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

# Robust Set Stabilization of Boolean Control Networks: an Efficient Approach based on Reverse Set Propagation

Shuhua Gao, Qian Wang, Yakun Li, Zezheng Li, and Cheng Xiang*

*Abstract*— **This paper investigates the robust set stabilization of nondeterministic Boolean control networks (BCNs) subject to random disturbance inputs. Although this problem has been previously addressed in the literature, we propose an alternative approach primarily to decrease the computational complexity of the algorithms. Our technique is inspired by the set propagation technique in reachability analysis but is applied in reverse order, identifying all the layered sets of states that reach a target set in a specific order. Two algorithms are developed: the first determines the largest robust control invariant subset, while the second handles time-optimal robust set stabilization using the results from the first algorithm. In particular, all time-invariant state feedback gain matrices are identified. Our approach achieves the lowest computational complexity ever known, even lower than the current methods designed solely for deterministic set stabilization without any disturbances. Numerical simulations with two biological networks demonstrate the significantly reduced processing time of our algorithms. Overall, this study presents a new approach for robust set stabilization with improved efficiency, capable of handling relatively large BCNs beyond the capabilities of existing techniques.**

## I. INTRODUCTION

A Boolean network (BN) is a discrete-time logical system model initially proposed to describe gene regulatory networks (GRNs). In a BN, the state of each node is binary and updated via Boolean interaction with each other. A BN with external binary inputs is called a *Boolean control network* (BCN). The control-theoretical study on BCNs has been booming in recent years mainly thanks to a novel mathematical tool called the semi-tensor product (STP) of matrices [1]. In this study, we focus on a canonical control-theoretical problem under uncertainties [2], namely robust set stabilization of BCNs under external disturbance inputs.

Set stabilization aims to steer a BCN into and keep it inside a given target set of states, termed $\mathcal{Z}$, by a proper control law [3]–[5]. Apparently, it turns into the normal stabilization of a BCN if $\mathcal{Z}$ contains only a single state [6], [7]. In the theoretical aspect, some important control problems like output tracking control [8], partial stability and stabilization [3], and network synchronization [9] can be recast into set stabilization problems. In the application aspect, set stabilization of BCNs may help design therapeutic intervention strategies [4], [10], [11].

A real-world system is always affected by disturbances. Compared with the well-studied robust stabilization of BCNs, there are considerably fewer studies on robust set stabilization due to its increased complexity and difficulties. A pioneering study characterized robust control invariance of a BCN in [12] but did not investigate its stabilization. In [6], the global robust stability of a BN was first investigated, and a state feedback pinning control strategy was proposed for robust stabilization to a fixed point or a limit cycle. Later, [13] proposed an event-triggered control (ETC) strategy for robust set stabilization. Still, the state feedback gain designed therein is time-variant, and the target set is assumed to be a robust control invariant one. A follow-up work studied ETC for time-optimal robust set stabilization [14]. Two more recent studies targeted probabilistic BCNs (PBCNs): the former investigated robust control invariance [15]; and the latter developed time-invariant state feedback control for robust set stabilization [16], respectively.

One common challenge in controlling BCNs is the high computational burden caused by their exponentially large state and control space [5], [17]. As a result, control problems related to BCNs are NP-hard in general [18]. This justifies the exponential time complexity of all robust set stabilization methods reviewed in this article. Although the NP-hardness makes it impossible to design polynomial-time algorithms, unless P=NP [19], it does not necessarily mean that we cannot improve upon existing methods. We consider a BCN with $m$ control inputs, $n$ state variables, and $q$ disturbance inputs. We define $N := 2^n$, $M := 2^m$, and $Q := 2^q$. Usually, the big-O time complexity of existing methods is generally a polynomial of $Q, M$ and $N$, such as $O(QMN^3)$ in [16, Algorithm 4.8]. We may naturally wonder if it is possible to decrease the degree of such a polynomial.

Existing robust set stabilization studies typically depend on computationally expensive matrix algebra operations like large matrix multiplications, resulting in excessively high time complexity. In view of this challenge, we develop an alternative method from an algorithmic perspective instead, which contains as few matrix multiplications as possible. Our methodology is mainly inspired by the set propagation technique in classical control theory, which computes the *destination* set of states reachable by a dynamic system from a given *source* set of states [20]. In contrast to the standard set propagation, we compute the source set for

Shuhua Gao and Zezheng Li are with School of Control Science and Engineering, Shandong University, Jinan 250100, China shuhuagao@sdu.edu.cn

Qian Wang is with Shandong Huate Magnet Technology Co.,Ltd., Weifang 262600, China qianwang@chinahuate.com

Yakun Li is with School of Mathematics, Shandong University, Jinan 250100, China liyakunmath@mail.sdu.edu.cn

Cheng Xiang (*corresponding author*) is with Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 elexc@nus.edu.sg

This work was supported by National Natural Science Foundation of China under Grant 62303277, Young Experts of Taishan Scholar Project (NO. tsqn202306067) and the Natural Science Fund of Shandong Province under Grant 2023HWYQ-003 and ZR2023QF048.

a given destination set instead, which we call *reverse set propagation* (RSP). As we will show, though the RSP idea appears straightforward, it helps simplify the algorithms and enhance their time efficiency significantly through lowering the polynomial degree of the big-O time complexity.

The main contributions of this paper are:

1) We develop two novel algorithms based on RSP to compute the largest robust control invariant subset (LRCIS) and to achieve time-optimal robust set stabilization, respectively. All time-invariant state feedback gain matrices for the two problems are identified.
2) The proposed approach is distinguished by its superior efficiency, attaining time complexity considerably lower than existing methods. Numerical simulations with a medium-sized biological network demonstrate the substantially reduced execution time of our approach.

