

Causal Search Guided by Metamorphic Information

Kaveh Aryan¹[0009–0003–7245–4677], Mahsa Varshosaz²[0000–0002–4776–883X],
and Mohammad Reza Mousavi¹[0000–0002–4869–6794]

¹ King’s College London, London, UK

{kaveh.aryan,mohammad.mousavi}@kcl.ac.uk

² IT University of Copenhagen, Copenhagen, Denmark
mahv@itu.dk

Abstract. Determining actual causality in complex systems is computationally challenging, with the decision problem being Σ_2^P or DP^1 -complete. We present a novel approach that leverages metamorphic relations to guide the search for actual causes, significantly improving computational efficiency. Our method exploits directional knowledge about how input changes affect system robustness, encoded as metamorphic relations, to prune the search space and prioritise promising causal candidates. We formalise both qualitative and quantitative monotonic metamorphic relations and introduce bundled relations for handling variables’ collective behaviour. We study three metamorphic-guided best-first search variants: approximate best-first search, NSGA-II Pareto search, and best-first search followed by exhaustive subset shrinkage. Experimental evaluation on two system types, a lift system and an autonomous vehicle system, compares these variants with unguided breadth-first search using runtime measurements and shows significant improvements. The ablation study evaluates the effectiveness of bundling on the lift system across different bundle sizes.

Keywords: metamorphic testing · causality analysis · actual causality

1 Introduction

Actual causality provides explanations of why a particular outcome, such as a failure, occurred in a specific system execution. Such causal explanations are valuable in software reliability, where identifying the root cause of a failure enables targeted debugging rather than exhaustive manual inspection. The Halpern–Pearl (HP) definition [14] offers a rigorous basis for such explanations through counterfactual reasoning, i.e. evaluating whether the outcome would have changed under a different input, within structural causal models (SCMs) that encode causal dependencies between system variables. However, despite its clarity, computing causes remains difficult: determining whether an event is a cause is DP^1 -complete [12], making exhaustive causal search infeasible in practice.

In many real systems, we have directional knowledge about how changing inputs affects system behaviour [1, 23]. Such monotonic tendencies are routinely encoded as metamorphic relations in software testing [5], capturing domain knowledge as input-output constraints that serve as test oracles. However, to the best of our knowledge, they have not been exploited in causal search.

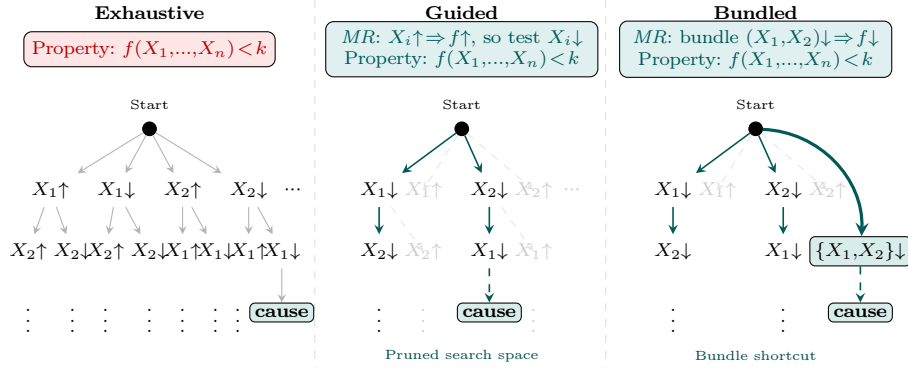


Fig. 1: Causal search starting from an initial state: exhaustive search (left) explores all intervention directions for every variable $X_1, \dots, X_n$, expanding exponentially many nodes before finding a cause. Considering the property $f(X_1, \dots, X_n) < k$, a metamorphic relation (MR) guides the search in changing variable values, pruning inconsistent directions at every level and reducing the search space (middle). Bundled relations additionally permit shortcut expansions that jump directly to coordinated interventions over variable groups, further reducing the search depth (right).

This creates an opportunity: if we translate the directional knowledge into formal constraints, then this information can guide the causal search. Instead of exhaustively exploring all variables, we can prioritise interventions, input changes, whose direction is consistent with changing the observed outcome. In effect, metamorphic relations give rise to an informed search heuristic over the search space. The following running example illustrates this.

Example 1 (Lift System). We use a lift (elevator) system as a running example throughout the paper. The system consists of lifts serving floors, with the goal of minimising the average waiting time (AWT) for passengers. The causal structure of the system is modelled using binary variables representing the working status of each lift. Metamorphic relations capture the inverse relationship between the number of working lifts and AWT. This monotonic relationship is the type of metamorphic information our approach exploits to guide causal search.

The contributions of this paper are as follows:

- A formal framework for expressing metamorphic relations in structural causal models,
- The concept of bundled metamorphic relations to efficiently handle the collective behaviour of variables,
- Metamorphic-guided best-first search variants that use approximate best-first search, NSGA-II Pareto search, and exhaustive subset shrinkage, and
- Runtime evaluation on lift and autonomous-vehicle subject systems.

The rest of the paper is organised as follows: Section 2 reviews the related work. Section 3 provides the necessary background on actual causality, robustness semantics,

and metamorphic relations. Section 4 details the metamorphic-relation syntax, its semantics, and their incorporation into the search algorithm. Section 5 evaluates our approach through experimental validation. Finally, Section 6 concludes the paper and discusses future work.

2 Related Work

Actual causality Actual causality, or token-level causation, concerns whether a particular event caused another particular event in a specific situation. Most accounts of actual causation appeal to counterfactual dependence in one form or another, but simple but-for dependence breaks down in cases of redundant causation (where multiple causes lead to an effect independently). The Halpern–Pearl (HP) framework addresses this by defining actual causality over structural causal models (SCMs) [20] and allowing counterfactual dependence to be evaluated under suitable contingencies [12, 14]. It is widely used in computer science, with applications ranging from databases to formal verification, explainable AI, and responsibility attribution [4, 6, 7, 18]. This makes HP one of the main formal accounts of actual causality in computer science, and the modified 2015 definition [12] provides the basis for our work. From an algorithmic perspective, however, even deciding whether a candidate event is an HP cause is computationally difficult, which motivates the search for more effective reasoning procedures.

