2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

New Results on Input-output Decoupling of Boolean Control Networks

Yiliang Li, Hongli Lyu, Jun-e Feng and Abdelhamid Tayebi

***Abstract*— We provide new necessary and sufficient conditions (with low computational complexity) for the input-output decoupling problem of Boolean control networks. Instrumental in our approach, the introduction of a new concept relying on the construction of some input-output-decoupling matrices that have to satisfy some conditions to ascertain whether a given Boolean control network is input-output decoupled. A numerical example is provided to illustrate our theoretical developments.**

I. Introduction

Boolean networks, which have been first introduced by Kauffman in [1], are known to be of great importance in the description and simulation of the behaviour of a variety of biological systems and physiological processes, such as genetic regulatory networks [2], cell growth, apoptosis and differentiation [3]. The success in the development of Boolean Networks (BNs) and Boolean Control Networks (BCNs) is mainly attributed to the powerful algebraic framework developed by Cheng and co-authors [4], that allows to represent these logical networks as linear state space representations with canonical state vectors. This allowed to recast most of the common systems and control problems into the BCN world (see, for instance, [5]–[18]).

The Input-output (IO)-decoupling problem, also known as Morgan's problem [19], is no exception, and is among the problems that have been dealt with from the BCN point of view. The first definition of BCN IO-decoupling was introduced by Valcher in [20], where some algebraic criteria for the characterization of IO-decoupling were proposed. However, the computational complexity of these criteria is quite high, and as such, it is difficult to apply them to BCNs with a large number of nodes. As an alternative, a graph theory based approach was presented in [21]. Unfortunately, this approach relies on a large observability matrix to determine the proper vertex partition, and does not provide a constructive procedure for the verification of the conditions required for IO-decoupling. Although the work in [22], [23] deals with IO-decoupling via state feedback, it does not provide a procedure that allows to check whether a given open-loop BCN system is IO-decoupled.

This work was supported in part by the National Natural Science Foundation of China under the grant 62273201, the Research Fund for the Taishan Scholar Project of Shandong Province of China under the grant tstp20221103, and the National Sciences and Engineering Research Council of Canada (NSERC), under the grant NSERC-DG RGPIN 2020-06270.

Yiliang Li and Jun-e Feng are with the School of Mathematics, Shandong University, Jinan, Shandong, P.R. China (E-mail: liyiliang1994@126.com, fengjune@sdu.edu.cn)

Hongli Lyu and Abdelhamid Tayebi are with the Department of Electrical Engineering, Lakehead University, Thunder Bay, Ontario, Canada (E-mail: {hlyu1;atayebi}@lakeheadu.ca)

The objective of the present paper is to provide verifiable necessary and sufficient conditions (with low time and space complexities) allowing to check whether an open-loop BCN system is IO-decoupled or not. These conditions rely on the construction of an appropriate matrix (from the the system's information) with space complexity less than or equal to $O(2^{2n+1})$, where n is the number of state nodes. Note that the space complexity of our approach is lower than that of the observability matrix obtained in [21], which is $O(2^{2^n+n})$. The time complexity of the verification condition proposed in [20] is $O(m2^{2^n})$, while that of our proposed algorithm is $O(m2^{3n})$, where m is the number of input nodes. Interestingly, our proposed approach can be applied, after minor modifications, to handle other problems such as disturbance decoupling.

The remainder of this paper is organized as follows. Section II presents the notations used throughout the paper, as well as some preliminaries on semi-tensor product (STP) of matrices and algebraic forms of BCNs. Section III presents our main results. In Section IV, a numerical example is given to illustrate the effectiveness of the proposed approach. Some concluding remarks are provided in Section V.

II. Preliminaries

A. Notations

The set of nonnegative integers is denoted by $\mathbb{N}$. The set of all $m \times n$ real matrices is denoted by $\mathcal{M}_{m\times n}$. The i-th column (row) of a matrix M is denoted by $\mathrm{Col}_i(M)$ ($\mathrm{Row}_i(M)$). The cardinality of a set Ω is denoted by $|\Omega|$. The set of all $m \times n$ Boolean matrices is denoted by $\mathcal{B}_{m\times n} := \{B \in \mathcal{M}_{m\times n} | B_{ij} \in \mathcal{D}\}$, where $\mathcal{D} = \{0, 1\}$. The set of all $m \times n$ logical matrices is denoted by $\mathcal{L}_{m\times n} := \{L \in \mathcal{B}_{m\times n} | \mathrm{Col}_i(L) \in \Delta_m, i = 1, \dots, n\}$, where $\Delta_m = \{\delta_m^i | \delta_m^i = \mathrm{Col}_i(I_m), i = 1, \dots, m\}$ and $I_m = \mathrm{diag}\{1, \dots, 1\}, m \geq 2$. We will denote Δ_2 by Δ, and the logical matrix $[\delta_m^{i_1} \ \cdots \ \delta_m^{i_n}]$ by $\delta_m[i_1 \ \cdots \ i_n]$. We define the scalar-valued function $\mathrm{sgn}(a)$ as a function that returns 0 if $a = 0$ and 1 if $a \neq 0$. For a given matrix $M = (M_{ij})_{p\times q}$, we define $\mathrm{sgn}(M) = (\mathrm{sgn}(M_{ij}))_{p\times q}$. We define $V(\Omega) := \delta_n^{i_1} + \delta_n^{i_2} + \cdots + \delta_n^{i_m}$, where $\Omega = \{\delta_n^{i_1}, \delta_n^{i_2}, \dots, \delta_n^{i_m}\}$. The kronecker product is denoted by $\otimes$, and the swap matrix with indices n and m is defined as $W_{[n,m]} = [I_m \otimes \delta_n^1 \ \cdots \ I_m \otimes \delta_n^n]$. We denote the n-dimensional vectors of ones and zeros by $\mathbf{1}_n := \sum_{i=1}^{n} \delta_n^i$ and $\mathbf{0}_n := [0 \ \cdots \ 0]^\top$, respectively.