## II. Preliminaries

### A. Notations Related to STP

1) Given integers $k, n$, $[k, n] := \{k, k+1, \cdots, n\}$.
2) $\mathcal{M}_{m\times n}$ denotes the set of all $m \times n$ matrices. Given a matrix $A$, $\mathrm{Col}_i(A)$ and $\mathrm{Row}_j(A)$ denote its $i$-th column and $j$-th row, respectively.
3) Let $I_n$ denote the $n$-dim identity matrix. Let $\delta_n^i := \mathrm{Col}_i(I_n)$ be the $i$-th column of $I_n$ that contains $n-1$ 0's and a single 1. Define $\Delta_n := \{\delta_n^i | i = 1, 2, \cdots, n\}$. A shorthand for $\{\delta_n^{i_1}, \delta_n^{i_2}, \cdots, \delta_n^{i_k}\}$ is $\delta_n\{i_1, i_2, \cdots, i_k\}$.
4) A matrix $L \in \mathcal{M}_{n\times q}$ with $\mathrm{Col}_i(L) \in \Delta_n, \forall i \in [1, q]$, is called a *logical matrix*. Let $\mathcal{L}_{n\times q} \subset \mathcal{M}_{n\times q}$ be the set of all $n \times q$ logical matrices.
5) Given a matrix $M \in \mathcal{M}_{n\times q}$, if $q$ is a multiple of $k \in \mathbb{N}$, divide $M$ from left to right into $k$ blocks and denote the $i$-th block by $\mathrm{Blk}_i^k(M) \in \mathcal{M}_{n\times(q/k)}, i \in [1, k]$.

### B. Algebraic Representation of BCNs

To be consistent with the majority of existing studies, we adopt the algebraic representation of a BCN in this study. Nonetheless, our approach applies to the raw logical form of a BCN equally well since it does not depend on matrix operations. Consider a general BCN with $n$ state variables $x_i \in \Delta_2, 1 \le i \le n$, $m$ control inputs $u_j \in \Delta_2, 1 \le j \le m$, and $q$ disturbance inputs $\xi_k \in \Delta_2, 1 \le k \le q$. Its dynamics is described by $n$ Boolean functions, one for each state variable:

$$x_i(t+1) = f_i(x_1, \cdots, x_n, u_1, \cdots, u_m, \xi_1, \cdots, \xi_q),$$

where $f_i : \Delta_2^{m+n+q} \to \Delta_2$ is the $i$-th Boolean function.

The foundation of the algebraic representation for a logical system is the semi-tensor product (STP), a generalization of the normal matrix product [1]. The ASSR of the BCN above in the STP framework is

$$x(t+1) = L\xi(t)u(t)x(t), \tag{1}$$

where $L \in \mathcal{L}_{N\times MNQ}$ is called the state transition matrix; $x(t) := \ltimes_{i=1}^{n} x_i(t) \in \Delta_N$, $u(t) := \ltimes_{i=1}^{m} u_i(t) \in \Delta_M$, and $\xi(t) := \ltimes_{i=1}^{q} \xi_i(t) \in \Delta_Q$ denote the vector form of the network state, control, and disturbance, respectively. The derivation of Eq. (1) has been detailed in many papers like [1], [6], [14], [16] and is omitted here to save space.

## III. Problem Formulation

In view of the stochastic disturbances, we formulate the problems in a feedback sense [2], [15], [16]. To ease subsequent illustrations, let $\Pi := \{\pi : \Delta_N \to \Delta_M\}$ denote the set of all time-invariant state feedback control laws. Obviously, $u = \pi(x)$ for any $\pi \in \Pi$ can be written equivalently as $u = F_\pi x$, where $F_\pi \in \mathcal{L}_{M\times N}$ is a unique feedback gain matrix associated with $\pi$. We aim to find out all feasible state feedback gain matrices for time-optimal robust set stabilization. We first list some necessary definitions.

*Definition 1:* Given a set $\mathcal{Z} \subseteq \Delta_N$ and an initial state $x^0 \in \Delta_N$, BCN (1) is *robustly set stabilizable* to $\mathcal{Z}$ from $x^0$, if and only if there exists a state feedback control law $\pi \in \Pi$ and a finite integer $T$ such that $x(t) \in \mathcal{Z}, \forall t \ge T$ holds for all $\xi(t) \in \Delta_Q, \forall t \ge 0$.

*Definition 2:* Following Definition 1, let the smallest value of $T$ be $T_{\mathcal{Z},\pi}(x^0)$, called the *shortest transient period* attained by $\pi$ towards $\mathcal{Z}$. Let $T^*_{\mathcal{Z}}(x^0) := \min_{\pi\in\Pi} T_{\mathcal{Z},\pi}(x^0)$ denote the shortest transient period among all possible state feedback control laws.

Obviously, BCN (1) is robustly set stabilizable from $x^0$ to $\mathcal{Z}$ if and only if $T^*_{\mathcal{Z}}(x^0) < \infty$. The *domain of attraction* (DoA) of $\mathcal{Z}$ contains all initial states from which BCN (1) is robustly set stabilizable to $\mathcal{Z}$, denoted by

$$\Omega_{\mathcal{Z}} := \{x \in \Delta_N | T^*_{\mathcal{Z}}(x) < \infty\}. \tag{2}$$

Let $T^{**}_{\mathcal{Z}} := \max_{x\in\Omega_{\mathcal{Z}}} T^*_{\mathcal{Z}}(x)$ denote the maximum shortest transient period among all initial states in $\Omega_{\mathcal{Z}}$.

Two problems are investigated in this study.

*Problem 1:* Given a target set $\mathcal{Z} \subseteq \Delta_N$ and an initial state $x^0 \in \Delta_N$, check whether BCN (1) is robustly set stabilizable from $x^0$ to $\mathcal{Z}$ via time-invariant state feedback control.

*Problem 2:* Given a target set $\mathcal{Z} \subseteq \Delta_N$ and an initial state $x^0 \in \Delta_N$, compute $T^*_{\mathcal{Z}}(x^0)$ and find all time-invariant state feedback gain matrices that attain $T^*_{\mathcal{Z}}(x^0)$.