Actual causal search Prior work on actual causal search spans both exact and approximate methods. Exact approaches include answer set programming (ASP) encodings [19] and ILP/MaxSAT formulations [15], but these target binary SCMs. Approximate approaches include beam search with a tunable precision–exhaustiveness trade-off [22] and SMT-based abstraction refinement for continuous systems [21]; however, the former does not provide a concrete heuristic construction, while the latter omits the AC3 minimality condition. Our method is positioned between these lines of work: like the continuous-system approach, it applies beyond binary models, but unlike the approximate methods it preserves completeness for the modified HP setting (under the shrinkage variant). Its main contribution is to provide a concrete source of heuristic information, via metamorphic relations, for guiding best-first causal search.

Metamorphic relations Metamorphic relations (MRs) relate changes in the inputs to the corresponding changes in the outputs. They have been used in software testing to generate test cases [5]. In the context of cyberphysical systems, MRs have been employed to verify system properties and detect faults [2, 23]. Clark et al. [8] use causal graphs to specify causal properties and generate metamorphic relations and tests. We proceed in the opposite direction, taking suitable relations as given and using them to prioritise the search for actual causes. Mazouni et al. [17] also use metamorphic relations together with mutation-guided state selection to detect non-optimal behaviour in AI planners. Their use of MRs remains within the testing setting, where follow-up test cases are generated and checked for optimality violations. Our work differs in its goal and mechanism: we use metamorphic relations not as test oracles, but as heuristic guidance for actual causal search in SCMs.

3 Background

3.1 Structural Causal Models

A *structural causal model* (SCM) [20] is a mathematical framework that formalises a system of interconnected variables and their causal relationships. The notion of SCMs is standard in formalisations of actual causality [3, 13]. Intuitively, an SCM captures how a system's variables depend on each other and on external inputs.

Definition 1. A structural causal model is a tuple $M = (U, V, P, R, E)$ in which U and V are two disjoint sets of variables called *exogenous* and *endogenous variables*, respectively, P is a set of types, $R : U \cup V \rightarrow P$ is a function that determines the ranges of variables in U and V , and E is the set of structural equations $\{Y = f_Y(X_1, \dots, X_n)\}_{Y \in V}$ where $X_i \in (U \cup V) - \{Y\}$ for $i = 1, \dots, n$. A particular setting for the exogenous variables is called a *context*, and a particular setting for the endogenous variables is called a *state*. When there is no cyclic dependence between the variables, the context $\mathbf{u}$ uniquely determines the state $\mathbf{v} = M[\mathbf{u}]$.

The reason to distinguish between exogenous and endogenous variables is to distinguish between system inputs and those influenced by the inputs, respectively. Structural *equations* specify how each endogenous variable depends on exogenous and other endogenous variables.

Example 2 (Lift System SCM). Consider the lift system in Example 1. We model this system using an SCM $M = (U, V, P, R, E)$ where:

- $U = \{U_1, U_2, \dots, U_n\}$ are exogenous variables representing external factors affecting each lift's operational status (e.g., power supply status and mechanical wear);
- $V = \{X_1, X_2, \dots, X_n, AWT\}$ are endogenous variables where $X_i \in \{0, 1\}$ indicates whether lift i is working (1) or not working (0), and $AWT \in \mathbb{R}^+$ represents the average waiting time for passengers;
- $P = \{\mathbb{R}^+, \{0, 1\}\}$ is the SCM's set of types, and R is the range function assigning a type to each variable in U and V : $R(U_i) = R(X_i) = \{0, 1\}$ for all $i \in \{1, \dots, n\}$, and $R(AWT) = \mathbb{R}^+$; and
- the structural equations, E , are:

$$X_i = f_{X_i}(U_i) \quad \text{for } i \in \{1, \dots, n\} \quad (1)$$

$$AWT = f_{AWT}(X_1, X_2, \dots, X_n) \quad (2)$$

The function f_{AWT} models the relationship between lift availability and passenger waiting time. The exogenous variables U_i capture external factors determining whether each lift is operational (e.g., power supply), with f_{X_i} mapping these factors to the operational status of each lift X_i , which in turn affects the overall lift system performance metric AWT .

An important notion for defining and quantifying causal effects is that of an *intervention*, which refers to deliberately changing the value of a variable to observe its subsequent effects on other variables, such as execution outcomes.

We use $\mathbf{X}$ to denote a set of variables and $\mathbf{a}$ the corresponding set of values, following the HP causality convention. Single variables are denoted by uppercase letters (e.g., X, Y) and their values by lowercase letters (e.g., a, b).

Definition 2. Let $M = (U, V, P, R, E)$ be an SCM, $X \in V$ and $a \in R(X)$. The intervention model $M[X = a]$ is the SCM (U, V, P, R, E') in which E' is identical to E except that the equation $X = f_X(Z_1, \dots, Z_n), Z_i \in (U \cup V) - \{X\}$, is replaced with $X = f'_X(Z_1, \dots, Z_n) = a$. This definition extends naturally to simultaneous interventions on multiple variables, written as $M[\mathbf{X} = \mathbf{a}]$.

Intuitively, intervening on X forces it to take value a regardless of its normal dependencies, i.e., fixing a system variable to a specific value to observe its effect on the rest of the system.

3.2 Actual Causality

The Halpern–Pearl (HP) definition of actual causality [12] determines whether an event $\mathbf{X} = \mathbf{a}$ caused an event φ by checking if: (1) both events actually occurred, (2) while holding certain variables fixed at their actual values, changing $\mathbf{X}$ prevents φ (counterfactual dependence), and (3) no smaller subset of $\mathbf{X}$ does so (minimality). We use the latest (*modified*) definition [12].

