Strongly Infinite-Step Opaque Boolean Networks

Spyros Reveliotis

School of Industrial & Systems Engineering
Georgia Institute of Technology, USA
(email: sr81@gatech.edu)

Abstract: Motivated by the increasing interest of the Discrete Event Systems (DES) community in the theory of Boolean networks (BNs), this work undertakes a prototypical investigation of the notion of opacity in the BN modeling framework. More specifically, we (i) adapt a particular version of this concept to the BN semantics and dynamics; (ii) provide an algorithm that assesses the resulting opacity concept while taking advantage of the special structure of the BN state spaces; and (iii) develop a methodology for enforcing the sought opacity, in some *optimal sense*, on BNs that do not possess this property. The closing part of the paper briefly discusses how this prototypical study can be extended to other versions of opacity studied by the DES community.

Keywords: Boolean networks; opacity verification and enforcement; Discrete Event Systems; optimal supervisory control theory

1. INTRODUCTION

Boolean networks (BNs) were introduced by Kauffman (Kauffman (1969)) in order to model, analyze and simulate the dynamics of the cellular networks studied by systems biology. This model provides a discrete-time, state-space-based representation of the traced dynamics in a spirit similar to the state-space approaches of the classical control theory; but all the involved variables are Boolean variables and the recursive functions that iterate these variables are Boolean functions. Also, more recently, Boolean networks have attracted a broader interest as a formal modeling framework, and there has been an extensive effort to develop a general theory for the analysis and the control of these networks that parallels the analysis and the control of more classical dynamical systems. A nice introduction to these developments is provided in Cheng et al. (2011).

At the same time, the discrete (binary) nature of the variables employed by the BN model places it in the class of *Finite State Automata (FSA)* (Cassandras and Lafortune (2021)). Therefore, the BNs can also be analyzed in terms of concepts and methods that have been developed by computer science and the theory of Discrete Event Systems (DES) (Cassandras and Lafortune (2021); Hadjicostis (2020)). This possibility has been recognized by, both, the BN researchers (e.g., Kauffman (1969); Cheng et al. (2011); Gao et al. (2018)) and the DES community (e.g., Cury and Baldissera (2012)). Furthermore, the BNs possess significant special structure, both, in their semantics and in the dynamics of the induced FSAs, that can enable extensive customization of the general DES theory to this particular setting. More specifically, the transitional dynamics of an *autonomous BN* place the underlying FSA to the class of *deterministic FSAs with a single transition available at every state*. In the following, we shall refer to this FSA class as *1-DFSAs*. An 1-DFSA has a unique trajectory when started from any given initial state, and this fact partitions the underlying state space into a number of noncommunicating subspaces that are defined by the *"attractors"* (i.e., *absorbing states* or *cycles*) that are present in the corresponding *state transition diagram (STD)*. The resulting topological structure can simplify very substantially the specification and the analysis of many properties that have been investigated for DES represented by more general FSAs.

Motivated by the above remarks, this paper undertakes an investigation of the notion of *opacity* in the context of autonomous BNs. Opacity is a concept that has drawn extensive attention within the DES community, and it implies the preservation of a *"secret"* from an external *"intruder"* that has complete knowledge of the system structure but only partial observation of the generated dynamics. It also comes in many variations, that include (i) language-based opacity, (ii) current-state opacity, (iii) initial-state opacity, (iv) weak K-step and infinite-step opacity, and (v) strong K-step and infinite-step opacity. The reader is referred to Jacob et al. (2016); Lafortune et al. (2018) and the monograph of Hadjicostis (2020) for some systematic introductions to all these concepts and their development within DES.

The current work seeks to adapt the notion of *strong infinite-step opacity* to the dynamics of autonomous BNs and their observational mechanisms. In more specific terms, we (i) adapt the original definition of this concept to the BN semantics and the standard analytical model that represents the dynamics of these systems; (ii) develop customized algorithms for assessing the strong infinite-step opacity of any given BN that take advantage of the special structure of the underlying state space described in the earlier part of this section; and also (iii) develop a methodology for enforcing strong infinite-step opacity, in some *optimal sense*, when the original BN does not

possess this property. All these results are meant to have a demonstrative, prototypical role and value, and, hence, in the last part of the paper we briefly discuss how they can be adapted and extended to address some other notions of opacity in the BN modeling framework.

The rest of the paper is organized as follows: The next section provides some background material from the FSA and the BN modeling frameworks that is necessary for the main developments of this work. It also introduces some new concepts and results that are necessary for these developments. Section 3 adapts the notion of strong infinite-step opacity to the context of the BN modeling framework, and develops a novel, customized algorithm for the verification of this concept for any given BN. Section 4 considers the problem of enforcing optimally strong infinite-step opacity on any given BN that does not possess this property. Finally, Section 5 concludes the paper and suggests some directions for future work.

2. PRELIMINARIES

Finite State Automata: For the needs of this work, we define a *finite state automaton (FSA)* G by a quintuple $\langle S, E, \delta, S_0, S_m\rangle$ where: (i) $S = \{s_1, s_2, \ldots, s_n\}$ is the set of *states* of G. (ii) $E = \{e_1, e_2, \ldots, e_m\}$ is the set of *events* taking place in G. (iii) $\delta : S \times E \to 2^S$ is the *state transition function* of G. For any pair $(s_i, e_j) \in S \times E$, $\delta(s_i, e_j) \neq \emptyset$ is the set of possible states that can result from the execution of event e_j at state s_i. On the other hand, $\delta(s_i, e_j) = \emptyset$ implies that event e_j is not feasible at state s_i. (iv) $S_0 \subseteq S$ is the set of the possible *initial states* of G. Finally, (v) $S_m \subseteq S$ is the set of the *marked states* of G.

