

Constrained Causal Decision Dilemmas

Abraham Moore Odell¹, Andrew Forney¹, Adrienne Raglin², Sunny Basak², and
Peter Khooshabeh²

¹ Loyola Marymount University, Los Angeles, CA 90045, USA
amooreod@lmu.edu, andrew.forney@lmu.edu

² DEVCOM Army Research Laboratory
{adrienne.raglin2.civ, peter.khooshabehadeh2.civ}@army.mil,
anjon.sunny@gmail.com

Abstract. Decision-makers in high-stakes, high-stress, and time-limited scenarios (such as emergency first-responders) face challenges in factoring in all relevant information, making optimal choices as a response, and avoiding choice paralysis when decisions are difficult or demand ethical considerations. This work attempts to better advise human deciders in these systems by introducing and formalizing such Constrained Causal Decision Dilemmas (C2D2s) as sequential decision-making scenarios characterized by (1) decisions that differently affect the environment vs. the agent’s perspective of the environment at different tiers of the Pearlian causal hierarchy, (2) restrictions in choices imposed by some conditions like a time limit, and (3) some possibly multi-dimensional utility describing multiple objectives desired in an outcome. To provide optimal recommendations in C2D2s, we propose a Causal Decision Network (CDN) used in companion with a novel type of Causal Expectimax Search (CES) that can plan for the best Course of Action (COA). Not only do simulations demonstrate the efficacy of these techniques compared to traditional, associative approaches, but also expose a number of interesting edge cases to causal decision-making that are handled by CDNs and CES: (a) the Ostrich Effect, by which agents may avoid investigating some variable should doing so reduce expected utility, (b) the Possum Effect, during which the best choice may be to make no choice at all, (c) the Double Check Effect, in which the best policy is to investigate the outcome of an intervention and repeat if the result is unsatisfactory, and (d) the Pseudo-counterfactual Effect, by which an optimal choice may involve changing the result of an investigation through intervention. Taken together, the contributions of this work include the formalization of a new class of decision problems, demonstration of causal tools in their solution, and discussion of applications to data-sparse domains.

Keywords: Causal Decision Support, Sequential Planning, Counterfactual Inference

1 Introduction

Decision-makers in high-stakes, time-sensitive environments such as first responder captains, police officers, and emergency medics encounter obstacles in making

both swift and optimal decisions: consolidating an array of relevant factors from often uncertain reports and sensors [32,31], managing quick judgments amidst conflicting priorities [21], and avoiding decision paralysis [36]. The complexity of these challenges is compounded by numerous potential human biases [34], such as the tendency of experts to rely on intuition when a more thorough investigation of important factors may be necessary [18], and reconciling implicit knowledge with increasingly intricate or changing standard operating procedures (SoPs) [23]. For instance, the optimal way to douse a fire may depend on knowing its fuel source, but taking the time to perform such an investigation may compete with the urgency of acting swiftly if a life is threatened.

Although real-time recommender systems may provide assistive intelligence in these domains, the ability to develop specialized decision-support technology is likewise complicated because either (a) there is no training data to emulate, or (b) if there is, emulation may be undesirable because the data may be contaminated with human biases and suboptimal decisions that are undesirable to mimic [12]. Moreover, the emulating ML algorithms might confuse correlation with causation of chosen actions, compromising the ability to make explainable, ethical, and optimal decisions that follow desired protocol. In lieu of such data, SOPs could provide a means to derive recommended Courses of Action (COAs), but fall short of their potential because (c) managing exceptions and uncertainty can complicate SOP clarity, which are also (d) often over-simplified so as not to overwhelm a human decider. As the decision-making sciences within each domain grows, SOPs risk becoming too complex to remain useful for deciders under pressure or too simplified and miss crucial variables in the state that lead to optimal choices.

This work provides a scalable alternative / translation of SOPs into the language of causality that is amenable to novel decision-support algorithms that consider the dynamics of investigating and intervening on a problem state to reach a desired outcome. It is situated amongst other work using causal tools to optimally choose interventions [19,8] (the closest of which is [20], but which examines a different problem space and does not consider constraints like time), connect causality to reward and utility [41], and more broadly work that examines causality in the role of sequential decision problems [14,5,10,37].

Similar efforts exist to use the tools of graphical causality to guide decision-making, like the application of Structural Causal Models (SCMs) and counterfactual inference in Dec-POMDPs [38,22,6]. However, this work examines a problem space that differs from those easily modeled in POMDPs on key details, viz., in our setting: (1) the agent may actively affect what observations are gathered in subsequent episodes and may make choices that invalidate previous observations, (2) interactions between choices to observe or intervene on a variable in the system demand specification for planning, e.g., choosing to observe or intervene on the same variable or observing the state of an intervention’s consequence.

Thus, the present paper uniquely considers the ramifications of choices whose effects are situated at different tiers of the Pearlian Causal Hierarchy (PCH) [42,13], and our specific contributions are: (1) to formalize a new problem space

of Constrained Causal Decision Dilemmas (C2D2s) and provide interesting C2D2 examples that challenge traditional decision procedures, (2) to define a Causal Decision Network (CDN) that both models C2D2s (as a potential alternative language for defining SOPs) and can be used in their solution, (3) to demonstrate how CDNs contribute to a new class of causal expectimax planning algorithms, and (4) to support the efficacy of these techniques in simulated settings.

2 Background

To effectively select optimal COAs in our problem space, we must employ a language capable of distinguishing different types of information the agent can gather or enact on the system: (1) how observations update beliefs about the best decisions, and (2) how interventions *affect* the environmental state; we thus employ the language of Structural Causal Models:

Definition 1. (*Structural Causal Model*) [27] A Structural Causal Model is a 4-tuple, $M = \langle U, V, F, P(u) \rangle$ such that: (1) U is a set $\{U_1, U_2, \dots, U_k\}$ of *exogenous* variables (also called *background*), that are determined by factors outside of the model. These variables represent the entry-points of noise into the system. (2) V is a set $\{V_1, V_2, \dots, V_n\}$ of *endogenous* variables that are determined by variables in the model, viz. variables in $U \cup V$. (3) F is a set of structural functions $\{f_1, f_2, \dots, f_n\}$ such that each f_i is a mapping from (the respective domains of) $U_i \cup PA_i$ to V_i where $U_i \subseteq U$ and $PA_i \subseteq V \setminus V_i$ and the entire set F forms a mapping from U to V . In other words, each f_i in $v_i = f_i(pa_i, u_i), i = 1, \dots, n$ assigns a value to V_i that depends on (the values of) a select set of variables. A variable that appears as a parameter in another’s function is considered one of its *direct causes*. (4) $P(u)$ is a probability function defined over the domain of U .

Each SCM induces a causal graph [26] depicting the causal relations of variables upon one another such that if $Y \leftarrow f_Y(X)$ in its structural equation, then $X \rightarrow Y$ in the causal graph. From this graph, we gain definitions of which variables are conditionally independent of others (d-separation [15]), how interventions differ from observations using the causal do-operator [24], and which variables will be affected by interventions through the concept of spurious vs. causal pathways [7]. Given that agents in the present problem space may choose to either observe the value of certain variables or not, familiarity with the phenomenon of Simpson’s Paradox (whereby the optimal decision can change depending upon which covariates are controlled) is likewise useful [9,29].

However, SCMs guide only the ability of the agent to understand the causal relationships between variables in their system; defining the best *sequence* of actions composing a COA for maximizing some desirable outcome requires additional tools. We thus expect readers to have some familiarity with utility theory for defining agent preferences [40], expectimax search for planning sequences of utility-driven actions [16], and the value of information for traditional approaches for assessing the potential gains in utility from investigations [11,32]. A cursory review of how different decision-making strategies define optimality at different

Table 1. Overview of Decision Theories as they pertain to layers ($\mathcal{L}$) in the Pearlian Causal Hierarchy (PCH) with maximization criteria defined for agent choice X , observed evidence Z , and expectation maximization of some desirable outcome Y .

