A Rational Distributed Process-level Account of Independence Judgment

Ardavan S. Nobandegani[1,2] **Ioannis N. Psaromiligkos**[1]
{ardavan.salehinobandegani@mail.mcgill.ca, ioannis.psaromiligkos@mcgill.ca}
[1]Department of Electrical & Computer Engineering, McGill University
[2]Department of Psychology, McGill University

Abstract

It is inconceivable how chaotic the world would look to humans, faced with innumerable decisions a day to be made under uncertainty, had they been lacking the capacity to distinguish the relevant from the irrelevant—a capacity which computationally amounts to handling probabilistic independence relations. The highly parallel and distributed computational machinery of the brain suggests that a satisfying process-level account of human independence judgment should also mimic these features. In this work, we present the first rational, *distributed*, message-passing, process-level account of independence judgment, called $\mathcal{D}^*$. Interestingly, $\mathcal{D}^*$ shows a curious, but normatively justified tendency for quick detection of dependencies, whenever they hold. Furthermore, $\mathcal{D}^*$ outperforms all the previously proposed algorithms in the AI literature in terms of worst-case running time, and a salient aspect of it is supported by recent work in neuroscience investigating possible implementations of Bayes nets at the neural level. $\mathcal{D}^*$ exemplifies how the pursuit of cognitive plausibility can lead to the discovery of state-of-the-art algorithms with appealing properties, and its simplicity makes $\mathcal{D}^*$ potentially a good candidate as a teaching tool.

Keywords: Rational process models; Distributed computing; Probabilistic independence judgment; Pearl's d-separation

1 Introduction

Is there any connection between the quality of your last night sleep and the color of the shirt your colleague happened to be wearing at work today? How about Mars' current weather and your mood today? We humans judge innumerable such possible connections a day rather effortlessly, appearing to be quite good at teasing apart pertinent from impertinent factors when making decisions. But how does the mind do that? The famous frame problem (Icard & Goodman, 2015; Nobandegani & Psaromiligkos, 2017), a puzzle in philosophy of mind and epistemology, further highlights this intriguing ability of the mind in distinguish the relevant from the irrelevant, and asks a closely related question: "How do we account for our apparent ability to make decisions on the basis only of what is relevant to an ongoing situation without having explicitly to consider all that is not relevant?" (Stanford Encyclopedia of Philosophy). Computationally, the mind's ability of distinguishing the relevant from irrelevant can be characterized in terms of handling probabilistic (in)dependence relations, with 'dependency' implying the existence of connection or relevance between factors and 'independence' the contrary (Pearl, 1986, 1988, 2000). For example, assuming that the random variable $\mathbf{x}$ encodes the quality of your sleep, and $\mathbf{y}$ the color of the shirt your colleague happened to wear the next day, the nonexistence of any connection between $\mathbf{x}$ and $\mathbf{y}$ (which seems to be a rational judgment) can be formally characterized using the notion of probabilistic independence: $\mathbf{x} \perp\!\!\!\perp \mathbf{y}$ (read $\mathbf{x}$ is independent of $\mathbf{y}$, and, by virtue of symmetry, $\mathbf{y}$ is independent of $\mathbf{x}$).

In this work, we are concerned with developing a plausible, process-level account of human independence judgment. Adopting causal Bayes nets (CBNs) (Pearl, 1988; Gopnik et al., 2004, *inter alia*) as a normative model to represent how the reasoner's internal causal model of the world is structured (i.e., reasoner's mental model), the aforesaid task computationally amounts to checking for independencies in the distribution encoded by a CBN. Interestingly, Pearl (1986) put forth a graph-theoretic notion called d-separation, allowing for reading off probabilistic independence relations from the mere structure of a CBN (Pearl, 1986).[1] Ever since its inception, d-separation has proved fundamental in a variety of domains in artificial intelligence, e.g., probabilistic reasoning (Pearl, 1988), causal reasoning (Pearl, 2000), decision making (Shachter, 1998; Koller & Friedman, 2009), and has played important roles in a broad range of areas, e.g., handling missing data (Mohan & Pearl, 2014), extrapolation across populations (Pearl & Bareinboim, 2014), and deep learning (Goodfellow et al., 2016). In that light, algorithms for implementing d-separation could potentially serve as a rational, process-level model of human independence judgment. But what should such a model look like? The highly parallel and distributed computational machinery of the brain suggests that a satisfying process-level account of human independence judgment should also mimic these features. Sadly enough, all past algorithms for the implementation of d-separation have been *sequential* (aka *serial*), i.e., without any parallelism in computation, and, arguably worse, *centralized*, i.e., their executions are fully coordinated by a supervisory unit, analogous to a homunculus (Geiger et al., 1989; Lauritzen et al., 1990; Shachter, 1998; Koller & Friedman, 2009; Butz et al., 2016). These features strongly call into question their psychological plausibility.

The notion of (conditional) probabilistic independence is a quintessential feature of CBNs, and, interestingly, the realization that probabilistic independence plays a crucial role in human cognition was a key element in the development of the CBN formalism (Pearl, 1986). In Pearl's (1986) words: "Whereas a person may show reluctance to giving a numerical estimate for a conditional probability $P(\mathbf{x}_i|\mathbf{x}_j)$, that person can usually state with ease whether $\mathbf{x}_i$ and $\mathbf{x}_j$ are dependent or independent, namely, whether or not knowing the truth of $\mathbf{x}_j$ will alter the belief in $\mathbf{x}_i$." He then continues: "Likewise,

[1]More accurately, Pearl's (1986) d-separation is equally valid for Bayes nets wherein the edges do not enjoy causal interpretations.