The transitional dynamics of FSA G can be represented graphically by a labelled directed graph (or digraph), $\mathcal{G} = (V, \mathcal{E}, \mathcal{L})$, that is known as the *state transition diagram (STD)* of G. The nodes $v \in V$ of $\mathcal{G}$ are in one-to-one correspondence with the states $s_i \in S$ of G. To emphasize this correspondence, in the following we shall set $V = S$. The edge set $\mathcal{E}$ of $\mathcal{G}$ is a subset of $S \times S$, with the pair (s_i, s_j) being in $\mathcal{E}$ if there exists an event $e_k \in E$ such that $s_j \in \delta(s_i, e_k)$. The label function $\mathcal{L}$ is defined on the edge set $\mathcal{E}$, and for every $(s_i, s_j) \in \mathcal{E}$, $\mathcal{L}(s_i, s_j) = \{e_k \in E : s_j \in \delta(s_i, e_k)\}$.

The state transition function of FSA G is extended to $(2^S \setminus \{\emptyset\}) \times E^*$ (i.e., to nonempty subsets $\hat{S}$ of state set S and strings q of the event set E) through the following recursion:[1] (i) $\forall s_i \in S,\ \delta(s_i, \epsilon) = s_i$; (ii) $\forall s_i \in S,\ \forall q = e_{[1]}e_{[2]}\ldots e_{[l]},\ \delta(s_i, q) = \bigcup_{s_k \in \delta(s_i, e_{[1]}e_{[2]}\ldots e_{[l-1]})} \delta(s_k, e_{[l]})$; (iii) $\forall \hat{S} \in 2^S \setminus \{\emptyset\},\ \forall q \in E^*,\ \delta(\hat{S}, q) = \bigcup_{s_i \in \hat{S}} \delta(s_i, q)$.

We shall refer to the set of strings $L(G) = \{q \in E^* : \delta(S_0, q) \neq \emptyset\}$ as the *language generated by* G. Also, $L_m(G) = \{q \in L(G) : \delta(S_0, q) \cap S_m \neq \emptyset\}$ is the *language recognized by* G.

For an event string $q \in L(G)$, with $q = e_{[1]}e_{[2]}\ldots e_{[l]}$, a *trace* – or *run* – of q in G is a state sequence $\langle s_{[0]}, s_{[1]}, \ldots, s_{[l]}\rangle$ such that (i) $s_{[0]} \in S_0$ and (ii) $\forall i = 1, \ldots, l,\ s_{[i]} \in \delta(s_{[i-1]}, e_{[i]})$. The set of traces of an event string $q \in L(G)$ in G is denoted by $tr(q; G)$.

For any FSA $G = \langle S, E, f, S_0, S_m\rangle$, we define the set of its *accessible* – or *reachable* –*states*, $ac(S)$, by $ac(S) = \{s_i \in S : \exists q \in E^*\ s.t.\ s_i \in \delta(S_0, q)\}$. Also, the FSA $ac(G) = \langle ac(S), E, ac(\delta), S_0, ac(S_m)\rangle$, where $ac(S_m) = S_m \cap ac(S)$ and $ac(\delta)$ is the restriction of f in $ac(S) \times E$, is the *accessible part* of G. These definitions imply that

$$L(ac(G)) = L(G) \quad \wedge \quad L_m(ac(G)) = L_m(G) \tag{1}$$

An FSA G is *complete* if $\forall (s_i, e_j) \in S \times E,\ \delta(s_i, e_j) \neq \emptyset$. Also, G is *deterministic (DFSA)* if (i) $\forall (s_i, e_j) \in S \times E,\ |\delta(s_i, e_j)| \leq 1$, and (ii) $|S_0| = 1$. Otherwise it is *nondeterministic (NFSA)*. The state transition function δ of a complete DFSA can be alternatively defined as a function $\delta : S \times E \to S$.

Boolean Networks: For the needs of this work, an *(autonomous) Boolean network (BN)* $\mathcal{B}$ is a discrete-time dynamical system where:

(1) the *state* x is a column vector $x = [x_1, x_2, \ldots, x_n]^T$ such that

$$\forall\ i = 1, \ldots, n,\quad x_i \in \mathbb{B} = \{0, 1\}. \tag{2}$$

(2) The dynamics of $\mathcal{B}$ are defined by
   (a) an *initial state* $x(0) \in \mathbb{B}^n$, and
   (b) the following recursion:

$$\forall t = 0, 1, \ldots,\ \forall i \in \{1, \ldots, n\},\ x_i(t+1) = f_i(x(t)) \tag{3}$$

   where $\forall i,\ f_i(\cdot) : \mathbb{B}^n \to \mathbb{B}$ (i.e., $f_i(\cdot)$ is an n-variate Boolean function).

(3) The *output* of $\mathcal{B}$ at any given state x is an m-dimensional binary column vector $o(x) = [o_1(x), o_2(x), \ldots, o_m(x)]^T$ where

$$\forall j \in \{1, \ldots, m\},\quad o_j(x) = g_j(x)\ \wedge\ g_j(\cdot) : \mathbb{B}^n \to \mathbb{B} \tag{4}$$

We also set: $f(x) = [f_1(x), f_2(x), \ldots, f_n(x)]^T = f^1(x)$; $\forall k \geq 2$, $f^k(x) = f(f^{k-1}(x))$; and $g(x) = [g_1(x), g_2(x), \ldots, g_m(x)]^T$. Furthermore, for a given BN $\mathcal{B}$ and any $x \in \mathbb{B}^n$, $\mathcal{B}(x)$ denotes the BN induced by $\mathcal{B}$ by setting its initial state $x(0) = x$.

