Optimal Correction Sets for Argumentative Causal Discovery

Fabrizio Russo[1], Giuseppe Mazzotta[2], Carmine Dodaro[2], Francesco Ricca[2] and Francesca Toni[1]

[1]*Imperial College London, London, United Kingdom*
[2]*University of Calabria, Rende, Italy*

Abstract
Causal Assumption-based Argumentation (ABA) has been proposed as a causal discovery method with increased guarantees on the correspondence of the discovered causal graphs to a subset of the input constraints that drive the search for the causal relations. Heuristics are currently used to identify the optimal subset of constraints and infer the most likely corresponding graphs. Minimal Unsatisfiable Sets (MUSes) and Minimal Correction Sets (MCSes) have been explored in the Answer Set Programming (ASP) literature to create explanations and repair logic programs (LPs). We leverage the correspondence of stable semantics between ABA and LPs to investigate the benefits and drawbacks of MUSes and MCSes when applied to the causal discovery task carried out by Causal ABA. We define the notion of optimal MCSes and show how they can be computed by leveraging standard optimisation constructs such as weak constraints. We then empirically show that optimal MCSes, integrated into Causal ABA for causal discovery, substantially (i) increase the identification rate of true constraints; (ii) reduce the number of compatible causal graphs in output; (iii) improve graph reconstruction according to standard metrics.

Keywords
causal discovery, assumption-based argumentation, answer set programming, minimal correction sets

1. Introduction

Causal discovery supports scientific understanding and decision making by turning observational and/or interventional data into hypotheses about the underlying causal mechanisms. It is a central ingredient in tasks such as explanation, prediction under interventions, and policy evaluation [1]. A wide range of approaches exists, including constraint-based methods relying on independence testing, score-based methods that optimise a goodness-of-fit criterion over graphs, and functional causal model approaches that leverage assumptions on the data-generating mechanism; see e.g. [2].

In the standard constraint-based setting, the Peter-Clark (PC) [3] algorithm recovers the true Markov equivalence class (MEC, the upper limit of what can be identified from observational data alone) [4] given correct conditional independence (CI) information and *faithfulness*, i.e., the assumption that the observed independences are exactly those entailed by the underlying causal graph (assumed to be directed and acyclic: DAG) [5]. Causal Assumption-based Argumentation (Causal ABA) [6] inherits analogous guarantees under faithful constraints, but, crucially, it *enforces* the correspondence between a selected set of facts and the output graphs: only graphs consistent with the enforced facts are admitted. A useful by-product is traceability, since each output can be linked back to the specific facts that were enforced. When the input facts are noisy and mutually conflicting, however, the encoding may admit no solution. Hence, selecting which facts to enforce and which to *release* is a central part of the pipeline; [6] address this via a strength-based heuristic that incrementally releases lower-ranked facts.

This paper builds upon methods from logic programs explanation and repair [7, 8, 9], in particular, Minimal Unsatisfiable Subsets (MUSes) and Minimal Correction Subsets (MCSes). Intuitively, MUSes expose *cores* of mutually conflicting rules explaining why an LP is incoherent, while MCSes allows to identify sets of rules whose removal restores consistency, both with minimality guarantees. We leverage these concepts to replace heuristic fact selection in Causal ABA with a more principled optimisation objective with optimality guarantees. Concretely, we: (i) cast Causal ABA's "released facts" step as a *repair* problem over constraints; (ii) define *optimal* MCSes to restore coherence while minimising a given cost function; (iii) show how this notion matches the objective optimised by weak constraints, yielding

KR / FLoC 2026 Workshop - LINDA 2026: Logical Approaches to Handling Inconsistent Data, July 24, 2026, Lisbon, Portugal

an efficient implementation to extract optimal MCSes from optimal stable models [10]; and (iv) integrate this optimisation in Causal ABA, evaluating OptABA-PC against ABA-PC. Optimisation-based causal discovery in (Max-)SAT/ASP is not novel [11, 12, 13, 14]; our contribution is to import optimal-MCS reasoning into Causal ABA while preserving traceability from accepted facts to returned graphs.

