most size 2^n. Then the size of $\mathbf{P}$ would be at least $\frac{3^n}{2^n} = 1.5^n$, which is exponential in n. □

Corollary 1. There is no polynomial p and function $f : \mathfrak{P} \to \mathcal{P}(\mathfrak{C})$ such that $|f(E_n)| \leq p(n)$ and f fulfills **Preservation of Consistent Subgraphs**, so **Theorem** 10 is true (with E_n as a counterexample).

Remark 3. This bound is $(\frac{3}{2})^{\frac{m}{3}} = \sqrt[3]{\frac{3}{2}^m} \approx 1.145^m$ for the number of vertices m in E_n, which is exponential but can probably be improved upon.

3.5.2 Polynomial Time Complexity and Preservation of Consistent Subgraphs are Incompatible

Similarly, it is not possible to have an algorithm that returns, for each inclusion-maximal consistent subgraph S, at least one consistent graph that contains S, *and* computes its output in polynomial time.

Theorem 14. Let $\mathtt{A}$ be an algorithm for resolving inconsistent graphs that implements an f which fulfills **Preservation of Consistent Subgraphs** for all graphs $G \in \mathfrak{P}$. Then there exists no polynomial p so that for all directed graphs $P_n \in \mathfrak{P}_n$ of size n it holds that $\mathtt{A}(P_n)$ computes its output in less than $p(n)$ steps.

Proof. Let $\mathbf{C} = \mathtt{A}(E_n)$. **Lemma** 13 shows that $\mathbf{C}$ is exponential in the number of vertices (by **remark** 3). Any $\mathtt{A}$ would at least need to enumerate all $C \in \mathbf{C}$, which would take exponential time. □

Remark 4. The set of inclusion-maximal consistent subgraphs on E_n can be compactly represented as the Cartesian product of the inclusion-maximal consistent subgraphs of the "levels" V_i:

$$\bigtimes_{i=1}^{n} \{\alpha_i \to \beta_i, \beta_i \to \gamma_i, \gamma_i \to \alpha_i\}$$

This might also allow for a compact representation of the result of f which includes all inclusion-maximal consistent subgraphs. We suspect there are counter-examples that don't allow for this, but haven't been able to find any.

4 Inconsistent Preferences over Lotteries

Von Neumann and Morgenstern formulate their famous theorem by defining some restriction on relations over lotteries [von Neumann and Morgenstern, 1947], as explained in Section 2.1.

Finding a mathematical structure which can encode all inconsistent preferences over lotteries *and* is still computationally tractable remains an open problem, but we propose two structures which can either tractably encode some subset of inconsistent preferences or are rich enough to encode all inconsistent preferences, but too complex to be compactly represented.

4.1 Violating the Axioms

Introducing lotteries allows for a large variety of violations of the von Neumann-Morgenstern axioms.

4.1.1 Discontinuity

Discontinuity in relations over lotteries can occur if we know that $l_1 \preceq l_2 \preceq l_3$, but there is no p so that $l_2 \sim [p : l_1, (1-p) : l_3]$. A discontinuous preference that fulfills $l_1 \preceq l_2 \preceq l_3$ could then state that for every $p \in (0; 1]$ it holds that $l_2 \succ [p : l_1, (1-p) : l_3]$ but $l_2 \prec l_3$: l_2 is strictly preferred over any mixture of l_1, l_3, but l_3 is still strictly preferred to l_2. The equivalent can occur if l_2 is strictly dispreferred to any mixture of l_1, l_3, but strictly preferred over l_1.

In humans, this can sometimes be observed as the certainty effect from prospect theory [Tversky and Kahneman, 1981], in which subjects systematically overrate the value of certain (deterministic) option, which leads to the Allais paradox [Allais, 1953].

A view under which discontinuities of this type make sense is if an agent has a specific aversion to lotteries, irrespective of the options they comprise of ([von Neumann and Morgenstern, 1947, 3.7.1] call the continuity axiom "excluding a "utility of gambling"", and state that "concepts like a "specific utility of gambling" cannot be formulated free of contradiction on this level." [ibid.]).

4.1.2 Dependence

Violating of the independence axiom ("dependence") occur if for two lotteries l_1, l_2 ($l_1 \preceq l_2$) there is an option l_3 and a $p \in [0; 1]$ so that

$[p : l_1, (1-p) : l_3] \succ [p : l_2, (1-p) : l_3]$: Mixing in l_3 in equal proportion to both l_1, l_2 causes the preference to switch.

Together with a strong preference for certainty it is used to construct the paradox in [Allais, 1953]: In experiments, the lottery $A_1 = [1 : \$1\text{mio.}]$ is preferred over the lottery $B_1 = [0.89 : \$1\text{mio.}, 0.01 : \$0, 0.1 : \$5\text{mio.}]$ by humans, but the lottery $B_2 = [0.9 : \$0, 0.1 : \$5\text{mio.}]$ is preferred over $A_2 = [0.89 : \$0, 0.11 : \$1\text{mio.}]$, even though by independence

$$
\begin{aligned}
& A_1 \preceq B_1 \\
& \Leftrightarrow [1 : \$1\text{mio.}] \preceq [0.89 : \$1\text{mio.}, 0.01 : \$0, 0.1 : \$5\text{mio.}] \\
& \Leftrightarrow [0.89 : \$1\text{mio.}, 0.11 : \$1\text{mio.}] \preceq [0.89 : \$1\text{mio.}, 0.01 : \$0, 0.1 : \$5\text{mio.}] \\
& \Leftrightarrow [1 : \$1\text{mio.}] \preceq [1/11 : \$0, 10/11 : \$5\text{mio.}] \\
& \Leftrightarrow [0.89 : \$0, 0.11 : \$1\text{mio.}] \preceq [0.9 : \$0, 0.1 : \$5\text{mio.}] \\
& \Leftrightarrow A_2 \preceq B_2
\end{aligned}
$$

4.2 Representing Inconsistencies

It is more difficult to find a mathematical structure to represent arbitrary inconsistent preferences over lotteries over some set of options Ω.