FSA-based representation of the BN dynamics: The BN $\mathcal{B}$ defined in the previous subsection induces the FSA $G(\mathcal{B}) = \langle S(\mathcal{B}), E(\mathcal{B}), \delta(\mathcal{B}), S_0(\mathcal{B}), S_m(\mathcal{B})\rangle$ where: (i) $S(\mathcal{B}) = \mathbb{B}^n$; (ii) $E(\mathcal{B}) = \{\tau\}$, i.e., a singleton with the unique event τ corresponding to the discrete-time advancement by one time unit; (iii) $\delta(\mathcal{B}) : \mathbb{B}^n \times E(\mathcal{B}) \to \mathbb{B}^n$ with $\delta(\mathcal{B})(x, \tau) = f(x)$; (iv) $S_0(\mathcal{B}) = \{x(0)\}$; and (v) $S_m(\mathcal{B}) = S(\mathcal{B}) = \mathbb{B}^n$. [2]

When combined with Eq. 3, the above definition of the FSA $G(\mathcal{B})$ implies that (i) it is complete; (ii) there is only one event available at each state $s \in S(\mathcal{B})$; and (iii) the corresponding transition is deterministic. We shall indicate these three properties of $G(\mathcal{B})$ by characterizing it as *1-DFSA*.

[1] We remind the reader that for any finite set Q, Q^* is the set containing all the finite-length sequences of elements of Q, including the empty sequence, and it is known as the *Kleene closure* of Q. The elements of Q^* are also known as the *(finite) strings* of Q. The *empty string* is denoted by ϵ and has zero length.

[2] In the rest of this work, the FSA $G(\mathcal{B})$ is used primarily as an alternative *generator* of the dynamics of the BN $\mathcal{B}$, and therefore, the specification of the set $S_m(\mathcal{B})$ is indifferent.

For any $t \geq 0$, let τ^t denote the t-length string $\tau\tau\ldots\tau$. An important implication of the 1-DFSA property of the FSA $G(\mathcal{B})$ is that, for any initialization $\mathcal{B}(x)$ of $\mathcal{B}$ and any period $t \geq 0$, the trace set $tr(\tau^t; G(\mathcal{B}(x))) \equiv tr(t; \mathcal{B}(x))$ is a *singleton*. With a slight abuse of notation, in the following we shall also use the notation $tr(t; \mathcal{B}(x))$ to denote the single element of this set. Furthermore, a trace $tr(t; \mathcal{B}(x)) = \langle x_{[0]}(= x), x_{[1]}, \ldots, x_{[t]}\rangle$ will be represented more compactly by $x_{[0]}(= x)x_{[1]}\ldots x_{[t]}$. Finally, for any state set $X \subseteq \mathbb{B}^n$, $tr(t; \mathcal{B}(X)) = \{tr(t; \mathcal{B}(x)) : x \in X\}$.

The singleton structure of the trace sets $tr(\tau^t; G(\mathcal{B}(x))) \equiv tr(t; \mathcal{B}(x))$, when combined with (i) the perpetuality of the dynamics of $\mathcal{B}N(x)$ in the discrete-time t, and (ii) the finiteness of the state space $S(\mathcal{B}(x))$, also imply that for, $t \geq 2^n$, any trace $tr(t; \mathcal{B}(x)) = x_{[0]}(= x)x_{[1]}\ldots x_{[t]}$ can be decomposed to (i) an initial, possibly empty, *transient segment*, followed by (ii) a *periodic segment*. In the STD of the DFSA $G(\mathcal{B}(x))$, the transient segment is represented by a *directed acyclic path* $\mathcal{P}(x)$ that starts from state x and leads to a *directed cycle* $\mathcal{C}(x)$ that supports the periodic segment of the behavior of $\mathcal{B}(x)$.[3] Also, in the following, $|\mathcal{P}(x)|$ (resp., $|\mathcal{C}(x)|$) will denote the *length* of path $\mathcal{P}(x)$ (resp., $\mathcal{C}(x)$), defined by the number of states in this structure. Finally, the above remarks further imply that for any BN $\mathcal{B}$ and any state $x \in \mathbb{B}^n$, the FSA $ac(G(\mathcal{B}(x)))$ is graphically represented by the subgraph of the STD of $G(\mathcal{B})$ that is induced by the path $\mathcal{P}(x)$ and the cycle $\mathcal{C}(x)$.

A trace $tr(t; \mathcal{B}(x))$ induces the corresponding *observation string* $o(t; \mathcal{B}(x))$ with $o_{[k]}(t; \mathcal{B}(x)) = o(x(k))$, for $k = 0, 1, \ldots, t$; i.e., string $o(t; \mathcal{B}(x))$ traces the output of the BN $\mathcal{B}(x)$ at each period k, from period 0 up to period t.

Definition 1. Two *traces* $tr(t; \mathcal{B}(x_1))$ and $tr(t; \mathcal{B}(x_2))$ are *indistinguishable* if $o(t; \mathcal{B}(x_1)) = o(t; \mathcal{B}(x_2))$. Also, *states* x_1 and x_2 are *indistinguishable* if

$$\forall t = 0, 1, 2, \ldots, \quad o(t; \mathcal{B}(x_1)) = o(t; \mathcal{B}(x_2)) \tag{5}$$

The next proposition provides a succinct test for state indistinguishability.