B. STP basics

Given two matrices $P \in \mathcal{M}_{m\times n}$ and $Q \in \mathcal{M}_{p\times q}$, the STP of P and Q is defined as

$$P \ltimes Q := (P \otimes I_{t/n})(Q \otimes I_{t/p}),$$

where t is the least common multiple of n and p. In this paper, the default matrix product is STP and the symbol $\ltimes$ is omitted. Define the one-to-one mapping φ from $\mathcal{D}$ to Δ, such that $\varphi(1) = \delta_2^1$ and $\varphi(0) = \delta_2^2$. For any logical variable $X_i \in \mathcal{D}, i = 1, \ldots, n$ and any n-ary Boolean function $f(X_1, \ldots, X_n)$, one has the following vector forms

$$\varphi(X_i) = \begin{bmatrix} X_i \\ 1 - X_i \end{bmatrix}$$

and

$$\varphi(f(X_1, \ldots, X_n)) = \begin{bmatrix} f(X_1, \ldots, X_n) \\ 1 - f(X_1, \ldots, X_n) \end{bmatrix},$$

where $i = 1, \ldots, n$. The following lemma shows that for any Boolean function, there is a structure matrix such that its vector form can be depicted via a linear form.

Lemma 1: [24] For an $n-$ary Boolean function $f(X_1, \ldots, X_n)$, there exists a unique structure matrix $L_f \in \mathcal{L}_{2\times 2^n}$ such that the vector form of $f(X_1, \ldots, X_n)$ is expressed as

$$\varphi(f(X_1, \ldots, X_n)) = L_f \ltimes_{i=1}^n x_i,$$

where $x_i = \varphi(X_i), i = 1, \ldots, n$.

In addition, the following lemma is useful for the introduction of BCN algebraic forms.

Lemma 2: [24] Assume

$$\begin{cases} y = M_y \ltimes_{i=1}^n x_i, \\ z = M_z \ltimes_{i=1}^n x_i, \end{cases}$$

where $x_i, y, z \in \Delta, i = 1, 2, \ldots, n, M_y, M_z \in \mathcal{L}_{2\times 2^n}$. Then

$$yz = (M_y * M_z) \ltimes_{i=1}^n x_i,$$

where $M_y * M_z = [\mathrm{Col}_1(M_y) \otimes \mathrm{Col}_1(M_z) \ \cdots \ \mathrm{Col}_{2^n}(M_y) \otimes \mathrm{Col}_{2^n}(M_z)]$.

C. Algebraic forms of BCNs

Consider the following BCN:

$$\begin{cases} X_i(t+1) = f_i(X(t), U(t)), \\ Y_j(t) = g_j(X(t)), \end{cases} \tag{1}$$

where $X(t) = (X_1(t), \ldots, X_n(t)), U(t) = (U_1(t), \ldots, U_m(t))$ and $Y(t) = (Y_1(t), \ldots, Y_m(t))$ are the state, input and output of the system, respectively, f_i and g_j are Boolean functions.

Let $x_i = \varphi(X_i), u_i = \varphi(U_i)$ and $y_i = \varphi(Y_i)$. From Lemma 1 and Lemma 2, BCN (1) is transformed into the following algebraic form:

$$\begin{cases} x(t+1) = Lu(t)x(t), \\ y_i(t) = H_i x(t), \end{cases} \tag{2}$$

where $x(t) = \ltimes_{i=1}^n x_i(t) \in \Delta_{2^n}$ and $u(t) = \ltimes_{i=1}^m u_i(t) \in \Delta_{2^m}$ are vector forms of $X(t)$ and $U(t)$, respectively, $y_i(t) \in \Delta, L \in \mathcal{L}_{2^n\times 2^{m+n}}$ and $H_i \in \mathcal{L}_{2\times 2^n}, i = 1, \ldots, m$.

III. Main Results

The IO-decoupling of BCN is defined in the following definition provided in [20].

Definition 1: BCN (2) is said to be IO-decoupled if for every index $i \in \{1, \ldots, m\}$ and every initial state $x(0) \in \Delta_{2^n}$, if $\{u(t)\}_{t=0}^{+\infty}$ and $\{\hat{u}(t)\}_{t=0}^{+\infty}$ are two input sequences characterized by the fact that their i-th entries coincide at every time instant, *i.e.*,

$$u_i(t) = \hat{u}_i(t), \forall t \in \mathbb{N},$$

then the output sequences $\{y(t)\}_{t=0}^{+\infty}$ and $\{\hat{y}(t)\}_{t=0}^{+\infty}$, generated by BCN (2) corresponding to $x(0), \{u(t)\}_{t=0}^{+\infty}$ and $\{\hat{u}(t)\}_{t=0}^{+\infty}$, respectively, satisfy

$$y_i(t) = \hat{y}_i(t), \forall t \in \mathbb{N}.$$

In the sequel, we will include all inputs with $u_i = \delta_2^j$ into the set $\Omega_{ij} = \{u | u = \delta_{2^{i-1}}^{k_1} \ltimes \delta_2^j \ltimes \delta_{2^{m-i}}^{k_2}, k_1 \in \{1, \ldots, 2^{i-1}\}, k_2 \in \{1, \ldots, 2^{m-i}\}\}$, where $i \in \{1, \ldots, m\}$ and $j = 1, 2$.