4.2.1 Edge-Weighted Graphs

Given Ω, some inconsistent preferences on lotteries on Ω can be represented by the set $\mathfrak{G}_\Omega$ of edge-weighted directed graphs on Ω, where edge weights can be expressed as functions $w : \Omega \times \Omega \to \mathbb{R}$.

The subset $\mathfrak{S}_\Omega \subset \mathfrak{G}_\Omega$ of consistent preferences on Ω is the set of all edge-weighted directed graphs that is **complete**, **transitive**, **irreflexive** and **weight-transitive**, where a graph is weight-transitive if for all edges $e \in E$ it holds that $w(\alpha \to \beta) = c_1, w(\beta \to \omega_3) = c_2 \Rightarrow w(\alpha \to \omega_3) = c_1 + c_2$.

An element from $\mathfrak{S}_\Omega$ assigns each element from Ω a cardinal value, equivalent to a utility function on Ω.

Edge-weighted directed graphs are not expressive enough to represent all relevant inconsistent preferences, though. As a trivial example, let $l_1 = [0.25 : \alpha, 0.75 : \beta] \prec l_2 = [0.75 : \alpha, 0.25 : \beta]$, but $l_3 = [0.3 : \alpha, 0.7 : \beta] \succ l_4 = [0.7 : \alpha, 0.3 : \beta]$. The first preference implies a positive weight for the edge $\alpha \to \beta$, but the second preference implies a negative weight for $\alpha \to \beta$.

Introducing two positively weighted edges between α, β (creating a two-cycle) is able to represent that such a preference between lotteries *is* present, but it doesn't allow reconstruction of which lotteries are preferred over which others: Given a preference of α over β by w_l, and of β over α by w_r doesn't enable reconstruction of whether $l_1 \prec l_2$ or $l_1 \succ l_2$.

4.2.2 Arbitrary Relations over the Lotteries

As [von Neumann and Morgenstern, 1947] uses lotteries on Ω as the set of options over which agents can have preferences, a natural instinct is to

use arbitrary relations over lotteries on Ω as the mathematical object to represent preferences.

However, if Ω has at least one element, such a relation can be uncountably large and without compact representation, making it impossible to be handled computationally.

Example 2. A pathological example would be a relation $\mathcal{R} \in \Delta(\Omega) \times \Delta(\Omega)$ on probability distributions of $\Omega = \{\alpha, \beta\}$ in which $[p : \alpha, (1-p) : \beta] \prec [q : \alpha, (1-q) : \beta]$ if and only if $p \in [0; 1]$ is an uncomputable real number and $q \in [0; 1]$ is a computable real number.

We were also unable to find a method for resolving such inconsistent preferences into their consistent versions.

4.3 Algorithms

After some search, we were able to identify `HodgeRank` from [Jiang et al., 2011] as a suitable algorithm for resolving an edge-weighted inconsistent graph into an edge-weighted consistent graph.

Some other possible candidates for methods for resolving inconsistent preferences over edge-weighted graphs were considered, and finally rejected.

One option was the `PageRank` algorithm [Bianchini et al., 2005], also mentioned in [Sun et al., 2017]. We rejected PageRank for the same reason as [Sun et al., 2017] do: In a directed acyclic graph, a unique greatest element does not necessarily receive the highest ranking. This problem extends to using other centrality measures for graphs such as degree centrality and betweenness centrality [Zhang and Luo, 2017]: In graphs that are already consistent, the greatest element usually receives a low centrality score, and elements closer to the center receive larger scores, which is counter to our criteria.

4.3.1 `HodgeRank`

`HodgeRank`, introduced in [Jiang et al., 2011], is an algorithm based on Hodge theory from algebraic geometry for decomposing a doubly edge-weighted, potentially not fully connected graph $G = (V, E, w : E \to \mathbb{R} \cup \{\texttt{nan}\}, l : E \to \mathbb{N}\})$ into the sum of three different edge weighted graphs:

- A gradient graph $G_g = (V, E, w_g : E \to \mathbb{R})$, in which W_g is derived from a potential function that assigns consistent values to $v \in V$: $p : V \to \mathbb{R}$ so that $g(e = (v_i, v_j)) = p(v_j) - p(v_i)$.
- A curl graph $G_c = (V, E, w_c : E \to \mathbb{R})$, where a function c assigns every 3-cycle in the graph a specific value, and the value $W_c(e)$ for an edge is the sum of the values c assigns to all the 3-cycles e is in.
- A harmonic graph $G_h = (V, E, w_h : E \to \mathbb{R})$.

Then $w(e) = w_g(e) + R(e) = w_g(e) + w_c(e) + w_h(e)$, where R is a residual.

[Jiang et al., 2011] develop `HodgeRank` from a social-choice theoretic perspective: Given a set of incomplete cardinal ratings $\mathcal{C} = (\mathbb{R} \cup \{\texttt{nan}\})^{n \times m}$

by a set $V = \{1, \ldots, m\}$ of voters on $A = \{1, \ldots, n\}$ alternatives, one can construct an edge-weighted graph $G_{\mathcal{C}} = (V, E, w, l)$ where the nodes are the options A and each edge weight is some combination of the cardinal votes on the options ω_1, ω_2 that comprise the edge.

An edge weight can be for example the arithmetic mean

$$w_{\mathcal{C}}(\omega_1 \to \omega_2) = \frac{\sum_{i=1}^{n} \mathcal{C}_{i,\omega_2} - \mathcal{C}_{i,\omega_1}}{|\{n | \mathcal{C}_{n,\omega_1}, \mathcal{C}_{n,\omega_2} \text{ both } \neq \texttt{nan}\}|}$$

though [Jiang et al., 2011] also discuss using other methods such as the geometric mean or the ratio of preference to dispreference.

If every voter assigns `nan` to both ω_1 and ω_2, there is no edge between the two options.