$\mathcal{L}_1$	Evidential Decision Theory (EDT) [4,1]: An agent’s choice X may <i>both</i> be evidence-of, and influence-on, the state of its environment, and so maximizes the observational expectation of Y : $x^* = \arg \max_{x \in X} E[Y X = x, Z = z]$
$\mathcal{L}_2$	Causal Decision Theory (CDT) [41,35]: An agent’s choice X is not indicative of the state of the environment, and is a rational act that maximizes the interventional expectation over Y : $x^* = \arg \max_{x \in X} E[Y \text{do}(X = x), Z = z]$
$\mathcal{L}_3$	Regret Decision Theory (RDT) [3]: An agent’s observational choice $X = x'$ serves as a context for the state of its environment in which to then act interventionally, thus maximizing the counterfactual expectation of Y : $x^* = \arg \max_{x \in X} E[Y_{X=x} X = x', Z = z]$

tiers of the Pearlian Causal Hierarchy (PCH) can be found in Tab. 1; these are relevant comparisons because, although [12] proved that $RDT \geq CDT \geq EDT$ in terms of maximizing utility, reward, or modeling transitions involving chosen acts, there are still recent works that do not use causal operators for action choices (e.g., [39,30]), including some textbooks [33].

3 Constrained Causal Decision Dilemmas (C2D2s)

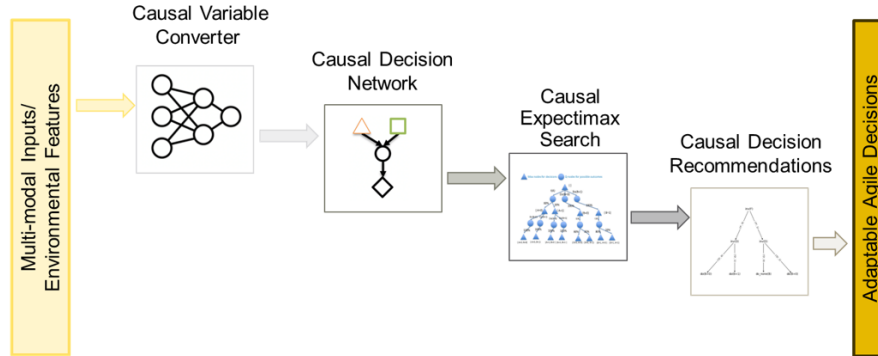


Fig. 1. Pipeline of causal decision-support leading (from left-to-right) from raw multi-modal data, from which some model (left for future directions) serves as a causal variable converter to extract variables and values encoded in a Causal Decision Network (CDN), which is then used during a form of Causal Expectimax Search (CES) to yield an interpretable Course of Action (COA).

In order to leverage the higher-order decision-tools afforded by the tools of causality, our broader goal is to develop a causal decision-support pipeline leading from raw, multi-modal data (e.g., drone footage or social media posts) to easily human-interpretable recommendations. In the following section, we describe the intermediary steps to accomplish this pipeline, though leave some components for future work (see outline in Fig. 1).

Although SCMs serve as oracles for the effects of interventions, it is generally assumed that these can be queried for any estimable causal effect or hypothetically obtained evidence; what is left is to connect these effects to what is realistically possible to either change or observe in reality, e.g., it is reasonable to imagine a firefighter wishing to observe the wind direction while being unable to alter it. We thus expand SCMs into this space through:

Definition 2. (*Causal Decision Networks (CDNs)*) A Causal Decision Network (CDN) is a 4-tuple $C = \langle M, I, A, Y \rangle$: an enhancement to traditional SCMs that are used for specifying not only how variables in the causal system interact, but also which are amenable to observation and/or intervention; a CDN consists of:

1. *SCM*, $M = \langle U, V, F, P(u) \rangle$, a traditional Structural Causal Model as defined in Def. 1 encoding the cause-effect relations between system variables.
2. *Investigations*, I , a set of variables in the SCM amenable to observation, i.e., whose value can be exposed via this choice by the agent. Represented graphically as triangles pointing to variables in $V \cup U$ that the investigation exposes.
3. *Acts* (AKA interventions), A , a set of variables in the SCM that can be intervened upon, i.e., whose value can be forced to a desired one despite the “natural” functions deciding its value. Represented graphically as squares pointing to variables in $V \cup U$ that the intervention affects. Interventions are implemented using the *do*-operator [28].
4. *Utility-scored Outcome Variables*, Y , whose values are scored by some utility function f deciding the quality of the outcome, conditional upon the state, i.e., $f(Y|S)$. Higher utility implies more desirable. Represented graphically as any variable in $V \cup U$ pointing to a diamond (i.e., the parents of any diamond are those scored by the utility function).

Definition 3. (*Constrained Causal Decision Dilemma (C2D2)*) A Constrained Causal Decision Dilemma (C2D2) formalizes the conditions under which choices in a CDN are constrained, which outcomes are considered optimal, and any other parameters under which a rational decision-maker would use to arrive at the best course of action (COA), consisting of:

1. *CDN*, modeling the variables and decisions of interest, primarily predicating where and how the agent has influence over the system and the effects of those influences.
2. *State* consists of an assignment of values to variables in the CDN, only portions of which may be known to the deciding agent; we thus distinguish two perspectives of the C2D2 state:

- (a) *Hidden / True State*, $X_t = \langle V_t, T \rangle$, a tuple consisting of $V_t = \{Var_1 : v_1, Var_2 : v_2, \dots\}$ a mapping of each variable $Var_i \in CDN$ to its current value, and remaining time available T_{rem} ; both are amenable to change through the agent's actions.
 - (b) *Agent State*, what the agent knows about the hidden state and its options for actions, $s = \langle e, a, T_{rem}, D \rangle$, a tuple storing the set of known evidence e (e.g., $civiliansPresent = 0$), applied interventions/acts a (e.g., $do(waterJets = 1)$) as well as the remaining time budget T_{rem} and available decisions D . This includes modeling the *initial agent state*, s_0 , which may begin with existing evidence and some max time budget T_{max} .
3. *Transition function*, $Tr(s, d) = s'$ defining how the state s changes to some next state s' with each decision d , as well as how much of a time cost c_d each investigation and act taxes the budget. We thus herein define the transitions of the *true / hidden state* and leave transitions of the agent state to the planning algorithms discussed in the following Sec. 3.2.
- (a) *Investigations* will reveal the current value h_t of variable H_t at time t where $H_t \in X_t$, denoted $inv(H)$, but do not alter the hidden state's values, i.e., $Tr(X_t, inv(H_t)) = \langle V_t, T - c_{inv(H_t)} \rangle$.
 - (b) *Acts / Interventions* will set a variable's state at time t to a chosen value, denoted $do(H_t = h_t)$, whose effect will cascade to any descendants of H in the CDN, denoted $desc(H)$, potentially changing the values of any variables along the causal path³ from H , i.e., $Tr(X_t, do(H_t = h_t)) = \langle V_{t+c_{do(H_t=h_t)}}, T - c_{do(H_t=h_t)} \rangle$ where $V_{t+c_{do(H_t=h_t)}}$ is the mutated next state composed of: (1) the intervened variable $H_t = h_t$ forced to its intervened value, (2) all descendants resampled from $desc(H) \sim P(desc(H) | (V_t - H_t) \cup H_t = h_t)$, and (3) all other variables from V_t untouched.
 - (c) *Null choices* are possible options for investigations and acts, denoted $do(X = \emptyset)$ or $inv(X = \emptyset)$, whereby, even if there is still time to take them, the agent may instead abstain at no cost in time nor change in state except to remove the choice from its possible decisions (see Sec. 3.2).
4. *Terminal test*, $term(s) \in \{True, False\}$ where a terminal state defines when decision-making ends, viz., when either all remaining actions take longer than time available, $c_i > t \ \forall i \in D$, or all possible decisions have been made, $D = \emptyset$.

Goal: Our objective is to construct a *C2D2 agent* that, given a problem specified above, produces a *Course of Action (COA)* that is some sequence of investigations and acts that optimally maximizes utility, i.e., $solver(C2D2) = COA$. Herein, order of choices matter for deriving the best COA, but for real-time decisions, it is the *first* action in any COA that is the most important to optimize, since it is assumed that realistic use of a C2D2 solver would happen within a

³ Note that this transition behavior requires knowledge of the causal structure to determine which variables are downstream effects of which others.

feedback loop, viz., making a choice, observing the state, and then planning again for the next choice given that updated state.

3.1 C2D2 Example Testbeds

To exemplify the definitions of CDNs and C2D2s and demonstrate some interesting quirks to deriving COAs in this domain, we examine the following toy scenarios that motivate our proposed solution. In each example, we make the following assumptions:

Assumption 31 *CDNs as Markovian Causal Bayesian Networks (CBNs)*

To show that the techniques herein hold even when the CDN’s SCM is not fully-specified, each example CDN will be modeled using a Markovian Causal Bayesian Network⁴ to describe the underlying SCM modeling the system of causes and effects. It is assumed that this SCM represents the “true” (however partial) model of reality—the structure and parameters of which are known by the deciding agent in advance—or at least are consistent with the SOPs deriving it. E.g., $C_1 \rightarrow D \leftarrow C_2$ might indicate that a drug D should be administered to stabilize a patient only when 2 conditions, C_1, C_2 , are met.