According to Definition 1, an IO-decoupled BCN requires that for each index $i \in \{1, \ldots, m\}$ and each initial state $x(0) \in \Delta_{2^n}$, $\{y_i(t)\}_{t=1}^{+\infty}$ remains unchanged for any input sequence $\{u(t)\}_{t=0}^{+\infty}$ satisfying $u(t) \in \Omega_{ij_t}, t \in \mathbb{N}, j_t = 1, 2$. It follows that, for each index $i \in \{1, \ldots, m\}$ and each initial state $x(0) \in \Delta_{2^n}$, $y_i(1)$ remains unchanged for any $u(0) \in \Omega_{ij_0}, j_0 = 1, 2$. Therefore, one has $H_i LV(\Omega_{ij_0})x(0) \in \{2^{m-1}\delta_2^1, 2^{m-1}\delta_2^2\}$, where $j_0 = 1, 2$, and consequently $\mathrm{sgn}(H_i LV(\Omega_{ij_0})x(0)) \in \Delta, j_0 = 1, 2$. Based on the above analysis, a necessary condition is provided for the IO-decoupling of BCNs.

Proposition 1: If BCN (2) is IO-decoupled, then for each index $i \in \{1, \ldots, m\}$, one has

$$\mathrm{sgn}(H_i L[V(\Omega_{i1}) \ V(\Omega_{i2})]) \in \mathcal{L}_{2\times 2^{n+1}}. \tag{3}$$

Proof: If there exists an index $i \in \{1, \ldots, m\}$ such that $\mathrm{sgn}(H_i L[V(\Omega_{i1}) \ V(\Omega_{i2})]) \notin \mathcal{L}_{2\times 2^{n+1}}$, then there exists a state $x \in \Delta$ such that $\mathrm{sgn}(H_i LV(\Omega_{i1})x) = \mathbf{1}_2$ or $\mathrm{sgn}(H_i LV(\Omega_{i2})x) = \mathbf{1}_2$. Without loss of generality, let us assume $\mathrm{sgn}(H_i LV(\Omega_{i1})x) = \mathbf{1}_2$. Consequently, there exist $u, \hat{u} \in \Omega_{i1}$ such that $H_i Lux \neq H_i L\hat{u}x$, which is a contradiction. Hence, any IO-decoupled BCN should satisfy $\mathrm{sgn}(H_i L[V(\Omega_{i1}) \ V(\Omega_{i2})]) \in \mathcal{L}_{2\times 2^{n+1}}$, where $i \in \{1, \ldots, m\}$. ■

Condition (3), which implies that, for each index $i \in \{1, \ldots, m\}$ and each initial state $x(0) \in \Delta_{2^n}$, $y_i(1)$ remains unchanged for any $u(0) \in \Omega_{ij_0}, j_0 = 1, 2$, is not enough to ascertain whether a given BCN is IO-decoupled or not. More conditions are required to verify whether $y_i(t), t > 1$ remains unchanged for any $u(t) \in \Omega_{ij_t}, j_t = 1, 2$. I what follows, we propose a procedure, based on the construction of m input-output-decoupling (IOD) matrices, namely $F_1, \ldots, F_m$, to verify whether a given BCN is IO-decoupled or not. The procedural construction of these IOD matrices, that reflect the output information, is shown in Algorithm 1. The procedure starts by finding the state subset associated to a given output. That is, for each index $i \in \{1, \ldots, m\}$ and $j = 1, 2$, the initial state subset is given by $\Gamma_{ij} =$

$\{x|H_ix = \delta_2^j\}$, in which all states satisfy $y_i = \delta_2^j$. For each index $i \in \{1,\dots,m\}$, the initial IOD matrix is given by $F_i = \text{sgn}(H_iL[V(\Omega_{i1})\ V(\Omega_{i2})])$. If there exists one initial IOD matrix $F_i \notin \mathcal{L}_{2\times 2^{n+1}}$, then the given BCN is not IO-decoupled as per Proposition 1, in which case Algorithm 1 ends the while loop. Otherwise, for each index $i \in \{1,\dots,m\}$ and each state $x(t) \in \Delta_{2^n}$, $y_i(t+1)$ remains unchanged for all $u(t) \in \Omega_{ij_t}, j_t = 1,2$. In this case, we check whether $y_i(t+1)$ generated by different states in Γ_{ij} remains unchanged, and partition the state subset Γ_{ij} accordingly. Specifically, for each index $i \in \{1,\dots,m\}$ and any two states $x(t), \hat{x}(t) \in \Gamma_{ij}, j = 1,2$, if $F_iW_{[2^n,2]}x(t) = F_iW_{[2^n,2]}\hat{x}(t)$, then the corresponding $y_i(t+1)$ and $\hat{y}_i(t+1)$, generated by $x(t)$ and $\hat{x}(t)$ respectively, are the same for all inputs $u(t)$ with the same $u_i(t)$, in which case we put $x(t)$ and $\hat{x}(t)$ into the same state subset. Otherwise, there are some inputs $u(t)$ with the same $u_i(t)$ such that the corresponding $y_i(t+1)$ and $\hat{y}_i(t+1)$, generated by $x(t)$ and $\hat{x}(t)$ respectively, are different, in which case we put $x(t)$ and $\hat{x}(t)$ into different state subsets. Executing this procedure for each state subset Γ_{ij}, a set of state subsets are obtained and denoted as $\Gamma_{i1},\dots,\Gamma_{i\mu_i}, i \in \{1,\dots,m\}$. Using the updated state subsets $\Gamma_{i1},\dots,\Gamma_{i\mu_i}$, F_i is recomputed via $\text{Row}_j(F_i) = \text{sgn}(\sum_{\delta_{2^n}^{\alpha}\in\Gamma_{ij}} \text{Row}_\alpha(L[V(\Omega_{i1})\ V(\Omega_{i2})]))$, where $i \in \{1,\dots,m\}$.

Next, we check whether the updated IOD matrices $F_1,\dots,F_m$ are logical matrices, and Algorithm 1 ends the while loop when one of the following cases occurs.

