Greedy Equivalence Search in the Presence of Latent Confounders

Abstract

We present Greedy PAG Search (GPS) for score-based causal discovery over equivalence classes, similar to the famous Greedy Equivalence Search algorithm [Chickering, 2002b], except now in the presence of latent confounders. It is based on a novel characterization of Markov equivalence classes for MAGs, that not only improves on state-of-the-art identification of Markov equivalence between MAGs, but also allows for efficient traversal over equivalence classes in the space of all MAGs. The resulting GPS algorithm is evaluated against several existing alternatives and found to show promising performance, both in terms of speed and accuracy.

1 INTRODUCTION

Ever since the advent in the early 90's of modern, principled methods for causal discovery from observational data, there have been two main paradigms that have been widely employed: constraint-based and score-based methodologies. Both start from the assumption that there is some underlying causal structure, typically in the form of a directed acyclic graph (DAG), that is responsible for the observed data distribution. The first class of methods then search for (conditional) in/dependence constraints between variables in the data, and use this information in combination with certain orientation rules to reconstruct the output causal model. Key assumptions include the *causal Markov assumption*, essentially stating that the structure of the underlying graph induces independence constraints in the observed data according to the d-separation criterion (see below), as well as the *causal faithfulness assumption*, stating that these are also the only observable independencies in the data. Other simplifying model assumptions like acyclicity and causal sufficiency (no latent confounders) can also be employed. When causal sufficiency does not apply the target causal model can be represented as a (maximal) ancestral graph (MAG, see below). The output then represents the so-called Markov equivalence class (MEC) of the underlying causal model, in the form of a partial ancestral graph (PAG) representing all causal graphs that satisfy the same independence model. Benchmark examples of algorithms in this tradition include PC and FCI [Spirtes et al., 2000], where the latter is sound and complete even in the presence of latent confounders and selection bias.

In contrast, score-based approaches define a metric that quantifies how well a certain graph structure captures the observed data, and then iteratively try to search for a graph that maximizes this score. The score is typically based on a (Bayesian) likelihood in combination with a penalty on model complexity, and usually assumes an underlying DAG structure with no unobserved confounders. A classic example is the K2 algorithm by Cooper and Herskovits [1992],

In many cases, it is possible to choose a score in such a way that all graphs in the same equivalence class obtain the same score [Heckerman et al., 1995]. As there can be a huge number of graph instances in the same equivalence class, this opens up the possibility of significantly speeding up the search by moving between equivalence classes rather than between individual graphs. This was the motivation behind algorithms like GBPS [Spirtes and Meek, 1995], and its famous successor GES (Greedy Equivalence Search) [Chickering, 2002b], as well as recent versions improving scaling behaviour and statistical efficiency [Ramsey et al., 2017, Chickering, 2020]. As a result, the output of these algorithms is a graph representing the equivalence class of the underlying causal model, similar to constraint-based methods. However, due to the global nature of the score, their output tends to be more robust than that of their constraint-based counterparts. Unfortunately, like PC, they also assume causal sufficiency, meaning that there is currently no available method that can employ the full potential of score-based equivalence search in the presence of latent confounders. Addressing this gap is the focus of this article.

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.

Towards equivalence search for MAGs

There have been several related score-based methods in recent years that try to go beyond the standard DAG search. For example Triantafillou and Tsamardinos [2016] consider the relative performance of constraint-based methods vs. MAG search using the BIC score for multivariate Gaussian distributions from [Richardson and Spirtes, 2002]. Their GSMAG algorithm employed a greedy search over the space of MAGs, where at each step all possible single edge modifications were evaluated. GSMAG was found to have promising performance, albeit at much greater running times.

More recently an alternative approach was taken by Ogarrio et al. [2016]. They managed to circumvent the MAG equivalence search by exploiting the original GES to first do equivalence search in the space of DAGs, and then to add a post-processing step using a modification of FCI that started from the GES output in order to obtain the final PAG. The result was a hybrid method (GFCI, short for Greedy FCI)) that showed promising performance over either method separately, but did not exploit the potential of full PAG search.

In the meantime many transformational characterizations of MAGs have been developed, see e.g. [Tian, 2012, Zhang and Spirtes, 2012], showing that we can reach all MAGs within the same equivalence class by a series of (covered) edge reversals to go from one MAG to the next where all are part of the same MEC. But as these characterizations are primarily concerned with transformations *within* the same equivalence class, they are not easy to generalize into an orthogonal search strategy *between* equivalence classes.

Our solution to this problem is based on a novel MEC characterization for MAGs that does not rely on complicated paths but on straightforward collider/noncollider triples. Any change to these triples implies a new MEC, which makes it easy to generate a collection of neighbouring MECs. In combination with an appropriate score this then forms the main engine in our Greedy PAG Search (GPS) algorithm for score-based equivalence search in the presence of latent confounders.

The rest of the article is organised as follows: section 2 introduces some basic concepts and terminology, section 3 describes the new characterization for Markov equivalence between MAGs, section 4 discusses how to use this for traversal between equivalence classes in the MAG space, ultimately leading to the GPS algorithm in section 5. Section 6 then shows the performance of GPS in practice compared to some state-of-the-art alternatives.

2 NOTATION AND TERMINOLOGY

A *mixed graph* $\mathcal{G}$ is a graphical model that can contain three types of edges between pairs of nodes: directed ($\rightarrow$), bi-directed ($\leftrightarrow$), and undirected ($-$). In a mixed graph, standard graph-theoretical notions, e.g. *child/parent*, *ancestor/descendant*, *directed path, cycle*, still apply, with natural extension to sets. A vertex z is a *collider* on a path $\pi = \langle \ldots, x, z, y, \ldots \rangle$ if there are arrowheads at z on both edges from x and y, otherwise it is a *noncollider*. A triple $x - z - y$ on a path is *unshielded* if x and y are not adjacent in $\mathcal{G}$. An unshielded collider is known as *v-structure*.

A mixed graph $\mathcal{G}$ is *ancestral* iff an arrowhead at x on an edge to y implies there is no directed path from x to y in $\mathcal{G}$, and there are no arrowheads at nodes with undirected edges. As a result, arrowhead marks can be read as 'is not an ancestor of'. In a mixed graph $\mathcal{G}$, a vertex x is *m-connected* to y by a path π, relative to a set of vertices Z, iff every noncollider on π is not in Z, and every collider on π is an ancestor of Z. If there is no such path, then x and y are *m-separated* by Z. An ancestral graph is *maximal* (MAG) if for any two nonadjacent vertices there is a set that m-separates them. A *directed acyclic graph* (DAG) is a special kind of MAG, containing only $\rightarrow$ edges, for which *m*-separation reduces to the standard *d*-separation criterion. For more details, see [Koller and Friedman, 2009, Spirtes et al., 2000].

A *causal DAG* $\mathcal{G}_C$ is a directed acyclic graph where the arcs represent direct causal interactions [Pearl, 2009]. In general, the independence relations between observed variables in a causal DAG can be represented in the form of a MAG [Richardson and Spirtes, 2002]. The (complete) partial ancestral graph (PAG) represents all invariant features that characterize the equivalence class $[\mathcal{G}]$ of such a MAG, with a tail '$-$' or arrowhead '$>$' end mark on an edge, iff it is invariant in all $[\mathcal{G}]$, otherwise it has a circle mark '$\circ$', see [Zhang, 2008].