Definition 3. Let $M = (U, V, P, R, E)$ be an SCM and $\varphi(V)$ a quantifier-free first-order (QFFO) formula over V (i.e., a condition over system variables such as $AWT < 10$). We omit V and write φ when it is clear from context and use $\varphi(M[\cdot])$ as shorthand for evaluating φ in the state determined by $M[\cdot]$. The event $\mathbf{X} = \mathbf{a}$ is called an HP cause of φ , iff:

- HP1. $M[\mathbf{u}][\mathbf{X}] = \mathbf{a}$ and $\varphi(M[\mathbf{u}])$, where $M[\mathbf{u}][\mathbf{X}] = \mathbf{a}$ denotes the value of $\mathbf{X}$ when the context is $\mathbf{u}$. This means both cause $\mathbf{X} = \mathbf{a}$ and outcome φ actually occurred.
- HP2. $\exists \mathbf{W} \subseteq V, \exists \mathbf{a}'$ such that $\neg \varphi(M[\mathbf{u}, \mathbf{X} = \mathbf{a}', \mathbf{W} = \mathbf{w}^*])$, where $\mathbf{w}^* = M[\mathbf{u}][\mathbf{W}]$ are the actual values of $\mathbf{W}$ in context $\mathbf{u}$, i.e., fixing $\mathbf{W}$ to $\mathbf{w}^*$ and changing $\mathbf{X}$ to $\mathbf{a}'$ negates φ .
- HP3. $\mathbf{X}$ is minimal, in the sense that no proper subset of $\mathbf{X}$ satisfies HP1 and HP2.

The tuple $(\mathbf{X}, \mathbf{a}', \mathbf{W})$ is called a witness to the fact that $\mathbf{X} = \mathbf{a}$ is a cause of φ .

Determining whether $\mathbf{X} = \mathbf{a}$ is a cause of φ is DP¹-complete for the modified HP definition [12]. This complexity arises from (i) searching the powerset of endogenous variables for a contingency set $\mathbf{W}$ (yielding 2^n possible contingencies for n variables) and (ii) verifying minimality (no proper subset of $\mathbf{X}$ satisfies HP1 and HP2). Evaluating counterfactual queries for each candidate $(\mathbf{X}, \mathbf{W})$ pair could be expensive. The *actual causal search problem*, finding one or all causes of φ , is at least as hard, requiring exponential search over all possible candidate sets $\mathbf{X} \subseteq V$. This exponential search motivates using metamorphic relations to guide and prune search.

3.3 Robustness Semantics of Logical Specifications

To use metamorphic relations to guide causal search, we use robustness semantics [11] to quantify how *close* a system state is to (un)satisfying a logical specification. This continuous measure captures how variable changes affect formula satisfaction, forming the basis of our metamorphic-guided search.

Definition 4. Let φ be a QFFO formula over a set of variables V . The extension of φ , denoted by $\text{ext}(\varphi)$, is the set of variable valuations that satisfy φ .

Given a specification formula φ , we quantify the *robustness* as the distance from the system state to the boundary of the extension set, positive when the state satisfies φ and negative when it does not, which is formally defined as follows.

Definition 5. Consider a metric space [24], (G, d) , where G is a vector space and d is a distance function over G (e.g., the Euclidean distance). Given some $\mathbf{x} \in G$ and $H \subseteq G$, we define:

- The distance from $\mathbf{x}$ to H is defined as $\text{dist}_d(\mathbf{x}, H) = \inf\{d(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in \bar{H}\}$, where $\bar{H}$ is the closure of H , that is, the intersection of all closed sets containing H ; and
- The depth of $\mathbf{x}$ in H is defined as $\text{depth}_d(\mathbf{x}, H) = \text{dist}_d(\mathbf{x}, G - H)$.
- The signed distance from $\mathbf{x}$ to S is defined as

$$\text{Dist}_d(\mathbf{x}, S) = \begin{cases} \text{depth}_d(\mathbf{x}, S) & \text{if } \mathbf{x} \in S \\ -\text{dist}_d(\mathbf{x}, S) & \text{if } \mathbf{x} \notin S \end{cases}$$

So the signed distance Dist_d measures how far a system state $\mathbf{x}$ is from satisfying or violating φ , positive when φ holds and negative when it does not.

We can now formally define the robustness of a point with respect to a logical specification:

Definition 6. Let φ be a QFFO formula over a set V and $\mathbf{x} \in V$ be a system state that is a valuation. The robustness of $\mathbf{x}$ with respect to φ is defined as:

$$\xi(\mathbf{x}, \varphi) = \text{Dist}_d(\mathbf{x}, \text{ext}(\varphi))$$

A higher robustness value indicates the system state is further from violating φ , while a lower value indicates it is closer to or already violating φ .

3.4 Search Algorithms

Search algorithms explore the solution space for instances with desired properties. Any search problem has several fundamental components [9]:

- **Search space:** the set of all possible configurations that the algorithm explores, in our case, all combinations of variables and their values.
- **Goal criteria:** conditions that define a successful solution. For actual causality, this means finding assignments that satisfy Definition 3.
- **Search strategy:** the systematic approach to traverse the search space, determining the order in which potential solutions are examined.

In the context of actual causality, we use breadth-first search as an uninformed baseline and best-first search variants as guided search procedures.

Breadth-First Search (BFS) explores nodes level by level, guaranteeing the shortest path to any goal is found first. Alternative baselines, such as SAT-augmented

strategies [16], do not apply to non-binary models such as our lift and vehicle systems. BFS provides a systematic but uninformed baseline against which the guided variants are compared.

Best-First Search is an informed search strategy that orders candidate nodes by a priority score derived from heuristic information. In our setting, metamorphic relations define this score so that candidates with greater expected potential to change the outcome are explored earlier.

NSGA-II is a genetic multi-objective optimisation algorithm based on non-dominated sorting and elitist selection [10]. We use it because causal-set search has two competing objectives: finding a smaller intervention set and achieving positive robustness. Instead of fully enumerating all subsets, NSGA-II samples candidate sets using the same metamorphic heuristic signal as best-first search: cardinality captures intervention cost, while robustness measures how far the intervention moves the system towards changing the specification outcome. The algorithm therefore approximates the trade-off between smaller causal sets and stronger robustness, and lets us select a positive-robustness candidate with minimal cardinality among the discovered non-dominated solutions.