1) *Not all constructed IOD matrices $F_1,\dots,F_m$ are logical matrices*: If there is an index $i \in \{1,\dots,m\}$ such that F_i is not a logical matrix, then there is a state $x \in \Delta_{2^n}$, which evolves to two different state subsets Γ_{ij_1} and Γ_{ij_2} under different inputs $u, \hat{u} \in \Delta_{2^m}$ with the same u_i. As a result, there are two input sequences $\{u(t)\}_{t=0}^{+\infty}$ and $\{\hat{u}(t)\}_{t=0}^{+\infty}$ satisfying $u_i(t) = \hat{u}_i(t), \forall t \in \mathbb{N}$ such that the corresponding $\{y_i(t)\}_{t=0}^{+\infty}$ and $\{\hat{y}_i(t)\}_{t=0}^{+\infty}$ are different. This contradicts with the definition of IO-decoupling.
2) *The constructed IOD matrices $F_1,\dots,F_m$ are no longer updated*: If each updated matrix F_i is the same as the previous matrix, *i.e.*, all state subsets are not partitioned further, then all states in the same Γ_{ij} have the same $\{y_i(t)\}_{t=0}^{+\infty}$ under any input sequence $\{u(t)\}_{t=0}^{+\infty}$ with the same $\{u_i(t)\}_{t=0}^{+\infty}$. This means that the given BCN is IO-decoupled.

If the updated IOD matrices $F_1,\dots,F_m$ are logical matrices and different from the previous IOD matrices, then the while loop continues, *i.e.*, repeat the above procedure to partition the state subset Γ_{ij} further.

Remark 1: For each index $i \in \{1,\dots,m\}$, the computational complexity of the construction of the matrices F_i is $O(\mu_i\nu_i^3)$, where $\nu_i = \max\{|\Gamma_{i1}|,\dots,|\Gamma_{i\mu_i}|\}$. Since the worst case is $\nu_i = 2^n$ or $\mu_i = 2^n$, the computational complexity of Algorithm 1 is $O(m2^{3n})$.

Based on the analysis above, the necessary and sufficient condition for IO-decoupling can be formulated depending on the nature of the IOD matrices $F_1,\dots,F_m$.

Algorithm 1 The construction of $F_1,\dots,F_m$

Input: L, H_i, $\Omega_{ij}, \Gamma_{ij}, i \in \{1,\dots,m\}, j = 1,2$.
Output: F_i and $\Gamma_{i1},\dots,\Gamma_{i\mu_i}, i \in \{1,\dots,m\}$.

for $i = 1$ to m, **do**
  compute $F_i = \text{sgn}(H_iL[V(\Omega_{i1})\ V(\Omega_{i2})])$;
  $M_i = \mathbf{0}_{2\times 2^{n+1}}$;
  $\mu_i = 2$;
  while $M_i \neq F_i$ and $F_i \in \mathcal{L}_{\mu_i\times 2^{n+1}}$, **do**
    let $M_i = F_i$;
    let $\Gamma'_\alpha = \{\delta_{2^n}^\alpha\}, \alpha \in \{1,\dots,2^n\}$;
    for $j = 1$ to μ_i, **do**
      for $k = 1$ to $|\Gamma_{ij}| - 1$, **do**
        if $\Gamma'_{\gamma_{ij}^k} \neq \emptyset$, **then**
          for $l = k+1$ to $|\Gamma_{ij}|$, **do**
            if $M_iW_{[2^n,2]}\delta_{2^n}^{\gamma_{ij}^k} = M_iW_{[2^n,2]}\delta_{2^n}^{\gamma_{ij}^l}$, **then**
              let $\Gamma'_{\gamma_{ij}^k} = \Gamma'_{\gamma_{ij}^k} \cup \{\delta_{2^n}^{\gamma_{ij}^l}\}$;
              let $\Gamma'_{\gamma_{ij}^l} = \emptyset$;
            end if
          end for
        end if
      end for
    end for
    let μ_i be the number of all nonemptyset subsets $\Gamma'_{\gamma_{ij}^k}$;
    rearrange these nonemptyset subsets $\Gamma'_{\gamma_{ij}^k}$, denoted by $\Gamma_{i1},\dots,\Gamma_{i\mu_i}$;
    recompute F_i according to $\text{Row}_j(F_i) = \text{sgn}(\sum_{\delta_{2^n}^{\alpha}\in\Gamma_{ij}} \text{Row}_\alpha(L[V(\Omega_{i1})\ V(\Omega_{i2})]))$;
  end while
  output F_i and $\Gamma_{i1},\dots,\Gamma_{i\mu_i}$.
end for

Theorem 1: Consider BCN (2) and let the IOD matrices $F_1,\dots,F_m$ be obtained from Algorithm 1. Then, the following statements are equivalent.

1) BCN (2) is IO-decoupled.
2) $F_i \in \mathcal{L}_{\mu_i\times 2^{n+1}}, i \in \{1,\dots,m\}$.
3) $\overline{F}_i \in \mathcal{L}_{\mu_i\times 2\mu_i}$, where $\text{Col}_j(\overline{F}_i\delta_2^k) = \text{sgn}(\sum_{\delta_{2^n}^{\alpha}\in\Gamma_{ij}} \text{Col}_\alpha(F_i\delta_2^k))$ and $i \in \{1,\dots,m\}, k = 1,2$.