people tend to judge the three-place relationships of conditional dependency (i.e., $\mathbf{x}_i$ influences $\mathbf{x}_j$ given $\mathbf{x}_k$) with clarity, conviction, and consistency. This suggests that the notions of dependence and conditional dependence are more basic to human reasoning than are the numerical values attached to probability judgments." Some psychological literature, however, does not fully embrace the statement "with clarity, conviction, and consistency" as Pearl put it. For example, the experimental work by Rehder (2014) suggests that adults exhibit deviations from the Markov condition (i.e., CBN's independencies entailed by d-separation). In contrast, drawing on the experimental studies of Park and Sloman (2013), Sloman and Lagnado (2015) conclude that people indeed uphold the Markov condition and the reason behind the observed deviations is that, under experimental conditions, people may not solely adhere to the information provided by the experimenter and may bring their own background knowledge into the experiment (see also Rehder & Waldmann, 2017). Specifically, Park and Sloman (2013) found strong support for their contradiction hypothesis followed by the mediating mechanism hypothesis, and finally concluded that people do conform to Markov condition once the causal structure people are using is correctly specified (i.e., people's mental causal models).

In this work, we present the first rational, *distributed*, process-level account of independence judgment, called $\mathcal{D}^*$. More formally, $\mathcal{D}^*$ is the first asynchronous, message-passing, distributed algorithm for implementing d-separation, with substantial parallelism in computation, and without any need for a supervisory unit to coordinate its execution (i.e., no synchrony is assumed in $\mathcal{D}^*$'s execution)—fully in the spirit of the celebrated parallel distributed processing (PDP) research program in brain and cognitive sciences (McClelland, 1989). Similar to the well-known belief propagation inference algorithm (Pearl, 1986, 1988), which has played important roles in the theoretical neuroscience literature (see e.g., Gershman & Beck, 2017; George & Hawkins, 2009; Litvak & Ullman, 2009; Rao, 2004; Lochmann & Deneve, 2011), $\mathcal{D}^*$ is a message-passing algorithm, wherein computation is carried out by propagating messages between computational units. Interestingly, $\mathcal{D}^*$ shows a curious, normatively justified tendency for quick detection of probabilistic dependencies, whenever they hold. Furthermore, $\mathcal{D}^*$ outperforms all the previously proposed algorithms in the AI literature in terms of worst-case running time, and a salient aspect of it is supported by recent work in neuroscience investigating possible implementations of Bayes nets at the neural level (e.g., Gershman & Beck, 2017; Lochmann & Deneve, 2011).

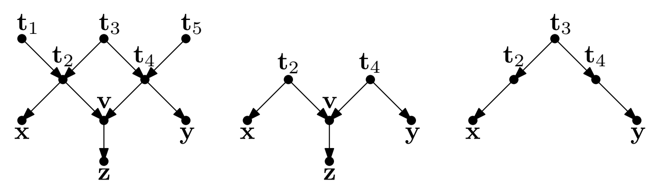
We provide a comprehensive analysis of the computational properties of $\mathcal{D}^*$, along with several refined time-complexity bounds. In the Discussion section, we provide a detailed comparison between $\mathcal{D}^*$ and previously proposed algorithms, and elaborate on the implications of the work presented here for neuroscience and psychology. Formal proofs of the results presented can be found in an extended version of this paper available on arXiv: `https://arxiv.org/abs/1801.10186`.

2 Preliminaries and Notations

Let us introduce the notation adopted in this work. Lower bold-faced letters (e.g., $\mathbf{x}$) denote random variables and upper bold-faced letters (e.g., $\mathbf{X}$) represent sets of random variables. A generic d-separation relation is denoted by $(\mathbf{A} \perp\!\!\!\perp \mathbf{B}|\mathbf{C})_G$ with $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ representing three mutually disjoint sets of variables belonging to the directed acyclic graph (DAG) G, where G represents the topology of the underlying CBN. Read $(\mathbf{A} \perp\!\!\!\perp \mathbf{B}|\mathbf{C})_G$ as follows: $\mathbf{C}$ d-separates $\mathbf{A}$ from $\mathbf{B}$ in DAG G. Similarly, $(\mathbf{A} \not\perp\!\!\!\perp \mathbf{B}|\mathbf{C})_G$ denotes that $\mathbf{C}$ does not d-separate $\mathbf{A}$ from $\mathbf{B}$ in DAG G. For ease of notation, we use $(\mathbf{A} \perp\!\!\!\perp \mathbf{B}|\mathbf{C})_G$ to denote both a d-separation relation (i.e., $\mathbf{C}$ d-separates $\mathbf{A}$ from $\mathbf{B}$ in DAG G) and to denote a d-separation query (i.e., does $\mathbf{C}$ d-separate $\mathbf{A}$ from $\mathbf{B}$ in DAG G?); the distinction should be clear from the context. Let also $G_{An(\mathbf{K})}$ denote the ancestral graph for the variables in set $\mathbf{K}$ belonging to the underlying DAG G (Lauritzen et al., 1990), i.e., the set of nodes for $G_{An(\mathbf{K})}$ comprises the nodes in $\mathbf{K}$ and all the ancestors of the nodes in $\mathbf{K}$ (hence, $G_{An(\mathbf{K})}$ is an induced subgraph of the underlying DAG G).

Informally speaking, throughout that paper, $(\mathbf{A} \perp\!\!\!\perp \mathbf{B}|\mathbf{C})_G$ should be interpreted as follows: "$\mathbf{A}$ and $\mathbf{B}$ are probabilistically independent of each other, given $\mathbf{C}$," and, in the query format, as follows: "Are $\mathbf{A}$ and $\mathbf{B}$ probabilistically independent of each other, given $\mathbf{C}$?" Likewise, $(\mathbf{A} \not\perp\!\!\!\perp \mathbf{B}|\mathbf{C})_G$ should be interpreted as follows: $\mathbf{A}$ and $\mathbf{B}$ are dependent, given $\mathbf{C}$.[2]

Next, a notion called refutation-module is introduced; this will be used later in our formal analysis of $\mathcal{D}^*$.