For simplicity (though without loss of generality), each variable is assumed binary, and unless otherwise stated in the environments below, each investigation and intervention *possible* may also be *abstained-from* via the null-choice $do(X = \emptyset)$, with all variable values as possible acts for variables that are amenable to intervention. Time will be tracked in arbitrary units $t \in \mathbb{N}$ with some per-example time-maximum T . Unless stated otherwise, the time-costs of each choice will be $c_{inv} = 1$ for investigations, $c_{do} = 2$ for acts, and $c_{\emptyset} = 0$ for do-nothing choices.

Assumption 32 *Temporal Intransience*. For decision at time t , d_t , and possible acts a , we assume that, unless acted upon through intervention, the hidden state at time t , X_t , will not change⁵ between time steps, viz. $Tr(X_t, d_t) = \langle V_t, t - c_{d_t} \rangle$ if $d_t \notin a$.

With the above assumptions, we now introduce our example C2D2 testbeds alongside our hypotheses of how agents practicing reasoning at different levels of the PCH (Tab. 1) are expected to perform.

Example 0 - Standard Confounding. CDN_1 in Fig. 2, demonstrates the traditional risk posed by confounding factors in the estimation of causal effects and the importance of decision-order in C2D2s. Herein, an observed confounder,

⁴ A Markovian CBN is a partially-specified SCM in which it is assumed that no unobserved confounders influence any variable, and thus that the model parameters are specified as family conditionals $P(V|parents(V))$ (see [25,17]). We invite future work to examine the role of confounding in C2D2s.

⁵ The generalizability of the techniques that follow do not ebb in violation of this assumption, but would require specification of a transition model between time steps that overcomplicates this introductory work.

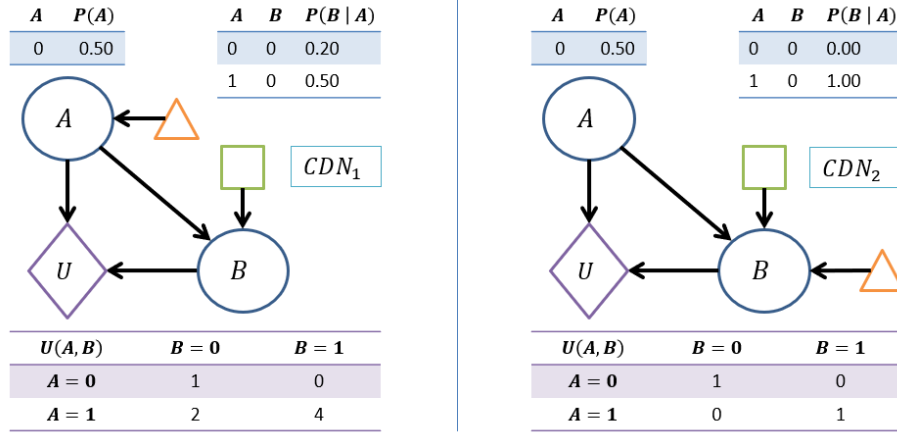


Fig. 2. CDNs used in Exs. 0-3 in which circles represent state variables, triangles pointing to variables amenable to investigation, squares pointing to variables amenable to intervention, and diamonds signifying a utility scoring the states of their parents.

A (amenable to investigation), introduces a spurious back-door path on the causal effect of B (amenable to intervention) on the resulting utility U . If given only $T = 2$, an agent using EDT will erroneously leave this backdoor path open, choosing suboptimally. However, if $T = 3$, all agents will be able to first $inv(A)$ before making an informed choice with B .

Example 1 - The Possum Effect. This scenario also occurs in CDN_1 but with two modifications: there is no longer an investigation possible on $inv(A)$, and we restrict choices on B to $do(B = 0)$ and $do(B = \emptyset)$. In this modified setting, EDT agents will still make the mistake of choosing $B = 0$, but CDT agents should discover that doing nothing yields a higher expected utility than does the utility-reducing $do(B = 0)$, thus giving this the name of the Possum Effect.

Example 2 - The Pseudo-Counterfactual Effect. CDN_2 enables agents to exploit something close to $\mathcal{L}_3$ (RDT) reasoning: when an investigation $inv(B)$ and intervention $do(B)$ are both possible on a single variable. Notably, the COA $[inv(B), do(B = b)]$ has some merit to explore, but not at the first two layers of reasoning: both associational ($\mathcal{L}_1 = \text{EDT}$) and interventional ($\mathcal{L}_2 = \text{CDT}$) agents would simply ignore the investigated value of B before considering which value to assign that maximizes expected utility. Considered through the lens of a $\mathcal{L}_3$ decider, observing the state of B provides information about the state of A through which to then best intervene on B because the utility function has an interaction between A, B .

Example 3 - The Double-Check Effect. In CDN_3 , we have an intervention possible on A that culminates in one of two outcomes, $B = 0$ (desirable) and $B = 1$ (undesirable). Much like an EMT could try to stabilize a patient, recheck

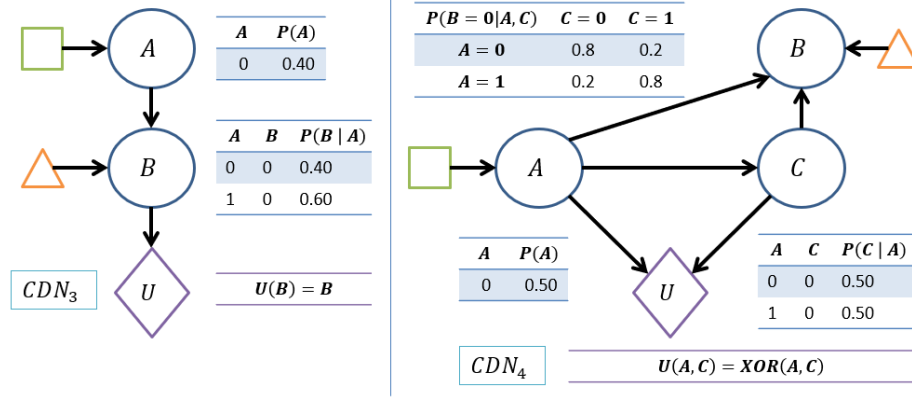


Fig. 3. CDNs used in Exs. 4 and 5 with graphical notation the same as in Fig. 2.

vitals, and then attempt to stabilize again if the vitals looked poor, so too does this scenario, for $T \geq 5$, allow the agent to repeat an intervention if the outcome was first not desirable. Interestingly, doing so here constitutes the optimal policy even though it requires controlling for a variable along the causal path of $do(A)$, generally forbidden by causal wisdom surrounding mediators [7].

Example 4 - The Ostrich Effect. In CDN_4 we have an intervention on A that affects two variables, B, C , and where the utility is an interaction between $f(A, C)$. This makes the state of C important to determine just as with the Double-Check Effect, but the only evidence of the state of C is through B : an investigation $inv(B)$ possible along a *spurious* path from the intervention $do(A)$. An optimal COA *should* solve this scenario as it did the Double-Check effect, but there are two main impediments: (1) once again in contrast to traditional causal wisdom, opening a spurious path when estimating the effect of an intervention can contaminate the results [7], and (2) (the namesake of this example), from a strictly utilitarian perspective, the agent would be disincentivized to $inv(B)$ because, no matter its outcome, the expected utility *after* observing B is unchanged compared to what it was *before* (due to the opening of the spurious path that was previously closed). An optimal C2D2 solver must thus avoid “sticking its head in the sand” and to take COAs whose investigations might not modify its expected utility.