In essence, Problem 1 ascertains robust set stabilizability, and Problem 2 targets time-optimal robust set stabilization. It deserves a mention that the solution to Problem 2 also answers Problem 1, since Problem 1 is equivalent to confirming whether $T^*_{\mathcal{Z}}(x^0) < \infty$. Moreover, our algorithms are general and not limited to any specific initial state.

## IV. Main Results

### A. Calculation of the LRCIS

An important concept in (robust) set stabilization is (robust) control invariance [3], [12], which is stated below.

*Definition 3:* Consider BCN (1) and a target set $\mathcal{Z} \subseteq \Delta_N$. A set $\mathcal{R} \subseteq \mathcal{Z}$ is called a *robust control invariant subset* (RCIS) of $\mathcal{Z}$, if for any state $x \in \mathcal{R}$, there exists a control input $u \in \Delta_M$ such that the succeeding state is still in $\mathcal{R}$, i.e., $L\xi ux \in \mathcal{R}$, for any disturbance $\xi \in \Delta_Q$. Denote the largest RCIS (LRCIS for short) of $\mathcal{Z}$ by $I_{\mathrm{C}}(\mathcal{Z})$.

Clearly, the union of any two RCIS's forms another RCIS. The LRCIS is the union of all RCIS's of $\mathcal{Z}$. Computing the LRCIS is usually the first task in robust set stabilization due to the critical role it plays as follows (see [2], [3], [5], [16]).

*Lemma 1:* Given an integer $T$, $x(t) \in \mathcal{Z}, \forall t \geq T$ holds if and only if $x(t) \in I_{\mathrm{C}}(\mathcal{Z}), \forall t \geq T$.

Since it is always possible to keep BCN (1) inside $I_{\mathrm{C}}(\mathcal{Z})$ forever by its definition, Lemma 1 implies that it is equivalent to steering the BCN from the initial state to any state in $I_{\mathrm{C}}(\mathcal{Z})$ for robust set stabilization. Unlike existing methods that compute the LRCIS directly (see, e.g., [3], [5], [16], [21]), we approach this problem via *reverse thinking* by first identifying the set of states in $\mathcal{Z}$ that do *not* belong to $I_{\mathrm{C}}(\mathcal{Z})$ instead, termed $\bar{I}_{\mathrm{C}}(\mathcal{Z}) := \mathcal{Z} \setminus I_{\mathrm{C}}(\mathcal{Z})$. The states in $\bar{I}_{\mathrm{C}}(\mathcal{Z})$ have an interesting property opposite to the property of states in $I_{\mathrm{C}}(\mathcal{Z})$, which is stated below.

*Proposition 1:* Consider BCN (1), a target set $\mathcal{Z}$, and an initial state $x(0) = x^0 \in \mathcal{Z}$. We have $x^0 \in \bar{I}_{\mathrm{C}}(\mathcal{Z})$ if and only if there exists a disturbance sequence and a finite integer $k \in [1, \infty)$ such that $x(k) \notin \mathcal{Z}$ is true.

*Proof:* We give a straightforward proof by contradiction. (*Sufficiency*) Supposing that such a disturbance sequence does not exist, there must exist a state feedback control law that ensures $x(t) \in \mathcal{Z}, \forall t \geq 0$ for all possible disturbance sequences. Lemma 1 further says $x(t) \in I_{\mathrm{C}}(\mathcal{Z}), \forall t \geq 0$, which conflicts with the precondition that $x(0) \in \bar{I}_{\mathrm{C}}(\mathcal{Z})$. Thus, a contradiction is formed, and the sufficiency is proved.

(*Necessity*) Suppose for the contradiction purpose that we have $x^0 \in I_{\mathrm{C}}(\mathcal{Z})$ instead. Then, by Lemma 1, we can always find a state feedback control law such that $x(t) \in \mathcal{Z}, \forall t \geq 0$ regardless of disturbances, which negates the existence of a disturbance sequence satisfying $x(k) \notin \mathcal{Z}, k \in [1, \infty)$. Therefore, the assumption $x^0 \in I_{\mathrm{C}}(\mathcal{Z})$ must be wrong, i.e., the "only if" clause is true. The necessity is proved. ■

Proposition 1 says that BCN (1) will finally leave $\mathcal{Z}$ at a certain time instant from an initial state $x \in \bar{I}_{\mathrm{C}}(\mathcal{Z})$. This proposition lays the foundation for our RSP based algorithm to locate the $\bar{I}_{\mathrm{C}}(\mathcal{Z})$. Intuitively, we first set up the base case, i.e., collect states from which the BCN leaves $\mathcal{Z}$ in at least one time step no matter what control is applied, termed $\mathcal{D}_1$. Then, we find states from which the BCN leaves $\mathcal{Z}$ in at least two time steps, termed $\mathcal{D}_2$. An equivalent construction of $\mathcal{D}_2$ is to get states from which the BCN reaches $\mathcal{D}_1$ in at least one time step regardless of control. We continue this iterative process to find all relevant states. Formally, we proceed as follows and propagate states reversely from $\mathcal{D}_i$ to $\mathcal{D}_{i+1}$:

$$\mathcal{D}_1 := \{x \in \mathcal{Z} | L\xi ux \notin \mathcal{Z}, \forall u \in \Delta_M, \exists \xi \in \Delta_Q\}, \quad (3)$$

$$\mathcal{D}_{i+1} := \{x \in \mathcal{Z} \setminus \cup_{l=1}^{i} \mathcal{D}_l | L\xi ux \in \cup_{l=1}^{i} \mathcal{D}_l, \forall u \in \Delta_M, \exists \xi \in \Delta_Q\},\ i \geq 1, \quad (4)$$

where $\mathcal{D}_i$ denotes the set of states in $\bar{I}_{\mathrm{C}}(\mathcal{Z})$ from which the BCN can leave $\mathcal{Z}$ in at least $i$ steps regardless of control inputs. Obviously, we have $\bar{I}_{\mathrm{C}}(\mathcal{Z}) = \cup_{i=1}^{\infty} \mathcal{D}_i$ by Proposition 1.

