Stability and Stabilization in Probability of Probabilistic Boolean Networks

Chi Huang, Jianquan Lu, *Senior Member, IEEE*, Guisheng Zhai, *Senior Member, IEEE*, Jinde Cao, *Fellow, IEEE*, Guoping Lu, and Matjaž Perc

***Abstract*— This article studies the stability in probability of probabilistic Boolean networks and stabilization in the probability of probabilistic Boolean control networks. To simulate more realistic cellular systems, the probability of stability/stabilization is not required to be a strict one. In this situation, the target state is indefinite to have a probability of transferring to itself. Thus, it is a challenging extension of the traditional probability-one problem, in which the self-transfer probability of the target state must be one. Some necessary and sufficient conditions are proposed via the semitensor product of matrices. Illustrative examples are also given to show the effectiveness of the derived results.**

***Index Terms*— Probabilistic Boolean network (PBN), semitensor product (STP), stability/stabilization in probability, state feedback control.**

I. Introduction

With the rapid development of biomolecular technologies, the research of complex regulatory systems has been a hot topic in recent years. To obtain a comprehensive understanding of these biological systems, mathematical models have become an efficient tool. Some static and dynamic methods, such as the Bayesian network [1], Boolean logic [2], [3], and differential equations [4], have been proposed. In addition to these approaches, special attention has been drawn to Boolean networks (BNs).

A BN is a sequential dynamical network, in which time and states are discrete. It was first introduced by Kauffman [5] to describe the intricate internal dynamics of gene-regulatory systems. In BNs, each component can be either active or inactive to represent a label "expressed" or "not expressed" to an individual gene. The highly structured interactions of genes can be described by Boolean functions, which determine the state of each gene by its neighboring genes using logical rules. Numerous works have shown that BNs are a powerful technique to investigate genetic networks [6]. Lately, the semitensor product (STP) of matrix was introduced [7] and became popular due to its remarkable effectiveness in the study of BNs, such as the topological structure [8], controllability [9], [10], and synchronization [11], [12]. The stability and stabilization, as important control-related issues, are studied via the STP method as well. In [13], the stability and stabilization of a fixed state of Boolean control networks (BCNs) were investigated. Later, the state feedback stabilization [14], output feedback stabilization [15], and set stabilization [16], [17] for a collection of BCNs were also discussed. Recently, Lyapunov stability theory, known as a powerful technique for nonlinear systems, has been introduced for the study of BNs. In [18] and [19], a new framework of the Lyapunov theory was established for BNs and BCNs. Thanks to it, the (control) Lyapunov function can be built to propose stability conditions.

In spite of the great success in applications, the obvious limitation of traditional BNs is their intrinsic determinism. However, due to the inescapable noises and much smaller number of samples compared with that of parameters to be inferred, it is irrational to assume that the state of a gene is determined by only one logical rule. The Boolean functions may have different modes to reflect the actual regulatory effects [20]. Consequently, probabilistic BNs (PBNs) and probabilistic BCNs (PBCNs) have gained widespread attention both in the academic literature and in practice. In [20], the PBN was first introduced to deal with the uncertainty. From then on, substantial efforts have been devoted to the research of PBNs and PBCNs, such as controllability and stabilizability [21]–[23], optimal control [24], [25], synchronization [26], [27], and some other applications [28], [29].

So far, most existing research (see [30]–[32] and references therein) of PBNs and PBCNs have focused on stability/stabilization with probability one. Unfortunately, deterministic events with probability one are relatively subtle in biological systems, due to the inherent randomness of gene expression [33]. For transcriptional regulatory networks, it has been repeatedly observed that a high proportion of globular proteins are neither highly stable nor unstable [34]. The protein

Manuscript received February 25, 2019; revised August 12, 2019 and November 28, 2019; accepted February 27, 2020. Date of publication March 24, 2020; date of current version January 5, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61603268, Grant 61573102, Grant 61272530, and Grant 61573096, in part by the Fundamental Research Funds for the Central Universities under Grant JBK190502, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20170019, and in part by Slovenian Research Agency under Grant J4-9302, Grant J1-9112, and Grant P1-0403. *(Corresponding author: Chi Huang.)*

Chi Huang is with the School of Economic Information and Engineering, Southwestern University of Finance and Economics, Chengdu 610074, China, and also with the School of Mathematics, Southeast University, Nanjing 210096, China (e-mail: huangchi@swufe.edu.cn).

Jianquan Lu and Jinde Cao are with the School of Mathematics, Southeast University, Nanjing 210096, China.

Guisheng Zhai is with the Department of Mathematical Sciences, Shibaura Institute of Technology, Saitama 337-8570, Japan.

Guoping Lu is with the School of Electrical Engineering, Nantong University, Nantong 226000, China.