Proof: ($1 \Rightarrow 2$) If there is an F_i satisfying $F_i \notin \mathcal{L}_{\mu_i\times 2^{n+1}}$, then there is a state x and input u_i such that $\mathbf{1}_{\mu_i}^\top(F_iu_ix) \neq 1$. Without loss of generality, assume $\text{Row}_{k_1}(F_iu_ix) = \text{Row}_{k_2}(F_iu_ix) = 1$. Since $\{y_i(t)\}_{t=0}^{+\infty}$ generated by the states in Γ_{ik_1} are different from those generated by the states in Γ_{ik_2}, there are input sequences $\{u(t)\}_{t=0}^{+\infty}$ and $\{\hat{u}(t)\}_{t=0}^{+\infty}$ with the same $\{u_i(t)\}_{t=0}^{+\infty}$ such that $\{y_i(t)\}_{t=0}^{+\infty}$ and $\{\hat{y}_i(t)\}_{t=0}^{+\infty}$ generated by the state x are different. This is a contradiction. Thus, for each index $i \in \{1,\dots,m\}$, F_i obtained from Algorithm 1 is a logical matrix if the given BCN is IO-decoupled.

($2 \Rightarrow 3$) If there is an $\overline{F}_i$ satisfying $\overline{F}_i \notin \mathcal{L}_{\mu_i\times 2\mu_i}$, then there is a state subset Γ_{ij} and u_i such that $\mathbf{1}_{\mu_i}^\top\text{Col}_j(\overline{F}_iu_i) \neq 1$. Without loss of generality, assume $(\overline{F}_iu_i)_{l_1j} = (\overline{F}_iu_i)_{l_2j} = 1$. Then states in Γ_{ij} can evolve to states in Γ_{il_1} and Γ_{il_2} under some inputs u with the same u_i. According to Algorithm 1, $\{y_i(t)\}_{t=0}^{+\infty}$ generated by the states in Γ_{il_1} are different from those generated by states in Γ_{il_2}. Thus, states in Γ_{ij} generate different output sequences $\{y_i(t)\}_{t=0}^{+\infty}$, in which case Algorithm 1 ends the while loop due to the fact that F_i is not a logical matrix. This is a contradiction. Hence, it follows from $F_i \in \mathcal{L}_{\mu_i \times 2\mu_i}$ that $\overline{F}_i \in \mathcal{L}_{\mu_i \times 2\mu_i}$ holds, where $i \in \{1, \ldots, m\}$.

($3 \Rightarrow 1$) For each index $i \in \{1, \ldots, m\}$, $\overline{F}_i \in \mathcal{L}_{\mu_i \times 2\mu_i}$ means that states in Γ_{ij} evolve to states in the same Γ_{ik} for all inputs $u \in \Delta_{2^m}$ with the same u_i. For each index $i \in \{1, \ldots, m\}$ and a given initial state $x(0) = \delta_{2^n}^{\alpha} \in \Delta_{2^n}$, a state subset Γ_{ij_0} satisfying $x(0) = \delta_{2^n}^{\alpha} \in \Gamma_{ij_0}$ can be found. Due to the fact that $\overline{F}_i \in \mathcal{L}_{\mu_i \times 2\mu_i}$, there is only one Γ_{ij_1} such that $x(0)$ evolves to states in Γ_{ij_1} for all $u(0) \in \Delta_{2^m}$ with the same $u_i(0) \in \Delta$. As a result, the corresponding $y_i(1)$ is the same for all $u(0) \in \Delta_{2^m}$ with the same $u_i(0)$. Assume that the given initial state $x(0)$ evolves to a state $x(1) \in \Gamma_{ij_1}$. Due to the fact that $\overline{F}_i \in \mathcal{L}_{\mu_i \times 2\mu_i}$, there is only one Γ_{ij_2} such that $x(1)$ evolves to states in Γ_{ij_2} for all $u(1) \in \Delta_{2^m}$ with the same $u_i(1) \in \Delta$. Thus, the corresponding output sequence $\{y_i(t)\}_{t=0}^{2}$ is the same for all $\{u(t)\}_{t=0}^{1}$ with the same $\{u_i(t)\}_{t=0}^{1}$. Repeating this process, one sees that for a given initial state $x(0) = \delta_{2^n}^{\alpha}$, the corresponding $\{y_i(t)\}_{t=0}^{+\infty}$ is the same for all $\{u(t)\}_{t=0}^{+\infty}$ with the same $\{u_i(t)\}_{t=0}^{+\infty}$. Due to the arbitrariness of i and $x(0)$, one can conclude that the given BCN is IO-decoupled. ■

Remark 2: Note that $\overline{F}_i$ is computed according to $\mathrm{Col}_j(\overline{F}_i \delta_2^k) = \mathrm{sgn}(\sum_{\delta_{2^n}^{\alpha} \in \Gamma_{ij}} \mathrm{Col}_{\alpha}(F_i \delta_2^k))$, where $i \in \{1, \ldots, m\}, k = 1, 2$. For each state subset $\Gamma_{ij}, i \in \{1, \ldots, m\}, j \in \{1, \ldots, \mu_i\}$, since all states in Γ_{ij} generate the same output sequence $\{y_i(t)\}_{t=0}^{+\infty}$, one has $\mathrm{Col}_{\alpha_1}(F_i \delta_2^k) = \mathrm{Col}_{\alpha_2}(F_i \delta_2^k)$ if $\delta_{2^n}^{\alpha_1}, \delta_{2^n}^{\alpha_2} \in \Gamma_{ij}$. Consequently, if $F_i \in \mathcal{L}_{\mu_i \times 2^{n+1}}$, one has $\mathbf{1}_{\mu_i}^{\top} \mathrm{Col}_j(\overline{F}_i \delta_2^k) = \mathbf{1}_{\mu_i}^{\top} \mathrm{sgn}(\sum_{\delta_{2^n}^{\alpha} \in \Gamma_{ij}} \mathrm{Col}_{\alpha}(F_i \delta_2^k)) = 1$. Due to the arbitrariness of j, k, we confirm that $\overline{F}_i \in \mathcal{L}_{\mu_i \times 2\mu_i}$.