(a) (b) (c)

Figure 1: Examples for refutation modules. (a) The underlying DAG G is depicted, for which $(\mathbf{x} \not\perp\!\!\!\perp \mathbf{y}|\mathbf{z})_G$. (b,c) Two refutation-modules for the query $(\mathbf{x} \perp\!\!\!\perp \mathbf{y}|\mathbf{z})_G$ are depicted. Note, $\mathbf{z} = \varnothing$ in (c).

Def. 1. (Refutation-Module) Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be three mutually disjoint sets belonging to a DAG G. Let also $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_G$. A connected subgraph of G, $\mathcal{M}_{(\mathbf{X}\not\perp\!\!\!\perp\mathbf{Y}|\mathbf{Z})_G}$, serves as a refutation-module for the query $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_G$, iff $\mathcal{M}_{(\mathbf{X}\not\perp\!\!\!\perp\mathbf{Y}|\mathbf{Z})_G}$ satisfies the following two conditions: (1) $\mathcal{M}_{(\mathbf{X}\not\perp\!\!\!\perp\mathbf{Y}|\mathbf{Z})_G}$ contains an active path P (Pearl, 1986) between a node $\mathbf{x} \in \mathbf{X}$ and a node $\mathbf{y} \in \mathbf{Y}$, and (2) for every head-to-head node $\mathbf{v}$ on P, $\mathcal{M}_{(\mathbf{X}\not\perp\!\!\!\perp\mathbf{Y}|\mathbf{Z})_G}$ contains a directed path between $\mathbf{v}$ and a node $\mathbf{z} \in \mathbf{Z}$. See Fig. 1 for some examples.

Def. 2. (Minimal Refutation-Module) Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be three disjoint sets of nodes belonging to a DAG G. Also, let

[2] Formally, the said interpretations are not fully granted; however, for all purposes of this work, they can be taken to be accurate enough characterizations (see Pearl, 2000, for a complete elaboration on the precise relation between d-separation and conditional independence.)

$(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_G$. Let $\mathcal{M}^*_{(\mathbf{X}\not\perp\!\!\!\perp\mathbf{Y}|\mathbf{Z})_G}$ denote the refutation-module for the d-separation query $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_G$ which possesses the smallest number of edges. We refer to $\mathcal{M}^*_{(\mathbf{X}\not\perp\!\!\!\perp\mathbf{Y}|\mathbf{Z})_G}$ as the *minimal* refutation-module in G for the query $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_G$.

It is easy to prove by construction that the minimal refutation-module $\mathcal{M}^*_{(\mathbf{X}\not\perp\!\!\!\perp\mathbf{Y}|\mathbf{Z})_G}$ need not be unique.

3 The Three-Color Algorithm $\mathcal{D}^*$

In this section, we show how the proposed algorithm $\mathcal{D}^*$ allows us to decide if a generic d-separation query of the form $(\mathbf{A} \perp\!\!\!\perp \mathbf{B}|\mathbf{C})_G$ holds in a DAG G; $\mathcal{D}^*$ is an asynchronous, distributed, message-passing algorithm. More specifically, in $\mathcal{D}^*$, nodes of the underlying DAG G—symbolizing computational units—autonomously engage in communicating messages to their immediate neighbors via the edges of the DAG G—symbolizing communication channels. We assume that communication channels are reliable, bidirectional, and first-in first-out (FIFO) (Lynch, 1996).

The proposed algorithm $\mathcal{D}^*$ is outlined next. Throughout an execution of $\mathcal{D}^*$, variables in $\mathbf{C}$ ignore all messages received from any of their children, and do not send any message to any of their children. The variables in the sets $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ initially activate in the states represented by colors green (●), red (●), and white (○), respectively. Following the prescriptions of the original Belief Propagation algorithm (Pearl, 1986, Sections 1.3 and 2.2.3), we assume that the variables in the sets $\mathbf{A}, \mathbf{B}, \mathbf{C}$ acquire their initial states in a *self-activated* manner. Assuming that a CBN's node can be represented at the neural level by a single (Deneve, 2008b,a) or a population of neurons (Ma et al., 2006), self-activation reflects the content-addressability of the corresponding memory traces. $\mathcal{D}^*$ begins with nodes in $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ sending their colors as messages to their parents. Node $\mathbf{x}$, upon receiving a message, follows two simple steps in the following order:

(i) If $\mathbf{x}$'s current color differs from that of the received message, $\mathbf{x}$ replies by sending back its own color as a message to the transmitter node. If $\mathbf{x}$ is in the state of having no color (denoted by $\varnothing$) prior to the receipt of the message, it does not send back any message to the transmitter node.

(ii) $\mathbf{x}$ updates its color in accord with the following primitive rules, altogether composing the Color Update Grammar (CUG):

$$(\varnothing, \textcolor{green}{\bullet}) \to \textcolor{green}{\bullet}, (\varnothing, \textcolor{red}{\bullet}) \to \textcolor{red}{\bullet}, (\varnothing, \circ) \to \circ,$$
$$(\textcolor{green}{\bullet}, \textcolor{green}{\bullet}) \to \textcolor{green}{\bullet}, (\textcolor{red}{\bullet}, \textcolor{red}{\bullet}) \to \textcolor{red}{\bullet}, (\circ, \circ) \to \circ,$$
$$(\circ, \textcolor{green}{\bullet}) \to \textcolor{green}{\bullet}, (\circ, \textcolor{red}{\bullet}) \to \textcolor{red}{\bullet},$$
$$(\textcolor{green}{\bullet}, \circ) \to \textcolor{green}{\bullet}, (\textcolor{red}{\bullet}, \circ) \to \textcolor{red}{\bullet},$$
$$(\textcolor{green}{\bullet}, \textcolor{red}{\bullet}) \to \text{clash}, (\textcolor{red}{\bullet}, \textcolor{green}{\bullet}) \to \text{clash},$$

where the syntax is: ($\mathbf{x}$'s current color, received message) $\to$ $\mathbf{x}$'s new color. If $\mathbf{x}$'s new color turns out to be different from its old color, with the exception of the transmitter node, $\mathbf{x}$ sends its new color as a message to all its parents, and only those children of $\mathbf{x}$ with which $\mathbf{x}$ has communicated before.