3.5 Metamorphic Relations

Metamorphic relations capture properties that must hold between related inputs and outputs of a system under test. Let $f: D_X \rightarrow D_Y$ be a function for the system under test, with input domain D_X and output domain D_Y . A metamorphic relation is a tuple (R, S) where $R \subseteq D_X \times D_X$ is an input relation and $S \subseteq D_Y \times D_Y$ is an output relation, such that: $(x_1, x_2) \in R \implies (f(x_1), f(x_2)) \in S$. Here x_1 and x_2 , related by R , are called the *source* and *follow-up* inputs, and $f(x_1)$, $f(x_2)$, related by S , the corresponding *source* and *follow-up* outputs.

For example, let N be a sequence of numbers and $\pi(N)$ a permutation of N . An MR (R, S) for a sorting function f captures that sorting any permutation of N yields the same result, i.e., $f(N) = f(\pi(N))$:

$$R = \{(N, \pi(N)) : \pi \text{ is a permutation}\}$$

$$S = \{(y, y) : y \in D_Y\}$$

In our approach, metamorphic relations capture the directional knowledge used to guide and prune the causal search.

4 Guided Actual Causal Search

4.1 Proposed Metamorphic Relations

We propose metamorphic relations capturing how variable assignment changes affect the robustness of the effect formulae. These relations guide causal search by encoding monotonic relationships between inputs and outputs. The relations use the following syntax and semantics.

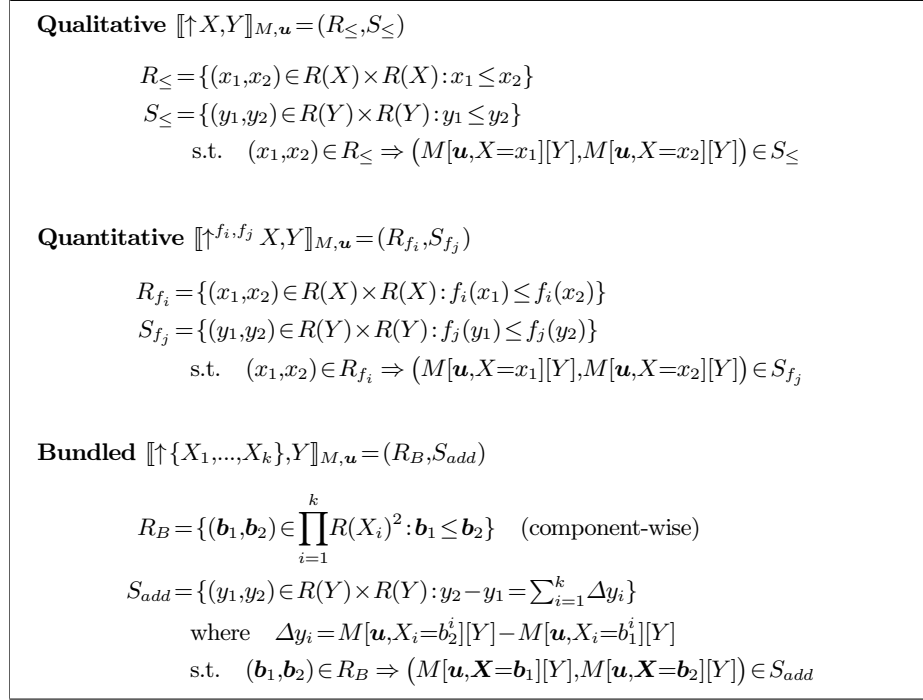


Fig. 2: Formal semantics of the three types of metamorphic relations for the increasing ($\uparrow$) case. Decreasing ($\downarrow$) relations are defined symmetrically by reversing the inequalities. The qualitative case is a special case of the quantitative case with identity transformation functions $f_i = f_j = \text{id}$.

Definition 7. Let $\mathcal{M} = (U, V, P, R, E)$ be an SCM and let $\{f_1, \dots, f_m\}$ be a set of functions over $V \cup U$. A metamorphic relation m is defined by the following grammar:

$$\begin{aligned}
m &\rightarrow op' \text{ var}, \text{ var} \mid op \text{ } \mathbf{var}, \text{ var} \\
op &\rightarrow \uparrow \mid \downarrow \\
op' &\rightarrow op \mid \uparrow^{f, f} \mid \downarrow^{f, f} \\
\text{var} &\rightarrow X_1 \mid \dots \mid X_n \\
\mathbf{var} &\rightarrow (\text{var}, \mathbf{var}) \mid (\text{var}, \text{var}) \\
f &\rightarrow f_1 \mid \dots \mid f_m
\end{aligned}$$

Given this syntax, a relation m takes one of three forms:

- **qualitative** ($op' \text{ var}, \text{var}$): a plain $\uparrow$ or $\downarrow$, encoding that two variables change in the same or opposite direction, respectively.

- **quantitative** ($op^{f,f} \text{ var}, \text{var}$): extends the qualitative case with two transformation functions capturing the magnitude of the relationship for input and output, respectively.
- **bundle** ($op \text{ var}, \text{var}$): a qualitative relation between a group of variables and a single output variable, capturing their collective additive effect on the output. In all cases, $\uparrow$ encodes same-direction monotonicity and $\downarrow$ encodes opposite-direction monotonicity; $X_i \in V$ are endogenous variables; and each f_i is a transformation function on variable values.

Next, we define the semantics of the operations in the syntax above.

Definition 8 (Semantics of a Metamorphic Relation). Let $\mathcal{M} = (U, V, P, R, E)$ be an SCM, $\mathbf{u}$ a context, and m a metamorphic relation. The semantics of m , denoted $\llbracket m \rrbracket_{\mathcal{M}, \mathbf{u}}$, is a relation:

$$\llbracket m \rrbracket_{\mathcal{M}, \mathbf{u}} \subseteq \bigcup_{X, Y \in V} (R(X) \times R(X)) \times (R(Y) \times R(Y))$$

Three cases are defined in Figure 2: qualitative, quantitative, and bundled. Qualitative relations capture same or opposite direction monotonicity between two variables. Quantitative relations additionally specify transformation functions capturing the magnitude of the relationship. Bundled relations capture the additive combined effect of a group of variables on a single output variable. Decreasing relations are defined symmetrically.