Remark 3: For the sake of simplicity, in the present work, we consider that a given BCN is IO-decoupled if the i-th output y_i is affected only by the i-th input u_i, where $i \in \{1, \ldots, m\}$. In the general case, where the i-th output y_i is affected by the j_i-th input u_{j_i}, our approach is still applicable. In fact, for each output $y_i, i \in \{1, \ldots, m\}$, we use Algorithm 1 to construct a set of IOD matrices, *i.e.*, $F_{i1}, \ldots, F_{im}$, and then we check, via an exhaustive process shown in Algorithm 2, whether there are m distinct indexes $j_1, \ldots, j_m$ such that $F_{1j_1}, \ldots, F_{mj_m}$ are logical matrices. If such IOD matrices exist, then the given BCN is IO-decoupled. In Algorithm 2, the constructed matrix $A \in \mathcal{M}_{m \times m!}$ satisfies

1) $A_{ij} \in \{1, \ldots, m\}$, where $i \in \{1, \ldots, m\}, j \in \{1, \ldots, m!\}$;
2) for $i \in \{1, \ldots, m!\}$, $A_{1i}, \ldots, A_{mi}$ are distinct;
3) any two columns of A are distinct.

For instance, if $m = 3$, then one has

$$A = \begin{bmatrix} 1 & 1 & 2 & 2 & 3 & 3 \\ 2 & 3 & 1 & 3 & 1 & 2 \\ 3 & 2 & 3 & 1 & 2 & 1 \end{bmatrix}$$

For a given BCN with $m = 3$, if $F_{1j_1}, \ldots, F_{mj_m}$ are logical matrices when $j_i = A_{i4}$, then we obtain

$$\begin{bmatrix} u_{j_1} \\ u_{j_2} \\ u_{j_3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

i.e., $u_{j_1} = u_2, u_{j_2} = u_3, u_{j_3} = u_1$.

Algorithm 2 The solvability algorithm

Input: L, H_i, $\Omega_{ij}, \Gamma_{ij}, i \in \{1, \ldots, m\}, j = 1, 2$.
Output: "R", if the given BCN is IO-decoupled; and "None", otherwise.

for $i = 1$ to m, **do**
  for $j = 1$ to m, **do**
    compute $F_{ij} = \mathrm{sgn}(H_i L[V(\Omega_{j1})\ V(\Omega_{j2})])$;
    let $F_i = F_{ij}$, and execute Steps 3-24 of Algorithm 1;
    denote $F_{ij} = F_i$;
  end for
end for
construct a matrix $A \in \mathcal{M}_{m \times m!}$ such that (i) $A_{ij} \in \{1, \ldots, m\}$, where $i \in \{1, \ldots, m\}, j \in \{1, \ldots, m!\}$, (ii) for $i \in \{1, \ldots, m!\}$, $A_{1i}, \ldots, A_{mi}$ are distinct, (iii) any two columns of A are distinct;
$r = 0$ and $k = 1$;
while $r = 0$ and $k \le m!$, **do**
  denote $j_i = A_{ik}, i \in \{1, \ldots, m\}$;
  if $F_{1j_1}, \ldots, F_{mj_m}$ are logical matrices; **then**
    $r = 1$;
    the mapping between $u_{j_1}, \ldots, u_{j_m}$ and $u_1, \ldots, u_m$ is as follows:

$$\begin{bmatrix} u_{j_1} \\ u_{j_2} \\ \vdots \\ u_{j_m} \end{bmatrix} = \begin{bmatrix} (\delta_m^{A_{1k}})^{\top} \\ (\delta_m^{A_{2k}})^{\top} \\ \vdots \\ (\delta_m^{A_{mk}})^{\top} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}$$

  end if
  $k = k + 1$;
end while
if $r = 1$ **then**
  output $R = [\delta_m^{A_{1k}}\ \delta_m^{A_{2k}}\ \cdots\ \delta_m^{A_{mk}}]^{\top}$;
else
  output "None".
end if

In the following, we provide a comparative discussion with the existing results in the literature.

- In light of the results provided in [20], a given BCN is IO-decoupled if and only if for each index $i \in \{1, \ldots, m\}$ and each $k \in \mathbb{N}, k \ge 1$, the matrix $\mathrm{sgn}(H_i(LV(\Omega_{ib_k})) \cdots (LV(\Omega_{ib_2}))(LV(\Omega_{ib_1})))$ is a logical matrix, where $b_j = 1, 2, j \in \{1, \ldots, k\}$. Hence, $m2^{2^n+1}$ matrices need to be checked, which means that the time complexity of the method is $O(m2^{2^n})$. In contrast to the results in [20], the time complexity of Theorem 1 is much lower (*i.e.*, $O(m2^{3n})$).
- The necessary and sufficient condition of in [21] for IO-decoupling, states that for each index $i \in \{1, \ldots, m\}$, the state transition graphs $\mathcal{G}_{i1}$ and $\mathcal{G}_{i2}$ must have a common *concolorous* and *perfect* vertex partition, where $\mathcal{G}_{ij}$ is derived from $\mathrm{sgn}(LV(\Omega_{ij})), j = 1, 2$. However,

reference [21] does not provide a constructive procedure to verify the partitioning conditions mentioned above.

- Although an observability matrix method has been provided in [21] to obtain a proper vertex partition, the space complexity of the observability matrix is $O(2^{2^n+n})$ which is higher than the space complexity of F_i in our approach, which is $O(\mu_i 2^{n+1})$.
- Reference [21] does not provide a procedure to test whether a vertex partition is concolorous and perfect. In our approach, the BCN is IO-decoupled if and only if $\overline{F}_i$ is a logical matrix. The computational complexity of the algorithm dealing with the construction of $\overline{F}_i$ is $O(\mu_i^2 \nu_i^4)$.
- The work in [22], [23] focused on the IO-decoupling via state feedback. The obtained results can be used to verify whether a given BCN is IO-decoupled. However, only some sufficient conditions are provided since the IO-decoupling considered in [22], [23] is a special case of the present paper, which has been proven in [21].