Proposition 1. Consider states x_1 and x_2 and set

$$T(x_1, x_2) = \max\{|\mathcal{P}(x_1)|, |\mathcal{P}(x_2)|\} + \text{lcm}\{|\mathcal{C}(x_1)|, |\mathcal{C}(x_2)|\} \tag{6}$$

where the function 'lcm$\{\cdot, \cdot\}$' returns the least common multiplier of its arguments. States x_1 and x_2 are *indistinguishable* if and only if

$$o(T(x_1, x_2); \mathcal{B}(x_1)) = o(T(x_1, x_2); \mathcal{B}(x_2)) \tag{7}$$

Proof: The necessity of the condition of Eq. 7 is obvious from the definition of the state indistinguishability. For the sufficiency part, first consider the period $\hat{t}(x_1, x_2) = \max\{|\mathcal{P}(x_1)|, |\mathcal{P}(x_2)|\}$. By this time, both automata $G(\mathcal{B}(x_1))$ and $G(\mathcal{B}(x_2))$ have reached their corresponding cycles $\mathcal{C}(x_1)$ and $\mathcal{C}(x_2)$. Furthermore, at period $T(x_1, x_2)$, each automaton $G(\mathcal{B}(x_i))$, $i = 1, 2$, will be at the same state in its cycle $\mathcal{C}(x_i)$ that it was at period $\hat{t}(x_1, x_2)$. More specifically, after period $\hat{t}(x_1, x_2)$, the state pair (x_1, x_2) evolves periodically and the time interval $\langle\hat{t}(x_1, x_2) + 1, \ldots, T(x_1, x_2)\rangle$ constitutes a cycle of this evolution. Hence, the indistinguishability of states x_1 and x_2 up to $T(x_1, x_2)$ also implies their indistinguishability over any longer observation interval. □

[3] Cycle $\mathcal{C}(x)$ may contain only a single state x, in which case state x is characterized as an *absorbing state* of $G(\mathcal{B})$ or a *fixed point* of $\mathcal{B}$.

3. THE CONSIDERED NOTION OF OPACITY AND ITS VERIFICATION

Consider a BN $\mathcal{B}$ defined as in Section 2 but with its initial state $x(0)$ selected arbitrarily from a subset of $\mathbb{B}^n$ defined by a Boolean function $\zeta(\cdot) : \mathbb{B}^n \to \mathbb{B}$; i.e.,

$$X_0 = \{x \in \mathbb{B}^n : \zeta(x) = 1\} \tag{8}$$

Also, let $\hat{X}$ be another *predicate set* of $\mathbb{B}^n$, defined by a Boolean function $\xi(\cdot) : \mathbb{B}^n \to \mathbb{B}$ through the equation

$$\hat{X} = \{x \in \mathbb{B}^n : \xi(x) = 1\} \tag{9}$$

$\hat{X}$ denotes a special set of states of $\mathcal{B}$ to be referred to as the *secret states*. Also, function $\xi(\cdot)$ is the *secret* of $\mathcal{B}$. BN $\mathcal{B}$ is observed by an agent $\mathcal{A}$ who (i) traces the output $o(t)$, $t = 0, 1, \ldots$, generated by $\mathcal{B}$, and also (ii) knows (a) the Boolean functions f, g, ζ and ξ, that determine the dynamics, the output, the initial-state set, and the secret-state set of $\mathcal{B}$. On the other hand, agent $\mathcal{A}$ does not know the exact initial state $x(0)$ of $\mathcal{B}$.

Definition 2. BN $\mathcal{B}$, with initial state set X_0, is *strongly infinite-step opaque with respect to the secret state set* $\hat{X}$, if for every period $t = 0, 1, 2, \ldots$, and trace $q \in tr(t; \mathcal{B}(X_0))$ that contains a secret state $x \in \hat{X}$, there exists a trace $q' \in tr(t; \mathcal{B}(X_0))$ that does not contain any secret state and it is indistinguishable from q.

Without loss of generality, in the following we assume that $\hat{X} \subseteq ac(S(\mathcal{B}(X_0)))$. Then, for every secret state $\hat{x} \in \hat{X}$, we define the set $X_0(\hat{x}) = \{x \in X_0 : \hat{x} \in ac(S(\mathcal{B}(x)))\}$. We also set $\hat{X}_0 = \bigcup_{\hat{x}\in\hat{X}} X_0(\hat{x})$ and $\tilde{X}_0 = X_0 \setminus \hat{X}_0$. The following theorem provides a necessary and sufficient condition for the strong infinite-step opacity of BN $\mathcal{B}$ with respect to the secret state set $\hat{X}$ when the initial state set is X_0.

Theorem 2. BN $\mathcal{B}$, with initial state set X_0, is *strongly infinite-step opaque with respect to the secret state set* $\hat{X}$ if and only if every state $x \in \hat{X}_0$ is indistinguishable from some state $x' \in \tilde{X}_0$.

Proof: The sufficiency of the condition of Theorem 2 for the strong infinite-step opacity of BN $\mathcal{B}$ is obvious. Next, we prove the necessity of this condition by contraposition. Hence, suppose that there is a secret state $\hat{x}$ for which there exists a state $x \in X_0(\hat{x})$ that is not indistinguishable from any state $x' \in \tilde{X}_0$. Then, Proposition 1 implies that, when starting BN $\mathcal{B}$ from state x, agent $\mathcal{A}$ will be able to infer that the initial state $x(0) \notin \tilde{X}_0$ by period $T(x) = \max_{x'\in\tilde{X}_0}\{T(x, x')\}$. But then, she will also know that a secret state has been visited by period $\hat{T} = \max_{x\in\hat{X}_0}\{|ac(S(\mathcal{B}(x)))|\}$. □

Theorem 2 decomposes the assessment of the strong infinite-step opacity of a BN $\mathcal{B}$ with a set of initial states X_0 and a secret-state set $\hat{X}$, to a set of tests assessing the indistinguishability of each state $x \in \hat{X}_0 = \bigcup_{\hat{x}\in\hat{X}} X_0(\hat{x})$ with the set $\tilde{X}_0 = X_0 \setminus \hat{X}_0$. Algorithm 1 outlines an

Algorithm 1 An algorithm for assessing the strong infinite-step opacity of an autonomous BN $\mathcal{B}$ with initial-state predicate ζ and secret-state predicate ξ.