The RSP based procedures instantiating the above idea for LRCIS calculation are presented in Algorithm 1, which calculates $\mathcal{D}_i$ one by one. Note that the state transition $L\xi ux$ in the algorithm can be quickly computed in $O(1)$ without using matrix products thanks to the special structure

**Algorithm 1:** Calculation of LRCIS

**Input:** BCN (1) and a target set $\mathcal{Z} \subseteq \Delta_N$
1 Compute $\mathcal{D}_1$ by Eq. (3) and initialize $\mathcal{D} \leftarrow \mathcal{D}_1$
2 $i \leftarrow 1$
3 **do**
4     Compute $\mathcal{D}_{i+1}$ by Eq. (4)
5     $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{i+1}$
6     $i \leftarrow i + 1$
7 **while** $\mathcal{D}_i \neq \emptyset$
8 **return** $\mathcal{Z} \setminus \mathcal{D}$

of logical vectors as follows,

$$L\delta_Q^i \delta_M^j \delta_N^k = \mathrm{Col}_k \left( \mathrm{Blk}_j^M \left( \mathrm{Blk}_i^Q (L) \right) \right). \quad (5)$$

*Theorem 1:* Consider Algorithm 1. We have

(1) Algorithm 1 terminates in at most $|\mathcal{Z}|$ iterations;
(2) $\bar{I}_{\mathrm{C}}(\mathcal{Z}) = \mathcal{D}$ and $I_{\mathrm{C}}(\mathcal{Z}) = \mathcal{Z} \setminus \mathcal{D}$.

*Proof:* (1) The size of $\mathcal{D}$ is increased at least by 1 at Line 5 unless $\mathcal{D}_{i+1}$ is empty. Recall that we have $|\mathcal{D}| \leq |\mathcal{Z}|$ because of $\mathcal{D} \subseteq \mathcal{Z}$. This fact means that $\mathcal{D} = \mathcal{Z}$ becomes true after the **do-while** loop runs at most $|\mathcal{Z}|$ times (if it has not terminated before that). According to how $\mathcal{D}$ is constructed in Algorithm 1, Eq. (4) at Line 4 is equivalent to

$$\mathcal{D}_{i+1} = \{x \in \mathcal{Z} \setminus \mathcal{D} | L\xi ux \in \mathcal{D}, \forall u \in \Delta_M, \exists \xi \in \Delta_Q\}.$$

In the case of $\mathcal{D} = \mathcal{Z}$, it is obvious that $\mathcal{D}_{i+1}$ is set empty at Line 4, and the loop condition fails immediately at Line 7. Therefore, Algorithm 1 terminates in at most $|\mathcal{Z}|$ iterations.

(2) Supposing that the **do-while** loop runs totally $H$ iterations in Algorithm 1, we have $\mathcal{D}_{H+1} = \emptyset$, and $\mathcal{D} = \cup_{i=1}^{H+1} \mathcal{D}_i$ by construction. Eq. (4) implies that $\mathcal{D}_i = \emptyset, \forall i \geq H + 1$. Thus, we have got $\mathcal{D} = \cup_{i=1}^{H+1} \mathcal{D}_i = \cup_{i=1}^{\infty} \mathcal{D}_i = \bar{I}_{\mathrm{C}}(\mathcal{Z})$. We further have $I_{\mathrm{C}}(\mathcal{Z}) = \mathcal{Z} \setminus \bar{I}_{\mathrm{C}}(\mathcal{Z}) = \mathcal{Z} \setminus \mathcal{D}$ by definition. ■

*Time complexity*: Computing Eq. (4) at Line 4 takes time $O(MQ|\mathcal{Z}|)$ in the worst case. Line 5 takes at most $O(|\mathcal{Z}|)$ because of $|\mathcal{D}_{i+1}| \leq |\mathcal{Z}|$. Since the **do-while** loop continues for no more than $|\mathcal{Z}|$ iterations, the worst-case time complexity of Algorithm 1 is $O(MQ|\mathcal{Z}|^2)$.

*Example 1:* Consider a tiny BCN for illustration purposes [16], which has three states, one control input, and one disturbance input ($\wedge$, conjunction; $\vee$, disjunction):

$$\begin{cases} x_1(t+1) = x_2(t) \vee x_3(t) \\ x_2(t+1) = x_1(t) \wedge u(t) \\ x_3(t+1) = u(t) \vee (\xi(t) \wedge x_1(t)) \end{cases}. \quad (6)$$

The transition matrix of the BCN in its ASSR form (1) is

$$L = \delta_8[1, 1, 1, 5, 3, 3, 3, 7, 3, 3, 3, 7, 4, 4, 4, 8, 1, 1, 1, 5, 3, 3, 3, 7, 4, 4, 4, 8, 4, 4, 4, 8]. \quad (7)$$

Given a set $\mathcal{Z} := \delta_8\{2, 4, 5, 7\}$, Algorithm 1 yields $\mathcal{D}_1 = \delta_8\{2\}$ and $\mathcal{D}_2 = \emptyset$. The correctness of $\delta_8^2$ is easily validated. Since there hold $L\delta_2^1\delta_2^1\delta_8^2 = \delta_8^3 \notin \mathcal{Z}$ and $L\delta_2^1\delta_2^2\delta_8^2 = \delta_8^3 \notin \mathcal{Z}$, there always exist a disturbance input $\delta_2^1$ that forces the BCN to leave the set $\mathcal{Z}$ for both control $\delta_2^1$ and $\delta_2^2$. The LRCIS is thus $I_{\mathrm{C}}(\mathcal{Z}) = \mathcal{Z} \setminus \{\delta_8^2\} = \delta_8\{4, 5, 7\}$.