3 CHARACTERIZING MARKOV EQUIVALENCE CLASSES

In this section we introduce a modified characterization for the Markov equivalence class (MEC) of (maximal) ancestral graphs (MAGs), that will form the basis for the equivalence search in the next section. It also leads to a simple method to establish Markov equivalence between MAGs.

3.1 MECS OF MAGS

For Markov equivalence between MAGs we start from the following characterization from Ali et al. [2009]:

Lemma 1 *Two MAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ belong to the same Markov equivalence class if and only if they have the same skeleton and the same colliders with order.*

This reflects the well known characterization for DAGs where two members are in the same equivalence class iff they have the same skeleton and *v*-structures, with the latter now generalized to 'collider triples with order':

Definition 1 *Let $\mathfrak{T}_i (i \geq 0)$ be the set of* triples of order i *in a MAG $\mathcal{G}$, defined recursively as:*

- *A triple $\langle a, b, c \rangle \in \mathfrak{T}_0$ if $a *\!\!-\!\!* b *\!\!-\!\!* c$ is in $\mathcal{G}$, with a and b not adjacent.*
- *A triple $\langle a, b, c \rangle \in \mathfrak{T}_{i \geq 1}$ if $\langle a, b, c \rangle \notin \mathfrak{T}_{j<i}$, and there is a discriminating path $\langle x, q_1, .., q_p, a, b, c \rangle$ for b in $\mathcal{G}$ (possibly $q_1 = a$), where the $p + 1$ colliders $\langle x, q_1, q_2 \rangle, ..., \langle q_{p-1}, q_p, b \rangle, \langle q_p, a, b \rangle \in \bigcup_{j<i} \mathfrak{T}_j$.*

Note that triples $\langle a, b, c \rangle$ and $\langle c, b, a \rangle$ are equivalent, and that triples with order $i \geq 1$ are triangles in $\mathcal{G}$. Also note that the final condition is only needed to uniquely determine the order i, but that the characterization itself does not depend on the actual value. This characterization leads to an algorithm for testing Markov equivalence between two MAGs with polynomial complexity $O(ne^4)$, with n the number of vertices and e the number of edges in the graph Ali et al. [2009].

More recently, Hu and Evans [2020] came up with another characterization in terms of a parameterizing set $\mathcal{S}_3(\mathcal{G})$ based on so-called *heads* and *tails* of the districts (connected bi-directed components) in $\mathcal{G}$, and the '3' indicates only sets of up to 3 nodes are required. In contrast with Ali et al. [2009] it does *not* rely on the discriminating path, and as a result leads to an even more efficient algorithm for checking equivalence that runs in $O(ne^2)$ for sparse graphs (when $n = O(e)$). Unfortunately, this characterization is difficult to translate into a comprehensive search strategy between equivalence classes.

However, it turns out that we can also circumvent the discriminating path in definition 1 in another way.

3.2 A NEW 'TRIPLES WITH ORDER' CHARACTERIZATION

On closer inspection of the second part of Definition 1 we see that every discriminating path (see Figure 1) can be viewed as a collection of collider and noncollider triples with order. More importantly, to know that a path $\langle x, q_1, .., q_p, z, y \rangle$ is a valid discriminating path for z in $\mathcal{G}$ it suffices to know that $\langle x, q_1, .., q_{p-1}, q_p, y \rangle$ is a valid discriminating path for noncollider q_p along the path, and that $\langle q_{p-1}, q_p, z \rangle$ is a collider, and that z and y are adjacent in $\mathcal{G}$. But that also means we do not actually need the full discriminating path, but we just need to know that $\langle q_{p-1}, q_p, y \rangle$ is a noncollider with order, and that $\langle q_{p-1}, q_p, z \rangle$ is a collider with order. This results in the following alternative characterization:

Definition 2 *Let $\mathfrak{C}_i$ resp. $\mathfrak{D}_i$ $(i \geq 0)$ be the set of collider- resp. noncollider triples with order i in a MAG $\mathcal{G}$, defined recursively as:*

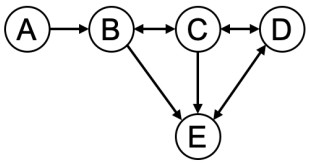

Figure 1: MAG with discriminating paths A-B-C-(D)-E.

k	$\mathfrak{C}$		
0	A	B	C
0	B	C	D
0	B	E	D
1	C	D	E

k	$\mathfrak{D}$		
0	A	B	E
1	B	C	E

Table 1: Corresponding 'triples with order' lists.

- *A triple $\langle a, b, c \rangle \in \mathfrak{C}_0$ (resp. $\mathfrak{D}_0$), if $a - b - c$ is an unshielded collider (resp. noncollider) in $\mathcal{G}$.*
- *A triple $\langle a, b, c \rangle \in \mathfrak{C}_i$ (resp. $\mathfrak{D}_i$), with $i \geq 1$, if $\langle a, b, c \rangle \notin \mathfrak{C}_{j<i}$ (resp. $\mathfrak{D}_{j<i}$), and*
  1. *$a - b - c$ is a collider (resp. noncollider) in $\mathcal{G}$,*
  2. *$\exists q : \langle q, a, b \rangle \in \mathfrak{C}_{k<i}$, and $\langle q, a, c \rangle \in \mathfrak{D}_{j<i}$.*

The connection to the original 'triple with order' definition follows from the next lemma (proof in the supplement):

Lemma 2 *In a MAG $\mathcal{G}$, a triple $\langle a, b, c \rangle$ is in $\mathfrak{C}_i$ (resp. $\mathfrak{D}_i$), if and only if $\langle a, b, c \rangle \in \mathfrak{T}_i$ and $\langle a, b, c \rangle$ is a collider (resp. noncollider) in $\mathcal{G}$.*

This motivates the following definition:

Definition 3 *The MEC $\mathcal{M}$ of a MAG $\mathcal{G}$, denoted $\mathcal{M}(\mathcal{G})$, is defined as the triplet $\langle \mathcal{S}, \mathfrak{C}, \mathfrak{D} \rangle$, with $\mathcal{S}$ the (undirected) skeleton of $\mathcal{G}$, and $\mathfrak{C}$ and $\mathfrak{D}$ the corresponding lists of collider resp. noncollider triples with order from Definition 2.*

Which leads to the straightforward implication:

Corollary 3 *Two MAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent if and only if $\mathcal{M}(\mathcal{G}_1) = \mathcal{M}(\mathcal{G}_2)$.*

From here on we will use the term MEC to denote this particular representation of the Markov equivalence class of a MAG $\mathcal{G}$.