The function $l : E \to \mathbb{R}$ denotes the number of voters which have a non-`nan` rating for both nodes in the edge. In the case where we do not take the social choice view, we can assume that $\forall e \in E : l(e) = 1$, which does not change the process of computing the output of `HodgeRank`.

Algorithm 4 Computing `HodgeRank` from an edge-weighted directed graph.

function HODGERANK($G = (V, E, w, l)$)
  Revert all $e \in E$ with $w(e) < 0$ so that they now have positive weight.
  $f \leftarrow (w(e_1), \ldots, w(e_k))$
  $L \leftarrow \text{diag}(l(e_1), \ldots, l(e_k))$ ▷ diag is the diagonal matrix of a vector
  $O \leftarrow \mathbf{0}^{|E| \times |V|}$
  for $e = (u, v) \in E$ **do**
    $O_{eu} \leftarrow -1$, $O_{ev} \leftarrow 1$
  end for
  $s \leftarrow -(O^\top L O)^+ O^\top L f$ ▷ A^+ is the Penrose-Moore pseudo-inverse of A
  return s
end function

Remark 5. One might ask, under the social choice view, whether it makes sense for some $v \in V$ to lie about their preferences over A in order to change the output of `HodgeRank` to correspond to their own ranking ordinally. In fact this is true and therefore `HodgeRank` is not strategy-free.

It is easy to find an example for this: Assume there are three options $A = \{a, b, c\}$, and three voters $V = \{1, 2, 3\}$, and let the cardinal values assigned to the options be $u_1(a) = 4, u_1(b) = 3, u_2(b) = 4, u_2(c) = 3, u_3(c) = 4, u_3(a) = 3$, with the rest of the values assigned to the options being `nan`. Then the values `HodgeRank` assigns to the options are $h(a) = h(b) = h(c) = 0$. But voter 1 can change their reported assignments to be $u_1'(a) = 5, u_1'(b) = 3, u_1'(c) = 1$, changing the outputs of `HodgeRank` to $h'(a) = 1, h'(b) = 0$ and $h'(c) = -1$, which is more compatible with their preferences.

It would be interesting to investigate the computational complexity of finding manipulations of existing preference of one voter to ordinally change the output of `HodgeRank` to more strongly conform to that voters' preferences.

Besides the disadvantage of allowing for strategic manipulation, the decomposition returned by `HodgeRank` appears to display many desirable properties as a method for resolving inconsistent preferences over edge-weighted graphs:

- **Existence**: It always exists.
- **Uniqueness**: This decomposition is unique up to an additive constant.
- **Polynomial time computability**: Finding W_g is equivalent to solving a $|V| \times |V|$ least-squares problem, which can be solved in $\mathcal{O}(n^3)$ time, for example by computing the Penrose-Moore inverse of a specific matrix. Finding W_h and W_c from R is more computationally intensive, but still polynomial: they are equivalent to solving a least-squares problem of size $\frac{|V|}{3} \approx \mathcal{O}(n^3)$, and can therefore be found in $\mathcal{O}(n^9)$.
- **Robustness to incomplete and cyclic data**: `HodgeRank` still returns a result, even if edges are missing or there are positive-valued cycles in the data.
- **Relation to known solution concepts from social choice theory**: If G has no missing edges and w is defined for every edge, `HodgeRank` returns an affine transformation of the result that the Borda count would return.

In the context of inconsistent preferences, `HodgeRank` can be interpreted as taking the observed preferences of an agent as an edge-weighted directed graph, and decomposing it so that the potential function p determines how much the agent values different elements in V. Here p can act as a utility function. The social-choice theoretic perspective offers an intriguing possibility of modeling humans as being comprised of subagents [Demski and Garrabrant, 2019] and Minsky [1988], which we will not pursue further here.

5 Applications

Equipped with a notion of how to represent inconsistent preferences and how to resolve them, one can examine problems that have come up in other contexts and apply the knowledge gained to them. I will examine one of those: The problem of changing a preference as the underlying set of options changes.

5.1 Ontology Identification, Ontological Shifts and Ontological Crises

The term "ontological crisis" was introduced in [De Blanc, 2011] and intuitively refers to a scenario in which an agent has preferences, defined over some world model, and then the world model changes without corresponding changes in the values.

An example of this can be observed in human values before and after exposure to philosophy: A human might have a value they would formulate as "I value the continuation of my life". However, after reading [Parfit, 1984, pt. 3], the view of personal identity that justifies a notion of "continuation" might seem much less defensible, as thought experiments around teleportation, the fusion and fission of persons, gradual replacement of the body or atom-by-atom recreation of the body all undermine the concept of a single fixed personal identity.

However, this person would likely not just give up their value of their continued existence, but instead attempt to "port it" to the new world model.

[Soares and Fallenstein, 2017] motivate the problem of ontological crises in the context of a problem they call **Ontology Identification**: Given a Turing machine using the atomic model of physics, they ask how one can identify which parts of the program and the tape represent atoms or macroscopic objects, and repeat the question for a Turing machine using a quantum-mechanical model of the world. The problem is further elaborated on outside of the academic literature in [Yudkowsky et al., 2016] and [Yudkowsky and Andreev, 2016].

It seems useful to disambiguate some terms that appear in the literature, to create clarity about what they mean:

- **Ontology Identification**: "Given goals specified in some ontology and a world model, how can the ontology of the goals be identified in the world model? What types of world models are amenable to ontology identification?" (Definition from [Soares and Fallenstein, 2017])
- **Ontological Shift**: Given some goals specified in some ontology and a world model in which those goals have already been identified, an ontological shift occurs if the world model changes but the ontology of the goals does not.
- **Ontological Crisis**: An ontological crisis is the *result* of an ontological shift, and the behavior of an agent after an ontological crisis could be undefined.

The word "ontology" here is a place-holder for a more crisply defined model, such as Markov Decision Processes (MDPs) or Partially Observable Markov Decision Processes (POMDPS).

5.1.1 Existing Approaches

[De Blanc, 2011] approaches the problem of ontological crises formally in the context of what they call "finite state models" (they neglect to give a full definition), and one can refine their problem statement and their approach to a solution by stating it in terms of Markov decision processes [Russell, 2010, ch. 17.1].