**Algorithm 2:** Time-Optimal Robust Set Stabilization

```
Input: BCN (1) and a target set Z ⊆ Δ_N
1  Compute the LRCIS I_C(Z) by Algorithm 1
2  If I_C(Z) = ∅ return end                  // infeasible
3  R_0 ← I_C(Z), and R ← R_0
4  i ← 0
5  do
6  |  Compute R_{i+1} based on R_i by Eq. (10)
7  |  U_tr(x) ← U(x, R_i), ∀x ∈ R_{i+1} by computing Eq. (9)
8  |  R ← R ∪ R_{i+1}
9  |  i ← i + 1
10 while R_i ≠ ∅
11 return U_tr, R, R_0, R_1, ··· , R_{i−1}
```

### B. Time-Optimal Robust Set Stabilization

To begin with, we say that a state $x$ reaches a set $\mathcal{S} \subseteq \Delta_N$ *robustly* in one step if it satisfies

$$L\xi ux \in \mathcal{S},\ \exists u \in \Delta_M, \forall \xi \in \Delta_Q. \tag{8}$$

Collect all states that enable Eq. (8) into a set $\mathcal{U}(x, \mathcal{S})$:

$$\mathcal{U}(x, \mathcal{S}) \coloneqq \{u \in \Delta_M | L\xi ux \in \mathcal{S}, \forall \xi \in \Delta_Q\}. \tag{9}$$

After the LRCIS $I_{\mathrm{C}}(\mathcal{Z})$ is obtained above, we develop an algorithm for robust set stabilization via RSP again. Denote the set of initial states for which the shortest transient period is $i$ by $\mathcal{R}_i$. Lemma 1 tells $\mathcal{R}_0 = I_{\mathrm{C}}(\mathcal{Z})$. Moreover, we have the following equation by definition

$$\mathcal{R}_{i+1} \coloneqq \{x \in \Delta_N \setminus \cup_{l=0}^{i} \mathcal{R}_l | L\xi ux \in \cup_{l=0}^{i} \mathcal{R}_l, \exists u \in \Delta_M, \forall \xi \in \Delta_Q\},\ i \geq 0. \tag{10}$$

That is, any state in $\mathcal{R}_{i+1}$ must reach $\mathcal{R}_i$ robustly in exactly one time step. Applying the RSP principle again, we start from $\mathcal{R}_0$ and calculate each non-empty $\mathcal{R}_i, i \geq 1$ in Algorithm 2.

*Theorem 2:* Supposing that Algorithm 2 finishes in $K$ iterations (i.e., $i = K$ at the end), we have

(1) $K \leq N$

(2) $T^*_{\mathcal{Z}}(x) = k, \forall x \in \mathcal{R}_k, k \in [0, K-1]$, and $\Omega_{\mathcal{Z}} = \mathcal{R}$.

*Proof:* (1) Assume that $\mathcal{R}_0 \neq \emptyset$; otherwise, the algorithm terminates at Line 2 with $K = 0$. Clearly, at least one extra state is inserted into the set $\mathcal{R}$ at Line 8 in each iteration until $\mathcal{R}_{i+1} = \emptyset$. Consequently, we get $\mathcal{R} = \Delta_N$ after at most $N-1$ iterations if the algorithm does not stop before that. In the next iteration, we must have $\mathcal{R}_{i+1} = \emptyset$ at Line 6 because there holds $\mathcal{R}_{i+1} \subseteq \Delta_N \setminus \mathcal{R}$ by Eq. (10). The algorithm thus stops in this iteration according to the condition at Line 10. Therefore, we always have $K \leq N$.

(2) A straightforward proof can be given via mathematical induction. The base that $T^*_{\mathcal{Z}}(x) = 0, \forall x \in \mathcal{R}_0$ is obviously true by Lemma 1. Supposing that $T^*_{\mathcal{Z}}(x) = i, \forall x \in \mathcal{R}_i$ is true, we get $T^*_{\mathcal{Z}}(x) = i+1, \forall x \in \mathcal{R}_{i+1}$ by the construction of Eq. (10). Therefore, we have $T^*_{\mathcal{Z}}(x) = k, \forall x \in \mathcal{R}_k, k \geq 0$. Note additionally that Algorithm 2 yields $\mathcal{R}_K = \emptyset$, and Eq. (10) further tells us that $\mathcal{R}_k = \emptyset, \forall k \geq K+1$. This fact means that the maximum shortest transient period is $T^{**}_{\mathcal{Z}} = K-1$. Hence, the DoA $\Omega_{\mathcal{Z}}$ is computed by $\Omega_{\mathcal{Z}} = \cup_{k=0}^{K-1} \mathcal{R}_k = \cup_{k=0}^{K} \mathcal{R}_k = \mathcal{R}$. ■

Theorem 2 also answers Problem 1: BCN (1) is robustly set stabilizable from $x^0$ to $\mathcal{Z}$ if and only if $x^0 \in \Omega_{\mathcal{Z}}$.

*Time Complexity*: First, Algorithm 2 runs Algorithm 1 once at the beginning, which takes $O(MQ|\mathcal{Z}|^2)$ time. Then, we proceed to the **do-while** loop. In the $i$-th iteration, $0 \leq i \leq N-1$, there are at most $N-1-i$ states to check for the computing of Eq. (9) and Eq. (10) at the first two lines of the loop, which takes at most $O(MQ(N-1-i))$ time. The subsequent Line 8 appends at most $N-1-i$ states to $\mathcal{R}$ in $O(N-1-i)$ time. Thus, the $i$-th iteration takes totally $O(MQ(N-1-i))$ time in the worst case. The runtime of the whole **do-while** loop is no more than

$$\sum_{i=0}^{N-1} O(MQ(N-1-i)) = O(MQN^2). \tag{11}$$

The total worst-case time complexity of Algorithm 2 is thus $O(MQ|\mathcal{Z}|^2) + O(MQN^2)$, which is simplified to $O(MQN^2)$ because of $|\mathcal{Z}| \leq N$.

*Remark 1:* The concept of RSP in the above algorithms is comparable to that of [2] in dealing with continuous-valued linear uncertain systems. However, we leverage the discrete nature and finite size of a BCN's state and control space to develop exact algorithms that *converge* within a specified number of iterations. In contrast, the algorithm proposed in [2] applies linear programming at its core and only obtains an *approximate* solution with respect to a tolerance parameter.