The following examples illustrate the use of these relations in the lift system.

Example 3 (Lift Quantitative Relations). Ayerdi et al. [1] introduced a metamorphic relation for the relationship between the number of active lifts and the Average Waiting Time (AWT) in a lift system.

$$\frac{AWT(\text{serve}(E_f, C))}{AWT(\text{serve}(E_s, C))} \gtrsim \frac{|E_s|}{|E_f|} \quad (3)$$

where E_f and E_s are sets of active lifts in follow-up and source executions; $|\cdot|$ denotes the size of a set, and C is a fixed set of passenger requests. This relation shows that increasing the number of active lifts decreases AWT proportionally. In our notation, this is a quantitative decreasing relation. If N is the number of active lifts:

$$\forall k \in \mathbb{R}^+ \downarrow^{\times k, \times \frac{1}{k}} N, AWT.$$

This metamorphic relation means multiplying N by k divides AWT by approximately k .

The bundled relations allow the algorithm to handle groups of variables simultaneously.

Example 4 (Lift Bundling). In the lift system, multiple lift indicators can be bundled by their joint effect on AWT. For any subset of m lifts, the combined effect on AWT is approximately the sum of individual effects³. Therefore, for any subset of m indicator variables $\{X_1, \dots, X_m\}$, we can define a bundled relation $\downarrow\{X_1, \dots, X_m\}, AWT$.

4.2 Heuristic Function

The mapping below defines the heuristic for each syntactic category of metamorphic relations.

Definition 9 (Metamorphic Heuristic Functions). Let M be an SCM, $\mathbf{u}$ a context, and $\mathbf{v} = M[\mathbf{u}]$ the actual state. Let $\mu_l = Q_f^l \cup Q_r^l$, with $Q_f^l = \{\uparrow X, \xi(\cdot, \varphi)\}$ and $Q_r^l = \{\downarrow X, \xi(\cdot, \varphi)\}$, be a set of qualitative increasing and decreasing metamorphic relations, where $\xi(\cdot, \varphi)$ denotes the robustness of the system state with respect to φ , used as the output variable of the heuristic relations. Let $\mu_t = Q_f^{t,+} \cup Q_f^{t,\times} \cup Q_r^{t,+} \cup Q_r^{t,\times}$, with $Q_f^{t,+} = \{\uparrow^{+k,+l} X, \xi(\cdot, \varphi)\}$, $Q_f^{t,\times} = \{\uparrow^{\times k, \times l} X, \xi(\cdot, \varphi)\}$, $Q_r^{t,+} = \{\downarrow^{+k,+l} X, \xi(\cdot, \varphi)\}$, and $Q_r^{t,\times} = \{\downarrow^{\times k, \times l} X, \xi(\cdot, \varphi)\}$, be a set of quantitative increasing and decreasing metamorphic relations for additive and multiplicative cases. Let $\mu_b = Q_f^b \cup Q_r^b$, with $Q_f^b = \{\uparrow B, \xi(\cdot, \varphi)\}$ and $Q_r^b = \{\downarrow B, \xi(\cdot, \varphi)\}$, be a set of bundled metamorphic relations. For a candidate set $\mathbf{X} \subseteq V$, the metamorphic heuristic function $h_\mu(\mathbf{X})$ is defined by the case analysis in Figure 3.

Rationale The heuristic prioritises candidate sets with the greatest potential to create counterfactuals where φ does not hold: (i) Each term captures normalised potential for changing variables in directions that affect robustness (“room for improvement”) (ii) Signs weight variables by potential to create counterfactual dependence in the correct direction (iii) The quantitative version uses transformation magnitudes through ratios such as $\frac{l}{k}$, for additive effects, and factors such as $(k-1)$, for multiplicative effects

³ From Example 3, AWT is approximately inversely proportional to the number k of active lifts, so let $AWT(k) \approx c/k$. If m lifts are activated from an initial state with k_0 active lifts, then

$$AWT(k_0) - AWT(k_0 + m) = c \left(\frac{1}{k_0} - \frac{1}{k_0 + m} \right).$$

Activating the lifts individually gives

$$\sum_{i=1}^m (AWT(k_0 + i - 1) - AWT(k_0 + i)) = c \sum_{i=1}^m \left(\frac{1}{k_0 + i - 1} - \frac{1}{k_0 + i} \right),$$

which telescopes to

$$c \left(\frac{1}{k_0} - \frac{1}{k_0 + m} \right).$$

Hence the bundled change equals the sum of the individual changes, satisfying the additive bundled relation $\downarrow_{X_1, \dots, X_m}, AWT$.

$$h_\mu(\mathbf{X}) = \begin{cases} \sum_{Y \in Q_f^l \cap \mathbf{X}} \frac{\max(R(Y)) - v[Y]}{\max(R(Y)) - \min(R(Y))} + \sum_{Y \in Q_r^l \cap \mathbf{X}} \frac{v[Y] - \min(R(Y))}{\max(R(Y)) - \min(R(Y))} & \text{(Qualitative)} \\ \sum_{Y \in Q_f^{t,+} \cap \mathbf{X}} \frac{l}{k} (\max(R(Y)) - v[Y]) + \sum_{Y \in Q_f^{t,\times} \cap \mathbf{X}} \left(\frac{\max(R(Y))}{v[Y]} \right)^{\frac{\log l}{\log k}} \\ + \sum_{Y \in Q_r^{t,+} \cap \mathbf{X}} \frac{l}{k} (v[Y] - \min(R(Y))) + \sum_{Y \in Q_r^{t,\times} \cap \mathbf{X}} \left(\frac{v[Y]}{\min(R(Y))} \right)^{\frac{\log l}{\log k}} & \text{(Quantitative)} \\ \sum_{B \in \mu_b} \left[\mathbf{1}_{B \cap \mathbf{X} \neq \emptyset} \sum_{Y \in B} \delta(Y) \right] & \text{(Bundled)} \end{cases}$$

Fig. 3: Metamorphic heuristic function $h_\mu(\mathbf{X})$ for qualitative, quantitative, and bundled metamorphic relations, where $\delta(Y)$ is the contribution of Y as defined for the quantitative case above. Each term captures the normalised potential for changing variables in directions that affect robustness. The qualitative case measures room for improvement; the quantitative case uses transformation magnitudes; and the bundled case exploits variables' collective behaviour.