3.3 FROM MAG TO MEC

Definition 2 implies that after we established the unshielded (non-)collider triples with order 0, we we only need to check the already constructed lists and a specific non/collider triple in the graph $\mathcal{G}$ in order to identify each higher order triple. This leads to the following **MAG-to-MEC** procedure:

Algorithm 1 MAG-to-MEC

Input: MAG $\mathcal{G}$
Output: MEC $\{\mathcal{S}, \mathfrak{C}, \mathfrak{D}\}$
phase 1: initialise, process unshielded triples
$\mathcal{S} \leftarrow Skeleton(\mathcal{G})$
$\mathfrak{C}_0/\mathfrak{D}_0 \leftarrow$ unshielded (non)colliders $\langle x, z, y\rangle \in \mathcal{G}$
for all $\langle x, z, y\rangle \in \mathfrak{D}_0$ **do**
  if $\exists q : \langle x, z, q\rangle \in \mathfrak{C}_0$ and $\mathcal{G}(q, y) > 0$ **then**
    $\mathfrak{L} \leftarrow \langle z, q, y\rangle$ {*initialise process list* $\mathfrak{L}$}
  end if
end for
phase 2: process candidate triples until no more left
repeat
  $\langle x, z, y\rangle \leftarrow Pop(\mathfrak{L})$
  if $x *\!\!\rightarrow z \leftarrow\!\!* y$ in $\mathcal{G}$ **then**
    add $\langle x, z, y\rangle$ to $\mathfrak{C}$
    $\forall q : \langle x, z, q\rangle \in \mathfrak{D}, \mathcal{G}(q, y) > 0$: add $\langle z, y, q\rangle$ to $\mathfrak{L}$
  else
    add $\langle x, z, y\rangle$ to $\mathfrak{D}$
    $\forall q : \langle x, z, q\rangle \in \mathfrak{C}, \mathcal{G}(q, y) > 0$: add $\langle z, y, q\rangle$ to $\mathfrak{L}$
  end if
until $\mathfrak{L}$ is empty
return $\mathcal{S}, \mathfrak{C}, \mathfrak{D}$

Algorithm 1 gives a high-level overview of the corresponding steps.[1]. It starts by identifying all unshielded triples (order 0) and allocating them to the appropriate collider or noncollider lists. After that all triples with order 1 are collected in list $\mathfrak{L}$, and processed one by one depending on whether they correspond to a collider or noncollider in the graph. Each allocated triple may give rise to new triples with order that are added to the end of the list $\mathfrak{L}$, until we have found them all. For each processed triple (allocated to $\mathfrak{C}$ or $\mathfrak{D}$) we only need to consider the existence of matching triples in the complementary list together with the presence of a specific edge in the MAG to find the new implied (higher order) triples. Table 1 shows the output $\mathfrak{C}$ and $\mathfrak{D}$ lists given the MAG in Figure 1.

3.4 FROM MEC BACK TO MAG

For the reverse **MEC-to-MAG** direction we can employ a more direct version of the standard FCI orientation procedure, where we first directly map all triples with order into specific (minimal) edge mark orientations to obtain the so-called **core PAG** (definition 4), and then propagate the remaining implied orientations using the rules $\mathcal{R}1 - \mathcal{R}3, \mathcal{R}5 - \mathcal{R}10$ from Zhang [2008] to obtain the completed PAG.

Definition 4 ***(core PAG)*** *For a MEC $\mathcal{M} = \langle \mathcal{S}, \mathfrak{C}, \mathfrak{D}\rangle$, the core PAG $\mathcal{P}^*$ is defined as the graph obtained from the skeleton $\mathcal{S}$ with all $\circ\!\!-\!\!\circ$ edges, in combination with*

- $\forall \langle x, z, y\rangle \in \mathfrak{C}_0$: *orient* $x *\!\!\rightarrow z \leftarrow\!\!* y$ *in* $\mathcal{P}^*$
- $\forall \langle x, z, y\rangle \in \mathfrak{C}_{k\geq 1}$: *orient* $z \leftarrow\!\!* y$ *in* $\mathcal{P}^*$
- $\forall \langle x, z, y\rangle \in \mathfrak{D}_{k\geq 1}$: *orient* $z —\!\!* y$ *in* $\mathcal{P}^*$

Each collider with order 0 becomes a *v*-structure, and each triple with order $k \geq 1$ corresponds to exactly one invariant edge mark (arrowhead or tail) in the graph. Note that in processing triples $\langle x, z, y\rangle$ with order $k \geq 1$, we rely on the fact that they are stored in the lists such that the y entry corresponds to the final node in a discriminating path, which is easily done when constructing the MEC.

The justification for the notion of a'core PAG' is that the resulting graph contains all invariant information needed to uniquely establish the full, completed PAG, by only propagating the graphical FCI orientation rules, i.e. *without* the *v*-structure rule $\mathcal{R}0$ and the discriminating path rules $\mathcal{R}4a/b$ in [Zhang, 2008] that both require a specific independence test result:

Algorithm 2 MEC-to-PAG

Input: MEC $\{\mathcal{S}, \mathfrak{C}, \mathfrak{D}\}$
Output: completed PAG $\mathcal{P}$
$\mathcal{P} \leftarrow \mathcal{P}^*(\mathcal{S}, \mathfrak{C}, \mathfrak{D})$ (*the core PAG from definition 4*)
run orientation rules $\mathcal{R}1 - \mathcal{R}3$ on $\mathcal{P}$ (*all arrowheads*)
run orientation rules $\mathcal{R}5 - \mathcal{R}10$ on $\mathcal{P}$ (*remaining tails*)
return $\mathcal{P}$

The following lemma ensures the output is indeed sound and complete:

Lemma 4 *For a valid MEC $\mathcal{M}$, algorithm 2 will output the corresponding completed PAG $\mathcal{P}$*

Note that the orientation rules also remain sound for arbitrary subsets $\mathfrak{C}' \subseteq \mathfrak{C}$, $\mathfrak{D}' \subseteq \mathfrak{D}$ of triples with order from the MEC, provided that all colliders with order zero (v-structures) are present in $\mathfrak{C}'_0$.

Finally, once we have the completed PAG we can obtain a matching MAG instance by following the arc-augmentation procedure in Theorem 2 of [Zhang, 2008] which will result in a fully oriented MAG in the same MEC with a minimum number of (invariant) bi-directed and undirected edges.

3.5 ALGORITHMIC COMPLEXITY

Checking for Markov equivalence between two MAGs simply corresponds to building the MEC for one, and verifying that the same steps apply to the other. This will induce at most a constant factor for each entry in the MEC (skeleton or triple lists), and so the algorithmic complexity for increasing graph sizes/densities is determined by the complexity of building the MEC from a given MAG.

[1] All software will be made available on publication.

To estimate the worst-case time complexity of algorithm 1 consider graphs over n nodes with e edges and max. node degree d. For sparse graphs with $d \leq k$ we have $e = O(n)$, whereas in general we can have $e = O(n^2)$.

Now the first phase of the algorithm requires finding all unshielded triples, which means selecting all pairs of two nodes from the neighbours of every node in the graph, leading to $n \cdot d \cdot (d-1) = O(nd^2)$ triples. For the initialization of the temporary triple list $\mathfrak{L}$ we need to check all triples $\langle x, y, z \rangle$ in $\mathfrak{C}_0$, and compare with specific entries in the complementary list $\mathfrak{D}_0$ (or vice versa) for nodes adjacent to z in $\mathcal{G}$. With appropriate indexing that implies an additional d candidates to check for each entry in the smaller of the two lists bringing the total for phase 1 to $O(nd^3)$.

Each entry in the temporary list is then processed and compared against d other candidates, each of which can be handled in constant time as it involves only verifying presence in one of the non/collider triple lists, which can again be done in constant time using appropriate indexing, and the presence of a specific edge in $\mathcal{G}$, also in constant time. Each combination added corresponds to a triangle in the graph, meaning there are at most $O(nd^2)$ triples to process, where each requires checking d entries, again leading to a combined total of $O(nd^3)$ steps for phase 2.