### C. State Feedback Control

In this section, we characterize all feasible time-invariant state feedback gain matrices for time-optimal robust set stabilization. Robust set stabilization of a BCN is composed of two phases in general: first, steer the BCN from its initial state to any state in the LRCIS, called the *transient* phase; second, keep the state inside the LRCIS, named the *steady* phase. The control inputs associated with each state in the first phase have been identified by Algorithm 2 (i.e., the returned $\mathcal{U}_{\mathrm{tr}}$), which are generally not unique. The capability to keep the BCN's state inside the LRCIS is implied by Definition 3, which guarantees that $\mathcal{U}(x, I_{\mathrm{C}}(\mathcal{Z}))$ is not empty for any state $x$ in $I_{\mathrm{C}}(\mathcal{Z})$. Let

$$\mathcal{U}_{\mathrm{st}}(x) \coloneqq \mathcal{U}(x, I_{\mathrm{C}}(\mathcal{Z})),\ \forall x \in I_{\mathrm{C}}(\mathcal{Z}). \tag{12}$$

We have the following result.

*Theorem 3:* Consider BCN (1), a target set $\mathcal{Z}$, and an arbitrary initial state from which the BCN is robustly set stabilizable. Time-optimal robust set stabilization is attained with time-invariant state feedback control $u = Fx$ if and only if

$$\mathrm{Col}_i(F) \in \begin{cases} \mathcal{U}_{\mathrm{tr}}(x),\ \text{if } x = \delta_N^i \in \Omega_{\mathcal{Z}} \setminus I_{\mathrm{C}}(\mathcal{Z}) & \text{(13a)} \\ \mathcal{U}_{\mathrm{st}}(x),\ \text{if } x = \delta_N^i \in I_{\mathrm{C}}(\mathcal{Z}) & \text{(13b)} \\ \Delta_M,\ \text{otherwise,} & \text{(13c)} \end{cases}$$

where $\mathcal{U}_{\mathrm{tr}}$ is given by Algorithm 2, and $\mathcal{U}_{\mathrm{st}}$ is filled in (12).

*Proof:* Let the initial state be $x^0$. Since the BCN is robustly set stabilizable from $x^0$ to $\mathcal{Z}$, we have $x^0 \in \Omega_{\mathcal{Z}}$, or equivalently, $x^0 \in \cup_{k=0}^{K-1}\mathcal{R}_k$ based on Theorem 2, where $K-1$ indicates the maximum shortest transient period (see the proof of Theorem 2). Assume $x^0 = \delta_N^i$ without loss of generality. The feedback control injected at state $x^0$ is computed by $u^0 = Fx^0 = \mathrm{Col}_i(F)$.

(*Sufficiency*) In the case of $x^0 \in \mathcal{R}_0$, or equivalently, $x^0 \in I_{\mathrm{C}}(\mathcal{Z})$ by construction (see Algorithm 2), Eq. (13b) is activated, and $u^0 \in \mathcal{U}_{\mathrm{st}}(x^0)$. Recall the definition in Eq. (12). It is clear that Eq. (13b) will keep the BCN's state inside $I_{\mathrm{C}}(\mathcal{Z})$ forever, which yields $T_{\mathcal{Z}}(x^0) = 0 = T^*_{\mathcal{Z}}(x^0)$. Next, consider the case with $x^0 \in \mathcal{R}_k, k \in [1, K-1]$. At first, Eq. (13a) is activated for state $x^0$ with $u^0 \in \mathcal{U}_{\mathrm{tr}}(x^0)$. By the construction of $\mathcal{U}_{\mathrm{tr}}$ in Algorithm 2, we must have $x^1 = L\xi^0 u^0 x^0 \in \mathcal{R}_{k-1}, \forall \xi^0 \in \Delta_Q$. If we still have $k-1 \geq 1$, the same analysis can show that the state after $x^1$, say $x^2$, must belong to $\mathcal{R}_{k-2}$. Hence, the state trajectory starting from the initial state $x^0$ will finally arrive at a state in $\mathcal{R}_0$, say $x^k \in \mathcal{R}_0$, after $k$ steps steered by feedback control (13a). This forms indeed the transient phase. After that, we return to the first case again, and Eq. (13b) is used instead, which traps the state within $I_{\mathrm{C}}(\mathcal{Z})$ forever. Therefore, we have $T_{\mathcal{Z}}(x^0) = k = T^*_{\mathcal{Z}}(x^0)$ for $x^0 \in \mathcal{R}_k, k \in [1, K-1]$ (note that the second equality is due to Theorem 2). A state $x \notin \Omega_{\mathcal{Z}}$ will never be encountered by definition; otherwise, the BCN cannot get robustly set stabilized. Consequently, the control corresponding to such a state does not matter, which indicates that Eq. (13c) is sufficient. The proof of sufficiency is thus finished for an arbitrary initial state $x^0 \in \Omega_{\mathcal{Z}}$.