(iv) The bundled version exploits variables' collective behaviour to explore groups efficiently; $\delta(Y)$ gives individual contributions. This design ensures that best-first search explores candidates with the highest potential for satisfying the HP causality conditions first, leading to more efficient discovery of actual causes.

4.3 Metamorphic-Guided Search Algorithm

Approximate best-first search. This variant treats each candidate pair $(\mathbf{X}, \mathbf{W})$, where $\mathbf{X} \subseteq V$ is a candidate intervention set and $\mathbf{W} \subseteq V \setminus \mathbf{X}$ is a contingency set, as a search node. For each candidate, the variables in $\mathbf{X}$ are assigned alternative values $\mathbf{a}'$, while the variables in $\mathbf{W}$ are fixed to their actual values $\mathbf{w}^* = M[\mathbf{u}][\mathbf{W}]$, as required by HP2. A node therefore satisfies the counterfactual criterion when

$$\neg \varphi(M[\mathbf{u}, \mathbf{X} = \mathbf{a}', \mathbf{W} = \mathbf{w}^*]).$$

The frontier is ordered primarily by the metamorphic heuristic $h_\mu(\mathbf{X})$ from Definition 9, since the heuristic estimates the potential of the intervention variables to change the outcome. With qualitative or quantitative relations, intervention-set expansion adds one variable at a time; with bundled relations, it may add a whole bundle B whose variables are expected to act collectively according to the bundled heuristic term. The algorithm expands the most promising candidate first and returns as soon as it finds a pair satisfying the causal-search criterion. This makes the method cheap, but approximate, because it is not guaranteed to find a minimal causal set.

NSGA-II Pareto search. This variant represents each candidate set as a bit-vector and uses NSGA-II to evolve a population of candidates. In the bundled setting, the bit-vector selects bundles rather than individual variables, and the selected bundles induce the candidate intervention set. Rather than exhaustively traversing the subset lattice, NSGA-II uses non-dominated sorting to balance small cardinality against the metamorphic heuristic score h_μ . After evolution, we choose a discovered candidate with minimum cardinality among those retained by the heuristic-guided Pareto search. This approach is more expensive than approximate best-first search, but it is more likely to find a minimal causal set.

Best-first search with shrinkage. This variant first uses the same best-first ordering to locate a causal candidate quickly, including bundle-level expansion when bundled metamorphic relations are available. It then performs an exhaustive subset-shrink step inside that candidate, enumerating subsets by increasing cardinality until it finds a smallest subset that still satisfies the causal-search criterion. Thus, bundling can speed up the discovery phase by moving through groups of related variables, while shrinkage checks whether the final explanation can be reduced below the initially discovered bundle-level candidate.

5 Experiments

We evaluate on 200 subject systems derived from two system types. We surveyed the literature for realistic cyberphysical systems with monotonic metamorphic relations. This led us to two system types: a lift system (our running example) and an autonomous vehicle system (AVS) [1,2,23]. We used these metamorphic relations without modification. From each type, we generated 100 subject systems of varying characteristics to answer the following research questions. The code is available at <https://github.com/kavaryan/actual-causality/>, under the `subprojects/metamorphic` directory.

5.1 Research Questions

RQ1: How does our metamorphic-guided approach scale compared to traditional exhaustive search? We evaluate scalability by comparing the execution time of metamorphic-guided and unguided BFS across problem sizes.

RQ2: What is the individual contribution of each component in our approach? (Ablation Study) We conduct an ablation study to assess contributions of the approach components, with two sub-questions:

RQ2a: What is the effect of removing bundling altogether? We compare unbundled best-first search (metamorphic heuristics only) to bundled best-first search to isolate bundling’s contribution.

RQ2b: How does bundle size affect performance? We experiment with different bundle sizes (e.g., $B=2,5$) to understand the trade-off between bundle size and search efficiency.

5.2 Baselines and Metrics

We compare our metamorphic-guided approach to unguided breadth-first search (BFS). While there is no obvious choice of baseline other than unguided search, BFS ensures minimality by exploring candidate causal sets in order of increasing size, satisfying minimality (HP3) in Definition 3. The primary metric for both RQs is actual wall-clock runtime in seconds, using a fixed cutoff $T=2s$. We do not use a Runtime Improvement Ratio (RIR) metric in these experiments because RIR requires the BFS runtime as the baseline numerator, but for larger numbers of variables, BFS often does not terminate by T ; its true runtime is therefore censored. Reporting actual runtime keeps completed and timed-out runs on the same measured scale.

5.3 Subject Systems

Subject systems were selected based on the availability of well-defined metamorphic relations and their relevance to real-world applications. We used metamorphic relations from original studies without modification.

Lift System [1] introduces metamorphic relations for how the number of active lifts influences the Average Waiting Time (AWT) in a lift system. The SCM has endogenous variables

$$V = \{X_1, \dots, X_n, N_{active}, AWT\}$$

where each X_i is a binary indicator for lift i functioning, $N_{active} = \sum_{i=1}^n X_i$, and AWT is computed by a lift simulator. The outcome formula is $\varphi \equiv AWT \geq \text{threshold}$, with the robustness defined as $\xi = AWT - \text{threshold}$. The metamorphic relations are:

$$\forall k \in \mathbb{R}^+ \quad \downarrow^{\times k, \times \frac{1}{k}} N_{active}, AWT \cdot \downarrow \{X^1, \dots, X^m\}, AWT.$$

(additive bundled relation for any subset $\{X^1, \dots, X^m\}$, with $1 < m \leq n$)

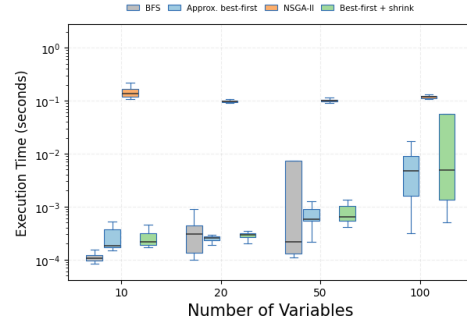
The lift simulator is parametrised by lift speed and call (congestion) density, generating diverse subject systems. We experimented on 100 subject systems from all combinations of 10 lift speeds and 10 call densities.