The rules given in the first row of the CUG correspond to white-, green-, and red-colored nodes sending their colors to their yet-uncolored parents. Rules in the second row ensure that the colors of white-, green-, and red-colored nodes persist upon interacting with nodes of the same color. Rules stated in the third row bear on the key understanding that the white color functions as a mere place-holder getting "replaced" by interacting with green-, or red-colored nodes. Rules in the fourth row guarantee the persistence of colors green and red upon interacting with white. Finally, rules given in the last row correspond to the clash event the implication of which is discussed in Remark 1 below.

Remark 1. A clash between colors green (●) and red (●) at a node, any time throughout an execution of $\mathcal{D}^*$, signals the falsity of the input d-separation query, upon which $\mathcal{D}^*$ decides that $(\mathbf{A} \not\perp\!\!\!\perp \mathbf{B}|\mathbf{C})_G$.

Note that the asynchrony of $\mathcal{D}^*$ stems from the fact that there exists no *global clock* for the system and hence any node, upon receiving a message, follows Steps (i) and (ii) *autonomously*, i.e., informally, without having to attend to what computations other nodes in G are performing.

Some of the computational properties of the proposed algorithm $\mathcal{D}^*$ are formally articulated in Proposition 1 below.

Proposition 1. *The following statements hold for $\mathcal{D}^*$.*

(1) For a given d-separation query $(\boldsymbol{A} \perp\!\!\!\perp \boldsymbol{B}|\boldsymbol{C})_G$ and DAG G,

"$\boldsymbol{C}$ does not d-separate $\boldsymbol{A}$ from $\boldsymbol{B}$ in G" $\Longleftrightarrow$
"Clash takes place during $\mathcal{D}^*$'s execution".

(2) $\mathcal{D}^*$'s message-passing is confined within the ancestral graph $G_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$.

(3) During $\mathcal{D}^*$'s execution, either a clash between colors red (●) and green (●) takes place (see Remark 1) upon which $\mathcal{D}^*$ decides that $(\boldsymbol{A} \not\perp\!\!\!\perp \boldsymbol{B}|\boldsymbol{C})$, or a state of equilibrium will be reached in $O(l_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})})$ time where $l_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$ denotes the length of the longest undirected path in the ancestral graph $G_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$.

(4) Message-passing terminates in $O(1)$ time after reaching the state of equilibrium, thereby guaranteeing the termination of $\mathcal{D}^*$.

(5) Message-complexity of $\mathcal{D}^*$ is $O(|E_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}|)$ where $E_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$ is the set of the edges of the ancestral graph $G_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$.

(6) Communication-complexity of $\mathcal{D}^*$ is $O(|E_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}|)$ bits where $E_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$ is the set of the edges of the ancestral graph $G_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$.

3.1 High-Level Understanding of $\mathcal{D}^*$

$\mathcal{D}^*$ has a simple machinery as we informally discuss here. Upon variables in $\mathbf{A} \cup \mathbf{B} \cup \mathbf{C}$ sending their colors to their parents, colors white (○), green (●), and red (●) begin to propagate in a *backwards* manner throughout the network. In the midst of this process, white-color nodes which have a neighboring node colored either red (●) or green (●), change their color to that of their neighbors, and if a clash ever occurs between colors red and green, $\mathcal{D}^*$ decides that the input d-separation query is false (i.e., it is a NO-instance d-separation query). Informally put, white-color nodes function as relays, which, by copying the colors of their neighbors, facilitate the possibility of a (permissible) collision between red and green.

3.2 A Note On The Termination of $\mathcal{D}^*$

According to Proposition 1, if the input d-separation query presented to $\mathcal{D}^*$ is true (i.e., it is a YES-instance d-separation query), the system reaches a state of equilibrium in $O(l_{An(\mathbf{A} \cup \mathbf{B} \cup \mathbf{C})})$ time and message-passing is guaranteed to terminate in $O(1)$ time after that. However, due to its local view, a node cannot know if such a global state has been reached. This is a fairly standard situation for an asynchronous distributed algorithm to find itself in (Mattern, 1987; Tel, 2000), leading to the introduction of the fundamental concept of Termination-Detection (TD) in the distributed systems literature; see Tel (2000, Ch. 8). There exist a variety of TD algorithms in the literature (e.g., Dijkstra et al., 1983; Mattern, 1987; Mittal et al., 2004, 2007). For example, Mittal et al. (2004) proposed two TD algorithms, each having detection latency of $O(D)$ where D is the diameter of the underlying graph G, and G is allowed to have an arbitrary topology.