3.2 Solving C2D2s

We now turn to implementation of a procedure for finding optimal COAs, beginning with a “baseline” implementation of a type of C2D2 Expectimax Search that can solve some, but not all, of the scenarios proposed in Sec. 3.1. Moreover, the formats of several computations in the following expectimax strategy will depend on the agent’s employed decision theory (outlined in Tab. 1). Enhancements will later be made on top of the following baseline approach:

Definition 4. (*Expectimax Search (ES) for C2D2s*) An expectimax search strategy for solving a C2D2 explores a type of expectimax tree that plans for all possible sequences of investigations and acts within the given time budget and consists of:

1. *Max Nodes* determining the points at which to make choices and whose value $V(s)$ is the maximum of any chance-node child value $Q(s, a)$ such that $V(s) = \max_a Q(s, a)$.
2. *Chance Nodes* determining the probability-weighted possible transitions / next states s' from a given max node state s with value $Q(s, a) = \sum_{s'} P(s'|s) * V(s')$.
3. *Transition probabilities* $P(s'|s)$ that are computed using the CDN and the current C2D2 agent state s such that:
 - (a) Investigation transitions, $inv(H)$, are computed using the following probabilities that are sensitive to the agent's decision strategy: $\mathcal{L}_1 = EDT \Rightarrow P(H|e, a)$, $\mathcal{L}_2 = CDT \Rightarrow P(H|e, do(a))$, and $\mathcal{L}_3 = RDT \Rightarrow P(H_a|e)$, where e is all current observed evidence in S and a all current acts.
 - (b) Act transitions, forcing some variable H to obtain value h , are assumed to be executed with certainty.
4. *Terminal nodes* are scored according to the expected utility $EU[Y|s] = \sum_{y \in Y} P(y|s) * f(y)$ of the outcome variables specified in the C2D2, Y , state gives $s = \{e, do(a)\}$, and agent's decision strategy: $\mathcal{L}_1 = EDT \Rightarrow EU[Y|e, a]$, $\mathcal{L}_2 = CDT \Rightarrow EU[Y|e, do(a)]$, and $\mathcal{L}_3 = RDT \Rightarrow EU[Y_a|e]$.

We thus propose Causal Expectimax Search (CES), which enhances ES for C2D2s via:

CES Enhancement - Intervention Planning: Given that C2D2s assume an underlying SCM dictating the causal mechanics of the system, interventions carry some side effects: (1) a newly applied intervention will cascade to any descendants along its causal path, thus invalidating any previous investigations and interventions on its descendants, (2) due to the temporal-intransience assumption, there is no need to investigate a variable after intervening upon it because we assume it obtains, and then retains, its intervened value unless it or an ancestor are later intervened upon, and (3) given examples like the Double Check Effect, there may be situations in which the optimal COA requires multiple interventions on the same variable. These insights thus specify and distinguish the transitions of the *agent state* $s = \langle e, a, T, D \rangle$ (see Sec. 3) associated with investigations vs. acts:

CES Investigation Transitions, for investigations $inv(H)$ adds observed value h that is not presently part of the agent state, i.e., $H \notin s$. $Tr(s, inv(H)) = \langle (e \cup \{H = h\}), a, T - c_{inv(H)}, D - \{inv(H)\} \rangle$

CES Act Transitions, for intervention $do(H = h)$, must erase any previous observations or interventions that are in the descendants of H , denoted $desc(H)$, given that acts will resample variables along the causal path from H , re-enabling them to be later investigated or intervened upon. $Tr(s, do(H = h)) = \langle (e - H) - desc(H), (a \cup \{H = h\}) - desc(H), t - c_{do(H=h)}, (D - \{do(H = h), inv(H)\}) \cup$

Input: $cdn, query_var, evidence, acts$
Output: Query result for $query_var$ in the updated model
Function pcc

```

     $pcc\_model \leftarrow cdn.copy()$ 
     $J \leftarrow ((cdn.all\_vars() - evidence) - acts) - cdn.causal\_paths(acts)$ 
    for  $j \in J$  do
        // Abduction
         $pcc\_model.update\_cpt(P(j \mid parents(j)), P(j \mid parents(j), evidence))$ 
    end
    // Action: Obtain interventional model  $M_{acts}$ 
     $pcc\_model \leftarrow pcc\_model.do(acts)$ 
    // Prediction: estimate query in  $M_{acts}$ 
    return  $pcc\_model.query(query\_var)$ 

```

Algorithm 1: Pseudocode for PCC queries

$og_desc(H)\rangle$ where $og_desc(H)$ denotes the original set of investigations and acts licensed from the CDN for only the descendants of H .

CES Null Choice Transitions, in order to advance CES closer to a terminal state (where $D = \emptyset$) without forcing it to make choices that lower utility, CES defines $Tr(s, d = do(X = \emptyset)) = Tr(s, d = inv(X = \emptyset)) = \langle e, a, t, D - d \rangle$

CES Enhancement - Pseudo-Counterfactual Computation [PCC]: Typically, computation of *structural counterfactuals* follows only from a fully-specified SCM and can be obtained through a 3-step process of abduction, action, and prediction [2]. However, for solving a C2D2, our goal is not exactly the same because we are not attempting to estimate the effect of some action contrary to reality at a fixed point in the past, but instead, the effect of an action *following* an observed context on the same variable. Because C2D2s allow for sequences of observation and intervention on the same variables (coupled with the temporal-intransience assumption in Assumption 32), we can adapt the traditional steps for counterfactual inference enabling agents to perform a type of pseudo-counterfactual *RDT* decision-making like is required for solving Example 2. In these cases, where the agent state $s = \{e, do(a)\}$ contains contrasting evidence and intervention⁶ on some $H = h \in e$ and $do(H = h') \in do(a)$, we instead replace all probabilistic queries on the CDN computations with an approximation of the 3-step counterfactual computation as given in Alg. 1.

CES Enhancement - Conditional COA (CCOA) Generation: Fig. 4 depicts CDN_1 and its associated CES solution. From the resulting CES tree, we can easily produce a COA through a max-path traversal, but with one quirk: after the optimal first step of $inv(A)$, the optimal *next* step is conditional on the result of that investigation. We thus must branch at each investigation to create a Conditional COA (as depicted in the figure), forming an interpretable decision-tree that can be communicated to a human advisee, or simply deliver the

⁶ Note that due to the CES Intervention Planning enhancement, this will only occur on CES paths in which a variable is *first* investigated $inv(H)$ and later intervened upon $do(H = h)$.

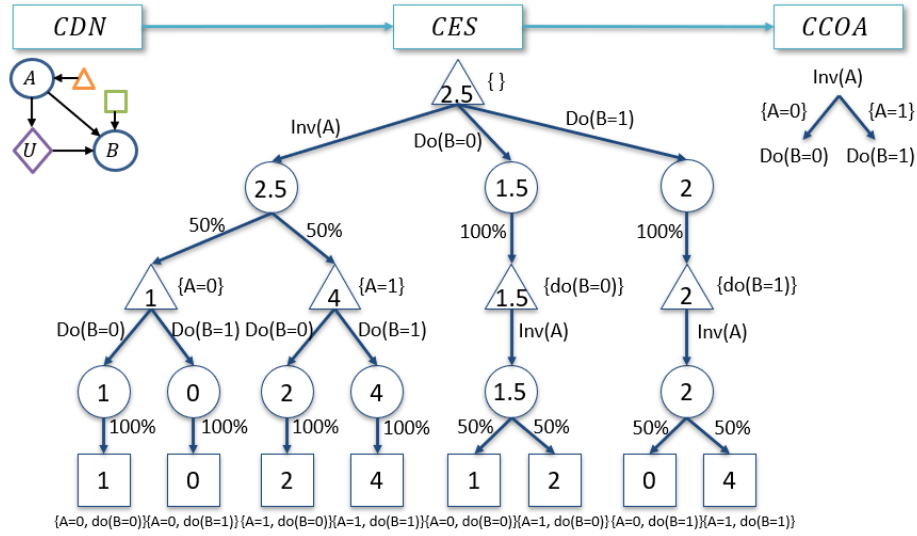


Fig. 4. CES values from CDN_1 with $T = 3$. Null actions omitted for brevity.

root of this CCOA as the next best action. One edge case that this enhancement handles through a simple tree traversal following the original CCOA generation: if all branches from an investigation are equivalent to its others, that investigation can be safely ignored and its branches merged (i.e., if the agent would make the same future choices in each possible observable outcome of $inv(A)$, the effort to investigate A is wasted). Functionally, this leads to the same expected utility but leads to more concise CCOAs that cost less total time.⁷

4 Method

To validate the approaches in Sec. 3.2, we conduct simulations to trace how different policies and their resultant COAs perform in the examples of Sec. 3.1.

Procedure. The `c2d2_sim` procedure in Alg. 2 details the iterated decision process through which policies are tested in repeated rounds of making choices, seeing the outcomes of those choices, and eventually reaching terminals whose utility scores are recorded for that round.⁸ These round utilities are averaged across some n Monte Carlo repetitions as indication of the agent’s expected performance for that CDN. ANOVAs are then used to find any significantly different performances between agents in each of the Example scenario CDNs posed in Sec. 3.1.

⁷ “Time” here refers to the time cost specified in the C2D2; a computational analysis is sketched in Appendix A.3.

⁸ All simulation code can be found at <https://bit.ly/3HJijZ2>, and instructions for executing in Appendix A.