Autonomous Vehicle System (AVS) For the AVS, we model an autonomous vehicle moving between two positions with several obstacles in between [23]. The SCM has endogenous variables

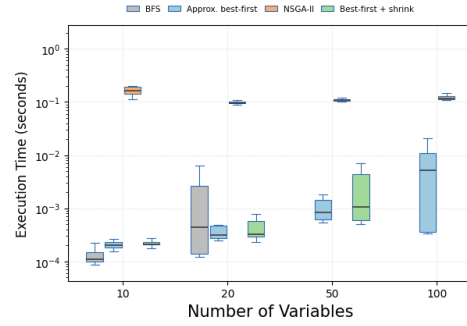
$$V = \{SP, P_a, P_b, OB_1, \dots, OB_n, TD\}$$

where SP is the vehicle nominal speed, P_a and P_b are starting and ending positions, respectively, each OB_i is a binary indicator variable for obstacle i 's presence, and TD is the time to destination, which we derive from a simulator. The metamorphic relations are $\uparrow OB_i, TD$ for each obstacle variable OB_i , $i = 1, \dots, n$. The outcome formula is $\varphi \equiv TD \geq \text{threshold}$ and robustness is $\xi = TD - \text{threshold}$.

The vehicle simulator is parametrised by vehicle nominal speed and travel distance, generating diverse subject systems. We experimented on 100 subject systems from all combinations of 10 vehicle speeds and 10 travel distances.



(a) Lift system



(b) Autonomous vehicle system

Fig. 4: RQ1 runtime results. At each value of N , boxes are ordered left-to-right as BFS, approximate best-first, NSGA-II, and best-first with shrinkage. A box is omitted when more than half of the runs for that method-size pair reached the $T=2s$ cutoff.

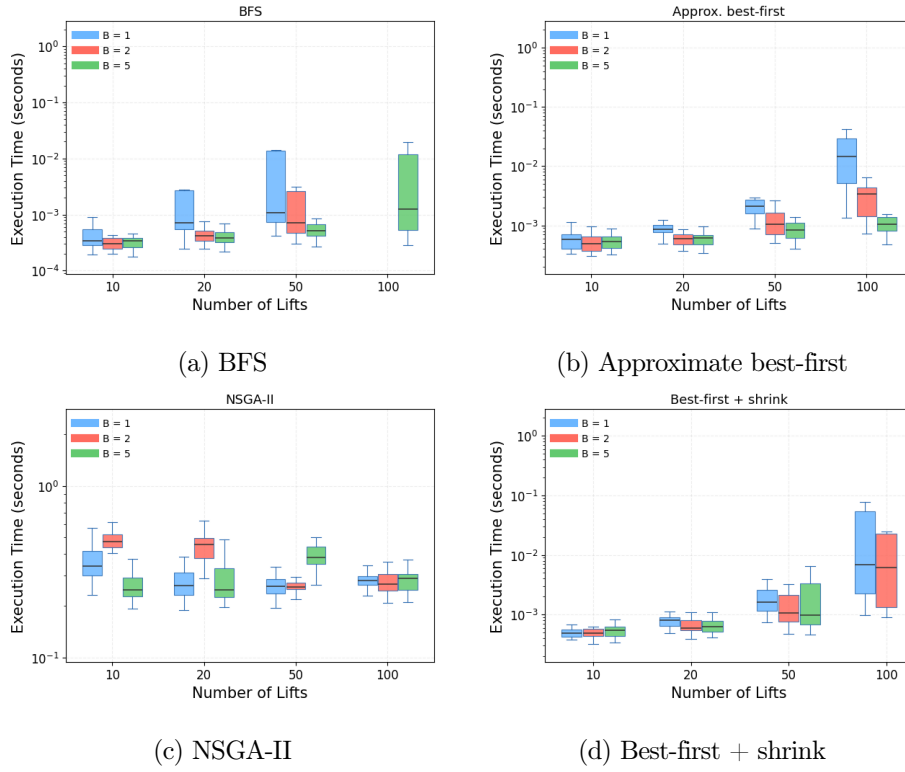


Fig. 5: RQ2 bundle-size results for the lift system. Each plot compares bundle sizes $B=1,2,5$ for one search method.

5.4 Implementation and experimental setup

The implementation is contained in the subprojects/metamorphic directory of the actual-causality repository at github.com/kavaryan/actual-causality. The metamorphic-guided and uninformed search algorithms were implemented directly in Python (3.12) because no existing causal-analysis tool supported the required combination of continuous SCMs, system simulators, and qualitative, quantitative, and bundled metamorphic relations. Supporting libraries were used for graph construction, numerical arrays, plotting, and data handling (NetworkX 3.5, NumPy 2.0, Matplotlib 3.10, and pandas 2.3).

The subject SCM, effect formula, simulator, and metamorphic relations are communicated to the algorithms as Python objects with standard interfaces, according to principles of modularity and separation of concerns.

The experiments were run on a 13th Gen Intel Core i7-1355U with 12 logical processors and 23 GiB RAM, on x86-64 Ubuntu 24.04 LTS.

5.5 Results

RQ1: Scalability To answer RQ1, we experimented on instances with varying numbers of variables ($N = 10, 20, 50, 100$) and different instance-specific parameters (lift speed and call density for the lift system; vehicle speed and travel distance for the AVS). In RQ1, no bundling was applied to either system. For both systems, a three-way ANOVA (instance parameter 1, instance parameter 2, number of variables) is applied to log wall-clock runtime. Therefore, the data were aggregated across instance parameters for the presented results.

The experimental results from both system types are shown in Figure 4.