(*Necessity*) Consider the case with $x^0 \in I_{\mathrm{C}}(\mathcal{Z})$, i.e., $x^0 \in \mathcal{R}_0$. Since time-optimal robust set stabilization is required, all succeeding states must stay in $I_{\mathrm{C}}(\mathcal{Z})$ regardless of disturbances to make $T_{\mathcal{Z}}(x^0) = T^*_{\mathcal{Z}}(x^0) = 0$. Clearly, we must have $u^0 \in \mathcal{U}(x^0, I_{\mathrm{C}}(\mathcal{Z}))$, written also as $u^0 \in \mathcal{U}_{\mathrm{st}}(x^0)$ based on Eq. (12). This case corresponds to Eq. (13b). Secondly, we proceed to the case where $x^0 \in \mathcal{R}_k, k \in [1, K-1]$ holds. Theorem 2 states $T^*_{\mathcal{Z}}(x^0) = k$, which implies that $u^0$ must drive the state from $x^0$ to a state in $\mathcal{R}_{k-1}$ in exactly one time step without being affected by disturbances. This condition effectively requires $u^0 \in \mathcal{U}(x^0, \mathcal{R}_{k-1})$, or more concisely, $u^0 \in \mathcal{U}_{\mathrm{tr}}(x^0)$ (see $\mathcal{U}_{\mathrm{tr}}$ in Algorithm 2), which corresponds to Eq. (13a). Note that the initial state $x^0 \in \Omega_{\mathcal{Z}}$ is arbitrarily chosen in the above discussion, which justifies the necessity of Eq. (13a) and (13b). As for a state $x$ not contained in $\Omega_{\mathcal{Z}}$, Eq. (13c) is always true no matter what control is injected at $x$. (In fact, as we have discussed above, the control for a state $x \notin \Omega_{\mathcal{Z}}$ does not matter.) Thus, Eq. (13) is correct, and the proof of necessity is completed. ∎

*Example 2:* Continuing Example 1, we build the state feedback control law for time-optimal robust set stabilization. Algorithm 2 produces $\mathcal{R}_0 = \delta_8\{4,5,7\}$, $\mathcal{R}_1 = \delta_8\{6,8\}$, and $\mathcal{R}_2 = \emptyset$. The maximum shortest transient period is thus $T^{**}_{\mathcal{Z}} = 1$, and the DoA is $\Omega_{\mathcal{Z}} = \delta_8\{4,5,6,7,8\}$. All time-optimal state feedback gain matrices are given by

$$F = \delta_2[\{1,2\},\{1,2\},\{1,2\},\{1\},\{2\},\{2\},\{2\},\{1\}],$$

where each column can take any value in the corresponding set. Take the state $x = \delta_8^4 \in I_{\mathrm{C}}(\mathcal{Z})$ for an example. It is easy to find out $\mathcal{U}(x, I_{\mathrm{C}}(\mathcal{Z})) = \delta_8\{1\}$, which is consistent with the fourth column of the above feedback gain matrix.

TABLE I

TIME COMPLEXITY COMPARISON FOR SET STABILIZATION TASKS OF *deterministic* BCNS WITH NO DISTURBANCES.

| Task | Method | Time complexity |
|---|---|---|
| Compute the LRCIS | [3, Proposition 2] | $O(MN + \lvert\mathcal{Z}\rvert N^3)$ |
| | [21, ] | $O(M\lvert\mathcal{Z}\rvert + \lvert\mathcal{Z}\rvert^3)$ |
| | [5, Algorithm 2] | $O(M\lvert\mathcal{Z}\rvert + \lvert\mathcal{Z}\rvert^2)$ |
| | [16, Algorithm 3.7] | $O(M\lvert\mathcal{Z}\rvert^3)$ |
| | Algorithm 1 | $O(M\lvert\mathcal{Z}\rvert^2)$ |
| Check set stabilizability | [3, Proposition 5] | $O(MN + N^4)$ |
| | [5, Theorem 2] | $O(MN + \lvert\mathcal{Z}\rvert^2)$ |
| | [16, Theorem 4.10] | $O(MN^3)$ |
| | Algorithm 2 | $O(MN^2)$ |
| Time-optimal set stabilization | [3, Proposition 6] | $O(MN^3 + \lvert\mathcal{Z}\rvert N^3)$ |
| | [5, Theorem 3] | $O(MN + \lvert\mathcal{Z}\rvert^2)$ |
| | [16, Algorithm 4.8] | $O(MN^3)$ |
| | Algorithm 2 | $O(MN^2)$ |

## V. COMPARISON WITH EXISTING METHODS

Currently, most studies in the literature are devoted to *deterministic* set stabilization without taking disturbances into account, and relatively few studies focus on robust set stabilization via regular state feedback control. To perform a more comprehensive comparison, we take the deterministic BCN as a special case of the nondeterministic one by fixing $\Delta_Q = \{\delta_1^1\}$ in view of the fact that $L\delta_1^1 ux \equiv Lux$. That is, algorithms developed for robust set stabilization apply to deterministic set stabilization as well but not vice versa. We then list the time complexity of ours and existing methods for deterministic set stabilization in Table I. Three tasks have been investigated, though the last two tasks are solved simultaneously with Algorithm 2. Only our approach and Wang's method [16] are capable of robust set stabilization problems in particular. Interestingly, our approach even beats most existing methods for deterministic set stabilization except Gao's [5] in terms of time complexity, highlighting its outstanding efficiency advantage clearly. Note however that the method in our previous work [5] is a graphical one, which depends on the static state transition graph and cannot handle nondeterministic state transitions, prohibiting its extension to robust set stabilization problems.

Next, we move to robust set stabilization for BCNs subject to stochastic disturbances. The main competitor is the STP based algebraic method proposed in the recent study [16]. The time complexity comparison is listed in Table II for three tasks. In addition, we also measured the actual runtime of different algorithms for two biological networks and report the results in Table II. The *ara* operon network is characterized by $n = 9, m = 4, q = 2$ with a target set $\mathcal{Z}$ composed of 203 states. The T-LGL network is a bit large with $n = 16, m = 4, q = 3$, and its target set contains 1094 states. More details of the two networks are available in [5]. We first see that our algorithms enjoy lower time complexity than those in [16] in all three tasks. One notable

TABLE II

TIME COMPLEXITY AND RUNTIME COMPARISON FOR ROBUST SET STABILIZATION OF TWO BIOLOGICAL NETWORKS.