2. Motivation and Background

We use a running example to illustrate how noisy CI tests can be mutually inconsistent, how Causal ABA enforces a *traceable* subset of CI facts, and how MUS/MCS reasoning casts "releasing facts" as principled repair. A CI fact records a dependence/independence claim; in Causal ABA, *enforcing* a fact means requiring every returned graph to satisfy it, whereas *releasing* a fact means allowing it to be dropped to regain coherence. We write $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ for CI between variables X and Y given conditioning set $\mathbf{Z}$ ($X \perp\!\!\!\perp Y$ if $\mathbf{Z}$ is empty). A MEC is represented by a *completed partially DAG* (CPDAG [15]), whose directed edges are shared by all DAGs in the class, while undirected edges are reversible.

Our focus is the realistic regime where finite-sample CI tests are noisy and possibly conflicting. The Majority-PC (MPC) algorithm [16], an improved version of PC for finite samples, runs a sequence of CI tests to build a skeleton and orient edges, which is efficient and sound with a CI oracle. With noisy data, however, the outcomes can be inconsistent.

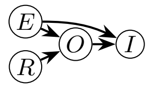

Figure 1: Example DAG.

Example 1 (Causal discovery)**.** *Consider a simple socioeconomic setting with Education (E), Race (R), Occupation (O) and Income (I) with ground-truth DAG in Fig. 1. When fed data generated from this DAG, MPC runs* 14 *CI tests (Ex. 3) and uses them sequentially to remove and orient edges. With finite-sample noise, it may report* mutually incompatible outcomes*. For instance, it includes the high-strength independences $E \perp\!\!\!\perp O \mid \{I\}$ and $O \perp\!\!\!\perp I$, even though it also reports dependencies such as $E \not\!\perp\!\!\!\perp O$ and $E \not\!\perp\!\!\!\perp I$. These four statements already cannot be simultaneously satisfied by any DAG: if E and O are dependent but become independent after conditioning on I, then I must block every active E–O path (so I must occur as a non-collider on such a path), which implies $O \not\!\perp\!\!\!\perp I$—contradicting $O \perp\!\!\!\perp I$. Nevertheless, MPC still returns a non-empty CPDAG (here, a single undirected edge $O - R$), without indicating which constraints were incompatible and effectively disregarded.*

2.1. Causal ABA

Example 1 highlights a core issue: enforcing *all* empirical CI facts may be impossible (no DAG satisfies them all), yet we still want a meaningful output and an explicit account of which facts were kept or released. Causal ABA [6] addresses this by encoding candidate edges and CI facts as an ABA framework [17] designed to enforce their correspondence through d-separation [18]. Under stable semantics [19], there is a one-to-one correspondence between stable models [20] of an Answer Set Program (ASP) and extensions of the ABA framework [21]. In Causal ABA, each stable extension/model corresponds to a DAG and its implied d-separations; *enforced* facts are guaranteed to hold in every returned graph, providing traceability from outputs back to the facts that support them [6, Prop. 3.18]. If the enforced facts are jointly inconsistent, the ASP program becomes incoherent (no stable models) and a repair step must decide which facts to release.

Example 2 (Causal ABA)**.** *Continuing Ex. 1, Causal ABA's grounded ASP encoding would include:*

```
collider(O,E,R)   :- arrow(E,O), arrow(R,O), E!=R.
nb(O,E,R,Z)       :- in(O,Z), collider(O,E,R).
ap(E,R,p_eor,Z)   :- arrow(E,O), arrow(R,O), nb(O,E,R,Z).
ap(E,R,p_eior,Z)  :- arrow(E,I), arrow(I,O), arrow(R,O), nb(O,E,R,Z).
dep(E,R,Z)        :- ap(E,R,_,Z).
indep(E,R,Z)      :- not ap(E,R,p_eor,Z), not ap(E,R,p_eior,Z).
```