Inputs:
A BN $\mathcal{B}$, the Boolean function $\zeta(\cdot)$ defining the set X_0 of the possible initial states of $\mathcal{B}$, and the function $\xi(\cdot)$ defining the secret-state set $\hat{X}$ of $\mathcal{B}$.

Outputs:
A decision of whether $\mathcal{B}$ is strongly infinite-step opaque with respect to the secret state set $\hat{X}$.

1) $X_0 := \{x \in \mathbb{B}^n : \zeta(x) = 1\}$.
2) $\forall x \in X_0$, compute the FSA $ac(G(\mathcal{B}(x)))$.
3) $X := \bigcup_{x \in X_0} ac(S(\mathcal{B}(x)))$.
3) $\hat{X} := \{x \in \mathbb{B}^n : \xi(x) = 1\} \cap X$.
4) $\hat{X}_0 := \{x \in X_0 : ac(S(\mathcal{B}(x))) \cap \hat{X} \neq \emptyset\}$.
5) $\tilde{X}_0 := X_0 \setminus \hat{X}_0$.
6) $T := \max_{(x_1,x_2) \in \hat{X}_0 \times \tilde{X}_0} \{T(x_1, x_2)\}$ (c.f. Eq. 6).
7) For each state $x \in \tilde{X}_0$, compute the observation string $o(T; \mathcal{B}(x))$.
8) For each state $x \in \hat{X}_0$, use the results of Step 7 to check whether $\exists x' \in \tilde{X}_0 : o(T; \mathcal{B}(x)) = o(T; \mathcal{B}(x'))$.
9) If $\exists x \in \hat{X}_0$ failing the test of Step 8, return with a negative decision; otherwise, return with a positive decision.

efficient way for contacting this assessment in view of the aforementioned decomposition. [4]

Also, an upper bound for the worst-case time complexity of Algorithm 1 can be obtained as follows: Let $M = |\bigcup_{x \in X_0} ac(S(\mathcal{B}(x)))|$. Then, Step 1 of Algorithm 1 has a time complexity of 2^n, while the worst-case time complexity of Steps 2-5 is $O(M)$. The worst-case time complexity of Step 6 is $O(M^2)$ and so is the value of the variable T that is computed at this step. Step 7 has worst-case time complexity $O(M \cdot T) = O(M^3)$. Finally, Step 8 has worst-case time complexity $O(M^2 \cdot T) = O(M^4)$. Hence, the worst-case time complexity of the entire algorithm is $\min\{2^n, O(M^4)\}$. But $M = O(2^n)$, where n the state dimensionality of $\mathcal{B}$. Therefore, eventually, the worst-case time complexity of Algorithm 1 is $O(2^{4n})$.

Example: We demonstrate the concepts introduced in Section 2 and in this section, and the execution of Algorithm 1, through a small but elucidating example. The considered BN $\mathcal{B}$ has a 4-dim state vector $x = [x_1, x_2, x_3, x_4]^T$ and a 3-dim output vector $o = [o_1, o_2, o_3]^T$. The state-updating function $f : \mathbb{B}^4 \to \mathbb{B}^4$ is defined by

$$\forall x \in \mathbb{B}^4, \quad f(x) = [\neg x_1, \; x_1 \bar{\vee} x_2, \; x_1 \wedge x_2, \; \neg x_4]^T \quad (10)$$

The function $g : \mathbb{B}^4 \to \mathbb{B}^3$, determining the system output, is defined by

$$\forall x \in \mathbb{B}^4, \quad g(x) = [\neg x_1, \; x_2 \vee x_3, \; x_1 \wedge x_4]^T \quad (11)$$

In Eqs 10 and 11, the binary operators '$\wedge$', '$\vee$' and '$\bar{\vee}$' denote, respectively, the AND, OR and XOR Boolean functions, and the unitary operator '$\neg$' denotes negation (or the NOT Boolean function). We also set [5]

$$X_0 = \{[0,0,0,0]^T, \; [0,0,0,1]^T, \; [1,1,1,0]^T, \; [1,1,1,1]^T\} \quad (12)$$

and

$$\hat{X} = \{[0,0,1,0]^T\} \quad (13)$$

The STD of the FSA $ac(G(\mathcal{B}(X_0)))$ is provided in Figure 1. The possible initial states $x \in X_0$ are depicted with bold black boundaries, while the secret state $\hat{x} \in \hat{X}$ is depicted with a bold red boundary. The reader should also notice that the entire STD consists of two disjoint subgraphs, with each subgraph containing a single directed cycle that functions as an attractor for the corresponding initial states. From this decomposition, it is clear that the state sets $\hat{X}_0$ and $\tilde{X}_0$ computed in Steps 4 and 5 of Algorithm 1, are

$$\hat{X}_0 = \{[0, \; 0, \; 0, \; 0]^T, \; [1, \; 1, \; 1, \; 1]^T\} \quad (14)$$

and

$$\tilde{X}_0 = \{[0, \; 0, \; 0, \; 1]^T, \; [1, \; 1, \; 1, \; 0]^T\} \quad (15)$$

Next, we compute the value of the variable T that appears in Step 6 of Algorithm 1. For this, first we notice that both of the directed cycles appearing in Figure 1 include four states. Therefore, the second term in the right-hand-side of Eq 6 is equal to 4. From Figure 1, it is also clear that, for every initial state $x \in X_0$, the first term in the right-hand-side of Eq 6 is equal to 1. Hence, in the considered example, $T = 5$.