Input: $cdn, agents, n$
Output: $results$
Function $c2d2Sim$

```

    // Stores agent results per MC repetition
    results ← {}
    for  $n \in N$  do
        // CDN generates a hidden sim state for sim  $n$ 
        true_state ←  $cdn.sample()$ 
        for  $ag \in agents$  do
            // Copy for agent with all vals hidden
            curr_state ←  $true\_state.get\_ag\_copy()$ 
            while  $\neg terminal(curr\_state)$  do
                // Agent chooses from given state
                choice ←  $ag.choose(cdn, curr\_state)$ 
                // Transition reveals part of state
                curr_state ←  $cdn.Tr(curr\_state, choice)$ 
            end
            // Log this MC repetition's terminal-state util
            results[ag].add( $f(curr\_state)$ )
        end
    end
    return results

```

Algorithm 2: Pseudocode for C2D2 simulation

Evaluated Agents. There are two agent-policy dimensions to compare as they become appropriate in each Example CDN: (1) the agent’s employed decision theory as outlined in Def. 4 (EDT, CDT, or RDT), and (2) standard ES vs. CES. We distinguish such permutations using the notation like: π_{CES}^{CDT} , an agent using CDT and CES search (notably absent: the combination of *RDT* and *ES* because *RDT* requires the intervention planning transitions of *CES*). Although many time-horizons are not evaluated due to brevity, the most important comparisons posed by the problems in each Example CDN are listed in the columns of Tab. 2, and randomized parameters to demonstrate robustness (including additional, larger simulation CDNs) are included in the Appendices C, D, respectively. Across all simulations, we average performance across $n = 10,000$ MC repetitions.

5 Results & Discussion

The results across all simulation environments are displayed in Tab. 2. The underlined values indicate the policy obtaining the highest average utility for that simulation row *or* those whose are equally, maximally high but not significantly different. Examining these results reveals that, in all examples, CES variants meet or exceed ES performance, and reiterate the risk of employing EDT.

In reference to each row’s CDN: **EX0:** Demonstrates not only the superiority of CDT and RDT over EDT in predicting the effects of interventions and avoiding spurious pathways (for $T = 2$), but also the effectiveness of CES in

Table 2. Simulation results from `c2d2_sim` for the Examples in Sec 3.1 comparing select policies.

	π_{ES}^{EDT}	π_{ES}^{CDT}	π_{CES}^{EDT}	π_{CES}^{CDT}	π_{CES}^{RDT}
<i>EX0</i> : $T = 2$	1.505 ± 0.005	2.021 ± 0.020	1.502 ± 0.005	2.021 ± 0.020	2.021 ± 0.020
<i>EX0</i> : $T = 3$	2.516 ± 0.015	2.516 ± 0.015	2.516 ± 0.015	2.516 ± 0.015	2.516 ± 0.015
<i>EX1</i> : $T = 2$	1.507 ± 0.005	1.625 ± 0.016	1.507 ± 0.005	1.625 ± 0.016	1.625 ± 0.016
<i>EX2</i> : $T = 3$	0.501 ± 0.005	0.501 ± 0.005	0.501 ± 0.005	0.501 ± 0.005	0.999 ± 0.005
<i>EX3</i> : $T = 3$	0.596 ± 0.005	0.595 ± 0.005	0.791 ± 0.005	0.798 ± 0.004	0.795 ± 0.004
<i>EX3</i> : $T = 4$	0.607 ± 0.005	0.599 ± 0.005	0.788 ± 0.004	0.791 ± 0.004	0.797 ± 0.004
<i>EX3</i> : $T = 5$	0.611 ± 0.005	0.596 ± 0.005	0.838 ± 0.004	0.835 ± 0.004	0.840 ± 0.004
<i>EX4</i> : $T = 5$	0.495 ± 0.005	0.497 ± 0.005	0.651 ± 0.005	0.649 ± 0.005	0.647 ± 0.005

preferring COAs that prioritize investigations that later inform optimal acts ($T = 3$). **EX1:** Likewise demonstrates the superiority of CDT and RDT over EDT in circumventing the Possum Effect. **EX2:** Shows the advantages of the PCC enhancement to CES, allowing RDT to rise to the top in the $T = 3$ case. **EX3:** Showcases the interesting behavior that begins at $T = 5$ wherein optimal agents must be capable of performing interventions multiple times if the first outcomes are undesirable, but only if the act is understood to follow as a causal intervention (as CES does, clearing any known evidence along the causal path). **EX4:** Illustrates the effectiveness of CES’ Intervention Planning in solving the Ostrich Effect compared to the ES alternatives.

This paper constitutes only one step in a larger pipeline of time-constrained decision-support; we envision a sequence of (1) expert translation of SOPs into CDNs, (2) computation and caching of CES trees and CCOAs from these CDNs, (3) deployment in real-time scenarios simply retrieving cached recommendations based on context, and (4) augmenting these real-time systems with automated investigation mechanisms, like translating drone footage into CDN variable values. In total, this pipeline promises to relieve decision-makers from the cognitive load of uniting complex features with complex SOPs, all while under the stress of a ticking clock. Outside of the field, the CCOAs can be used to vet the efficacy and intended consequences of SOPs as well as in helping novices learn optimal decisions or determine wherein their COAs deviated from the optimal. We leave it to future work to examine how well SOPs translate to CDNs, how well CCOAs aid actual human decisions, and how automated sensors could be used to assess urgency in order to set a time horizon.

6 Conclusion

In this work, we formalized the novel problems of Constrained Causal Decision Dilemmas (C2D2s, Sec. 3) as sequential decision-problems in which agents must choose some order of investigations and interventions so as to maximize the expectation of a desired goal. By motivating solutions to C2D2s through a variety of canonical example CDNs (Sec. 3.1), we developed enhancements to

traditional expectimax searchers (ES) via Causal Expectimax Search (CES) (Sec. 3.2) alongside challenges to some conventional causal wisdom. Simulation support validates the effectiveness of these approaches (Sec. 5) and causal decision-theories.

References

1. Ahmed, A.: Evidence, Decision and Causality. Cambridge University Press (2014), <https://books.google.com/books?id=7qcdBAAQBAJ>
2. Balke, A., Pearl, J.: Counterfactual probabilities: Computational methods, bounds and applications. In: Uncertainty in artificial intelligence. pp. 46–54. Elsevier (1994)
3. Bareinboim, E., Forney, A., Pearl, J.: Bandits with unobserved confounders: A causal approach. In: Advances in Neural Information Processing Systems. pp. 1342–1350 (2015)
4. Briggs, R.: Normative theories of rational choice: Expected utility. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, spring 2017 edn. (2017)
5. Browne, A., Forney, A.: Exploiting causal structure for transportability in online, multi-agent environments. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems. pp. 199–207 (2022)
6. Buesing, L., Weber, T., Zwols, Y., Racaniere, S., Guez, A., Lespiau, J.B., Heess, N.: Woulda, coulda, shoulda: Counterfactually-guided policy search. arXiv preprint arXiv:1811.06272 (2018)
7. Cinelli, C., Forney, A., Pearl, J.: A crash course in good and bad controls. Sociological Methods & Research p. 00491241221099552 (2020)
8. Deng, Z., Jiang, J., Long, G., Zhang, C.: Causal reinforcement learning: A survey. arXiv preprint arXiv:2307.01452 (2023)
9. Dong, J.: Simpson’s paradox. In: Armitage, P., Colton, T. (eds.) Encyclopedia of Biostatistics, pp. 4108–4110. J. Wiley, New York (1998)
10. Everitt, T., Ortega, P.A., Barnes, E., Legg, S.: Understanding agent incentives using causal influence diagrams. part i: Single action settings. arXiv preprint arXiv:1902.09980 (2019)
11. Felli, J.C., Hazen, G.B.: Sensitivity analysis and the expected value of perfect information. Medical Decision Making **18**(1), 95–109 (1998)
12. Forney, A., Bareinboim, E.: Counterfactual randomization: Rescuing experimental studies from obscured confounding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33-01, pp. 2454–2461. AAAI, Honolulu, Hawaii (2019)
13. Forney, A., Mueller, S.: Causal inference in ai education: A primer. Journal of Causal Inference **10**(1), 141–173 (2022)
14. Forney, A., Pearl, J., Bareinboim, E.: Counterfactual data-fusion for online reinforcement learners. In: International Conference on Machine Learning. pp. 1156–1164. PMLR (2017)
15. Geiger, D., Verma, T., Pearl, J.: d-separation: From theorems to algorithms. In: Machine Intelligence and Pattern Recognition, vol. 10, pp. 139–148. Elsevier (1990)
16. Hauk, T.G.: Search in trees with chance nodes (2004)
17. Heckerman, D.: A bayesian approach to learning causal networks. arXiv preprint arXiv:1302.4958 (2013)
18. Highhouse, S.: Stubborn reliance on intuition and subjectivity in employee selection. Industrial and Organizational Psychology **1**(3), 333–342 (2008)