Intuitively, `arrow(x,y)` *encodes a directed edge $X \rightarrow Y$;* `in(v,Z)` *encodes membership in the conditioning set;* `collider` *and* `nb` *characterise whether a node blocks a path given $\mathbf{Z}$; and* `ap(x,y,p,Z)` *witnesses an active path, where labels such as* `p_eor` *and* `p_eior` *are mnemonic identifiers for the corresponding path shapes. The atoms* `dep`/`indep` *are then derived from* `ap`*. Empirical CI tests are provided as input facts (e.g.,* `dep(O,R,{})` *for $O \not\!\perp\!\!\!\perp R$ and* `indep(R,I,{O})` *for $R \perp\!\!\!\perp I$ given $\{O\}$) and enforced by rejecting any stable model whose derived* `dep`/`indep` *disagree with the input. E.g., conditioning on the collider O makes* `ap(E,R,p_eior,{O})` *true, hence* `dep(E,R,{O})` *is derived.*

ABA-PC [6] is a concrete instantiation of Causal ABA in which the defeasible constraints are CI facts obtained from MPC. ABA-PC assigns each fact a *strength* derived from its test p-value [22] and then incrementally releases low-strength facts until coherence is restored: a simple baseline repair strategy.

Example 3 (ABA-PC)**.** *Continuing Ex. 1, consider an ABA-PC run on variables (E, R, O, I) whose CI tests (using MPC) yield the following* `dep`/`indep` *facts, sorted by descending strength S and labelled as true (T) or false (F) w.r.t. the ground-truth DAG in Fig. 1:*

```
indep(E,O,{I}) S=1.00 (F)     dep(E,I,{R})   S=0.56 (T)
indep(R,I,{O}) S=1.00 (F)     dep(R,I,{})    S=0.54 (T)
                              ------------------------
dep(O,R,{})    S=0.99 (T)     indep(E,I,{O}) S=0.53 (F)
dep(O,R,{E})   S=0.95 (T)     indep(R,I,{E}) S=0.53 (F)
dep(E,O,{})    S=0.92 (T)     dep(O,R,{I})   S=0.52 (T)
dep(E,O,{R})   S=0.92 (T)     indep(O,I,{})  S=0.51 (F)
dep(E,I,{})    S=0.84 (T)     indep(E,R,{})  S=0.50 (T)
```

MPC uses all 14 tests (including 5 wrong ones), producing a CPDAG with a single undirected edge ($O - R$). ABA-PC accepts the top 9 facts (7 true, 2 false, above the dashed line), yielding one compatible DAG with one correct edge ($O \to I$) and two wrong ones ($O \to R$, $I \to E$).

2.2. MUSes and MCSes

Explaining and repairing incoherence is a classic theme in diagnosis of ASP programs [7, 8], and it naturally applies here since CI tests play the role of defeasible constraints.

MUSes and MCSes are defined relative to a base ASP program P and a distinguished set O of *objective atoms*. For $\mathrm{S} \subseteq \mathrm{O}$, let $\mathsf{enforce}(\mathrm{P}, \mathrm{S}, \mathrm{O})$ denote the program obtained from P by (i) adding a choice rule over atoms in O and (ii) forcing every atom in S to be true. Concretely:

$$\mathsf{enforce}(\mathrm{P}, \mathrm{S}, \mathrm{O}) \;:=\; \mathrm{P} \,\cup\, \{\, \{o\}. \mid o \in \mathrm{O}. \,\} \cup \{\, \texttt{:- not } o. \mid o \in \mathrm{S} \,\}.$$

where $\{o\}.$ denotes choice over o. Then S is an *unsatisfiable set* if $\mathsf{enforce}(\mathrm{P}, \mathrm{S}, \mathrm{O})$ is incoherent [9].

A MUS is an unsatisfiable set $\mathrm{U} \subseteq \mathrm{O}$ such that no strict subset $\mathrm{U}' \subset \mathrm{U}$ is an unsatisfiable set. Dually, a *correction set* $\mathrm{R} \subseteq \mathrm{O}$ is a set whose *removal* restores coherence, i.e., $\mathsf{enforce}(\mathrm{P}, \mathrm{O} \setminus \mathrm{R}, \mathrm{O})$ is coherent; it is a *minimal correction set* (MCS) if no strict subset of R is a correction set. Intuitively, MUSes pinpoint small "conflict cores", while MCSes identify minimal "repairs"; in principle, both can be enumerated, but their number can grow quickly.