Together that means for sparse graphs we have worst case linear complexity of $O(n)$ (!), whereas in general this leads to $O(n^4)$. This is actually a significant improvement over the $O(ne^2)$ complexity reported by Hu and Evans [2020], corresponding to $O(n^3)$ for sparse graphs and $O(n^5)$ for arbitrary density (when $e = O(n^2)$).

Note that these complexity results relate to the worst-case scaling behaviour, and that in practice the typical performance may scale much better. For example the empirical complexity for sparse graphs in Hu and Evans [2020] seemed much closer to our linear result, meaning that in practice the two characterizations may be expected to perform very similar (see section 6.1). Apart from the nice-to-have worst-case scaling guarantee, the main contribution of our new representation therefore lies in the way it will allow us to traverse the MEC space in the next section.

4 MOVING BETWEEN EQUIVALENCE CLASSES

The main goal of this article is to find a search strategy that will allow us to move directly from one equivalence class to another as the basis for a score-based causal discovery algorithm, analogous to GES for DAGs [Chickering, 2002a], but now in the presence of latent confounders.

A natural line of attack might be to consider the complete PAG representation of the Markov equivalence class of a MAG, containing all invariant edge marks, and directly making changes to that in order to obtain a new PAG. However, this turns out to be less straightforward than maybe anticipated. For example, many of the invariant edge marks in the PAG are themselves implied by other invariant edge marks, and cannot be changed in isolation without invalidating the PAG. More tricky is the fact that it is not a priori clear whether changing, say, an invariant arrowhead, would imply an invariant tail or a circle mark. Alternatively, certain circle marks cannot be changed into either an invariant tail or arrowhead mark, whereas others can. Even then there may be cases where a mark can only be changed when other marks are also changed at the same time, but then the question becomes how to determine which one(s).

Going to MAG instances of a specific PAG does not immediately solve this problem, as for example certain invariant marks can be changed in some MAG instances of the PAG, but not others. This holds in particular for changes from invariant to non-invariant edge marks and vice versa. So even though it is possible to validate afterwards whether a certain modified PAG is valid, it is not easy to guarantee or verify whether all possible neighbouring instances are/can be reached.

This is where the new 'triples with order' characterization from section 3.2 comes in. It gives a characterization of the MEC $\mathcal{M}$ of a MAG $\mathcal{G}$ in terms of the triple $\langle \mathcal{S}, \mathfrak{C}, \mathfrak{D} \rangle$ corresponding to the skeleton and the lists of collider resp. noncollider triples with order. Any MAG with a different skeleton $\mathcal{S}'$ or a triple that is not present in either $\mathfrak{C}$ or $\mathfrak{D}$ corresponds to a different MEC, and so belongs to a different equivalence class. Moreover, if for a given triple with order the source collider and noncollider triples in point 2. of Definition 2 are still present in the MEC, then changing the triple corresponds to swapping the entry from $\mathfrak{C}$ to $\mathfrak{D}$ or vice versa. Doing so cannot affect (invalidate) lower order triples leading up to the changed triple, but it can in turn invalidate and/or create other/new triples with (higher) order.

Therefore we can move directly between equivalence classes by making single element changes to the MEC, provided that these still correspond to a valid MAG/MEC.

4.1 OPERATORS FOR MEC SPACE TRAVERSAL

Analogous to GES [Chickering, 2002a] we define a set of operators on a MEC $\mathcal{M} = \langle \mathcal{S}, \mathfrak{C}, \mathfrak{D} \rangle$ that will allow us to traverse MEC space directly. They are also complete in the sense that they suffice to move from any MEC of a MAG over n nodes to any other in a tight upper-bounded number of steps.

- **AddEdge** - insert an edge between two nodes in $\mathcal{S}$,
- **DeleteEdge** - remove a single edge from $\mathcal{S}$,
- **MakeNoncollider** - move a triple $\langle x, z, y \rangle$ in $\mathfrak{C}$ to $\mathfrak{D}$,
- **MakeCollider** - move a triple $\langle x, z, y \rangle$ in $\mathfrak{D}$ to $\mathfrak{C}$.

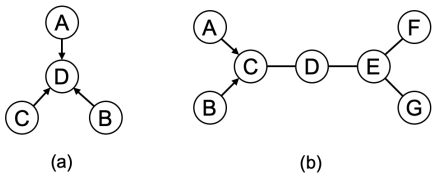

Figure 2: (a) Two colliders implying a third, (b) Core PAG with potential 'long distance' inconsistency on changing triple F-E-G.

Each operator by definition leads to a new MEC. The first two allow to move between MECs with different skeletons, whereas the later two allow to move between equivalence classes on the same skeleton. A single application of one of the operators above can lead to many implied changes, creating as well as destroying multiple 'triples with order'. It means 'neighbouring' MECs can be very different, allowing for large(r) steps to be taken, which should help avoid getting stuck in local optima.

However, we also need to take care of some subtle complications. In particular we need to ensure that the result of each step ultimately represents a valid MEC.

For example it is possible that one triple is implied by others, e.g. in Figure 2(a) two *v*-structures imply the third, and so one cannot be altered in isolation without being overruled or introducing an inconsistency. The same holds for changes that may be ok locally, but can lead to long-distance conflicts, e.g. in Figure 2(b) changing noncollider triple $\langle F, E, G \rangle$ into a collider implies that there must be an additional collider somewhere in between. Also, certain combinations of collider/noncollider triples can imply a MAG that is no longer ancestral, e.g. introducing arrowheads at nodes with undirected edges, or creating an (almost) directed cycle. Finally, a newly created 'triple with order' can often be assigned as either collider or noncollider, and so we have to choose on a default extension (noncollider), or consider both options.

Inevitably it means that post-processing in the form of an additional validity check is needed to ensure the changes result in different but valid MECs.

4.2 USING THE CORE PAG TO GENERATE NEIGHBOURING MECS

We can use the core PAG from 4 to simplify keeping track of which changes to the MEC imply which others. As indicated before, there is a one-to-one correspondence between invariant edge marks in the core PAG and entries in the collider/noncollider lists in the MEC. Each of the operators corresponds to altering a specific element in the core PAG, depending on the order of the triple:

Operator $AddEdge(x, y)$ inserts a non-committed edge $x \circ\!\!-\!\!\circ y$ in the core PAG $\mathcal{P}^*$, while $DeleteEdge(x, y)$ removes the edge between x and y from $\mathcal{P}^*$. The $MakeCollider(x, z, y)$ operator orients $x *\!\!\rightarrow z \leftarrow\!\!* y$ in $\mathcal{P}^*$ if it is an unshielded triple in $\mathfrak{D}_0$, or orients $z \leftarrow\!\!* y$ for triples in $\mathfrak{D}_{k\geq 1}$. Similarly, $MakeNoncollider(x, z, y)$ puts a tail mark at $z —\!\!* y$ in $\mathcal{P}^*$. If $\langle x, y, z \rangle \in \mathfrak{C}_0$, it also considers putting a tail mark at $x *\!\!— z$ (both destroying the original collider in $\mathcal{P}^*$). In addition, each time an arrowhead is introduced at a node z with undirected edges in the starting PAG, the tail marks at z on those edges are also turned into arrowheads, as by definition a MAG cannot have arrowheads at nodes with undirected edges (see section 2).