Despite the fact that the role of Algorithm 1 consists in determining whether a given BCN is IO-decoupled, the idea might also be leveraged to handle other problems. For instance, consider the disturbance decoupling problem of the following BN with disturbances

$$\begin{cases} x(t+1) = F\xi(t)x(t), \\ y(t) = Hx(t), \end{cases} \tag{4}$$

where $x(t) \in \Delta_{2^n}, \xi(t) \in \Delta_{2^q}, y(t) \in \Delta_{2^p}$ are the state, disturbance and output of the system respectively, $F \in \mathcal{L}_{2^n \times 2^{n+q}}$ and $H \in \mathcal{L}_{2^p \times 2^n}$. BN (4) is disturbance decoupled if for each initial state $x(0) \in \Delta_{2^n}$, the output sequence $\{y(t)\}_{t=0}^{+\infty}$ is the same for any disturbance sequence $\{\xi(t)\}_{t=0}^{+\infty}$ with $\xi(t) \in \Delta_{2^q}$. It is clear that Algorithm 1 can be used to classify all states into several subsets according to the output information. In fact, Algorithm 1 can be employed to construct a matrix $\widehat{H}$, each column of which reflects whether the output sequence of the corresponding state $x \in \Delta_{2^n}$ is affected by the disturbance sequence. Meanwhile, some state subsets are obtained, and all states in the same state subset have the same output sequence under any disturbance sequence. Based on the constructed matrix $\widehat{H}$ and the obtained state subsets, the disturbance decoupling of BN (4) can be studied.

IV. A NUMERICAL EXAMPLE

This section provides a numerical example to show the effectiveness of the proposed approach.

Example 1: Consider a reduced sub-network of signal transduction networks [25], that regulate fundamental biological processes, which can be modelled as the following BCN:

$$\begin{cases} X_1(t+1) = U_1(t) \wedge \neg X_3(t) \wedge U_2(t), \\ X_2(t+1) = U_2(t), \\ X_3(t+1) = \neg(X_2(t) \vee U_1(t)) \wedge U_2(t), \end{cases} \tag{5}$$

where $X_1, X_2, X_3 \in \mathcal{D}$ are the state nodes expressing genes *Atrboh*, *Ros* and *ABL1*, respectively, and $U_1, U_2 \in \mathcal{D}$ are the input nodes. The output equations, as given in [21], are as follows:

$$\begin{cases} Y_1(t) = (X_1(t) \wedge (X_2(t) \to X_3(t))) \\ \qquad\quad \vee(\neg X_1(t) \wedge \neg X_2(t) \wedge X_3(t)), \\ Y_2(t) = (X_1(t) \wedge (X_2(t) \to X_3(t))) \\ \qquad\quad \vee(\neg X_1(t) \wedge \neg X_2(t)), \end{cases} \tag{6}$$

where $Y_1, Y_2 \in \mathcal{D}$.

Let $u_i = \varphi(U_i), x_j = \varphi(X_j)$ and $y_k = \varphi(Y_k)$, where $i, k \in \{1, 2\}, j \in \{1, 2, 3\}$. Let $u = u_1 \ltimes u_2$ and $x = \ltimes_{i=1}^3 x_i$. Using Lemma 1 and Lemma 2, the algebraic forms of (5) and (6) are obtained as follows:

$$\begin{cases} x(t+1) = Lu(t)x(t), \\ y_1(t) = H_1 x(t), \\ y_2(t) = H_2 x(t), \end{cases} \tag{7}$$

where

$$\begin{aligned} L = \delta_8[&6 \quad 2 \quad 6 \quad 2 \quad 6 \quad 2 \quad 6 \quad 2 \\ &8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \\ &6 \quad 6 \quad 5 \quad 5 \quad 6 \quad 6 \quad 5 \quad 5 \\ &8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8], \\ H_1 = \delta_2[&1 \quad 2 \quad 1 \quad 1 \quad 2 \quad 2 \quad 1 \quad 2], \\ H_2 = \delta_2[&1 \quad 2 \quad 1 \quad 1 \quad 2 \quad 2 \quad 1 \quad 1]. \end{aligned}$$

To begin with, we use Algorithm 1 to construct matrices F_1 and F_2. The initial matrices are

$$\begin{aligned} F_1 = \delta_2[&2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \\ &2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2], \\ F_2 = \delta_2[&2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \quad 2 \\ &1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]. \end{aligned}$$

The initial state subsets are $\Gamma_{11} = \{\delta_8^1, \delta_8^3, \delta_8^4, \delta_8^7\}$, $\Gamma_{12} = \{\delta_8^2, \delta_8^5, \delta_8^6, \delta_8^8\}, \Gamma_{21} = \{\delta_8^1, \delta_8^3, \delta_8^4, \delta_8^7, \delta_8^8\}$ and $\Gamma_{22} = \{\delta_8^2, \delta_8^5, \delta_8^6\}$. For any two states $\delta_8^{\alpha_1}, \delta_8^{\alpha_2} \in \Delta_8$, $\alpha_1 \neq \alpha_2$, one has

$$\begin{aligned} F_1 W_{[8,2]} \delta_8^{\alpha_1} = F_1 W_{[8,2]} \delta_8^{\alpha_2} = \delta_2[2 \ 2], \\ F_2 W_{[8,2]} \delta_8^{\alpha_1} = F_2 W_{[8,2]} \delta_8^{\alpha_2} = \delta_2[2 \ 1]. \end{aligned}$$

Thus, all state subsets $\Gamma_{11}, \Gamma_{12}, \Gamma_{21}, \Gamma_{22}$ are not partitioned after the first while loop. As a result, the matrices F_1, F_2 obtained after the completion of Algorithm 1 are the initial matrices. Due to $F_i \in \mathcal{L}_{2\times 16}, i = 1, 2$, BCN (7) is IO-decoupled as per Theorem 1.