Duality can be stated as follows: every MCS must *hit* every MUS (otherwise an entire MUS would remain enforced, keeping the program incoherent). Moreover, letting $MUS(\mathrm{P}, \mathrm{O})$ and $MCS(\mathrm{P}, \mathrm{O})$ denote the sets of MUSes and MCSes for a program P and objective atoms O, $MCS(\mathrm{P}, \mathrm{O})$ coincides with the *minimal hitting sets* of $MUS(\mathrm{P}, \mathrm{O})$, and vice versa [9, Thm. 5.3]. This MUS/MCS-hitting-set connection and its use in optimisation are well established, including in core-guided and implicit hitting-set search for exact solutions [13].

Example 4 (MUS/MCS intuition)**.** *Consider a program* P *and objective* $\mathrm{O} = \{\texttt{o(1)}, \texttt{o(2)}, \texttt{o(3)}, \texttt{o(4)}\}$*:*

```
a :- not b, o(1).   b :- not c, o(2).   c :- not a, o(3).
:- not d, o(4).
```

Enforcing all of O *makes* $\mathsf{enforce}(\mathrm{P}, \mathrm{O}, \mathrm{O})$ *incoherent, but for* two independent reasons*: enforcing* `o(4)` *forces* `d` *even though it cannot be derived, and enforcing* `o(1)`, `o(2)`, `o(3)` *activates the negative loop over* `a`, `b`, `c`*. Hence the MUSes are* $\{\texttt{o(4)}\}$ *and* $\{\texttt{o(1)}, \texttt{o(2)}, \texttt{o(3)}\}$*, while the MCSes are the minimal hitting sets* $\{\texttt{o(1)}, \texttt{o(4)}\}$, $\{\texttt{o(2)}, \texttt{o(4)}\}$*, and* $\{\texttt{o(3)}, \texttt{o(4)}\}$.

In our causal discovery setting, the objective atoms correspond to candidate CI facts (Ex. 3), and an MCS corresponds exactly to the facts we *release* to recover at least one stable model. Enumeration can already explode: the 14 CI facts in Ex. 3 admit 103 MUSes and 25 MCSes. Two examples are:

```
{indep(R,I,{O}),dep(R,I,{}),indep(O,I,{})} %MUS        {indep(R,I,{E}),indep(O,I,{}),indep(E,O,{I})} %MCS
```

where only the `dep` is true in the MUS, while all facts in the MCS are false w.r.t. the DAG in Fig. 1.

3. Optimal Correction Sets

Even small incoherent programs can admit many alternative MCS repairs (in our small example, 25), so selecting a repair by full enumeration may be undesirable. We therefore refine the notion of MCS with a cost criterion: objective atoms carry positive weights and we seek repairs that minimise the total weight of the released atoms. To select a desirable repair *without* enumerating all repairs, we associate each objective atom $o \in \mathrm{O}$ with a positive weight w_o and minimise the total cost of the released facts. More formally, let P be a program, O a set of objective atoms, and $\mathrm{W} = \{(o, w_o) \mid o \in \mathrm{O},\ w_o > 0\}$ be weights over O. For any $\mathrm{S} \subseteq \mathrm{O}$, define its cost as $\mathcal{C}(\mathrm{S}, \mathrm{W}) = \sum_{(o,w_o)\in \mathrm{W},\ o\in \mathrm{S}} w_o$.

Definition 1. *Let* P *be a program,* O *a set of objective atoms, and* W *weights over* O*. An MCS* $\mathrm{S} \in MCS(\mathrm{P}, \mathrm{O})$ *is* optimal *(w.r.t.* W*) if for every* $\mathrm{S}' \in MCS(\mathrm{P}, \mathrm{O})$ *it holds that* $\mathcal{C}(\mathrm{S}, \mathrm{W}) \leq \mathcal{C}(\mathrm{S}', \mathrm{W})$*.*

Example 5 (Optimal MCS)**.** *Let* P *and* O *be, respectively, the program and the objective atoms from Ex. 4. Then* $MCS(\mathrm{P}, \mathrm{O}) = \{M_1, M_2, M_3\}$*, with* $M_1 = \{o(1), o(4)\}$*,* $M_2 = \{o(2), o(4)\}$*, and* $M_3 = \{o(3), o(4)\}$*. Let* $\mathrm{W} = \{(o(1), 5), (o(2), 10), (o(3), 15), (o(4), 20)\}$*. Then* $\mathcal{C}(M_1, \mathrm{W}) = 25$*,* $\mathcal{C}(M_2, \mathrm{W}) = 30$*, and* $\mathcal{C}(M_3, \mathrm{W}) = 35$*, hence* M_1 *is the unique optimal MCS by Def. 1.*