| Task | Method | Time complexity | Running time | |
|---|---|---|---|---|
| | | | *ara* operon | T-LGL |
| Compute the LRCIS | [16, Algorithm 3.7]<br>Algorithm 1 | $O(MQ\|\mathcal{Z}\|^3)$<br>$O(MQ\|\mathcal{Z}\|^2)$ | 1.93 s<br>6.4E-4 s | 269 s<br>3.1E-2 s |
| Check robust set stabilizability | [16, Theorem 4.10]<br>Algorithm 2 | $O(MQN^3)$<br>$O(MQN^2)$ | 2.26 s<br>4.2E-3 s | $>$ 24 hours<br>1.7 s |
| Time-optimal robust set stabilization | [16, Algorithm 4.8]<br>Algorithm 2 | $O(MQN^3)$<br>$O(MQN^2)$ | 2.26 s<br>4.2E-3 s | $>$ 24 hours<br>1.7 s |

observation is that, though the degree of $N$ is lower only by 1 in the big-O time complexity of our approach, the running time difference in practice for a medium-sized T-LGL network can be astonishing, reduced from more than 24 hours to roughly only 1.7 s in Table II. This huge difference is consistent with the exponential nature of $N$ in view of $N = 2^n$. The success of the efficiency enhancement in our approach is mainly attributed to the removal of expensive matrix operations. Indeed, our RSP based algorithms only involve remarkably cheap computations in each step aside from their simple structures (as implied by the few number of lines in each algorithm).

## VI. CONCLUSIONS

In this study, we proposed a novel approach for time-optimal robust set stabilization of BCNs based on reverse set propagation (RSP). This approach first sets up a destination set of states and then locates other sets of states that can reach the destination set in order of the required number of time steps. All feasible state feedback gain matrices for time-optimal robust set stabilization have been constructed by our algorithms. Comprehensive comparison showed that our approach attained the lowest known time complexity and was thousands of times faster than its competitor when handling a medium-sized 16-node network. It deserves a final mention that, although our approach can manage *relatively* large BCNs beyond the capacity of existing methods, it still runs in exponential time in accordance with the NP-hardness of the problems. Consequently, our approach is still computationally intractable when the network has many nodes. A promising workaround is to combine the RSP technique with the network aggregation framework [22] for treatment of really large-scale BCNs of a special structure.

## REFERENCES


[1] D. Cheng and H. Qi, "A linear representation of dynamics of boolean networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2251–2258, 2010.

[2] F. Blanchini, "Minimum-time control for uncertain discrete-time linear systems," in *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, Dec. 1992, pp. 2629–2634 vol.3.

[3] Y. Guo, P. Wang, W. Gui, and C. Yang, "Set stability and set stabilization of Boolean control networks based on invariant subsets," *Automatica*, vol. 61, pp. 106–112, Nov. 2015.

[4] F. Li, H. Li, L. Xie, and Q. Zhou, "On stabilization and set stabilization of multivalued logical systems," *Automatica*, vol. 80, pp. 41–47, 2017.

[5] S. Gao, C. Xiang, and T. H. Lee, "Set Invariance and Optimal Set Stabilization of Boolean Control Networks: A Graphical Approach," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 400–412, Mar. 2021.

[6] J. Zhong, D. W. C. Ho, J. Lu, and W. Xu, "Global robust stability and stabilization of Boolean network with disturbances," *Automatica*, vol. 84, pp. 142–148, Oct. 2017.

[7] Q. Zhu and W. Lin, "Stabilizing Boolean networks by optimal event-triggered feedback control," *Systems & Control Letters*, vol. 126, pp. 40–47, Apr. 2019.

[8] X. Zhang, Y. Wang, and D. Cheng, "Output Tracking of Boolean Control Networks," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2730–2735, Jun. 2020.

[9] J. Zhong, J. Lu, Y. Liu, and J. Cao, "Synchronization in an Array of Output-Coupled Boolean Networks With Time Delay," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2288–2294, 2014.

[10] S. Sherekar and G. A. Viswanathan, "Boolean dynamic modeling of cancer signaling networks: Prognosis, progression, and therapeutics," *Computational and Systems Oncology*, vol. 1, no. 2, p. e1017, 2021.

[11] Q. Zhang, J.-e. Feng, Y. Zhao, and J. Zhao, "Stabilization and set stabilization of switched Boolean control networks via flipping mechanism," *Nonlinear Analysis: Hybrid Systems*, vol. 41, p. 101055, Aug. 2021.

[12] H. Li, L. Xie, and Y. Wang, "On robust control invariance of Boolean control networks," *Automatica*, vol. 68, pp. 392–396, Jun. 2016.

[13] Y. Li, H. Li, and W. Sun, "Event-triggered control for robust set stabilization of logical control networks," *Automatica*, vol. 95, pp. 556–560, Sep. 2018.

[14] S. Zhu and J. Lu, "Robust Set Stabilization of Boolean Control Networks: An Event-Triggered Control Approach," in *2019 Chinese Control Conference (CCC)*, Jul. 2019, pp. 683–686.

[15] L. Lin, J. Cao, and L. Rutkowski, "Robust Event-Triggered Control Invariance of Probabilistic Boolean Control Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 1060–1065, Mar. 2020.

[16] J. Wang, W. Liu, S. Fu, and J. Xia, "On robust set stability and set stabilization of probabilistic Boolean control networks," *Applied Mathematics and Computation*, vol. 422, p. 126992, Jun. 2022.

[17] Y. Yu, J.-e. Feng, J. Pan, and D. Cheng, "Block Decoupling of Boolean Control Networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3129–3140, 2019.

[18] T. Akutsu, M. Hayashida, W.-K. Ching, and M. K. Ng, "Control of Boolean networks: Hardness results and algorithms for tree structured networks," *Journal of Theoretical Biology*, vol. 244, no. 4, pp. 670–679, 2007.

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009.

[20] M. Althoff, G. Frehse, and A. Girard, "Set Propagation Techniques for Reachability Analysis," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 369–395, 2021.

[21] R. Liu, J. Lu, W. X. Zheng, and J. Kurths, "Output feedback control for set stabilization of Boolean control networks," *IEEE transactions on neural networks and learning systems*, 2019.

[22] Y. Zhao, B. K. Ghosh, and D. Cheng, "Control of Large-Scale Boolean Networks via Network Aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 7, pp. 1527–1536, Jul. 2016.