4 $\mathcal{D}^*$ in Action: A Case Study

In this section, we present an example to illustrate an execution and highlight the simplicity of $\mathcal{D}^*$. Consider the CBN depicted in Fig. 2(a). Let the posed d-separation query be $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_G$ where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2\}$, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2\}$, and $\mathbf{Z} = \{\mathbf{z}\}$. According to the d-separation criterion (Pearl, 1988), observation of $\mathbf{z}$ activates the path $\mathbf{x}_1 \leftarrow \mathbf{t}_1 \leftarrow \mathbf{t}_2 \leftarrow \mathbf{t}_3 \rightarrow \mathbf{t}_4 \leftarrow \mathbf{t}_5 \rightarrow \mathbf{t}_6 \rightarrow \mathbf{t}_7 \rightarrow \mathbf{y}_1$, thereby yielding the falsity of the d-separation query $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_G$ (hence, the input is a NO-instance query); see Fig. 2(a). An execution of $\mathcal{D}^*$ is illustrated using successive *snapshots* shown in Figs. 2(b-f) with each figure depicting the global state of the system (i.e., nodes' colors) at some instance in global time (aka system's *configuration*). As depicted in Fig. 2(b), variables in sets $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$ initially self-activate in the states represented by colors green (●), red (●), and white (○), respectively. Also recall that, as explicated in Sec. 3, variables in $\mathbf{Z}$ ignore any message received from any of their children, and also do not send any message to any of their children—depicting the downlinks of the variables in $\mathbf{Z}$ in a dash-dotted format simply illustrates this statement pictorially in Fig. 2(b). The colors green (●), red (●), and white (○) propagate in a backwards manner (Figs. 2(c-d)). Also, the color of a white node gets replaced by green or red once a neighboring node acquires such colors (Figs. 2(d-f)). Eventually, in the configuration depicted in Fig. 2(f), a clash takes place between colors green and red at a node (circled node in Fig. 2(f)), upon which $\mathcal{D}^*$ decides that $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_G$.

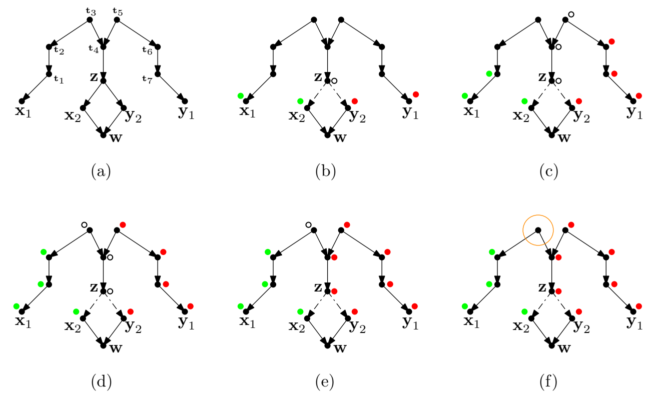


Figure 2: Illustrative example. The underlying DAG G is shown in (a). The initial configuration of the system is portrayed in (b), wherein variables in sets $\mathbf{X}, \mathbf{Y}$, $\mathbf{Z}$ self-activate in the states represented by green (●), red (●), and white (○), respectively. Depicting the downlinks of the variables in $\mathbf{Z}$ in a dash-dotted format simply symbolizes that the variables in $\mathbf{Z}$ ignore any message received from any of their children, and also do not send any message to any of their children. $\mathcal{D}^*$ begins by nodes in $\mathbf{X}, \mathbf{Y}$, $\mathbf{Z}$ sending their colors as messages to their parents and proceeds as shown in (c-f) with each figure depicting a snapshot of the global state of the system at some instance in global time. Eventually, upon occurrence of a clash between colors green and red (at the circled node in (f)), $\mathcal{D}^*$ decides that $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_G$. A better-quality version of this figure can be found on arXiv: `https://arxiv.org/abs/1801.10186`

Notice that, since $\mathbf{w}$ is unobserved (Fig. 2(a)), the path $\mathbf{x}_2 \rightarrow \mathbf{w} \leftarrow \mathbf{y}_2$ indeed remains blocked (Pearl, 2000); this is nicely captured by the machinery of $\mathcal{D}^*$. Algorithm $\mathcal{D}^*$ prevents $\mathbf{x}_2$ and $\mathbf{y}_2$ from sending their colors in the forward direction (i.e., along the edges pointing to $\mathbf{w}$), thereby guaranteeing the occurrence of no clash along the blocked path $\mathbf{x}_2 \rightarrow \mathbf{w} \leftarrow \mathbf{y}_2$. Also notice that, since $\mathbf{z}$ is observed (Fig. 2(a)), the path $\mathbf{x}_2 \leftarrow \mathbf{z} \rightarrow \mathbf{y}_2$ is blocked as well (Pearl, 2000). Once again the machinery of $\mathcal{D}^*$, due to $\mathbf{z}$ refraining from engaging in message-exchange with its children, ensures that no clash takes place due to the blocked path $\mathbf{x}_2 \leftarrow \mathbf{z} \rightarrow \mathbf{y}_2$.

5 Technical Discussion

A number of algorithms for the implementation of d-separation are proposed in the literature (Geiger et al., 1989; Lauritzen et al., 1990; Shachter, 1998; Koller & Friedman, 2009; Butz et al., 2016). Assuming $|E| \geq |V|$, to decide if $(\mathbf{A} \perp\!\!\!\perp \mathbf{B} | \mathbf{C})_G$ holds in G, the worst-case running time of Geiger et al.'s, Koller and Friedman's, Shachter's, and Butz et al.'s is $O(|E|)$ and that of Lauritzen et al.'s algorithm[3] is $O(|V|^2)$ where $|V|$ and $|E|$ denote the number of the nodes and the edges of the underling DAG G, respectively. Note that, since for any DAG G, $|E| \leq |V|^2$, an $O(|E|)$-time algorithm (e.g., Geiger et al.'s) outperforms an $O(|V|^2)$-time algo-

[3]The reader is referred to Geiger et al. (1989) for a detailed analysis of the running-time of Lauritzen et al.'s algorithm.

rithm (e.g., Lauritzen et al.'s) in terms of worst-case runtime[4] (see Geiger et al., 1989, for more discussions on this). According to Proposition 1, the time-complexity of the proposed algorithm $\mathcal{D}^*$ is $O(l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})})$ where $l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})}$ denotes the length of the longest undirected path in the ancestral graph $G_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})}$. Since, for any DAG G, $l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})} \leq |E| \leq |V|^2$, the proposed algorithm $\mathcal{D}^*$ outperforms all the previously proposed algorithms in terms of the worst-case running time.[5] Particularly, the gain is significant in dense DAGs. Note that, in the limit as the underlying DAG G gets denser, the worst-case runtime performances of the previously proposed algorithms become identical, i.e., $O(|V|^2)$.