19. Kaddour, J., Lynch, A., Liu, Q., Kusner, M.J., Silva, R.: Causal machine learning: A survey and open problems. arXiv preprint arXiv:2206.15475 (2022)
20. Lee, S., Bareinboim, E.: Characterizing optimal mixed policies: Where to intervene and what to observe. *Advances in neural information processing systems* **33**, 8565–8576 (2020)
21. Mukherjee, K.: A dual system model of preferences under risk. *Psychological review* **117**(1), 243 (2010)
22. Oberst, M., Sontag, D.: Counterfactual off-policy evaluation with gumbel-max structural causal models. In: *International Conference on Machine Learning*. pp. 4881–4890. PMLR (2019)
23. Park, J., Jung, W., Ha, J., Park, C.: The step complexity measure for emergency operating procedures: measure verification. *Reliability Engineering & System Safety* **77**(1), 45–59 (2002)
24. Pearl, J.: A probabilistic calculus of actions. In: *Uncertainty in artificial intelligence*. pp. 454–462. Elsevier (1994)
25. Pearl, J.: From bayesian networks to causal networks. In: *Mathematical models for handling partial knowledge in artificial intelligence*, pp. 157–182. Springer (1995)
26. Pearl, J.: Causal inference in statistics: An overview (2009)
27. Pearl, J.: *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, second edn. (2009)
28. Pearl, J.: Causal inference. *Causality: objectives and assessment* pp. 39–58 (2010)
29. Pearl, J.: Comment: understanding simpson’s paradox. In: *Probabilistic and causal inference: The works of judea Pearl*, pp. 399–412 (2022)
30. Primanita, A., Khalid, M.N.A., Iida, H.: Computing games: Bridging the gap between search and entertainment. *IEEE Access* **9**, 72087–72102 (2021)
31. Raglin, A., Emlet, A., Caylor, J., Richardson, J., Mittrick, M., Metu, S.: Uncertainty of information (uoi) taxonomy assessment based on experimental user study results. In: *International Conference on Human-Computer Interaction*. pp. 290–301. Springer (2022)
32. Raglin, A., Moraffah, R., Liu, H.: Causality and uncertainty of information for content understanding. In: *2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI)*. pp. 109–113. IEEE (2020)
33. Russell, S.J., Norvig, P.: *Artificial Intelligence: A Modern Approach*, Global Edition. Pearson (2021)
34. Saposnik, G., Redelmeier, D., Ruff, C.C., Tobler, P.N.: Cognitive biases associated with medical decisions: a systematic review. *BMC medical informatics and decision making* **16**(1), 1–14 (2016)
35. Savage, C.W.: *Scientific Theories*. No. v. 14 in *Minnesota studies in the philosophy of science*, University of Minnesota Press (1990), <https://books.google.com/books?id=GowI4u5Bd6MC>
36. Schwartz, B.: The paradox of choice. *Positive psychology in practice: Promoting human flourishing in work, health, education, and everyday life* pp. 121–138 (2015)
37. Triantafyllou, S., Singla, A., Radanovic, G.: Actual causality and responsibility attribution in decentralized partially observable markov decision processes. In: *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*. pp. 739–752 (2022)
38. Tsirtsis, S., Rodriguez, M.: Finding counterfactually optimal action sequences in continuous state spaces. *Advances in Neural Information Processing Systems* **36** (2024)

- 39. Vignon, C., Rabault, J., Vinuesa, R.: Recent advances in applying deep reinforcement learning for flow control: Perspectives and future directions. *Physics of fluids* **35**(3) (2023)
- 40. Von Neumann, J., Morgenstern, O.: *Theory of games and economic behavior*, 2nd rev (1947)
- 41. Weirich, P.: Causal decision theory. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2016 edn. (2016)
- 42. Zečević, M., Dhami, D.S., Kersting, K.: Not all causal inference is the same. *Transactions on Machine Learning Research* (2023)

A Simulation Instructions

A.1 Defining CDNs

The CDN class is encoded in the file "cdn.py." This object is parameterized by (nodes, edges, cpds, actions, util_func, util_nodes). A CDN contains a pgmpy Bayesian Network (text), as well as other values useful for computing decision problems.

- nodes: a list of capital letters denoting the nodes in the Bayesian Network
- edges: a list of edges in (parent, child) format, where parent and child are capital letter node names
- cpds: a list containing pgmpy CPDs for each node in the network (conditional probability distributions): text
- actions: a list of legal actions in the decision problem. Each action is a dictionary object containing the corresponding node name, action type, set value for intervention (do) actions, and associated time cost. action types include: do (intervene), inv (investigation), do_none, and inv_none
- util_func: a function parameterized by a dictionary with key:value for each parameter in the model's utility function
- util_nodes: a list of capital letter node names for the nodes evaluated by the utility function

```

1 def example_util(states):
2     return {(0, 0): 1, (0, 1): 0, (1, 0): 2, (1, 1): 4}
3         [(states['A'], states['B'])]
4
5 example_model = CDN(
6     nodes=['A', 'B'],
7     edges=[('A', 'B')],
8     cpds=[TabularCPD('A', 2, [[0.5], [0.5]]),
9           TabularCPD('B', 2, [[0.2, 0.5],
10                                [0.8, 0.5]],
11                                evidence=['A'], evidence_card=[2])],
12     actions=[
13         {"node": 'A', "action_type": 'inv', "time_cost": 1},
14         {"node": 'B', "action_type": 'do', "value": 0, "
15         time_cost": 2},
16         {"node": 'B', "action_type": 'do', "value": 1, "
17         time_cost": 2},
18         {"node": 'B', "action_type": 'do_none', "time_cost":
19         0}],
20     util_func=example_util,
21     util_nodes=['A', 'B'])

```

Listing 1.1. Syntax for defining a CDN.

A.2 Running Simulations

To run a simulation and generate a Microsoft Excel (.xlsx) table output of the results navigate to "test.py".

To modify the test parameters, first navigate to and expand the "generate_test_table_ES_vs_CES" function. Within this function the simulation's CDNs, and simulation parameters are instantiated. "columns" holds the algorithms used in the simulation.

- 'EDT_ES': Evidential Decision Theory, Expectimax Search
- 'CDT_ES': Causal Decision Theory, Expectimax Search
- 'EDT_CES': Evidential Decision Theory, Causal Expectimax Search
- 'CDT_CES': Causal Decision Theory, Causal Expectimax Search
- 'RDT_CES': Regret Decision Theory, Causal Expectimax Search

"num_trials" is the integer number of Monte Carlo repetitions to run "secondary_table_limit" is the number of trials to include in the more detailed, algorithm by algorithm secondary sheets "table_rows" is a list of 3-tuples (name, model, time_limit) of models and time limits to test for each algorithm "table_columns" is a list of 4-tuples (name, search_method, obs_only, pcc) the search method can be either the string "ES" for Expectimax Search, or "CES" for Causal Expectimax Search "obs_only" is the boolean for observation only (True if the model can collect only investigative evidence, False if it can collect both investigative and interventional evidence) "pcc" is the boolean for Pseudo-Counterfactual (True if RDT, False otherwise) To control if the simulation randomizes each model's CPDs, set "randomize_models=True" in the call of sim_trials

CALL THE "generate_test_table_ES_vs_CES" FROM WITHIN "test.py" TO GENERATE THE TEST TABLE.

To simply run the suite of pytest tests covering the code, first comment out or delete any call to "generate_test_table_ES_vs_CES" within "test.py", then navigate the src and run "pytest test.py".

A.3 Computation

For exact inference, queries on a CBN using something like variable elimination are exponential in the induced width of the network where, for n vars in the network, a being the number of possible actions, and w the induced width: $O(nd^{w+1})$. This can be reduced by sampling like with Gibbs sampling to $O(Nna)$ for N samples. This would be required for all $O(a^T)$ expectations computed at leaf nodes in the CES tree, but note that each query of this type need be computed exactly once for each unique agent state before it can be cached and reused. We reflect again that the bulk of this computation would happen offline during the planning / SOP generation stage, with cached results ready when needed for live decision-making.