The observation strings $o(5; \mathcal{B}(x)), \; x \in \hat{X}_0 \cup \tilde{X}_0$, are provided in Table 1. More specifically, the first two columns of Table 1 list the six-period observation strings for the initial states $x \in \hat{X}_0$ while the last two columns provide the corresponding observation strings for the initial states $x' \in \tilde{X}_0$. The perusal of this table reveals that every state in $x \in \hat{X}_0$ is distinguishable from every state $x' \in \tilde{X}_0$, and therefore, the considered BN is not strongly infinite-step opaque with respect to the secret state $\hat{x} = [0, \; 0, \; 1, \; 0]^T$.

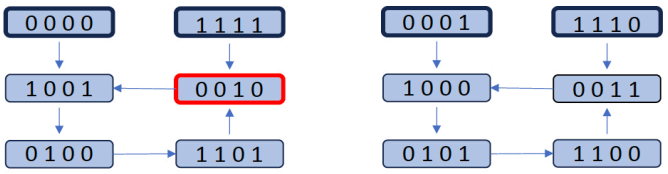

Fig. 1. The STD of the FSA $ac(G(\mathcal{B}(X_0)))$ in the example.

Table 1. The observation strings $o(5; \mathcal{B}(x))$, $x \in \hat{X}_0 \cup \tilde{X}_0$, for the considered example.

Init. state	0 0 0 0	1 1 1 1	0 0 0 1	1 1 1 0
t	o_1 o_2 o_3	o_1 o_2 o_3	o_1 o_2 o_3	o_1 o_2 o_3
0	1 0 0	0 1 1	1 0 0	0 1 0
1	0 0 1	1 1 0	0 0 0	1 1 0
2	1 1 0	0 0 1	1 1 0	0 0 0
3	0 1 1	1 1 0	0 1 0	1 1 0
4	1 1 0	0 1 1	1 1 0	0 1 0
5	0 0 1	1 1 0	0 0 0	1 1 0

[4] Theorem 2 also relates the considered notion of opacity to the notion of *"BN observability"*. We refer the reader to Cheng et al. (2011) for further discussion on this concept.

[5] For brevity, we provide an explicit enumeration of the sets X_0 and $\hat{X}$ instead of the corresponding Boolean functions ζ and ξ.

4. OPTIMAL ENFORCEMENT OF THE SOUGHT OPACITY

In this section we consider the problem of enforcing the strong infinite-step opacity of a BN $\mathcal{B}$ with a set of possible initial states X_0 and a secret-state set $\hat{X}$, when Algorithm 1 returns with a negative outcome. Obviously, a first straightforward remedy of this problem is to remove from the initial-state set X_0 all states $x \in \hat{X}_0$ that fail the test of Step 8 in Algorithm 1. Using the notation introduced in Algorithm 1, this set of states is formally characterized by

$$X_0^B \equiv \{x \in \hat{X}_0 : \forall x' \in \tilde{X}_0,\ o(T;\mathcal{B}(x)) \neq o(T;\mathcal{B}(x'))\} \tag{16}$$

and it is the *minimal* set of initial states that must be blocked in order to establish the sought opacity.

In the rest of this section, we consider an alternative mechanism for rendering BN $\mathcal{B}$ strongly infinite-step opaque with respect to the secret-state set $\hat{X}$. This mechanism does not block any initial states $x \in X_0$, but it tries to satisfy the condition of Step 8 in Algorithm 1 by preventing agent $\mathcal{A}$ from observing some of the variables o_j in the output vector o of BN $\mathcal{B}$. In more technical terms, for any index set $J \subseteq \{1,\ldots,m\}$, we define

$$o(J) \equiv \big[o_j : j \in \{1,\ldots,m\} \setminus J\big]^T \tag{17}$$

Also, $\mathcal{B}(J)$ denotes the BN induced from BN $\mathcal{B}$ by replacing its output vector o by the binary vector $o(J)$ defined in Eq. 17. The index set J is characterized as *effective* if BN $\mathcal{B}(J)$ is strongly infinite-step opaque with respect to the initial-state set X_0 and the secret-state set $\hat{X}$. The set collecting all the effective index sets J is denoted by $\mathcal{J}$. Finally, we assume that the blockage of an output variable $o_j,\ j = 1,\ldots,m$, incurs a cost $c_j > 0$, and we want to identify an effective index set $J^* \in \mathcal{J}$ such that

$$J^* = \arg\min_{J \in \mathcal{J}} \sum_{j \in J} c_j \tag{18}$$

The optimization problem defined by Eq. 18 is conceptually similar to the "*static-mask synthesis*" problem for opacity-enforcement that is investigated in Cassez et al. (2012). But the technical elements that are involved in the detailed specification of this problem and its underlying dynamics, are substantially different from their counterparts in the developments of Cassez et al. (2012), due to the BN nature of the controlled plant.