Another noteworthy property of $\mathcal{D}^*$ is its tendency toward quick detection of false d-separation queries (i.e., NO-instance queries), manifested in an occurrence of a clash according to Remark 1. For a NO-instance d-separation query, Proposition 2, below, gives a more refined upper-bound:

Proposition 2. *Let* $\boldsymbol{A} = \{\boldsymbol{a}_i\}_i$, $\boldsymbol{B} = \{\boldsymbol{b}_j\}_j$, $\boldsymbol{C} = \{\boldsymbol{c}_k\}_k$ *be three disjoint sets of nodes belonging to a DAG G. Let* $l^d_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$ *denote the length of the longest directed path in the ancestral graph* $G_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$*, and* $l^{ij}_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$ *the length of the shortest unblocked path between the nodes* $\boldsymbol{a}_i$ *and* $\boldsymbol{b}_j$ *in* $G_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}$*. As a convention, if all paths between* $\boldsymbol{a}_i$ *and* $\boldsymbol{b}_j$ *are blocked,* $l^{ij}_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})} = \infty$*. If* $(\boldsymbol{A} \not\perp\!\!\!\perp \boldsymbol{B}|\boldsymbol{C})_G$ *then a clash between colors green (•) and red (•) occurs in time* $O\big(l^d_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})} + \min_{i,j} l^{ij}_{An(\boldsymbol{A}\cup\boldsymbol{B}\cup\boldsymbol{C})}\big)$*, upon which* $\mathcal{D}^*$ *decides that* $(\boldsymbol{A} \not\perp\!\!\!\perp \boldsymbol{B}|\boldsymbol{C})_G$*.*

In Sec. 2, we formally defined a notion called refutation-module (see Def. 1). In the language of computational complexity and theorem-proving, a refutation-module $\mathcal{M}_{(\mathbf{X}\not\perp\mathbf{Y}|\mathbf{Z})_G}$ can serve as a *certificate* (or *witness*) for disproving a d-separation query $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_G$. This interpretation is related to the verifier-based definition of the complexity class *coNP*. Next, in Proposition 3, we provide an even more refined upper-bound on the time required for an occurrence of a clash, thereby strengthening our claim as to $\mathcal{D}^*$'s tendency toward quick detection of false d-separation queries.

Proposition 3. *Let* $\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}$ *be three disjoint sets of nodes belonging to a DAG G. Also, let* $(\boldsymbol{X} \not\perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z})_G$*. Let* $\mathcal{M}_{(\boldsymbol{X}\not\perp\boldsymbol{Y}|\boldsymbol{Z})_G}$ *denote a refutation-module for the query* $(\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z})_G$ *with* $l^d_{\mathcal{M}}$ *and* $|P_{\mathcal{M}}|$ *denoting the length of the longest directed path and the shortest unblocked path in* $\mathcal{M}_{(\boldsymbol{X}\not\perp\boldsymbol{Y}|\boldsymbol{Z})_G}$*, respectively. Finally, let* $\mathcal{M}^*_{(\boldsymbol{X}\not\perp\boldsymbol{Y}|\boldsymbol{Z})_G}$ *denote the minimal refutation-module for the query* $(\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z})_G$*, with* $E_{\mathcal{M}^*_{(\boldsymbol{X}\not\perp\boldsymbol{Y}|\boldsymbol{Z})_G}}$ *denoting the set of the edges of* $\mathcal{M}^*_{(\boldsymbol{X}\not\perp\boldsymbol{Y}|\boldsymbol{Z})_G}$*. Then the following statement holds true: A clash between colors green (•) and red (•) occurs in time* $O(\min_{\mathcal{M}_{(\boldsymbol{X}\not\perp\boldsymbol{Y}|\boldsymbol{Z})_G}}\{l^d_{\mathcal{M}} + |P_{\mathcal{M}}|\}) \leq O(|E_{\mathcal{M}^*_{(\boldsymbol{X}\not\perp\boldsymbol{Y}|\boldsymbol{Z})_G}}|)$*, upon which* $\mathcal{D}^*$ *decides that* $(\boldsymbol{X} \not\perp\!\!\!\perp \boldsymbol{Y}|\boldsymbol{Z})_G$*.*

Finally, we would like to point out an interesting property of the CUG, referred to as *order-invariance*, which is characterized informally as follows: *The order according to which nodes in the network receive their messages is irrelevant.*

6 General Discussion

The Algorithm $\mathcal{D}^*$, in the spirit of Pearl's (1986) belief propagation scheme, employs the edges of the underlying CBN as the medium through which message-passing between nodes takes place. The latter echos Pearl's (1986) insight when he advocated the idea that a CBN must *not* be viewed as "merely a passive parsimonious code for storing factual knowledge but also a computational architecture for reasoning about that knowledge." $\mathcal{D}^*$ adheres to this idea. Recent literature in neuroscience investigating possible implementation of CBNs at the neural level supports Pearl's idea (see Lochmann & Deneve, 2011; Gershman & Beck, 2017). Lochmann and Deneve (2011) advocate the idea that a CBN's node can be represented at the neural level by a single (Deneve, 2008a,b) or a population of neurons (Ma et al., 2006) with the neural network resembling a "mirror image" of the CBN it implements—though sometimes not a 'perfect' mirror (see Fig. 1 in Lochmann and Deneve, 2011)—and the links of the neural network providing the medium for inference to be carried out, either in the form of belief propagation or sample-based methods like Gibbs sampling.