Definition 5.1. A finite **Markov decision process** (MDP)
$\mathcal{M} = (S, A, P, R, I)$ is a tuple of five elements, where S is a set of states (in this case finite, with $|S| = n$), A is a set of actions (also finite, with $|A| = m$), $P(s, a, s') : S \times A \times S \to [0, 1] = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$ is

a function that returns the probability of transitioning from s to s' via the action a, $R : S \to \mathbb{R}$ is a reward function that returns a real-numbered value for reaching a certain state[5], and $I : S \to [0, 1]$ is a probability distribution for the states that the agent is initially in.

Given some ordering of the states $s_1, \ldots, s_n$, P from $\mathcal{M}$ can also be represented as a family of right stochastic matrices $\mathbf{T}(a)$ (the transition matrices), R can be encoded as a real-numbered vector with size n, and I can be described as real-numbered vector of size n in which the elements sum to 1.

$$\mathbf{T}(a) : [0,1]^{n \times n} = \begin{pmatrix} P(s_0|a, s_0) & \cdots & P(s_0|a, s_n) \\ \vdots & \ddots & \vdots \\ P(s_n|a, s_0) & \cdots & P(s_n|a, s_n) \end{pmatrix}$$

$$\mathbf{R} : \mathbb{R}^n = \begin{pmatrix} R(s_0) \\ \vdots \\ R(s_n) \end{pmatrix}$$

$$\mathbf{I} : \mathbb{R}^n = \begin{pmatrix} I(s_0) \\ \vdots \\ I(s_n) \end{pmatrix}$$

Consider two MDPs $\mathcal{M}_1 = (S_1, A, P_1, R_1, I_1)$ and $\mathcal{M}_2 = (S_2, A, P_2, R_2, I_2)$, but with R_2 being unknown. An agent who starts with $\mathcal{M}_1$, but who discovers that a better model of the environment has a different set of states and transition probabilities (however, the set of actions stays the same) and thereby now wants to operate in $\mathcal{M}_2$ has the problem of defining R_2.

Definition 5.2. The method [De Blanc, 2011] use to find R_2 is to find two linear maps $\phi : \mathbb{R}^{n_1 \times n_2}$ and $\psi : \mathbb{R}^{n_2 \times n_1}$ ($n_1 = |S_1|, n_2 = |S_2|$) such that ϕ, ψ can be used to "translate" between $\mathcal{M}_1$ and $\mathcal{M}_2$. Then, for any $a \in A$, ϕ and ψ should be selected so that for any $a \in A$, it holds that $\psi \times \mathbf{T}_1(a) \times \phi \approx \mathbf{T}_2(a)$, and equivalently $\phi \times \mathbf{T}_2(a) \times \psi \approx \mathbf{T}_1(a)$.

[De Blanc, 2011] don't name ϕ, ψ, but we will call such ϕ, ψ for MDPs a **de Blanc bisimulation**.

Definition 5.3. ϕ and ψ are found by minimising the Kullback-Leibler divergences of the columns of the transition matrices and of the distribution over initial states with a hill-climbing algorithm from random initial values:

$$\begin{aligned} \text{BisimulateShift}(\mathcal{M}_1, \mathcal{M}_2) = \underset{\phi,\psi}{\operatorname{argmin}} & \sum_{a \in A} \sum_{i=1}^{n_1} D_{KL}((\mathbf{T}(a)_2)_{i,*} || (\psi \mathbf{T}(a)_1 \phi)_{i,*}) + \\ & \sum_{a \in A} \sum_{j=1}^{n_2} D_{KL}((\mathbf{T}(a)_1)_{j,*} || (\phi \mathbf{T}(a)_2 \psi)_{j,*}) + \\ & D_{KL}(I_2 || I_1^\top \phi) + D_{KL}(I_1 || I_2^\top \psi) \end{aligned}$$

[5] [Russell, 2010, ch. 17] notes that sometimes R takes actions into account as well: $R : S \times A \times S \to \mathbb{R}$ (with different rewards for transitioning to a state with different actions), but also notes that this merely simplifies the description of some environments, but doesn't change which environments can be described.

We call a function that returns a de Blanc bisimulation for two MDPs by minimizing the Kullback-Leibler divergence between the MDPs **BisimulateShift**.

[De Blanc, 2011] notes that both products of the matrices ϕ, ψ should be close to equal to the identity matrix, $\phi \times \psi \approx \mathbf{1}_{n_1}, \psi \times \phi \approx \mathbf{1}_{n_2}$, which implies that mapping from $\mathcal{M}_1$ to $\mathcal{M}_2$ and back loses little information.

Given ϕ and ψ, it is possible to infer $\mathbf{R}_2$ using ϕ and ψ: $\mathbf{R}_2 = \mathbf{R}_1^\top \phi$.

Advantages There are some advantages to taking this approach for resolving ontological crises. One is that it does not presuppose a known mapping between S_1 and S_2, and can infer the mapping solely from the transition behavior of $\mathcal{M}_1$ and $\mathcal{M}_2$.

Another advantage is that for an exact solution found by BisumlateShift, the expected reward of repeating any action in $\mathcal{M}_2$ only depends on the expected reward of executing the same action in $\mathcal{M}_2$ with a linear transformation of the initial state distribution.

Proposition 4. Let $\mathcal{M}_1, \mathcal{M}_2$ be two MDPs, and let ϕ, ψ be two matrices found by BisimulateShift, so that $\phi \times \psi = \mathbf{1}_{n_1}, \psi \times \phi = \mathbf{1}_{n_2}$ and $\psi \mathbf{T}_1(a) \phi = \mathbf{T}_2(a)$. For an action $a \in A$, let $r_2(a, k)$ be the expected average reward of executing a in $\mathcal{M}_2$ $k \in \mathbb{N}$ times, and $r_1(a, k)$ the equivalent for $\mathcal{M}_1$.