Next, we provide an integer programming (IP) formulation for the optimization problem of Eq. 18. First consider a state pair $(x, x') \in X_0^B \times \tilde{X}_0$. According to Algorithm 1, an output variable o_j distinguishes state x from state x' if $o_j(x(t)) \neq o_j(x'(t))$ for some $t \in \{0,\ldots,T\}$. Hence, the output-variable subset $O(x,x')$ containing all the variables that distinguish state x from x', can be computed as follows:

$$O(x,x') = \bigvee_{t=0}^{T} \Big(o(x(t);\mathcal{B}(x)) \ \oplus\ o(x(t);\mathcal{B}(x'))\Big) \tag{19}$$

In Eq. 19, the operators '$\bigvee$' and '$\oplus$' denote, respectively, the 'OR' and the 'XOR' Boolean functions, and they must be applied in a component-wise sense upon their m-dimensional binary-vector arguments. The outcome of this computation, $O(x,x')$, is an m-dimensional binary vector with its '1'-valued components indicating the output variables o_j that must be blocked in order to establish the indistinguishability of x and x'.

Let $I_{\{\mathcal{C}\}}$ denote the *indicator function* of some condition $\mathcal{C}$, i.e., $I_{\{\mathcal{C}\}}$ is a binary variable that is equal to '1' if condition $\mathcal{C}$ is true and '0' otherwise. Then, an index set $J \subseteq \{1,\ldots,m\}$ will render a state $x \in X_0^B$ indistinguishable from the state set $\tilde{X}_0$ if and only if

$$\bigvee_{x' \in \tilde{X}_0} I_{\{O(x,x') \subseteq J\}} \tag{20}$$

Furthermore, the index set J is effective if and only if

$$\bigwedge_{x \in X_0^B} \bigvee_{x' \in \tilde{X}_0} I_{\{O(x,x') \subseteq J\}} \tag{21}$$

Equations 19–21 enable a characterization of the target set $\mathcal{J}$ through a SATISFIABILITY (SAT) formula (Papadimitriou (1995)). In order to turn this characterization into a set of binary constraints, consider the binary variables $Y_j,\ j = 1,\ldots,m$, with

$$Y_j = I_{\{j \in J\}} \tag{22}$$

Also, for all $(x,x') \in X_0^B \times \tilde{X}_0$, define

$$J(x,x') = \big\{j \in \{1,\ldots,m\} : O(x,x')[j] = 1\big\} \tag{23}$$

Then, the expression of Eq. 20 can be rewritten as follows:

$$\bigvee_{x' \in \tilde{X}_0} \bigwedge_{j \in J(x,x')} Y_j \ = \bigwedge_{\hat{J} \in \times_{x' \in \tilde{X}_0} J(x,x')} \bigvee_{l=1}^{|\tilde{X}_0|} Y_{\hat{J}[l]} \tag{24}$$

Eq. 24 results from the distributivity of '$\vee$' with respect to '$\wedge$'. Eqs 21, 22 and 24 enable the characterization of the set $\mathcal{J}$ by means of the binary variables $Y_j,\ 1,\ldots,m$, and the following constraint set:

$$\forall x \in X_0^B,\ \forall \hat{J} \in \times_{x' \in \tilde{X}_0} J(x,x'),\ \sum_{l=1}^{|\tilde{X}_0|} Y_{\hat{J}[l]} \ \geq\ 1.0 \tag{25}$$

But then, an IP formulation for the computation of an *optimal effective index set* J^* is as follows:

$$\min \sum_{j=1}^{m} c_j \cdot Y_j \tag{26}$$

s.t. Constraint 25 and the requirement $Y_j \in \mathbb{B},\ \forall j$.

The above IP has m binary variables and $O(|X_0| \cdot m^{|X_0|})$ technological constraints. Furthermore, using the representation of the solution set $\mathcal{J}$ that is provided by Eqs 19–21, it can be easily shown that the combinatorial optimization problem defined by Eq. 18 subsumes the *set covering problem* (Papadimitriou (1995)), and therefore it is NP-hard. [6] The affinity of the considered optimization problem with the set covering problem also suggests that it might be possible to adapt to this problem some of the

[6] More specifically, an instance of the set covering problem with universe set $U = \{u_1,\ldots,u_k\}$, and a collection of subsets of U, $\mathcal{S} = \{S_1,\ldots,S_m\}$, with associated costs $c_j,\ j = 1,\ldots,m$, can be polynomially reduced to the problem defined by Eq. 18, by setting: (i) $\{o_j,\ j = 1,\ldots,m+1\} = \mathcal{S} \cup \{d\}$ with (ii) variable $o_j,\ j = 1,\ldots,m$, having cost c_j, and variable o_{m+1} having cost $c_{m+1} = \infty$; (iii) $X_0^B = U$; and (iv) $\tilde{X}_0 = \mathcal{S}$. Also, for any pair $(u_i, S_j) \in X_0^B \times \tilde{X}_0$, we set $J(u_i,S_j) = \{j\}$, if $u_i \in S_j$; otherwise, we set $J(u_i,S_j) = \{m+1\}$.

Table 2. The index sets $J(x, x')$ of Eq. 23 for the considered example.

$\hat{X}_0 \setminus \tilde{X}_0$	0 0 0 1	1 1 1 0
0 0 0 0	3	1, 2, 3
1 1 1 1	1, 2, 3	3

existing heuristics for the set covering problem; but due to the imposed space limitations, we defer the systematic investigation of this possibility to our future work on this problem. It is also possible to simplify the constraint set of Eq. 25 by identifying (i) redundant variables in the involved summations due to variable repetition in these summations, and (ii) redundant constraints due to certain set inclusions among the various sets $J(x, x')$ that are defined by Eq. 23. In fact, these simplifications can be iterated to the point that they might even lead to an optimal solution of the underlying optimization problem instance without having to formulate and solve the corresponding IP; we demonstrate this possibility with the following example.