Starting from Def. 1, MCSes can be computed by iteratively searching for stable models in which at least k objective atoms are false, for increasing k, while blocking previously found models to avoid repeats and supersets [9, Alg. 5]. In principle, such enumeration can be adapted to compute an *optimal* MCS (Def. 1) by tracking the best cost encountered so far and using it as an improving upper bound to prune subsequent searches. However, such approach would explore the candidate MCSes by a fixed order (i.e. for increasing value of k) and so it may slow down the computation. To overcome this limitation, we can avoid the incremental approach and minimise the cost across all candidate MCSes. More precisely, let P be a program, O objective atoms, and W weights. We define the *opt* program $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ as P extended with (i) choice rules over O and (ii) a weak constraint per objective atom, penalising when they are *not* selected:

$$\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W}) = \mathrm{P} \cup \{ \{o\}. \mid o \in \mathrm{O}\} \cup \{ :\sim \ \mathrm{not}\ o.\ [w_o@1, o] \mid (o, w_o) \in \mathrm{W} \}.$$

where $:\sim$ represents a weak constraint, and $[w@p, t]$ its weighting function [23]: w is the penalty weight, p a priority level, and t a term used to identify the violated constraint. Then, if M is an optimal stable model of $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$, the released set $\mathrm{S} = \mathrm{O} \setminus M$ is an optimal MCS of P w.r.t. O and W.

Example 6 (MCSes and Weak Constraints)**.** *Let* P *and* O *be, respectively, the program and objective atoms from Ex. 4, and let* W *be as in Ex. 5. Then* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ *is:*

```
{o(1);o(2);o(3);o(4)}.
a :- not b, o(1).  b :- not c, o(2).  c :- not a, o(3).
:- not d, o(4).
:~ not o(1). [5@1,o(1)]   :~ not o(2). [10@1,o(2)]
:~ not o(3). [15@1,o(3)]  :~ not o(4). [20@1,o(4)]
```

An optimal stable model of $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ *is* $M = \{o(2), o(3), c\}$*, making* $\mathrm{O} \setminus M = \{o(1), o(4)\}$ *an optimal MCS for* P *w.r.t.* O, W.

Intuitively, let M be a stable model of $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ and let $\mathrm{S} = \mathrm{O} \setminus M$. Then all objective atoms in $\mathrm{O} \setminus \mathrm{S}$ are true in M, hence $\mathsf{enforce}(\mathrm{P}, \mathrm{O} \setminus \mathrm{S}, \mathrm{O})$ is coherent and S is a correction set for P. Furthermore, the cost associated with M corresponds to the sum of the weights of the objective atoms that are false w.r.t. M. Therefore, the cost of M coincides with the cost of S, that is, $\mathcal{C}(\mathrm{S}, \mathrm{W})$. Thus, minimising the cost of stable models of $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ amounts to minimising $\mathcal{C}(\mathrm{S}, \mathrm{W})$ over correction sets S, hence optimal stable models correspond to optimal MCSes for P w.r.t. O and W. This correspondence builds a clean parallel between optimal MCSes and solver-supported optimisation; our implementation and experimental material are available at https://github.com/briziorusso/ArgCausalDisco/tree/MCS.

OptABA-PC. Starting from the ABA-PC instantiation of Causal ABA [6] we can leverage optimal MCSes to derive the cost-aware minimal set of CI facts incompatible with any DAG. Specifically, take P being the ASP encoding Causal ABA [6], O to be the CI facts produced by MPC [16], and W their corresponding weights as calculated in [6]. Intuitively, these weights are release costs, so lower-confidence facts are cheaper to drop. Our proposed OptABA-PC computes *optimal* MCSes by solving $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ with a standard solver (*clingo* [23]).