Assuming all non-null-choice decisions have integer time cost $c_d > 0$, and that there are $|D|$ total decisions possible, naïve computational cost of the expectimax

tree reaches the typical exponential for a depth-first tree generation: $O(|D|^t)$. However, during the course of its generation, many probabilistic queries and subtrees will repeat, allowing caching to reduce the work to only the number of unique agent states at the product of investigations and acts.

Computation needs on small models is fairly trivial; on a 2021 Apple M1 Macbook Pro with 16 GB of RAM, and not caching node values due to comparisons across different policies, 1k rounds of all experimental models took roughly 7 minutes for all 5 of the compared policies in total.

B Conditional COAs

The following are all CCOAs generated in the simulations run from the Results Sec. 5, separated by policy. In some cases, agents that would normally take the null action may instead opt to simply investigate a node, which does not change the state but provides information about it.

B.1 Evidential Decision Theory Expectimax Search (EDT-ES)

```

1 EDT_ES - EX0_t2 - T = 2
2 [do(B=0), ]
3
4 EDT_ES - EX0_t3 - T = 3
5 [inv(A), {'A', 0): [do(B=0), ]}, {'A', 1): [do(B=1), ]}, ]
6
7 EDT_ES - EX1_t2 - T = 2
8 [do(B=0), ]
9
10 EDT_ES - EX2_t2 - T = 2
11 [inv(B), {'B', 0): []}, {'B', 1): []}, ]
12
13 EDT_ES - EX2_t3 - T = 3
14 [inv(B), {'B', 0): []}, {'B', 1): []}, ]
15
16 EDT_ES - EX3_t3 - T = 3
17 [do(A=0), inv(B), {'B', 0): []}, {'B', 1): []}, ]
18
19 EDT_ES - EX3_t4 - T = 4
20 [do(A=0), inv(B), {'B', 0): [do_none(A), ]}, {'B', 1): [
    do_none(A), ]}, ]
21
22 EDT_ES - EX3_t5 - T = 5
23 [do(A=0), inv(B), {'B', 0): [do(A=0), ]}, {'B', 1): [do(A
    =0), ]}, ]
24
25 EDT_ES - EX4_t5 - T = 5
26 [inv(B), {'B', 0): [do(A=0), do(A=0), ]}, {'B', 1): [do(A
    =0), do(A=0), ]}, ]

```

Listing 1.2. CCOAs generated by this policy.

B.2 Causal Decision Theory Expectimax Search (CDT-ES)

```

1 CDT_ES - EX0_t2 - T = 2
2 [do(B=1), ]
3
4 CDT_ES - EX0_t3 - T = 3
5 [inv(A), {( 'A' , 0): [do(B=0), ]}, {( 'A' , 1): [do(B=1), ]}, ]
6
7 CDT_ES - EX1_t2 - T = 2
8 [do_none(B), ]
9
10 CDT_ES - EX2_t2 - T = 2
11 [do(B=0), ]
12
13 CDT_ES - EX2_t3 - T = 3
14 [do(B=0), inv(B), {( 'B' , 0): []}, {( 'B' , 1): []}, ]
15
16 CDT_ES - EX3_t3 - T = 3
17 [do(A=0), inv(B), {( 'B' , 0): []}, {( 'B' , 1): []}, ]
18
19 CDT_ES - EX3_t4 - T = 4
20 [do(A=0), inv(B), {( 'B' , 0): [do_none(A), ]}, {( 'B' , 1): [
    do_none(A), ]}, ]
21
22 CDT_ES - EX3_t5 - T = 5
23 [do(A=0), inv(B), {( 'B' , 0): [do(A=0), ]}, {( 'B' , 1): [do(A
    =0), ]}, ]
24
25 CDT_ES - EX4_t5 - T = 5
26 [inv(B), {( 'B' , 0): [do(A=0), do(A=0), ]}, {( 'B' , 1): [do(A
    =0), do(A=0), ]}, ]

```

Listing 1.3. CCOAs generated by this policy.

B.3 Evidential Decision Theory Causal Expectimax Search (EDT-CES)

```

1 EDT_CES - EX0_t2 - T = 2
2 [do(B=0), ]
3
4 EDT_CES - EX0_t3 - T = 3
5 [inv(A), {( 'A' , 0): [do(B=0), ]}, {( 'A' , 1): [do(B=1), ]}, ]
6
7 EDT_CES - EX1_t2 - T = 2
8 [do(B=0), ]
9
10 EDT_CES - EX2_t2 - T = 2
11 [inv(B), {( 'B' , 0): []}, {( 'B' , 1): []}, ]
12

```

```

13 EDT_CES - EX2_t3 - T = 3
14 [inv(B), {'B', 0}: [], {'B', 1}: []], ]
15
16 EDT_CES - EX3_t3 - T = 3
17 [inv(B), {'B', 0}: [do(A=0), []], {'B', 1}: [do_none(A), []],
18   ]
19 EDT_CES - EX3_t4 - T = 4
20 [inv(B), {'B', 0}: [do(A=0), do_none(A), []], {'B', 1}: [
21   do_none(A), []], ]
22 EDT_CES - EX3_t5 - T = 5
23 [do(A=0), inv(B), {'B', 0}: [do(A=0), []], {'B', 1}: [
24   do_none(A), []], ]
25 EDT_CES - EX4_t5 - T = 5
26 [inv(B), {'B', 0}: [do(A=0), do(A=0), []], {'B', 1}: [
27   do_none(A), []], ]

```

Listing 1.4. CCOAs generated by this policy.

B.4 Causal Decision Theory Causal Expectimax Search (CDT-CES)

```

1 CDT_CES - EX0_t2 - T = 2
2 [do(B=1), ]
3
4 CDT_CES - EX0_t3 - T = 3
5 [inv(A), {'A', 0}: [do(B=0), []], {'A', 1}: [do(B=1), []], ]
6
7 CDT_CES - EX1_t2 - T = 2
8 [do_none(B), ]
9
10 CDT_CES - EX2_t2 - T = 2
11 [do(B=0), ]
12
13 CDT_CES - EX2_t3 - T = 3
14 [do(B=0), inv(B), {'B', 0}: [], {'B', 1}: []], ]
15
16 CDT_CES - EX3_t3 - T = 3
17 [inv(B), {'B', 0}: [do(A=0), []], {'B', 1}: [do_none(A), []],
18   ]
19 CDT_CES - EX3_t4 - T = 4
20 [inv(B), {'B', 0}: [do(A=0), do_none(A), []], {'B', 1}: [
21   do_none(A), []], ]
22 CDT_CES - EX3_t5 - T = 5
23 [do(A=0), inv(B), {'B', 0}: [do(A=0), []], {'B', 1}: [
24   do_none(A), []], ]

```

```

24
25 CDT_CES - EX4_t5 - T = 5
26 [inv(B), {('B', 0): [do(A=0), do(A=0), ]}, {('B', 1): [
    do_none(A), ]}, ]

```

Listing 1.5. CCOAs generated by this policy.

B.5 Regret Decision Theory Causal Expectimax Search (RDT-CES)

```

1 RDT_CES - EX0_t2 - T = 2
2 [do(B=1), ]
3
4 RDT_CES - EX0_t3 - T = 3
5 [inv(A), {('A', 0): [do(B=0), ]}, {('A', 1): [do(B=1), ]}, ]
6
7 RDT_CES - EX1_t2 - T = 2
8 [do_none(B), ]
9
10 RDT_CES - EX2_t2 - T = 2
11 [do(B=0), ]
12
13 RDT_CES - EX2_t3 - T = 3
14 [inv(B), {('B', 0): [do(B=1), ]}, {('B', 1): [do(B=0), ]}, ]
15
16 RDT_CES - EX3_t3 - T = 3
17 [inv(B), {('B', 0): [do(A=0), ]}, {('B', 1): [do_none(A), ]},
    ]
18
19 RDT_CES - EX3_t4 - T = 4
20 [inv(B), {('B', 0): [do(A=0), do_none(A), ]}, {('B', 1): [
    do_none(A), ]}, ]
21
22 RDT_CES - EX3_t5 - T = 5
23 [do(A=0), inv(B), {('B', 0): [do(A=0), ]}, {('B', 1): [
    do_none(A), ]}, ]
24
25 RDT_CES - EX4_t5 - T = 5
26 [inv(B), {('B', 0): [do(A=0), do(A=0), ]}, {('B', 1): [
    do_none(A), ]}, ]

```

Listing 1.6. CCOAs generated by this policy.