Matjaž Perc is with the Faculty of Natural Sciences and Mathematics, University of Maribor, 2000 Maribor, Slovenia, also with the Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, Taiwan 40402, China, and also with the Complexity Science Hub Vienna, 1080 Vienna, Austria.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2020.2978345

concentration controlled by similar genes fluctuates greatly depending on exact conditions. It indicates that the transcriptional process can randomize expression rates, which implies that predicting cell dynamics in a deterministic manner is very unlikely [35]. For example, the apoptotic pathway, which allows an organism to remove damaged or unwanted cells, can be activated by binding TNFα to death receptor TNFR1. In the absence of tumor necrosis factor (TNF), the cell can be bistable at two distinct states: survival and initiating apoptosis [36]. However, the decision of one or the other state mainly depends on the initial conditions, which randomly vary from cell to cell. Thus, all the cells are very unlikely to stabilize at either state with a probability of 100%. It is reasonable to compute the probability of convergence to each steady state [37]. Another fact is that genes have different impacts on the predictor. A significant task is to distinguish which genes have a major impact. It is highly desirable to calculate the probability of a gene or the joint probability of several selected genes which will be expressed in the long run [38]. As for an example of PBCNs, the function of anticancer drugs is to break the originally stable cellular state of a tumor cell and enforce it into the apoptosis or back into the differentiation state by intervening with some other genes. Both results are probable. Neither of them can be achieved definitely [39]. Therefore, it is extremely desirable to consider a more general likelihood of the dynamics of PBNs and PBCNs, instead of absolute determinism. This is the main purpose of this article to study stability/stabilization in probability, which has been greatly overlooked despite its wide practical uses in various fields, such as predicting the possibility of a spreading contagion in a financial crisis [40] and modeling manufacturing systems to improve design reliability and system resilience [41].

Stability in probability was first introduced for the stochastic systems with continuous value [42], and has been intensively studied for delayed systems [43], discrete-time systems [44], and so on. Recently, this concept has been extended to BNs. In [45], asymptotic stability in probability was investigated for stochastic BNs, based on some Lyapunov functions. As for PBNs, the synchronization in probability was first introduced in [46] for master–slave PBNs to deal with the situation that there exists one possible trajectory of the slave PBN coinciding with the trajectory of the master BN. After that, some other works, such as [26] and [47], were devoted to synchronization in probability for realization-dependent PBNs and asynchronous PBNs, respectively. However, these works just discussed the synchronization problem for PBNs with special structures. Is it possible to investigate stability in probability for more general PBNs? Besides, these researches handled the in-probability problem strikingly similar to the probability-one problem. Is the former one just a trivial extension of the latter one, or can we point out their essential differences and give more accurate characterization? Moreover, these works did not involve the control issue. Thus, how can we design a state-feedback controller for a PBN, when it cannot be stabilized in probability by itself?

Motivated by the above-mentioned challenges, in this article, we aim to study the stability and stabilization in probability for PBNs and PBCNs. For both problems, the whole network is needed to stabilize to a target state with a positive probability, instead of a strict one. The addressed problems cannot be seemed as a trivial extension of stability/stabilization with probability one. The main difficulty lies in the fact that even if the target state has zero probability of transferring to itself, the network still has a chance to stabilize at the target state. It is quite different from the probability-one problem, in which the target state is required to transfer to itself with probability one. Thus, the indeterminate process becomes a definite one, which makes the probability-one problem much easier. To overcome this difficulty, we construct a sequence of reachable sets from the target state according to probability transferring. Although the inclusion relation may not exist at the beginning of the sequence, we show that it must start from some point providing that stability in probability can be achieved.

Overall, the novelties of this article can be summarized as the following four aspects: 1) the stability in probability is first studied for PBNs. Some necessary and sufficient conditions are given. 2) Compared with the probability-one problem, the difficulty of the in-probability problem is discussed. That is, the target state may be impossible to transfer to itself. If such transferring is possible, a more feasible condition can be proposed. 3) The state-feedback controller is designed. Thanks to it, some necessary and sufficient conditions are derived to ensure the stabilization in the probability of PBCNs. Similarly, if the target state is possible to transfer to itself, a more efficient method can be proposed to design the controller. 4) Two real examples are used to verify our theory. The coincidence between simulation and the observed results indicates the effectiveness.