Example 7 (OptABA-PC)**.** *Continuing Ex. 3, our proposed OptABA-PC computes an optimal MCS over the same CI facts. Of the* 14 *facts, OptABA-PC excludes only* 4 *of out of the* 14 *facts (all wrong w.r.t. the DAG in Fig. 1). The corresponding optimal stable model of* $\mathsf{opt}(\mathrm{P}, \mathrm{O}, \mathrm{W})$ *contains* 9 *correct facts and* 1 *wrong (*`indep(E,I,{O})`*). As a result, the number of compatible DAGs is reduced from* 4 *(ABA-PC) to just* 1*, with better orientation:* $E \rightarrow O$*,* $R \rightarrow O$*,* $O \rightarrow I$ *(3 correctly oriented edges vs. 1 for ABA-PC).*

Table 1
Results on bnlearn and synthetic benchmarks (10 repetitions). We report OptABA-PC − ABA-PC as Δ (relative change in parentheses). $\Delta n^T_{\mathrm{acc}}$ is the change in the number of accepted true constraints and $\Delta\mathrm{FactF1}$ is the change in the F1 of fact classification. All graphical metrics (#CPDAG, SHD, and worst-case SHD_w) are computed on the returned CPDAG. Δt is in seconds (OptABA-PC time divided by ABA-PC in parentheses).

Dataset (nodes)	$\Delta n^T_{\mathrm{acc}} \uparrow$	$\Delta\mathrm{FactF1} \uparrow$	$\Delta\#\mathrm{CPDAG} \downarrow$	$\Delta\mathrm{SHD} \downarrow$	$\Delta\mathrm{SHD}_w \downarrow$	$\Delta t \downarrow$
Cancer (5)	$+3.9\,(+10.7\%)$	$+0.048\,(+5.2\%)$	$-7.4\,(-88.1\%)$	$-1.707\,(-56.8\%)$	$-2.600\,(-66.7\%)$	$+0.010\,(\times 1.65)$
Survey (6)	$+7.1\,(+16.0\%)$	$-0.019\,(-2.3\%)$	$-168.7\,(-99.4\%)$	$-2.039\,(-32.2\%)$	$-4.600\,(-51.7\%)$	$+4.109\,(\times 88.69)$
Asia (8)	$+31.7\,(+29.2\%)$	$+0.124\,(+14.4\%)$	$-63.0\,(-97.7\%)$	$-5.403\,(-59.0\%)$	$-7.500\,(-65.2\%)$	$+15.072\,(\times 5.37)$
ER (5)	$+47.7\,(+486.7\%)$	$+0.693\,(+263.3\%)$	$-69.8\,(-98.6\%)$	$-0.266\,(-3.7\%)$	$-1.600\,(-18.8\%)$	$+0.105\,(\times 5.50)$
ER (8)	$+329.0\,(+1455.8\%)$	$+0.839\,(+655.7\%)$	$-14691.8\,(-99.5\%)$	$-3.304\,(-17.9\%)$	$-7.667\,(-31.7\%)$	$+127.933\,(\times 1.69)$

4. Experiments

We evaluate whether our optimal-MCS-guided optimisation improves the quality of ABA-PC outputs, while inherently facilitating increased transparency. We focus on the following questions: (Q1) Does selecting an optimal MCS improve constraint selection quality with respect to ground-truth constraints implied by the data-generating DAG via d-separation? (Q2) Does it reduce the number of compatible CPDAGs (MECs) in output? (Q3) Does it improve graph reconstruction quality? (Q4) What are the runtime implications compared to heuristic selection?

Setup. We evaluate on the four bnlearn benchmark networks from [6] (Cancer, Earthquake, Survey, Asia) and on synthetic Erdős–Rényi (ER) networks [24] with 5 and 8 nodes. Each benchmark is run for 10 seeds. We compare ABA-PC with OptABA-PC, using an improved encoding with up to $\approx 90\%$ average solve-time reduction. At $n = 8$, considering all CI tests already gives 1792 possible CI constraints, so naive repair enumeration spans 2^{1792} subsets before coherence checking. We cap solving and CPDAG evaluation at 300s and report solutions and metrics obtained within budget.