C Randomized Examples

Table 3 demonstrates once more the superiority of CES over ES and the risks of employing EDT compared to CDT and RDT; these are consistent empirical findings that are likewise congruent with the proofs of causal decision theory superiority in [14]. Some experiments, viz., Ex2, have these effects washed out in

Table 3. Simulation results from `c2d2_sim` for the Examples in Sec. 3.1, repeated but with randomized parameters. Underlined values highlight the policies with the highest average utility or those that share the highest and are not significantly different.

	π_{ES}^{EDT}	π_{ES}^{CDT}	π_{CES}^{EDT}	π_{CES}^{CDT}	π_{CES}^{RDT}
$EX0 : T = 2$	1.442 ± 0.006	1.466 ± 0.005	1.442 ± 0.006	1.466 ± 0.005	1.466 ± 0.005
$EX0 : T = 3$	2.082 ± 0.005	2.082 ± 0.005	2.082 ± 0.005	2.082 ± 0.005	2.082 ± 0.005
$EX1 : T = 2$	1.465 ± 0.005	1.465 ± 0.005	1.465 ± 0.005	1.465 ± 0.005	1.465 ± 0.005
$EX2 : T = 3$	0.637 ± 0.002	0.637 ± 0.002	0.637 ± 0.002	0.637 ± 0.002	0.637 ± 0.002
$EX3 : T = 3$	0.612 ± 0.002	0.614 ± 0.002	0.830 ± 0.001	0.830 ± 0.001	0.828 ± 0.001
$EX3 : T = 4$	0.612 ± 0.001	0.612 ± 0.002	0.829 ± 0.001	0.830 ± 0.001	0.831 ± 0.001
$EX3 : T = 5$	0.613 ± 0.002	0.614 ± 0.002	0.853 ± 0.001	0.851 ± 0.001	0.850 ± 0.001
$EX4 : T = 5$	0.461 ± 0.002	0.458 ± 0.002	0.673 ± 0.001	0.670 ± 0.001	0.669 ± 0.001

the randomized variant compared to the original CDN from the main paper, but perform equally as well (and no worse) than the traditional approaches, while offering examples where CES are still superior.

D Additional Scenarios

We include one additional, large CDN used to demonstrate how CCOAs can change dramatically depending on available time; see structure in Fig. 5.

```

1 # mega model
2 def ex5_util(states):
3     return {(0, 0): 0, (0, 1): 1, (1, 0): 1, (1, 1): 0}
4     [states['E'], states['F']]
5 ex5_model = CDN(
6     nodes=['A', 'B', 'C', 'D', 'E', 'F'],
7     edges=[('A', 'B'), ('A', 'D'), ('A', 'C'), ('A', 'F'),
8            ('B', 'E'), ('D', 'E'), ('C', 'F')],
9     cpds=[
10         TabularCPD('A', 2, [[0.4], [0.6]]),
11         TabularCPD('B', 2, [[0.2, 0.6],
12                             [0.8, 0.4]],
13                  evidence=['A'], evidence_card
14                  =2),
15         TabularCPD('C', 2, [[0.3, 0.7],
16                             [0.7, 0.3]],
17                  evidence=['A'], evidence_card
18                  =2),
19         TabularCPD('D', 2, [[0.5, 0.6],
20                             [0.5, 0.4]],
21                  evidence=['A'], evidence_card
22                  =2),
23         TabularCPD('E', 2, [[0.2, 0.6, 0.5, 0.4],
24                             [0.8, 0.4, 0.5, 0.6]],

```

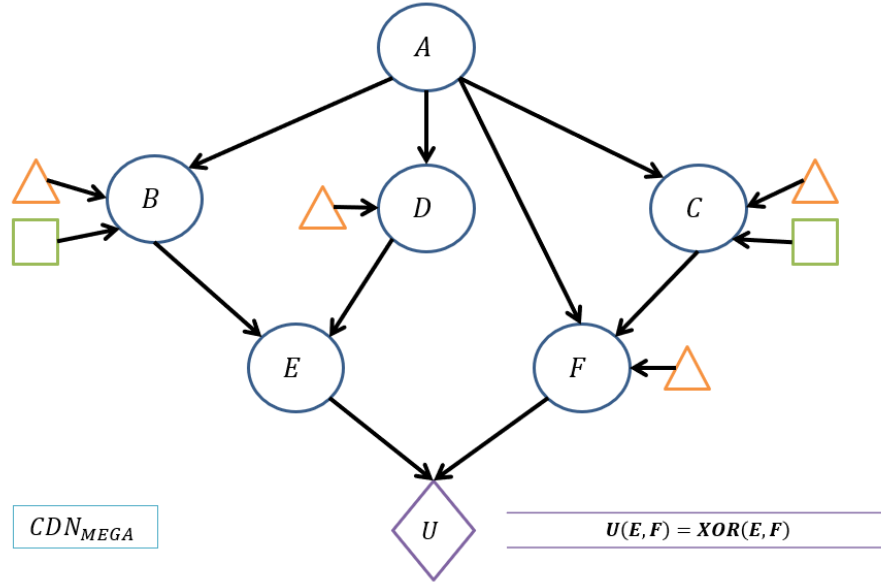


Fig. 5. CDN with many decision choices whose parameters can be found in the simulation source, repeated below.

```

22         evidence=['B', 'D'],
23         evidence_card=[2, 2]),
24     TabularCPD('F', 2, [[0.7, 0.4, 0.6, 0.5],
25                          [0.3, 0.6, 0.4, 0.5]],
26               evidence=['A', 'C'],
27               evidence_card=[2, 2]),
28 ],
29 actions=[
30     {"node": 'B', "action_type": 'do', "value": 0, "
31     time_cost": 2},
32     {"node": 'B', "action_type": 'do', "value": 1, "
33     time_cost": 2},
34     {"node": 'B', "action_type": 'do_none', "time_cost":
35     0},
36     {"node": 'B', "action_type": 'inv', "time_cost": 1},
37     {"node": 'B', "action_type": 'inv_none', "time_cost":
38     0},
39     {"node": 'C', "action_type": 'do', "value": 0, "
40     time_cost": 2},
41     {"node": 'C', "action_type": 'do', "value": 1, "
42     time_cost": 2},
43     {"node": 'C', "action_type": 'inv', "time_cost": 1},

```

```

38     {"node": 'C', "action_type": 'inv_none', "time_cost":
    0},
39     {"node": 'D', "action_type": 'inv', "time_cost": 1},
40     {"node": 'D', "action_type": 'inv_none', "time_cost":
    0},
41     {"node": 'F', "action_type": 'inv', "time_cost": 1},
42     {"node": 'F', "action_type": 'inv_none', "time_cost":
    0},
43 ],
44 util_func=ex5_util,
45 util_nodes=['E', 'F'],
46 )

```

Listing 1.7. CDN Definition of Mega Model.

Examining some interesting CCOAs for this model across different time thresholds, we recognize the necessity for recommender systems to help human decision-makers navigate C2D2s, as even with a small number of decisions, the optimal path can become a winding road of conditions that a human decider would be hard-pressed to faithfully follow in stressful, time-limited scenarios.

```

1  ===== t3 =====
2  ['inv(F)',
3   {'(F', 0): ['do(B=0)'], ('F', 1): ['do_none(B)', {}]}]
4
5  ===== t5 =====
6  ['inv(F)',
7   {'(F', 0): ['inv(C)',
8   {'(C', 0): ['inv(D)',
9   {'(D', 0): ['do(B=0)'], ('D', 1): ['do(B=1)']}]}],
10  {'(C', 1): ['inv(D)',
11  {'(D', 0): ['do(B=0)'], ('D', 1): ['do(B=1)']
12  ]}]},
13  {'(F', 1): ['inv(D)',
14  {'(D', 0): ['do_none(B)', {}], ('D', 1): ['inv(C)',
15  {'(C', 0): ['do(B=0)'], ('C', 1): ['do(B=0)']
16  ]}]}}}]]]
17
18  ===== t9 =====
19  ['inv(D)',
20  {'(D', 0): ['do(C=0)', 'inv(F)',
21  {'(F', 0): ['do(B=0)', 'do(B=0)', 'do_none(B)', {}],
22  {'(F', 1): ['do(C=0)', 'inv(B)',
23  {'(B', 0): ['do(C=0)'], ('B', 1): ['do(C=0)']
24  ]}]},
25  {'(D', 1): ['do(C=0)', 'inv(F)',
26  {'(F', 0): ['do(C=0)', 'do(B=1)', 'do_none(B)', {}],
27  {'(F', 1): ['do(C=0)', 'do(B=0)', 'do_none(B)',
28  {}]}]}]]]

```

Listing 1.8. CCOAs for the mega model with various time thresholds.