Each operator can create and/or destroy multiple other triples with order. As a result, some of the invariant edge marks in $\mathcal{P}^*$ may no longer be invariant, or new invariant edge marks may be introduced. To recognise which ones we can simply rebuild the new MEC from the modified core PAG. Doing so we have the option to either pick a single default (noncollider) for new, undetermined triples with order, or add one new MEC for each possible option.

The result of each operator by default is a new candidate MEC that needs to be validated to ensure it corresponds to a valid MAG. For that we can use Algorithm 2 to rebuild the corresponding completed PAG, which we can then expand into an arc-augmented MAG to check it is valid. The overhead introduced by this entire validation step is significant, but in practice still less than the contribution of the subsequent scoring step (section 5.1).

For a given starting MEC we can therefore consider all possible operator applications: changing each edge in the skeleton $\mathcal{S}$, and/or moving each single entry in $\mathfrak{C}$ to $\mathfrak{D}$ and vice versa, to generate a collection of neighbouring MECs, that each correspond to a valid MAG from a different equivalence class. The resulting procedure is shown in Algorithm 3.

Algorithm 3 MEC Candidate Neighbours

Input: MEC $\mathcal{M}$, active Operators
Output: collection of MEC $\{Neighbours\}$
$\mathcal{P}^* \leftarrow$ core PAG for $\mathcal{M}$
for all active Operators, target edges/triples **do**
  $\mathcal{P}'^* \leftarrow Modify(\mathcal{P}^*, operator, target)$ *(generate)*
  $\mathcal{M}' \leftarrow PAG\_to\_MEC(\mathcal{P}'^*)$
  $\mathcal{P}' \leftarrow MEC\_to\_PAG(\mathcal{M}')$ *(validate)*
  $\mathcal{G}' \leftarrow PAG\_to\_MAG(\mathcal{P}')$ *(arc-augmented)*
  if $IsValidMAG(\mathcal{G}')$ **then**
    $Neighbours\{end + 1\} \leftarrow \{\mathcal{M}', \mathcal{P}', \mathcal{G}'\}$ *add*
  end if
end for
return $\{Neighbours\}$

A key result for this approach is that the 4 operators suffice to traverse the entire MEC space, where any MEC can be reached from any other MEC over the same nodes in an upper bounded number of steps, and where each intermedi-

ate step corresponds to a different equivalence class, (see Theorem 1 in Appendix B).

5 GREEDY PAG SEARCH

Having an efficient representation for equivalence classes in the form of the MEC (2), in combination with a procedure to obtain different neighbouring MECs (Algorithm 3), all that remains to turn this into an effective search procedure is a means to score individual MECs in order to find the optimal PAG $\mathcal{P}$ that best describes the data. For simplicity, in this article we will assume a multivariate Gaussian model.

5.1 SCORING MECS

When moving between MECs, algorithm 3 maps each MEC to an arc-augmented MAG instance to verify validity. Given that for multivariate Gaussian models Richardson and Spirtes [2002] already introduced a well-established MAG score, we will rely on that as an associated score for the corresponding MEC.

Due to space limitations, and because it is already part of the literature, we will relegate the description of the score itself to Appendix C in the supplement. For details see also [Nowzohour et al., 2017, Triantafillou and Tsamardinos, 2016].

5.2 THE GPS ALGORITHM

Having developed all the necessary tools we can now put them together into the (baseline) Greedy PAG Search (GPS) algorithm below, where the 'PAG' in the name signifies it searches between equivalence classes. It starts from an empty model and each time greedily tries to find different, neighbouring MECs that will improve the score the most, until no more improvements can be found.

Algorithm 4 Greedy PAG Search

Input: Gaussian covariance Σ over N variables
Output: optimal matching PAG $\mathcal{P}$, top score s
Initialise: $\mathcal{M} \leftarrow$ empty MEC over N variables, $s \leftarrow 0$
repeat
  $\{\mathbf{M}\} \leftarrow Candidate\_Neighbours(\mathcal{M})$
  for all $\mathcal{M}_i \in \mathbf{M}$ **do**
    $s_i \leftarrow Score(\mathcal{M}_i)$
    if $s_i > s$ **then** $(\mathcal{M}, s) \leftarrow (\mathcal{M}_i, s_i)$
  end for
until no more improvement
return $\mathcal{P} \leftarrow MEC\_to\_PAG(\mathcal{M}), s$

There are many ways to improve the search strategy itself, in particular to speed up the overall search, and/or avoid getting stuck in (suboptimal) local maxima. Typical strategies involve tabu-search, multiple restarts, simulated annealing etc. Another option is to pursue the two-stage GES approach in [Chickering, 2020], first expanding the graph using insert operators, followed by a second, statistically efficient backward equivalence search to arrive at the optimal model. Alternatively, we could start from the output PAG of a constraint-based causal discovery algorithm like FCI/FCI+ [Spirtes et al., 2000, Claassen et al., 2013], and then run GPS to try and improve on that.

Many more options are available, however, due to space limitations we will leave most of these and other strategies as future work to explore.

Finally, note that the current GPS algorithm is only limited to Gaussian models due to the choice for this particular MAG score. However the search strategy itself applies to any Markov equivalence class. In particular we want to mention current work on incorporating a general equivalence score that also handles selection bias and cyclic interactions.

6 EXPERIMENTAL EVALUATION

6.1 MAG-TO-MEC COMPLEXITY

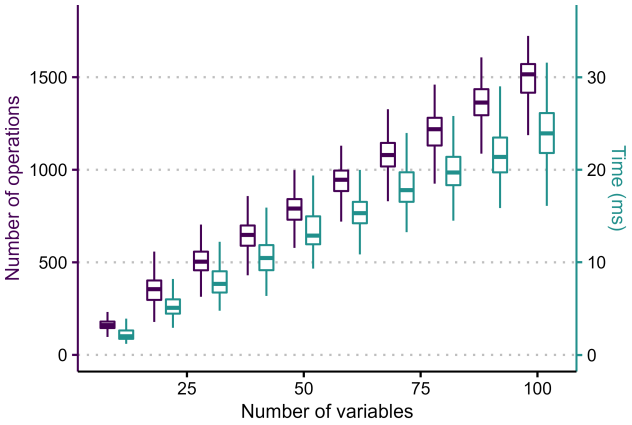

Figure 3: Empirical complexity **MAG-to-MEC**.

A crucial part of the proposed methodology is the new MEC characterization in terms of 'triples with order'. In Section 3.2, we derived that for sparse graphs the theoretical complexity of the **MAG-to-MEC** algorithm is $O(n)$. Figure 3 confirms this via the empirical complexity on random MAGs of size $n = \{10, 20, ..., 100\}$, each averaged over 250 graphs. Similar to the simulation in Hu and Evans [2020] the MAGs are generated to have approximately $e = 3n$ edges, (corresponding to $d = 6$), while each edge is (independently) either directed or bi-directed with probability $p = 0.5$. The results demonstrate a strong linear trend (even slightly better), both in terms of 'elementary operations' (purple) and raw computational time (cyan).