Example: We apply the methodology that was developed in this section, in order to enforce the strong infinite-step opacity for the BN $\mathcal{B}$ of the example that was presented in Section 3. In that example it was found that every state $x \in \hat{X}_0$ is distinguishable from some state in $\tilde{X}_0$; therefore, according to Eq. 16, $X_0^B = \hat{X}_0$. Then, applying the formula of Eq. 19 to the state pairs $(x, x') \in \hat{X}_0 \times \tilde{X}_0$, by means of Table 1, we obtain the index sets $J(x, x')$ that are tabulated in Table 2. Associating some costs c_j, $j = 1, 2, 3$, with the corresponding output variables o_j, we can use Table 2 to determine the IP that is defined by Eqs 25–26. After some obvious simplifications of the constraint set that is generated by Eq. 25, this IP can be written as follows:

$$\min \sum_{j=1}^{3} c_j \cdot Y_j \tag{27}$$

s.t.

$$Y_1 + Y_3 \geq 1.0 \; ; \quad Y_2 + Y_3 \geq 1.0 \; ; \quad Y_3 \geq 1.0 \tag{28}$$

$$\forall j \in \{1, 2, 3\}, \; Y_j \in \mathbb{B} \tag{29}$$

Looking at the constraints of this IP, and also considering the fact that $c_j > 0$, $\forall j$, we can see that the optimal solution for it is $Y_1 = Y_2 = 0 \; \wedge \; Y_3 = 1$; i.e., $J^* = \{3\}$. This result could also have been obtained directly from the information that is provided in Table 2, reasoning according to either of the following two arguments:

Argument 1: The appearance of an index j in every cell of a row in Table 2 corresponding to some state $x \in X_0^B$, implies that the output variable o_j distinguishes state x from every state $x' \in \tilde{X}_0$. Therefore, index j must be included in J^*. In the considered example, index 3 is such an index for both rows of Table 2. Furthermore, in this case, the blockage of o_3 renders (i) the state $[0, \; 0, \; 0, \; 0]^T$ indistinguishable from the state $[0, \; 0, \; 0, \; 1]^T$, and (ii) the state $[1, \; 1, \; 1, \; 1]^T$ indistinguishable from the state $[1, \; 1, \; 1, 0]^T$. Hence, $J^* = \{3\}$.

Argument 2: Since $c_j > 0$, $\forall j$, for any $x \in X_0^B$ and $x', x'' \in \tilde{X}_0$, $J(x, x') \subset J(x, x'')$ implies that trying to render state x indistinguishable from state x'' is not a competitive option, and the corresponding cells in Table 2 should be dropped from further consideration. In the context of the considered example, the elimination of these noncompetitive cells leaves the blockage of variable o_3 as the only competitive option for rendering states $[0, \; 0, \; 0, \; 0]^T$ and $[1, \; 1, \; 1, \; 1]^T$ indistinguishable from the state set $\tilde{X}_0$. Hence, $J^* = \{3\}$.

5. CONCLUSION

This paper has adapted the notion of strong infinite-step opacity in the operational context of autonomous BNs, and it has provided (i) an algorithm for the assessment of this property and (ii) some mechanisms for its enforcement in a maximally permissive manner. Instrumental in these developments are (a) the 1-DFSA structure of the underlying dynamics, and (b) the mechanism generating the output that is traced by the external observer. With these elements in mind, one can analyze and enforce additional notions of state-based opacity that were listed in the introductory section. This analysis can be extended even to controlled Boolean networks when the external observer traces the controls that are applied at each iteration. On the other hand, the case of controlled BNs with unobservable controls at each iteration gives rise to observational structures and dynamics that are closer to the typical FSA-based representations of the opacity assessment and enforcement problems that have been investigated by the DES community. Finally, another issue that can render more complete the developments that are presented in this work, is the formal characterization of the worst-case computational complexity of the opacity assessment problem that is addressed in Section 3.

REFERENCES

Cassandras, C.G. and Lafortune, S. (2021). *Introduction to Discrete Event Systems (3rd ed.)*. Springer, NY,NY.

Cassez, F., Dubreil, J., and Marchand, H. (2012). Synthesis of opaque systems with static and dynamic masks. *Formal Methods in Systems Design*, 40, 88–115.

Cheng, D., Qi, H., and Li, Z. (2011). *Analysis and Control of Boolean Networks*. Springer, London.

Cury, J.E.R. and Baldissera, F.L. (2012). Some perspectives and challenges in the (discrete) control of cellular systems. In *Proc. of the 11th International Workshop on Discrete Event Systems*, 1–3. IFAC.

Gao, Z., Chen, X., and Basar, T. (2018). Stability structures of conjunctive Boolean networks. *Automatica*, 89, 8–20.

Hadjicostis, C.N. (2020). *Estimation and Inference in Discrete Event Systems: A Model-Based Approach with Finite Automata*. Springer, Switzerland.

Jacob, R., Lessage, J.J., and Faure, J.M. (2016). Overview of Discrete Event Systems Opacity: models, validation and quantification. *Annual Reviews in Control*, 41, 135–146.

Kauffman, S.A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22, 437–467.

Lafortune, S., Lin, F., and Hadjicostis, C.N. (2018). On the history of diagnosability and opacity in discrete event systems. *Annual Reviews in Control*, 45, 257–266.

Papadimitriou, C.H. (1995). *Computational Complexity*. Addison-Wesley, Reading, MA.