Metrics. We report: (i) number of accepted true constraints n^T_{acc} and fact-classification F1 score (FactF1); (ii) number of CPDAGs compatible with the accepted constraints; (iii) graph reconstruction quality computed against the ground-truth DAG (Structural Hamming Distance (SHD) [25], including worst-case SHD over the returned solution set); and (iv) runtime and timeouts. Reported runtimes reflect the capped budgets above.

Results. We answer (Q1–Q4) with Table 1. (Q1) OptABA-PC accepts more true constraints ($\Delta n^T_{\mathrm{acc}} > 0$) and typically improves FactF1 (with a small drop on Survey). (Q2) It consistently reduces the number of compatible CPDAGs, often by orders of magnitude. (Q3) This improves reconstruction quality (lower SHD and lower worst-case SHD over the returned CPDAG set) on all shown benchmarks; notably, the consistent improvement in worst-case SHD indicates increased robustness when multiple MEC-consistent graphs remain, which is important when causal discovery is used to support high-stakes decisions and we want conservative, trustworthy outputs. Results on Earthquake are omitted because both methods are identical. (Q4) These gains come with increased runtime. Timeouts occur only on ER(8): OptABA-PC hits the 300s solver cap in all runs (vs. 3/10 for ABA-PC) because it solves an exact repair problem over a large search space. ER(8) outputs are therefore best-effort under budget; scaling will likely require approximation or more targeted search, e.g. core-guided or hybrid strategies.

5. Conclusion

We cast causal discovery from noisy data using Causal ABA as a principled *repair* problem: identify which CI constraints are wrong, and exclude them transparently via an optimal correction set. By aligning optimal MCSes with weak-constraint optimisation, OptABA-PC provides a principled, solver-supported approach, yielding a trustworthy alternative to heuristic selection [6]. Empirically, this improves robustness by retaining more true constraints while excluding additional wrong ones, which in turn shrinks the set of compatible CPDAGs and increases their accuracy. Future work includes richer cost functions from data and expert priors, multiple distinct optimal repairs, and interactive causal analysis where repairs can serve as explanations. More broadly, the same repair-via-optimisation view applies to inconsistent defeasible knowledge bases beyond causal discovery.

Acknowledgments

Russo was funded by the ERC under the ERC-POC programme (grant number 101189053), Toni under the EU's Horizon 2020 research and innovation programme (grant number 101020934) while Mazzotta, Dodaro and Ricca were partially funded by the Italian Ministry of Industrial Development (MISE) under project EI-TWIN n. F/310168/05/X56 CUP B29J24000680005 and also partially by projects SIGENERA (CUP J29I24001770005) and MOZART (J49I24001740005) selected within the framework of the PR FESR – FSE Calabria 2021/2027 and implemented with the support of the Italian State and the Calabria Region.