6.2 SIMULATION EXPERIMENTS

In this section, we evaluate the speed and accuracy of the **GPS** algorithm. We compare our method against the GSMAG algorithm proposed by Triantafillou and Tsamardinos [2016] and the GFCI algorithm proposed by Ogarrio et al. [2016], while also showing the results obtained with FCI as a baseline. We generated 100 MAGs for each graph size $n \in \{5, 10, 15, 20\}$, such that the average node degree was $d = 3$, the maximum node degree was $d_{\max} = 10$, and the probability of an edge being bi-directed (as opposed to directed) was $p = 0.2$.

We used the following metrics to evaluate the algorithm performance: 1. the Structural Hamming Distance (*SHD*), counting the number of different edges and/or edge marks between the output PAG and the ground truth PAG; 2. the Bayesian information criterion (*BIC*) score for MAGs as proposed by Triantafillou and Tsamardinos [2016]; and 3. the *accuracy* of edge marks, obtained as a Jaccard similarity coefficient, by dividing the number of correct edge marks in the output PAG by the total number of edge marks in the (skeleton) union of output and ground truth PAG.

Algorithm		GPS		GSMAG		GFCI	FCI
n	metric	empty	FCI	empty	FCI	N/A	N/A
5	SHD	9.180	9.040	9.730	8.600	10.360	10.640
	BIC	12.838	12.717	12.414	12.552	12.994	13.052
	accuracy	0.530	0.522	0.517	0.549	0.447	0.424
10	SHD	25.420	25.250	38.450	31.020	21.510	22.770
	BIC	30.616	30.438	28.923	28.753	31.716	31.735
	accuracy	0.470	0.460	0.322	0.416	0.481	0.454
15	SHD	35.700	37.800	62.900	53.640	29.990	34.100
	BIC	37.102	36.234	32.832	32.308	38.739	39.177
	accuracy	0.494	0.458	0.308	0.381	0.501	0.430
20	SHD	47.464	48.890	94.061	74.460	36.820	42.720
	BIC	60.542	59.368	54.524	54.226	63.534	63.904
	accuracy	0.523	0.508	0.292	0.389	0.552	0.462

Table 2: Algorithm accuracy comparison

The accuracy results are summarized in Table 2. For both GPS and GSMAG, we considered two different starting points for the greedy search, namely the empty graph and the PAG obtained by running the FCI algorithm. We used the BIC score for MAGs [Triantafillou and Tsamardinos, 2016] as the objective function in the greedy optimization. We ran FCI and GFCI using the Tetrad library [Glymour et al., 2014] with default parameters, where Fisher's z-test was used for finding conditional independences, and the BIC score was used for the score-based component of GFCI. Even though GSMAG manages to achieve a better local minimum for the objective function, the greedy PAG search finds graphs that are closer to the ground truth, as shown by the increased accuracy and lower SHD. On closer inspection this turns out to result from unstable BIC scores in GSMAG where the RICF fitting step fails to converge properly for many graphs considered by GSMAG. In contrast the equivalence class candidates proposed in GPS tend to be easier to compute leading to more reliable output. GPS performs slightly worse relative to GFCI in terms of SHD and accuracy, while obtaining a better likelihood fit for the data. This suggests further tweaks to the MEC score could be beneficial. Likewise going beyond the current baseline search for GPS (single run) should provide further improvements. When it comes to speed, GPS arrives at a result much faster than GSMAG due to the more efficient search through the space of equivalence classes, as shown in 3. This difference becomes more obvious when comparing the average time needed to run each algorithm, as GSMAG considers (and rejects) many more candidate graphs in each step of the search.

Algorithm		GPS		GSMAG	
n	metric	empty	FCI	empty	FCI
5	iterations	8.430	2.410	8.280	3.000
	time (s)	0.427	0.465	1.595	1.240
10	iterations	20.260	6.620	21.140	10.720
	time (s)	15.970	16.192	50.659	55.060
15	iterations	28.480	11.090	33.040	17.980
	time (s)	60.192	70.174	321.615	360.386
20	iterations	41.062	16.650	48.919	26.600
	time (s)	201.906	213.508	926.256	863.413

Table 3: Algorithm speed comparison

7 CONCLUSION

In this article we presented GPS, the first score-based equivalence search algorithm in the presence of latent confounders. It was based on a simple new MEC characterization for MAGs that brings establishing Markov equivalence between sparse graphs down to linear complexity. Experimental 'proof of concept' results on the baseline GPS version, confirmed our hopes/expectations that equivalence search could traverse the MAG space faster than single-edge MAG modifications, while arriving at slightly better models. It also showed that computing the BIC score itself for MAGs was in general considerably trickier than for DAGs, and that additional gains can be expected by incorporating more comprehensive search strategies like tabu-search and multiple restarts. Looking forward, we are working on extending GPS by including a more robust equivalence score that can also handle selection bias and cyclic interactions, and investigating how to optimize the global search.

References

R Ayesha Ali, Thomas S Richardson, and Peter Spirtes. Markov equivalence for ancestral graphs. *The Annals of Statistics*, 37(5B):2808–2837, 2009.

David Maxwell Chickering. Learning equivalence classes of bayesian-network structures. *The Journal of Machine Learning Research*, 2:445–498, 2002a.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002b.

Max Chickering. Statistically efficient greedy equivalence search. In *Conference on Uncertainty in Artificial Intelligence*, pages 241–249. PMLR, 2020.

Tom Claassen, Joris M Mooij, and Tom Heskes. Learning sparse causal models is not np-hard. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 172–181, 2013.

Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

Mathias Drton and Thomas S Richardson. Graphical methods for efficient likelihood inference in gaussian covariance models. *Journal of Machine Learning Research*, 9: 893–914, 2008.

Steven B Gillispie and Michael D Perlman. The size distribution for markov equivalence classes of acyclic digraph models. *Artificial Intelligence*, 141(1-2):137–155, 2002.

Clark Glymour, Richard Scheines, and Peter Spirtes. *Discovering causal structure: Artificial intelligence, philosophy of science, and statistical modeling*. Academic Press, 2014.

Yangbo He, Jinzhu Jia, and Bin Yu. Counting and exploring sizes of markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 16 (1):2589–2609, 2015.

David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

Zhongyi Hu and Robin Evans. Faster algorithms for markov equivalence. In *Conference on Uncertainty in Artificial Intelligence*, pages 739–748. PMLR, 2020.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Christopher Nowzohour, Marloes H Maathuis, Robin J Evans, and Peter Bühlmann. Distributional equivalence and structure learning for bow-free acyclic path diagrams. *Electronic Journal of Statistics*, 11(2):5342–5374, 2017.

Juan Miguel Ogarrio, Peter Spirtes, and Joe Ramsey. A hybrid causal search algorithm for latent variable models. In *Conference on probabilistic graphical models*, pages 368–379. PMLR, 2016.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International journal of data science and analytics*, 3(2): 121–129, 2017.

Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

Peter Spirtes and Christopher Meek. Learning bayesian networks with discrete variables from data. In *KDD*, volume 1, pages 294–299, 1995.

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

Jin Tian. Generating markov equivalent maximal ancestral graphs by single edge replacement. *arXiv preprint arXiv:1207.1428*, 2012.

Sofia Triantafillou and Ioannis Tsamardinos. Score-based vs constraint-based causal learning in the presence of confounders. In *Cfa@ uai*, pages 59–67, 2016.

Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17): 1873–1896, 2008.

Jiji Zhang and Peter L Spirtes. A transformational characterization of markov equivalence for directed acyclic graphs with latent variables. *arXiv preprint arXiv:1207.1419*, 2012.

A REMARK ON SIZE OF MECS

One may wonder whether searching between equivalence classes is actually worth the trouble, given the famous conclusion from Gillispie and Perlman [2002] that the average size of equivalence classes for DAGs is bounded below 4, even as n goes to infinity. This was all the more surprising given that experimental findings from e.g. [Chickering, 2002b] reported encountering huge sized equivalence classes.

As demonstrated by He et al. [2015], the main contribution to this bound comes from graphs with a high average density of around $n/2$ that account for the vast majority of graphs over n nodes, and for which nearly every instance is almost fully determined. But for sparse graphs with a density bounded by some constant $d \ll n$ the size of each individual equivalence class can become truly huge as n gets larger. For example [He et al., 2015] report an average MEC size of $3.5e19$ for DAGs over 50 nodes with average edge density of 4. Therefore despite some potential overhead, searching over equivalence classes rather than individual MAGs can still bring a sizeable improvement in efficiency.

B PROOFS

Below the proof details for the theoretical results in the main paper.

Lemma 2 *In a MAG $\mathcal{G}$, a triple $\langle a, b, c\rangle$ is in $\mathfrak{C}_i$ (resp. $\mathfrak{D}_i$), if and only if $\langle a, b, c\rangle \in \mathfrak{T}_i$ and $\langle a, b, c\rangle$ is a collider (resp. noncollider) in $\mathcal{G}$.*

Proof Clearly the definitions coincide for triples of order 0. First from old to new: if $\langle a, b, c\rangle \in \mathfrak{T}_1$ then there is a discriminating path $\langle x, a, b, c\rangle$ in $\mathcal{G}$ for which $\langle x, a, b\rangle$ is a collider triple with order 0, hence $\langle x, a, b\rangle \in \mathfrak{C}_0$, and $\langle x, a, c\rangle$ is a noncollider triple with order 0, $\langle x, a, c\rangle \in \mathfrak{D}_0$. Therefore all conditions for order $i = 1$ in the new definition are satisfied, and so $\langle a, b, c\rangle \in \mathfrak{C}_1$ resp. $\mathfrak{D}_1$, depending on whether the triple is a collider or noncollider in $\mathcal{G}$. By induction, suppose the mapping is valid up to order i, and let $\langle a, b, c\rangle \in \mathfrak{T}_{i+1}$. Then there is a discriminating path $\langle x, q_1, .., q_p, a, b, c\rangle$ in $\mathcal{G}$ for which $\langle q_p, a, b\rangle$ is a collider triple with order $k \leq i$, hence $\langle q_p, a, b\rangle \in \mathfrak{C}_k$, and for which $\langle q_p, a, c\rangle$ is a noncollider triple with order $j \leq i$, hence $\langle q_p, a, c\rangle \in \mathfrak{D}_j$. Therefore all conditions for order $i + 1$ in the new definition are satisfied, and so $\langle a, b, c\rangle \in \mathfrak{D}_{i+1}$ resp. $\mathfrak{C}_{i+1}$), again depending on whether the triple is a noncollider or collider in $\mathcal{G}$.

For the reverse, from new to old: at order $i = 1$, if $\langle a, b, c\rangle \in \mathfrak{D}_1$ then by definition there is a $\exists x : \langle x, a, c\rangle \in \mathfrak{D}_0$ as noncolllider triple, and also as collider triple $\langle x, a, b\rangle \in \mathfrak{C}_0$. But that implies $\langle x, a, b, c\rangle$ is a discriminating path in $\mathcal{G}$, and so $\langle a, b, c\rangle \in \mathfrak{T}_1$ as we already saw $\langle x, a, b\rangle \in \mathfrak{T}_0$. Similarly when $\langle a, b, c\rangle \in \mathfrak{C}_1$. Again by induction assuming the mapping is valid up to order i, and let $\langle a, b, c\rangle \in \mathfrak{D}_{i+1}$. Then $\exists q_p : \langle q_p, a, c\rangle \in \mathfrak{D}_{j\leq i}$ and $\langle q_p, a, b\rangle \in \mathfrak{C}_{k\leq i}$. If $j > 0$, then again there is a $q_{p-1} : \langle q_{p-1}, q_p, c\rangle \in \mathfrak{D}_{m<j}$ and $\langle q_{p-1}, q_p, a\rangle \in \mathfrak{C}_{n<k}$. The same holds for all subsequent triples until we arrive at some triple with order 0 for which $\langle x, q_1, c\rangle \in \mathfrak{D}_0$ and $\langle x, q_1, q_2\rangle \in \mathfrak{C}_r$. Then $\langle x, q_1, .., q_p, a, b, c\rangle$ is a discriminating path, where all required collider triples are of lower order than i and so also in $\bigcup \mathfrak{C}_{j<i}$. This implies $\langle a, b, c\rangle \in \mathfrak{T}_i$, which proves the lemma. ■

Corollary 3 *Two MAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent if and only if $\mathcal{M}(\mathcal{G}_1) = \mathcal{M}(\mathcal{G}_2)$.*

Proof Lemma 2 implies a MEC $\mathcal{M}(\mathcal{G})$ is unique and in a one-to-one correspondence with Lemma 1 which guarantees 'if and only if' Markov equivalence. ■

Lemma 4 *For a valid MEC $\mathcal{M}$, algorithm 2 will output the corresponding completed PAG $\mathcal{P}$.*

Proof (Rules following the notation in [Zhang, 2008].) Given the core PAG, all *v*-structures from rule $\mathcal{R}0$ are already included. In the eliminated discriminating path rule $\mathcal{R}4$, for the final 3 nodes $\langle .., \alpha, \beta, \gamma\rangle$ along a discriminating path all invariant edge marks at β on the edge to γ are also already covered in the core PAG via triples with order $k \geq 1$.

All other elements oriented by rule $\mathcal{R}4$ will get oriented by $\mathcal{R}2$. In particular: both $\mathcal{R}4a/b$ will also orient an arrowhead at γ on the edge to β, but this also follows directly from $\mathcal{R}2b$, as $\langle \alpha, \beta, \gamma\rangle$ together with already established arc $\alpha \rightarrow \gamma$ satisfy the precondition for R2b with the roles of α and β reversed, leading to the invariant arrowhead $\beta *\!\!\rightarrow \gamma$. For the remaining arrowhead orientation at $\alpha *\!\!\rightarrow \beta$ from rule $\mathcal{R}4b$, the final three nodes also satisfy the precondition for $\mathcal{R}2a$, except now with the roles of β and γ reversed.