Then $r_2(a, k)$ is equal to $r_1(a, k)$ with a linear transformation to the probability distribution over initial states of $\mathcal{M}_1$, I_1.

Proof. In matrix notation, the expected average reward of executing a infinitely often under the two MDPs is

$$r_1(a, k) = \frac{1}{k} \sum_{i=1}^{k} \mathbf{R}_1^\top \times (\mathbf{T}_1(a))^i \times \mathbf{I}_1$$

and

$$r_2(a, k) = \frac{1}{k} \sum_{i=1}^{k} (\mathbf{R}_1^\top \phi) \times \mathbf{T}_2(a)^i \times \mathbf{I}_2$$

$r_2(a)$ can be further expanded and simplified to

$$\begin{aligned} r_2(a) = & \\ & \frac{1}{k} \sum_{i=1}^{k} (\mathbf{R}_1^\top \phi) \times (\psi \mathbf{T}_1(a) \phi)^i \times (\mathbf{I}_1^\top \phi)^\top = \\ & \frac{1}{k} \sum_{i=1}^{k} \mathbf{R}_1^\top \times \mathbf{T}_1(a)^i \times \phi \phi^\top \times \mathbf{I}_1 \end{aligned}$$

□

Conjecture 6. There exists a linear function $f(x) = ax + b$ so that for any $a \in A$, $k \in \mathbb{N}$, $r_2(a, k) = f(r_1(a, k))$.

Disadvantages However, the approach [De Blanc, 2011] outline has some limitations. As they remark, their setting of what they call "finite state models" is a fairly restrictive class of computational models of the environment. Similarly, MDPs are also not able to represent some environments, especially ones in which observations of states carry uncertainty.

They also remark that BisimulateShift "is not computationally tractable for large ontologies", and their lack of clarity on the exact algorithm used (as well as the absence of an analysis of the computational complexity of the problem) makes it difficult to judge the computational complexity of the problem. It might be fruitful to study the convergence behavior of using different optimization procedures for finding ϕ and ψ to make further statements about the computational complexity of BisimulateShift.

Finally, the setting of a "finite state model" or an MDP can't encode certain types of consistent preferences. Let $\mathcal{M} = (S = \{s, s'\}, A = \{a_1, a_2\}, I, P, R)$, where $P(s, a_1, s') = P(s', a_1, s) = P(s, a_2, s) = P(s', a_2, s') = 1$ (a_1 causes the agent to switch states, and a_2 is the action where the agent stays in the same state).

Let now $t_1, t_2 \in (S \times A)^k \times S$ be two trajectories in $\mathcal{M}$, namely $t_1 = (s, a_1, s', a_1, s, a_2, s)$ and $t_2 = (s, a_2, s, a_1, s', a_1, s)$. Then the cumulative reward of both trajectories is equal: $R(t_1) = R(s, a_1, s') + R(s', a_1, s) + R(s, a_2, s) = R(s, a_2, s) + R(s, a_1, s') + R(s', a_1, s) = R(t_2)$. However, intuitively there should way a way to differently value these two trajectories: It should be possible to value be in s' earlier rather than later.

5.1.2 Using Inconsistent Preferences to Represent Ontological Crises

The framework of representing preferences as edge-weighted directed graphs on a set Ω of vertices, and consistent preferences as the set of edge-weighted acyclic tournaments on a set of deterministic options Ω, can be used to represent ontological shifts.

Definition 5.4. Given a consistent edge-weighted graph $G = (\Omega, E_G, w)$, a **graph-based ontological shift** is as a function from Ω to subsets of a new set of options Ξ, together with coefficients: $s : \Omega \to \mathcal{P}(\Xi \times [0, 1])$, where $(\xi, c) \in s(\omega)$ means that $\omega \in \Omega$ in the old set of options turned out to be $\xi \in \Xi$ to the degree c, the larger c, the more ω is ξ.

In this text, I will assume that $\forall \omega \in \Omega : 0 \leq \sum_{(\xi,c) \in s(\omega)} c \leq 1$.

If the coefficients of the image of ω sum to 1, that means that ω has been completely "ported over" to Ξ. If they sum to less than 1, that means that ω was a (partially) confused concept, if the coefficients in the image sum to 0 (or $s(\omega) = \emptyset$), that means that ω was a wholly confused concept and does not actually exist. If the sum of the coefficients are > 1, that means that ω turned out to be "more real" than in the old set of options.

Definition 5.5. Given G, the result $G^\star = (\Xi, E^\star, w^\star)$ after a graph-based ontological shift s is an edge-weighted graph with with a function $t : \Xi \times \Xi \to \mathbb{R}$, where t is a combination of the weights w of G and the coefficients of s (for all ω_1, ω_2):

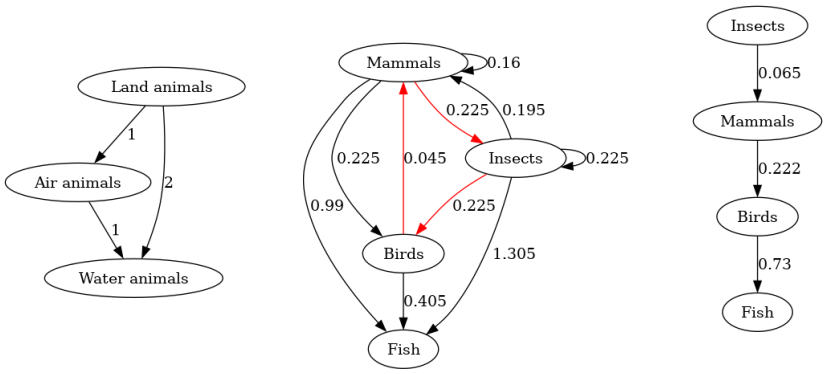
$$t(\xi_1, \xi_2) = \sum_{(\omega_1,\omega_2)\in E} \sum_{(\xi_1,c_1)\in s(\omega_1),(\xi_2,c_2)\in s(\omega_2)} c_1 \cdot c_2 \cdot w(\omega_1, \omega_2)$$

Example 3. Let $\Omega = \{L(\text{Land animals}), A(\text{Air animals}), W(\text{Water animals})\}$, and the current preference prefer land animals over air animals over water animals, that is $E_G = \{L \xrightarrow{1} A, L \xrightarrow{1} W, A \xrightarrow{2} W\}$.