Interestingly, the peculiar tendency of $\mathcal{D}^*$ toward quick detection of NO-instance d-separation queries is consistent with our pre-theoretical intuition that humans tend to detect possible dependencies between concepts and propositions rather swiftly, once such dependencies do exist. The following question then presents itself: Could this tendency be supported based on any rational grounds? In what follows we provide an argument supporting the rationality of the foregoing tendency. (†) Assuming that the mind incurs a higher rate of loss (defined as incurred cost per unit of time) for discovering a dependency when one does exist, compared to the condition wherein one does not exist and the mind recognizes that, we formally show that the foregoing tendency is simply a consequence of the mind acting as a boundedly-rational satisficer (Simon, 1957), trying to attain good performance in terms of expected accumulative cost. But why should the rate of loss under the condition wherein a dependency does exist be higher? Informally put, why should the mind be so hasty in detecting dependencies under that condition? One possible explanation is that it is crucial for the mind to swiftly detect dependencies under that condition, with the rationale being that delay in detecting those dependencies could be harmful

[4] The gain is particularly significant in sparse graphs.

[5] According to Proposition 1, a NO-instance d-separation query can be decided by $\mathcal{D}^*$ in time $O(l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})})$. The upper-bound $O(l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})})$ is an improvement over the worst-case runtime of all the previously proposed algorithms. Also note that, adopting a TD-algorithm with detection latency of $O(D)$ (see Mittal et al., 2004, 2007, for such TD-algorithms), a YES-instance d-separation query can be decided by $\mathcal{D}^*$ in time $O(l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})} + D)$ where D is the diameter of G. Once again, since $l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})} \leq |E|, D \leq |E|, |E| \leq |V|^2$, the upper-bound $O(l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})} + D)$ is an improvement over the worst-case runtime of all the previously proposed algorithms. (Notice that, for any DAG G, $\frac{1}{2}(l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})} + D) \leq |E|$, hence follows $|E| = \Omega(l_{An(\mathbf{A}\cup\mathbf{B}\cup\mathbf{C})} + D)$.)

to the reasoner and potentially jeopardize their life, hence important from an evolutionary standpoint. Furthermore, given the prominent role that explanation and inference play in human cognition (see Lombrozo, 2016), it is crucial for the mind to promptly detect those factors deemed relevant to the task faced by the reasoner.

Let us formally characterize a general condition under which the aforesaid tendency can be given a rational basis. Let $\mathbf{C}_{\mathcal{A}}$ denote the accumulative cost of an algorithm $\mathcal{A}$ implementing d-separation criterion, π_{YES} and π_{NO} denote the prior probability of the input being a YES-instance and NO-instance d-separation query, respectively. Let also $\mathbf{T}_{\mathcal{A}}^{\text{YES}}$ and $\mathbf{T}_{\mathcal{A}}^{\text{NO}}$ denote the worst-case runtime of $\mathcal{A}$ on YES-instance and NO-instance d-separation queries, respectively. Finally, let $\mathcal{L}_{\text{YES}}, \mathcal{L}_{\text{NO}} \in \mathbb{R}^{>0}$ denote the cost per unit of time incurred by $\mathcal{A}$ for delay in detecting a YES-instance and NO-instance d-separation query, respectively. Then, for any DAG G, the following holds true: $\mathbb{E}[\mathbf{C}_{\mathcal{A}}] \leq \mathcal{L}_{\text{YES}} \mathbf{T}_{\mathcal{A}}^{\text{YES}} \pi_{\text{YES}} + \mathcal{L}_{\text{NO}} \mathbf{T}_{\mathcal{A}}^{\text{NO}} \pi_{\text{NO}}$, where the expectation $\mathbb{E}[\cdot]$ is taken with respect to the (unknown) distribution of all d-separation queries. It is then easy to show that, under the condition $(*)$ $\mathcal{L}_{\text{NO}} \pi_{\text{NO}} \geq \mathcal{L}_{\text{YES}} \pi_{\text{YES}}$, it is rational for the mind trying to attain good performance in terms of expected accumulative cost to demonstrate the said tendency toward quick detection of NO-instance d-separation queries. The setting portrayed in $(\dagger)$ above is a special case of Condition $(*)$: It corresponds to Condition $(*)$ subject to the assumptions $\pi_{\text{NO}} = \pi_{\text{YES}}$ (reflecting the reasoner's uninformative, *a priori* expectation that YES- and NO-instance queries are equiprobable) and $\mathcal{L}_{\text{NO}} \geq \mathcal{L}_{\text{YES}}$ (reflecting a higher rate of loss for erring on NO-instance queries, as alluded to earlier). Future work should experimentally investigate if humans demonstrate the forgoing normatively justified tendency in probabilistic (in)dependence judgment tasks, or that, on the contrary, they systematically deviate from that.

Also interestingly, the forgoing tendency of $\mathcal{D}^*$ toward focusing its search on the minimal refutation module can be taken as evidence for its least-effort-like characteristic, and is fully consistent with recently proposed frameworks which seek rational understanding of the mind at the algorithmic level of analysis by appealing to the notion of economical use of limited computational and cognitive resources (in our case, by striving for minimizing the size of the module required to be investigated for refuting a false d-separation query); see Nobandegani (2017) and Griffiths et al. (2015). Although we briefly discussed the idea of termination detection for asynchronous distributed algorithms, a boundedly-rational agent may decide to only run an asynchronous distributed algorithm for a period of time which is justified based on the opportunity cost incurred by delaying another task. In that light, the boundedly-rational agent may plausibly decide to adopt termination detection algorithms only in settings wherein the opportunity costs involved would be relatively low. Also notably, $\mathcal{D}^*$ exemplifies how the pursuit of cognitive plausibility can lead to the discovery of state-of-the-art algorithms.