All other individual orientation rules remain sound, so that all other rules triggered in creating the PAG by FCI can/will also be triggered when starting from the MEC, which means the output PAG is also sound and complete. ■

Theorem 1 *The MEC operators (in combination with post-validation) are sound and complete.*

Proof Soundness is trivially ensured by the final validity check on the resulting implied arc-augmented MAG. Completeness follows from the fact that we can transform any source MAG into an undirected MAG on the same skeleton by iteratively turning all *v*-structures (colliders with order zero) into non-colliders, each time choosing a

v-structure with no other *v*-structures among its ancestors in the current implied graph instance. Each step corresponds to moving a triple with order zero from $\mathfrak{C}$ to $\mathfrak{D}$ until no more colliders or higher order triples are left, and the MEC consists of only noncolliders with order zero. After that we can arbitrarily add and/or remove undirected edges to obtain an undirected MAG on the target skeleton. In the process of adding/removing edges, all newly created triples-with-order default to 'noncollider', meaning that the end result at each step remains an undirected graph, corresponding to a MEC with the same skeleton and only noncollider triples with order zero. Finally, starting with the invariant undirected components in the PAG implied by the current instance, we can iteratively turn noncolliders with order zero into colliders matching the target MAG (corresponding to moving an entry from $\mathfrak{D}$ to $\mathfrak{C}$ in the MEC, possibly creating new, by default noncollider triples with order $k \geq 1$), and repeat for increasing order of k until we reach the target MEC. ■

Naturally this is unlikely to be the optimal strategy, but it does serve to show that the operators in themselves are flexible enough to reach all possible instances of the entire MEC space. Furthermore, it also gives a straightforward upper bound on the number of steps required to transform one MAG into another

Corollary 5 *For two different MAGs $\mathcal{G}_1$ and $\mathcal{G}_2$, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the corresponding MECs in accordance with 2. Let ΔE be the number of edge differences in the skeleton between $\mathcal{G}_1$ and $\mathcal{G}_2$, then there is a sequence of at most $\Delta E + |\mathfrak{C}_1(0)| + |\mathfrak{C}_2|$ single step MEC operators that will transform $\mathcal{M}_1$ into $\mathcal{M}_2$ via a series of intermediate valid MECs.*

Proof Follows immediately from the strategy described above, while noting that each step in each of the three phases reduces the amount of differences for that stage by at least one while not introducing new differences for that stage, and at the end the two MECs should be equal. This results in an upper bound of $|\mathfrak{C}_1(0)|$ steps for phase 1 to obtain the noncollider MEC (undirected skeleton of $\mathcal{M}_1$), then exactly ΔE edge insertion/deletions in phase 2 (undirected skeleton of $\mathcal{M}_2$, followed by at most $|\mathfrak{C}_2|$ to introduce the required collider triples in phase 3 to end up at the target MEC. ■

In particular it also holds when $\mathcal{M}_1$ is our starting empty MEC, and $\mathcal{M}_2$ the target optimal/true MEC. Naturally again it does not imply that any search strategy (incl. GPS) is guaranteed to find that optimal solution in this amount of steps, but merely that in theory it is possible to reach the optimal solution within a reasonable number of steps. This bound can be tightened even further by finding an improved strategy that keeps as many of the shared collider triples as possible (avoiding the intermediate fully undirected graph), but that goes beyond the scope of the current article.

C SCORING MECS

This section describes the details behind the BIC score for MAGs [Richardson and Spirtes, 2002], used to score MECs as indicated in section 5.1.

To connect a MAG to a linear Gaussian model, we can associate a MAG $\mathcal{G}$ over $n = |\mathbf{V}|$ variables with a collection of $n \times n$ matrices of structural parameters $\mathbf{B}(\mathcal{G})$, with $B_{ij} = 0$ iff $i = j$ or $j \to i \notin \mathcal{G}$, and a collection of positive definite covariance matrices of error/noise terms $\mathbf{\Omega}(\mathcal{G})$, where $\Omega_{ij} = 0$ iff $i \neq j$ and $i \leftrightarrow j \notin \mathcal{G}$. Then the system of (normal) linear equations $\mathbf{V} = B\mathbf{V} + \boldsymbol{\epsilon}$ with $B \in \mathbf{B}(\mathcal{G})$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \Omega \in \mathbf{\Omega}(\mathcal{G}))$ implies a multivariate Gaussian distribution over $\mathbf{V}$ with covariance matrix $\Sigma = (I - B)^{-1}\Omega(I - B)^{-\mathrm{T}}$.

For any given choice of B and Ω we can compute the likelihood of the observed sample covariance matrix S. But for a given MAG $\mathcal{G}$ we only have the structure, not the parameters. As a reasonable approximation, for a given graph $\mathcal{G}$ we therefore compute the parameters that maximize this likelihood. For DAGs this boils down to straightforward regression, but for MAGs in general no such expression exists, even though they are uniquely identifiable. Instead we can employ the *residual iterative conditional fitting* (RICF) method developed by Drton and Richardson [2008] which iteratively finds the maximum likelihood solution for the parameters in the model given the graph $\mathcal{G}$ and observed sample covariance matrix S, and outputs the implied covariance matrix $\hat{\Sigma}$, from which we can compute the (log) likelihood of the sample covariance matrix S under the model covariance $\hat{\Sigma}$ for $\mathcal{G}$.

An attractive property, as shown by Nowzohour et al. [2017], is that this log-likelihood can be decomposed into a sum of distinct contributions over the separate districts (connected bi-directed components) in the graph $\mathcal{G}$. With each district D_k a so-called *c-component* C_k is associated, consisting of the subgraph $\mathcal{G}_k$ of $\mathcal{G}$ over the nodes in $D_k \cup \mathrm{pa}_{\mathcal{G}}(D_k)$, but with all edges between $\mathrm{pa}(C_k) \equiv \mathrm{pa}_{\mathcal{G}}(D_k) \setminus D_k$ removed. With this the log-likelihood given N samples becomes:

$$l(S|\hat{\Sigma}_{\mathcal{G}}) = -\frac{N}{2}\sum_k \Big(|C_k|\log 2\pi + \log\frac{|\Sigma_{\mathcal{G}_k}|}{\prod_{j\in \mathrm{pa}(C_k)}\sigma^2_{kj}} + \frac{N-1}{N}\,\mathrm{tr}(\Sigma^{-1}_{\mathcal{G}_k}S_{\mathcal{G}_k} - |\,\mathrm{pa}(C_k)|)\Big) \quad (1)$$

As a result, when computing the score for a modified MEC we only need to recompute the score for the c-components that changed relative to the source MEC, providing a signi-

ficant speed improvement for the overall computational cost. Note that here the use of the arc-augmented MAG extension for a PAG minimizes the size of the districts, which also benefits the speed and convergence of the RICF step for each district in the computation of the score.

To avoid overfitting the negative log-likehood is typically regularized by adding a complexity penalty for the number of free parameters. For that we will use the BIC score for MAGs from [Richardson and Spirtes, 2002], with n and e resp. the number of variables and edges in $\mathcal{G}$; see also [Triantafillou and Tsamardinos, 2016].

$$BIC(\hat{\Sigma}, \mathcal{G}) = 2l(S|\hat{\Sigma}_{\mathcal{G}}) - \log(N)(2n + e) \qquad (2)$$

Two final remarks: in practice, the score 2 is not guaranteed to be a fully equivalent score, as different MAG instances in the same equivalence class can have different sized districts, making it harder for the RICF step in 1 to converge to the same value. However, in theory in the large sample limit any MAG instance from the true equivalence class should obtain a higher score than any MAG that is not. Secondly, the current likelihood score 1 is only defined for directed graphs, meaning that MAGs with invariant undirected edges (identifiable selection bias) cannot be scored and are therefore skipped in the evaluation. It is possible to extend the score to include selection bias as well, but that is left to another article.