Let now $\Xi = \{M(\text{Mammals}), B(\text{Birds}), F(\text{Fish}), I(\text{Insects})\}$ be a set that better represents the available options, and let s be

$$\begin{aligned} s(L) &= \{(M, 0.5), (I, 0.5)\} \\ s(A) &= \{(B, 0.45), (I, 0.45), (M, 0.1)\}) \\ s(W) &= \{(F, 0.9), (M, 0.1)\} \end{aligned}$$

(Ignoring, for the sake of simplicity of the example, exocoetidae[6] and aquatic insects).



(a) The initial preference G, as an edge-weighted graph.

(b) $G^\star$, after applying the ontological shift s and determining the edge weights using t. Positive 3-cycle in red.

(c) G' after applying a procedure for resolving the inconsistent preference $G^\star$, in this case using `HodgeRank`.

Figure 7: Undergoing an ontological shift s and then resolving the ontological crisis using `HodgeRank`. In the right image transitive correctly weighted edges are ommitted for readability.

The procedure for resolving ontological crises by representing them as inconsistent preferences can be written as **algorithm** 5. The algorithm takes a consistent edge-weighted graph G, a graph-based ontological shift

[6] Also know as flying fish.

s mapping elements from Ω to a new set Ξ, together with coefficients, and a method for resolving inconsistent preferences on edge-weighted graphs.

It then creates a new graph $G^\star$, mapping all nodes using s and creating new edges using the existing weights and coefficients with the function t explained above. Finally, $G^\star$ is resolved into a consistent preference with the method `Resolve` (which may be specified externally, e.g. by using `HodgeRank` or dropping the weights and using `EGEDmin`).

Algorithm 5 Resolving an ontological shift s on a edge-weighted directed graph.

function ResolveShift($G = (\Omega, E, w), s : \Omega \to \mathcal{P}(\Xi \times [0,1]), \texttt{Resolve}$)
  $E^\star \leftarrow \emptyset, w^\star = 0$ ▷ $w^\star$ is set to 0 for all possible inputs.
  for $(\omega_1, \omega_2) \in E$ **do**
    for $(\xi_1, c_1) \in s(\omega_1), (\xi_2, c_2) \in s(\omega_2)$ **do**
      $w^\star(\xi_1, \xi_2) \leftarrow w^\star(\xi_1, \xi_2) + c_1 \cdot c_2 \cdot w(\omega_1, \omega_2)$
      $E^\star \leftarrow E^\star \cup \{(\xi_1, \xi_2)\}$
    end for
  end for
  $G' \leftarrow \texttt{Resolve}(G^\star = (\Xi, E^\star, w^\star))$
  return G'
end function

Advantages An advantage of `ResolveShift` over BisimulateShift is the set of preferences that can be represented by G and G'. If Ω is the set of all finite sequences of state-action pairs $((S \times A)^k \times S)_{k \geq 0}$ then $t_1 = (s, a_1, s', a_1, s, a_2, s)$ and $t_2 = (s, a_2, s, a_1, s', a_1, s)$ are two different elements in Ω, and a preference of t_1 over t_2 can be represented e.g. with an edge $t_1 \to t_2$ in E.

A further advantage of `ResolveShift` is that it has a polynomial runtime complexity of $\mathcal{O}(|E| \cdot m^2) \subseteq \mathcal{O}(n^2 \cdot m^2)$ ($n = |\Omega|, m = |\Xi|$), unlike BisimulateShift, which offers no such guarantees.

Disadvantages If the dynamics (e.g. the transition function) of the elements of Ξ are known, then BisimulateShift is able to use this information to construct R_2. Additionally, if no mapping s from Ω to Ξ exists (that is, only Ω and Ξ are known, but their relations are not), then `ResolveShift` is not applicable.

Definition 5.6. Let $f : \mathfrak{G} \to \mathfrak{S}$ be a method for resolving inconsistent preferences represented by edge-weighted graphs, and let $s_1, s_2, \ldots, s_n$ ($s_i : \Omega_i \to \mathcal{P}(\Omega_{i+1}) \times [0,1]$) be a family of functions describing ontological shifts.

Let $g_1, g_2, \ldots, g_n$ be a family of functions that return the result of `ResolveShift` using the shift function s_i for g_i, but without executing a resolution procedure: $g_i(G_i) = \texttt{ResolveShift}(G_i, s_i, \mathrm{id})$, where $\mathrm{id} : \mathfrak{P}_{\Omega_{i+1}} \to \mathfrak{P}_{\Omega_{i+1}}$ is the identity function.

Let $G_1 = (\Omega_1, E_1, w_1)$ be any arbitrary consistent preference on Ω_1.

Then f is **distributive over ontological shifts** if and only if

$$(f \circ g_n \circ \cdots \circ g_2 \circ g1)(G_1) = (f \circ g_n \circ f \circ \cdots \circ f \circ g_2 \circ f \circ g_1)(G_1)$$

Intuitively, this condition says that it shouldn't matter whether an agent changes their mind on which things exist to have preferences over multiple times, and then resolves the resulting preferences into consistent ones, *or* resolve their preferences after each time they undergo an ontological shift s_i.

Proposition 5. `HodgeRank` is not **distributive over ontological shifts**.

Proof. It is easy to find examples where `HodgeRank` is not **distributive over ontological shifts**.

Let $G_1 = (\Omega = \{a, b\}, E = \{(a \xrightarrow{1} b)\})$. Let $s_1(a) = \{(d, 0.28)\}$, $s_1(b) = \{(c, 0.57), (e, 0.43)\}$. And let $s_2(c) = \{(f, 0.014)\}$, $s_2(d) = \{\}$, and $s_2(e) = \{(f, 0.34), (g, 0.66)\}$.