Perhaps the biggest limitation of $\mathcal{D}^*$ (and, likewise, of belief propagation) is the assumption that communication channels are faultless, allowing for reliable message exchange. The brain's neural circuits involve much stochasticity and response variability (e.g., Ma & Jazayeri, 2014; Ma, Beck, and Pouget, 2008; Summerfield & Tsetsos, 2015), undermining this assumption. Future work should investigate extensions of $\mathcal{D}^*$ that are more robust to neural noise. While many questions remain open, we hope to have made some progress toward understanding human probabilistic (in)dependence judgment at the algorithmic level, a capacity without which the world would seem too chaotic for humans to live by.

Acknowledgments: We would like to thank Michael Pacer for providing constructive comments on an earlier draft of this work, and, Tom Shultz and Luc Devroye for helpful discussions. This work was supported in part by NSERC under grant RGPIN 262017.

References

Butz, C. J., Dos Santos, A. E., & Oliveira, J. S. (2016). Relevant path separation: A faster method for testing independencies in bayesian networks. *Proceedings of the Eighth International Conference on Probabilistic Graphical Models (PGM)*, 74–85.

Deneve, S. (2008a). Bayesian spiking neurons II: Learning. *Neural Computation*, *20*(1), 118–145.

Deneve, S. (2008b). Bayesian spiking neurons I: Inference. *Neural Computation*, *20*(1), 91–117.

Dijkstra, E. W., Feijen, W. H., & Van Gasteren, A. M. (1983). Derivation of a termination detection algorithm for distributed computations. *Information Processing Letters*, *16*(5), 217–219.

Geiger, D., Verma, T., & Pearl, J. (1989). d-separation: from theorems to algorithms. *Fifth Workshop on Uncertainty in Artificial Intelligence*, pp. 118–125.

George, D., & Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLoS Computational Biology*, *5*(10), e1000532.

Gershman, S. J., & Beck, J. M. (2017). Complex probabilistic inference: From cognition to neural computation. In *Computational Models of Brain and Behavior,* ed A. Moustafa.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Griffiths, T. L., Lieder, F., & Goodman, N. D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in Cognitive Science*, *7*(2), 217–229.

Icard, T. F., & Goodman, N. D. (2015). A resource-rational approach to the causal frame problem. *Proc. of the 37th Annual Meeting of the Cognitive Science Society*.

Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT.

Lauritzen, S. L., Dawid, A. P., Larsen, B. N., & Leimer, H. G. (1990). Independence properties of directed markov fields. *Networks*, *20*(5), 491–505.

Litvak, S., & Ullman, S. (2009). Cortical circuitry implementing graphical models. *Neural Computation*, *21*(11), 3010–3056.

Lochmann, T., & Deneve, S. (2011). Neural processing as causal inference. *Current Opinion in Neurobiology*, *21*(5), 774–781.

Lombrozo, T. (2016). Explanatory preferences shape learning and inference. *Trends in Cognitive Sciences*, *20*(10), 748–759.

Lynch, N. A. (1996). *Distributed Algorithms*. Morgan Kaufmann.

Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nature Neuroscience*, *9*(11), 1432–1438.

Ma, W. J., Beck, J. M., & Pouget, A. (2008). Spiking networks for bayesian inference and choice. *Current Opinion in Neurobiology*, *18*(2), 217–222.

Ma, W. J., & Jazayeri, M. (2014). Neural coding of uncertainty and probability. *Annual Review of Neuroscience*, *37*, 205–220.

Mattern, F. (1987). Algorithms for distributed termination detection. *Distributed Computing*, *2*(3), 161–175.

McClelland, J. L. (1989). Parallel distributed processing: Implications for cognition and development. In: Morris mgm, editor. parallel distributed processing, implications for psychology and neurobiology. , Oxford: Clarendon Press; pp. 8-45.

Mittal, N., Venkatesan, S., & Peri, S. (2004). Message-optimal and latency-optimal termination detection algorithms for arbitrary topologies. In *International symposium on distributed computing (disc)* (pp. 290–304).

Mittal, N., Venkatesan, S., & Peri, S. (2007). A family of optimal termination detection algorithms. *Distributed Computing*, *20*(2), 141–162.

Mohan, K., & Pearl, J. (2014). On the testability of models with missing data. In *AISTATS*.

Nobandegani, A. S. (2017). The Minimalist Mind: On Mininality in Learning, Reasoning, Action, & Imagination. McGill University, PhD Dissertation.

Nobandegani, A. S., & Psaromiligkos, I. N. (2017). The causal frame problem: An algorithmic perspective. *Proc. of the 39th Annual Meeting of the Cognitive Science Society*.

Park, J., & Sloman, S. A. (2013). Mechanistic beliefs determine adherence to the markov property in causal reasoning. *Cognitive Psychology*, *67*(4), 186–216.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, *29*(3), 241–288.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*.

Pearl, J. (2000). *Causality*. Cambridge university press.

Pearl, J., & Bareinboim, E. (2014). External validity: From do-calculus to transportability across populations. *Statistical Science*, *29*(4), 579–595.

Rao, R. P. (2004). Bayesian computation in recurrent neural circuits. *Neural Computation*, *16*(1), 1–38.

Rehder, B. (2014). Independence and dependence in human causal reasoning. *Cognitive Psychology*, *72*, 54–107.

Rehder, B., & Waldmann, M. R. (2017). Failures of explaining away and screening off in described versus experienced causal learning scenarios. *Memory & Cognition*, *45*, 245-260.

Shachter, R. D. (1998). Bayes-ball: Rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)* (pp. 480–487).

Simon, H. A. (1957). *Models of Man*. Wiley.

Sloman, S. A., & Lagnado, D. (2015). Causality in thought. *Annual Review of Psychology*, *66*, 223–247.

Summerfield, C., & Tsetsos, K. (2015). Do humans make good decisions? *Trends in Cognitive Sciences*, *19*(1), 27–34.

Tel, G. (2000). Introduction to Distributed Algorithms. Cambridge University Press.