The y-axis is logarithmic, so vertical separation between boxes corresponds to orders of magnitude in execution time. For small instances ($N = 10, 20$), BFS and the two best-first variants complete quickly on both systems, while NSGA-II appears as a nearly horizontal band around 10^{-1} seconds because population evolution has a relatively fixed overhead. As N increases, the unguided BFS boxes become unstable and then disappear according to the omission rule in Figure 4. The approximate best-first variant remains the cheapest guided method in most completed runs, whereas best-first with shrinkage has similar behaviour on smaller instances but becomes more variable at larger N , reflecting the additional subset enumeration performed after the initial guided search.

RQ2: Ablation Study For the ablation study, we experimented with no bundling (i.e., $B = 1$), and bundle sizes $B = 2, 5$. As shown in Figure 5, bundling has method-dependent effects. For BFS and approximate best-first search, larger bundles generally reduce runtime as N grows, because the search can move through groups of related lift variables rather than expanding one variable at a time; this is most visible at $N = 50$ and $N = 100$. For NSGA-II, the boxes remain comparatively high and do not improve monotonically with bundle size, indicating that the evolutionary population overhead dominates any reduction in the number of bundle selectors. For best-first with shrinkage, bundling helps at moderate sizes, but at $N = 100$ the shrinkage phase becomes the bottleneck: the $B = 5$ box is omitted because more than half of those

runs reach the cutoff. Thus, bundling is useful when it reduces frontier traversal, but its benefit is limited when the dominant cost comes from population evolution or exhaustive subset shrinking.

6 Conclusion

We presented a novel approach to actual causal search leveraging metamorphic relations to guide the search. Incorporating directional knowledge of variable effects on system robustness, we transformed the prohibitive exhaustive search into efficient guided search.

Our key contributions are: (1) a formal framework for metamorphic relations in structural causal models, including both qualitative and quantitative monotonic relations; (2) bundled metamorphic relations that capture variables' collective behaviour for efficient group-wise exploration; (3) metamorphic-guided best-first search variants using approximate best-first search, NSGA-II Pareto search, and exhaustive subset shrinkage; and (4) runtime evaluation on lift and autonomous-vehicle subject systems.

Experimental results on lift systems confirm scalability, with the bundled variant providing additional benefits when variables show non-interaction patterns.

Future work includes extending the approach to handle more complex metamorphic relations and developing automated techniques for discovering metamorphic relations from system observations.

References

1. Ayerdi, J., Segura, S., Arrieta, A., Sagardui, G., Arratibel, M.: QoS-aware Metamorphic Testing: An Elevation Case Study. In: 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE). pp. 104–114 (2020)
2. Ayerdi, J., Terragni, V., Arrieta, A., Tonella, P., Sagardui, G., Arratibel, M.: Generating metamorphic relations for cyber-physical systems with genetic programming: an industrial case study. In: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 1264–1274. ESEC/FSE 2021, Association for Computing Machinery, New York, NY, USA (2021)
3. Beckers, S., Vennekens, J.: A principled approach to defining actual causation. *Synthese* **195**(2), 835–862 (2018)
4. Beer, I., Ben-David, S., Chockler, H., Orni, A., Trefler, R.: Explaining counterexamples using causality. *Formal Methods in System Design* **40**(1), 20–40 (2012)
5. Chen, T.Y., Kuo, F.C., Liu, H., Poon, P.L., Towey, D., Tse, T.H., Zhou, Z.Q.: Metamorphic testing: a review of challenges and opportunities. *ACM Computing Surveys* **51** (2018)
6. Chockler, H., Halpern, J.Y.: Responsibility and blame: a structural-model approach. *Journal of Artificial Intelligence Research* **22**, 93–115 (2004)
7. Chockler, H., Kroening, D., Sun, Y.: Explanations for occluded images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
8. Clark, A.G., Foster, M., Walkinshaw, N., Hierons, R.M.: Metamorphic testing with causal graphs. In: IEEE Conference on Software Testing, Verification and Validation (ICST). pp. 153–164. IEEE (2023)

9. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms, fourth edition. MIT Press (2022)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002)
11. Fainekos, G.E., Pappas, G.J.: Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science* **410**(42), 4262–4291 (2009)
12. Halpern, J.Y.: A modification of the Halpern-Pearl definition of causality. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. pp. 3022–3033. IJCAI’15, AAAI Press (2015)
13. Halpern, J.Y.: Actual causality. MIT Press (2016)
14. Halpern, J.Y., Pearl, J.: Causes and explanations: a structural-model approach. part i: causes. *The British Journal for the Philosophy of Science* **56**(4), 843–887 (2005)
15. Ibrahim, A., Pretschner, A.: From Checking to Inference: Actual Causality Computations as Optimization Problems. In: *Formal Methods*, pp. 343–359. Springer International Publishing (2020)
16. Ibrahim, A., Rehwald, S., Pretschner, A.: Efficiently Checking Actual Causality with SAT Solving (2019)
17. Mazouni, Q., Gotlieb, A., Spieker, H., Acher, M., Combemale, B.: Mutation-guided metamorphic testing of optimality in ai planning. *Software Testing, Verification and Reliability* **35**, e1898 (2025)
18. Meliou, A., Gatterbauer, W., Halpern, J., Koch, C., Moore, K., Suciu, D.: Causality in Databases. *IEEE Data Eng. Bull.* **33**, 59–67 (2010)
19. Özcan, D., Alrajeh, D., Craven, R.: Reasoning about actual causality in answer set programming. In: *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*. vol. 22, pp. 610–620 (2025)
20. Pearl, J.: Causality: models, reasoning, and inference. Cambridge University Press (2009)
21. Rafieioskouei, A., Bonakdarpour, B.: Efficient Discovery of Actual Causality Using Abstraction Refinement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **43**(11), 4274–4285 (2024)
22. Reynd, S., Diaconescu, A., Dessalles, J.L.: Searching for actual causes: Approximate algorithms with adjustable precision. In: *NeurIPS 2025 Workshop on CauScien: Uncovering Causality in Science* (2025)
23. Valle, P.: Metamorphic Testing of Autonomous Vehicles: A Case Study on Simulink. In: *2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. pp. 105–107 (2021)
24. Willard, S.: General topology. Courier Corporation (2012)