V. CONCLUSION

Two necessary and sufficient conditions have been proposed to determine whether a given BCN is IO-decoupled. These conditions rely on the nature of some IOD matrices constructed (via Algorithm 1) from the system's output information. The time and space complexities of the proposed approach are lower than the those of the existing methods (that we are aware of) in the literature. Moreover, the proposed approach can be used to solve other problems such the disturbance decoupling. It is worth pointing out that the authors in [22], [23] dealt with the IO-decoupling problem via state feedback, relying on the existence of the IO decomposed form. However, the authors in [21] proved that an IO-decoupled BCN may not necessarily have an IO-decomposed form. Therefore, the IO-decoupling problem via state feedback for BCNs needs to be investigated further. This will be part of our future research investigations. It is undeniable that the computational complexity of the proposed approach is still large, limiting the potential application of the present work to large-scale BCNs. References [16]–[18] provided some excellent work related to large-scale BCNs, which might be helpful for our future research work on IO-decoupling of large-scale BCNs.

REFERENCES

[1] S. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[2] T. Ideker, T. Galiski, and L. Hood, "A new approach to decoding life: Systems biology," *Annual Review of Genomics and Human Genetics*, vol. 2, no. 1, pp. 343–372, 2001.

[3] S. Huang and D. Ingber, "Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks," *Experimental Cell Research*, vol. 261, no. 1, pp. 91–103, 2000.

[4] D. Cheng, "Semi-tensor product of matrices and its application to Morgan's problem," *Science in China Series F: Information Sciences*, vol. 44, no. 3, pp. 195–212, 2001.

[5] Y. Yu, M. Meng, J. Feng, and G. Chen, "Observability criteria of Boolean networks," *IEEE Transactions on Automatic Control*, vol. 67, no. 11, pp. 6248–6254, 2022.

[6] K. Zhang, "Synthesis for observability of logical control networks," *Automatica*, vol. 144, pp. 110 481:1–110 481:9, 2022.

[7] Y. Liu, J. Zhong, D. Ho, and W. Gui, "Minimal observability of Boolean networks," *Science China Information Sciences*, vol. 65, no. 5, pp. 152 203:1–152 203:12, 2021.

[8] R. Zhou, Y. Guo, X. Liu, and W. Gui, "Stabilization of Boolean control networks with state-triggered impulses," *Science China Information Sciences*, vol. 65, pp. 132 202:1–132 202:15, 2022.

[9] H. Li, X. Yang, and S. Wang, "Robustness for stability and stabilization of Boolean networks with stochastic function," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1231–1237, 2021.

[10] S. Zhu, J. Lu, J. Zhong, Y. Liu, and J. Cao, "Sensors design for large Boolean networks via pinning observability," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4162–4169, 2022.

[11] Y. Yao and J. Sun, "Optimal control of multi-task Boolean control networks via temporal logic," *Systems & Control Letters*, vol. 156, pp. 105 007:1–105 007:9, 2021.

[12] M. Meng, G. Xiao, and D. Cheng, "Self-triggered scheduling for Boolean control networks," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 8911–8921, 2022.

[13] J. Zhong, Z. Yu, Y. Li, and J. Lu, "State estimation for probabilistic Boolean networks via outputs observation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4699–4711, 2021.

[14] R. Li, Q. Zhang, and T. Chu, "Bisimulations of probabilistic Boolean networks," *SIAM Journal on Control and Optimization*, vol. 60, no. 5, pp. 2631–2657, 2022.

[15] F. Li and Y. Tang, "Multi-sensor fusion Boolean Bayesian filtering for stochastic Boolean networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[16] S. Zhu, J. Lu, L. Lin, and Y. Liu, "Minimum-time and minimum-triggering observability of stochastic Boolean networks," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1558–1565, 2022.

[17] S. Zhu, J. Lu, S. Azuma, and W. Zheng, "Strong structural controllability of Boolean netwoeks: Polynomial-time criteria, minimal node control, and distributed pinning strategies," *IEEE Transactions on Automatic Control*, 2022.

[18] S. Zhu, J. Cao, L. Lin, J. Lam, and S. Azuma, "Towards stabilizable large-scale Boolean networks by controlling the minimal set of nodes," *IEEE Transactions on Automatic Control*, 2023.

[19] B. Morgan, "The synthesis of linear multivariable systems by state-variable feedback," *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pp. 405–411, 1964.

[20] M. Valcher, "Input/output decoupling of Boolean control networks," *IET Control Theory & Applications*, vol. 11, no. 13, pp. 2081–2088, 2017.

[21] Y. Li and J. Zhu, "Necessary and sufficient vertex partition conditions for input-output decoupling of Boolean control networks," *Automatica*, vol. 137, pp. 110 097:1–110 097:8, 2022.

[22] S. Fu, Y. Wang, D. Cheng, and J. Liu, "Morgan's problem of Boolean control networks," *Control Theory and Technology*, vol. 15, no. 4, pp. 316–326, 2017.

[23] J. Pan, J. Feng, J. Yao, and J. Zhao, "Input-output decoupling of Boolean control networks," *Asian Journal of Control*, vol. 20, no. 6, pp. 2185–2194, 2018.

[24] D. Cheng, H. Qi, and Y. Zhao, *An Introduction to Semi-tensor Product of Matrices and Its Application*. Singapore: World Scientific, 2011.

[25] S. Li, S. Assmann, and R. Albert, "Predicting essential components of signal transduction networks: A dynamic model of guard cell abscisic acid signaling," *PLOS Biology*, vol. 4, no. 10, pp. e312:1–e312:17, 2006.