References

[1] J. Pearl, Causality, 2 ed., Cambridge University Press, 2009. doi:10.1017/CBO9780511803161.
[2] C. Glymour, K. Zhang, P. Spirtes, Review of causal discovery methods based on graphical models, Frontiers in Genetics 10 (2019) 524. doi:10.3389/fgene.2019.00524.
[3] P. Spirtes, C. Glymour, R. Scheines, Causation, Prediction, and Search, Second Edition, Adaptive computation and machine learning, MIT Press, 2000. URL: https://www.cs.cmu.edu/afs/andrew/scs/cs/15-381/archive/OldFiles/lib/cvsub/.g/group/sdss/.g/group2/g/scottd/fullbook.pdf.
[4] T. S. Richardson, P. Spirtes, Ancestral graph markov models, The Annals of Statistics 30 (2002) 962–1030. doi:10.1214/aos/1031689015.
[5] J. Peters, D. Janzing, B. Schölkopf, Elements of Causal Inference: Foundations and Learning Algorithms, MIT Press, Cambridge, MA, USA, 2017. URL: https://mitpress.mit.edu/9780262037310/elements-of-causal-inference/.
[6] F. Russo, A. Rapberger, F. Toni, Argumentative causal discovery, in: Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, 2024, p. 938–949. doi:10.24963/KR.2024/88.
[7] R. Reiter, A theory of diagnosis from first principles, Artificial Intelligence 32 (1987) 57–95. doi:10.1016/0004-3702(87)90062-2.
[8] M. H. Liffiton, K. A. Sakallah, Algorithms for computing minimal unsatisfiable subsets of constraints, Journal of Automated Reasoning 40 (2008) 1–33. doi:10.1007/s10817-007-9084-z.
[9] M. Alviano, C. Dodaro, S. Fiorentino, A. Previti, F. Ricca, ASP and subset minimality: Enumeration, cautious reasoning and muses, Artificial Intelligence 320 (2023) 103931. doi:10.1016/j.artint.2023.103931.
[10] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, M. Maratea, F. Ricca, T. Schaub, Asp-core-2 input language format, Theory and Practice of Logic Programming 20 (2020) 294–309. URL: https://doi.org/10.1017/S1471068419000450. doi:10.1017/S1471068419000450.
[11] A. Hyttinen, F. Eberhardt, M. Järvisalo, Constraint-based Causal Discovery: Conflict Resolution with Answer Set Programming, in: Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, 2014, pp. 340–349. URL: https://www.cs.helsinki.fi/u/mjarvisa/papers/hyttinen-eberhardt-jarvisalo.uai14.pdf.
[12] Zhalama, J. Zhang, F. Eberhardt, W. Mayer, Sat-based causal discovery under weaker assumptions, Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017 (2017). URL: http://auai.org/uai2017/proceedings/papers/234.pdf.
[13] A. Hyttinen, P. Saikko, M. Järvisalo, A core-guided approach to learning optimal causal graphs, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, ijcai.org, 2017, pp. 645–651. doi:10.24963/ijcai.2017/90.
[14] K. Rantanen, A. Hyttinen, M. Järvisalo, Discovering causal graphs with cycles and latent confounders: An exact branch-and-bound approach, Int. J. Approx. Reason. 117 (2020) 29–49. doi:10.1016/j.ijar.2019.10.009.
[15] D. M. Chickering, Learning equivalence classes of bayesian-network structures, Journal of Machine Learning Research 2 (2002) 445–498. URL: https://jmlr.org/papers/v2/chickering02a.html.

[16] D. Colombo, M. H. Maathuis, Order-independent constraint-based causal structure learning, Journal of Machine Learning Research 15 (2014) 3741–3782. doi:10.5555/2627435.2750365.

[17] A. Bondarenko, P. M. Dung, R. A. Kowalski, F. Toni, An abstract, argumentation-theoretic approach to default reasoning, Artificial Intelligence 93 (1997) 63–101. doi:10.1016/S0004-3702(97)00015-5.

[18] J. Pearl, Probabilistic reasoning in intelligent systems - networks of plausible inference, Morgan Kaufmann, 1989. doi:10.1016/C2009-0-27609-4.

[19] K. Cyras, X. Fan, C. Schulz, F. Toni, Assumption-based argumentation: Disputes, explanations, preferences, IfCoLog Journal of Logics and their Applications 4 (2017). URL: http://www.collegepublications.co.uk/downloads/ifcolog00017.pdf.

[20] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: R. Kowalski, Bowen, Kenneth (Eds.), Proceedings of International Logic Programming Conference and Symposium, 1988, pp. 1070–1080. URL: http://www.cs.utexas.edu/users/ai-lab?gel88.

[21] A. Rapberger, M. Ulbricht, F. Toni, On the correspondence of non-flat assumption-based argumentation and logic programming with negation as failure in the head, in: Proceedings of the 22nd International Workshop on Nonmonotonic Reasoning (NMR 2024), volume 3835 of *CEUR Workshop Proceedings*, 2024, pp. 112–121. URL: https://ceur-ws.org/Vol-3835/paper12.pdf.

[22] G. Casella, R. Berger, Statistical Inference, Thomson Learning, 2002. doi:10.1201/9781003456285.

[23] M. Gebser, R. Kaminski, T. Schaub, Complex optimization in answer set programming, Theory and Practice of Logic Programming 11 (2011) 821–839. doi:10.1017/S1471068411000329.

[24] P. Erdős, A. Rényi, On random graphs I, Publicationes Mathematicae Debrecen 6 (1959) 18.

[25] I. Tsamardinos, L. E. Brown, C. F. Aliferis, The max-min hill-climbing bayesian network structure learning algorithm, Machine Learning 65 (2006) 31–78. doi:10.1007/s10994-006-6889-7.