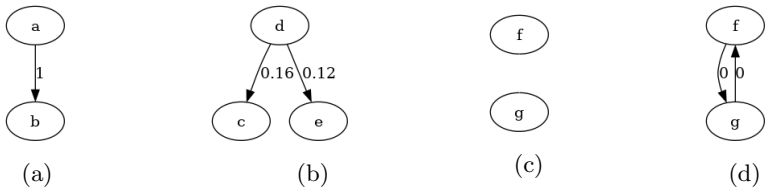


Figure 8: In (a) we have the initial preference G_1, as an edge-weighted graph. (b) shows the unresolved preference $g_1(G_1)$, and (c) shows $g_2(g_1(G_1))$, which has no edges. As a result, finally resolving $g_2(g_1(G))$ using `HodgeRank` results in a graph in which there is indifference between the vertices f and g.

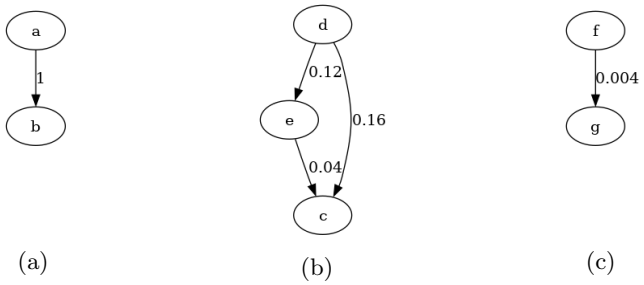


Figure 9: (a) again shows the initial preference G_1. (b) is $\texttt{HodgeRank}(g_1(G_1))$, which has an edge between e and c, unlike the result of just $g_1(G_1)$. The final preference shown in (c), $(\texttt{HodgeRank} \circ g_2 \circ \texttt{HodgeRank} \circ g_1)(G_1)$ is *not* indifferent between f and g, and slightly prefers f.

□

This example works because d gets "deleted" from the set of options, so having all preferences depend on d without resolving the incomparability

between c and e results in there being no preference, while resolving retains a slight preference of e over c, which remains with f and g.

Conjecture 7. There is a resolution function f for edge-weighted graphs that is **distributive over ontological shifts** in this framework.

6 Conclusion

In this investigation, we have identified the problem of resolving preferences that are inconsistent under the von Neumann-Morgenstern framework.

We first examined the restricted case of preferences over deterministic options, using directed graphs as an underlying mathematical structure to represent inconsistent preferences. We proposed two algorithms: `EGEDmin` and `HodgeResolve` (based on the `HodgeRank` algorithm). We analyzed both algorithms on several different criteria, with no clear winner.

We also proved that the criteria **Resolution to Polynomially Many Preferences** and **Preservation of Consistent Subgraphs** are incompatible, as well as **Resolution to Polynomially Many Preferences** and **Polynomial Time Complexity**.

For inconsistent preferences over lotteries, we examined a representation using edge-weighted directed graphs. This representation is inadequate, as it can not encode all possible inconsistent preferences, most notably the violation of independence observed in [Allais, 1953].

We nevertheless reviewed the `HodgeRank` algorithm that allows for resolving inconsistent edge-weighted directed graphs into utility functions, and observe that `HodgeRank` has several desirable properties, and that it also fails to conform to the (hard to fulfill) criterion of strategy-freeness from social choice theory.

We then connected inconsistent preferences to the little-explored issue of ontological crises, and offered a new perspective on what to do after a change with a set of objects that a preference was defined over, opening up many questions we didn't have the time to solve.

6.1 Further Research

We believe that the topics discussed in this text offer some fruitful lines of inquiry into the mathematical structure of desire.

On a concrete level we stated several conjectures and questions we were not able to prove, but might be relatively easy to answer. Of these, **conjecture** 5 on whether `HodgeResolve` fulfills **Preservation of Complete Domination** is most relevant, but **conjecture** 1 and **conjecture** 2 might also be interesting from graph-theoretic perspective.

Additionally, we only analysed two methods of mapping from directed graphs to acyclic tournaments, but are convinced that there are many other methods that could be investigated, specifically methods that use different methods of evaluating graph similarity or ones that result in weak orderings, or methods that are selected to preserve as many inclusion-maximal consistent subgraphs as possible.

Resolving inconsistent graphs could also be approached from a different perspective using random walks on the graph, breaking cycles and completing edges as they are encountered. An extension of this setup could involve two agents: One trying to resolve its preferences through a process of breaking cycles as it traverses the graph representing its preferences, and an adversary attempting to money-pump the agent. This setup also is amenable for an analysis of money-pumping under the light of computational complexity: which violations of the von Neumann-Morgenstern axioms are computationally easy or hard to find, and what is the attack/defense balance between finding and exploiting such violations?

In the context of preferences over lotteries, we are left with no satisfactory mathematical structure that we can use: edge-weighted graphs are not expressive enough, and arbitrary relations over all lotteries too unwieldy. Finding such a structure or finding a method for resolving arbitrary relations over lotteries would be helpful for further progress. Inspiration could be found in models of human decision making from mathematical psychology, such as the Priority Heuristic and the Random Utility Model from [El Gamal, 2013] and the BEAST model from [Erev et al., 2017], as well as alternatives to the utility framework from decision theory, such as risk-weighted utility maximization or the Jeffrey-Bolker axioms [Buchak, 2013], [Jeffrey, 2004].

The problem of ontological crises appears under-researched. As a first step, BisimulateShift could be extended to POMDPs, but finding out how real-world systems change their internal representations during learning could be valuable, with [Nanda et al., 2023] being a fascinating analysis of the toy case of modular addition in neural networks. This question could also be interesting for social scientists (discovering how humans manage ontological crises in practice) and philosophers.

We would also like to see further exploration of value-learning [Dewey, 2011] of inconsistent preferences, perhaps extending [Evans et al., 2016] to allow for a larger diversity of inconsistent preferences.

References

George Ainslie and R.J. Herrnstein. Preference reversal and delayed reinforcement. *Animal Learning & Behavior*, 9(4):476–482, 1981.

Michael Aird and Justin Shovelain. Using vector fields to visualise preferences and make them consistent. 2020. URL `https://www.lesswrong.com/posts/ky988ePJvCRhmCwGo/`.

Maurice Allais. Le comportement de l'homme rationnel devant le risque: critique des postulats et axiomes de l'école américaine. *Econometrica: journal of the Econometric Society*, pages 503–546, 1953.

Stuart Armstrong and Sören Mindermann. Occam's razor is insufficient to infer the preferences of irrational agents. *Advances in neural information processing systems*, 31, 2018.

Kenneth J Arrow. A difficulty in the concept of social welfare. *Journal of political economy*, 58(4):328–346, 1950.

David Austen-Smith and Jeffrey S. Banks. *Positive Political Theory I*. The University of Michigan Press, 2000.

David K Backus, Bryan R Routledge, and Stanley E Zin. Exotic preferences for macroeconomists. *NBER Macroeconomics Annual*, 19:319–390, 2004.

Ali Baharev, Hermann Schichl, Arnold Neumaier, and Tobias Achterberg. An exact method for the minimum feedback arc set problem. *Journal of Experimental Algorithmics (JEA)*, 26:1–28, 2021.

Monica Bianchini, Marco Gori, and Franco Scarselli. Inside pagerank. *ACM Transactions on Internet Technology (TOIT)*, 5(1):92–128, 2005.

Lara Buchak. *Risk and rationality*. OUP Oxford, 2013.

Bernard Caillaud and Bruno Jullien. Modelling time-inconsistent preferences. *European Economic Review*, 44(4-6):1116–1124, 2000.

Peter De Blanc. Ontological crises in artificial agents' value systems. *arXiv preprint arXiv:1105.3821*, 2011.

Abram Demski and Scott Garrabrant. Embedded agency. *arXiv preprint arXiv:1902.09469*, 2019.

Daniel Dewey. Learning what to value. In *International conference on artificial general intelligence*, pages 309–314. Springer, 2011.

Aly El Gamal. On the structural consistency of preferences. 2013.

Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.

Ido Erev, Eyal Ert, Ori Plonsky, Doron Cohen, and Oded Cohen. From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience. *Psychological review*, 124 (4):369, 2017.

Owain Evans, Andreas Stuhlmüller, and Noah Goodman. Learning the preferences of ignorant, inconsistent agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Wulf Gaertner. *A Primer in Social Choice Theory*. Oxford University Press, 2009.

Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica: journal of the Econometric Society*, pages 587–601, 1973.

L. Green, A. F. Fry, and J. Myerson. Discounting of delayed rewards: A life span comparison. *Psychological Science*, 5(1)):33–36, 1994.

Johan E Gustafsson. Money-pump arguments. *Elements in Decision Theory and Philosophy*, 2022.

Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill™: A bayesian skill rating system. *Advances in Neural Processing Systems (NIPS)*, pages 569–576, 2007.

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. *arXiv preprint arXiv:1906.01820*, 2019.

Olivier Hudry. On the complexity of slater's problems. *European Journal of Operational Research*, 203(1):216–221, 2010.

Richard Jeffrey. *Subjective probability: The real thing*. Cambridge University Press, 2004.

Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 127(1): 203–244, 2011.

Arthur B Kahn. Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562, 1962.

Jan Kirchner. Inferring utility functions from locally non-transitive preferences. 2022. URL `https://www.lesswrong.com/posts/QZiGEDiobFz8ropA5/`.

William MacAskill, Krister Bykvist, and Toby Ord. *Moral Uncertainty*. Oxford University Press, 2020.

I. McLean. The borda and condorcet principles: Three medieval applications. *Social Choice and Welfare*, 7:99–108, 1990.

Nicholas R. Miller. Graph-theoretical approaches to the theory of voting. *American Journal of Political Science*, 2:769–803, 1977.

Marvin Minsky. *Society of mind*. Simon and Schuster, 1988.

John W Moon and Leo Moser. On cliques in graphs. *Israel journal of Mathematics*, 3:23–28, 1965.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Derek Parfit. *Reasons and Persons*. Oxford University Press, 1984.

Martin Peterson. *An introduction to decision theory*. Cambridge University Press, 2017.

Stuart J Russell. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

Neil JA Sloane et al. The on-line encyclopedia of integer sequences, 2003.

Nate Soares and Benya Fallenstein. Agent foundations for aligning machine intelligence with human interests: a technical research agenda. *The technological singularity: Managing the journey*, pages 103–125, 2017.

Jiankai Sun, Deepak Ajwani, Patrick K. Nicholson, Alessandra Sala, and Srinivasan Parthasarathy. Breaking cycles in noisy hierarchies. In *Proceedings of the 2017 ACM on Web Science Conference*, WebSci '17, pages 151–160, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4896-6. doi: 10.1145/3091478.3091495. URL `http://doi.acm.org/10.1145/3091478.3091495`.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2020.

Steven Tadelis. *Game theory: an introduction*. Princeton university press, 2013.

Amos Tversky and Daniel Kahneman. The framing of decisions and the psychology of choice. *science*, 211(4481):453–458, 1981.

John von Neumann and Oskar Morgenstern. Theory of games and economic behavior, 2nd rev. 1947.

Abraham Wald. An essentially complete class of admissible decision functions. *The Annals of Mathematical Statistics*, pages 549–555, 1947.

Eliezer Yudkowsky and Alexei Andreev. Ontology identification problem: Technical tutorial. 2016. URL `https://arbital.com/p/ontology_identification/`.

Eliezer Yudkowsky, Eric Bruylant, and Eric Rogstad. Rescuing the utility function. 2016. URL `https://arbital.com/p/rescue_utility/`.

Junlong Zhang and Yu Luo. Degree centrality, betweenness centrality, and closeness centrality in social network. In *2017 2nd international conference on modelling, simulation and applied mathematics (MSAM2017)*, pages 300–303. Atlantis press, 2017.