II. Problem Formulation and Preliminaries

A. Notations and Definitions

$\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}_{m\times n}$ is the set of all $m\times n$ real matrices, $|A|$ represents the cardinal number of the set A, I_n represents the $n\times n$ identity matrix, $\mathbf{1}_n$ is an n dimensional vector with all elements 1, and $\mathscr{D} := \{0,1\}$, $\Delta_n := \{\delta_n^k : k=1,\ldots,n\}$, where δ_n^k denotes the kth column of the identity matrix I_n. An $n\times m$ matrix M is called a logical matrix, if $M = [\delta_n^{i_1}\ \delta_n^{i_2},\ \ldots,\ \delta_n^{i_m}]$, which can be briefly denoted as $M = \delta_n[i_1\ i_2,\ \ldots,\ i_m]$; $\mathscr{L}_{n\times m}$ represents the set of all $n\times m$ logical matrices; $\mathscr{B}_{n\times m}$ represents the set of $n\times m$ Boolean matrices, i.e., all of the matrices $X=(x_{ij})$ with $x_{ij}\in\mathscr{D}$. Assume that $X=(x_{ij})$ and $Y_{ij}=(y_{ij})\in\mathscr{B}_{m\times n}$. Then, $X\wedge Y := (x_{ij}\wedge y_{ij})$, and the Boolean addition is defined as $X +_{\mathscr{B}} Y := (x_{ij}\vee y_{ij})$; $(\mathscr{B})\sum_{i=1}^{n} X_i = X_1 +_{\mathscr{B}} X_2 +_{\mathscr{B}} \cdots +_{\mathscr{B}} X_n\ \forall X_i\in\mathscr{B}_{m\times n}$. Supposing $X=(x_{ij})\in\mathscr{B}_{n\times m}$ and $Y=(y_{ij})\in\mathscr{B}_{m\times p}$, the Boolean product is defined as $X\times_{\mathscr{B}} Y = (\bigvee_{l=1}^{m} x_{il}\wedge y_{lj})\in\mathscr{B}_{n\times p}$. $\wedge$ and $\vee$ are the symbols of logic operation AND and OR, respectively. For any $\mathbf{F}\in\mathscr{B}_{m\times n}$, a logical matrix $F\in\mathscr{L}_{m\times n}$ is called a logical submatrix of $\mathbf{F}$, if $F\wedge\mathbf{F}=F$; $\mathscr{T}(\mathbf{F})$ denotes the set of all of the logical submatrices of $\mathbf{F}$, i.e., $\mathscr{T}(\mathbf{F}) := \{F\in\mathscr{L}_{m\times n} | F\wedge\mathbf{F}=F\}$. For convenience, define $\mathscr{T}^{\top}(x) := \mathscr{T}(x^{\top})$ for any $x\in\mathscr{B}_{1\times n}$. $Col_i(A)$ denotes the ith column of the matrix A, $P\{\cdot\}$ is the probability of a random variable, and $P\{A|B\}$ is the conditional probability of event A, under the condition that event B is occurred.

Definition 1 [7]: Let $A \in \mathbb{R}_{m\times n}$ and $B \in \mathbb{R}_{p\times q}$. Then, the STP of A and B is defined as

$$A \ltimes B \triangleq (A \otimes I_{t/n})(B \otimes I_{t/p})$$

where $t = \operatorname{lcm}(n, p)$ is the least common multiplier of n and p; $\otimes$ is the Kronecker product of matrices.

Remark 1: When $n = p$, $A \ltimes B = (A \otimes I_1)(B \otimes I_1) = AB$. Thus, the STP degenerates to the conventional matrix product AB. The symbol $\ltimes$ may be omitted without causing confusion. Similarly, the Boolean STP can be defined as $A \ltimes_{\mathscr{B}} B := (A \otimes I_{\frac{t}{n}}) \times_{\mathscr{B}} (B \otimes I_{\frac{t}{p}})$, where $t = 1\ \mathrm{cm}(n, p)$.

By identifying $1 \sim \delta_2^1$ and $0 \sim \delta_2^2$, we have $\Delta \sim \mathscr{D}$, where "$\sim$" denotes two different forms of the same object. Usually, we use δ_2^1 and δ_2^2 to express the two logical values and call them the vector form of logical variables.

Lemma 1 [7]: Let $Y = f(X)$: $\mathscr{D}^m \to \mathscr{D}^n$ be a logical function, where $X = (x_1, x_2, \dots, x_m) \in \mathscr{D}^m$ and $Y = (y_1, y_2, \dots, y_n) \in \mathscr{D}^n$. Then, there exists a unique matrix $M_f \in \mathscr{L}_{2^n\times 2^m}$, called the structural matrix of f, such that

$$\ltimes_{i=1}^n y_i = M_f \ltimes_{j=1}^m x_j, \quad x_j, y_i \in \mathscr{D}. \tag{1}$$

Lemma 2: For $\forall a \in \mathscr{B}_{2^n\times 1}$ and $\forall b \in \mathscr{L}_{2^n\times 1}$, $a^\top b = 0$ if and only if $b \in \Delta_{2^n}\backslash \mathscr{T}(a)$.

Proof: $b \in \Delta_{2^n}\backslash\mathscr{T}(a) \Leftrightarrow b \in \mathscr{T}(\mathbf{1}_{2^n} - a) \Leftrightarrow (\mathbf{1}_{2^n} - a)^\top b = 1 \Leftrightarrow 1 - a^\top b = 1 \Leftrightarrow a^\top b = 0$. □

B. Algebraic Forms of PBNs and PBCNs

Consider the following PBN with n nodes:

$$X(t + 1) = f(X(t)), \quad t = 1, 2, \dots \tag{2}$$

where $X(t) = [x_1(t), x_2(t), \dots, x_n(t)]^\top \in \mathscr{D}^n$ is the states of the PBN, f: $\mathscr{D}^n \to \mathscr{D}^n$ is chosen from the set $\{f_1, f_2, \dots, f_r\}$ at every time step, and $P\{f = f_i\} = p_i > 0$, where f_i: $\mathscr{D}^n \to \mathscr{D}^n$, $i = 1, 2, \dots, r$ are given logical functions, and $\sum_{i=1}^r p_i = 1$.

Using the vector form of logical variables, we set $x(t) = \ltimes_{i=1}^n x_i(t) \in \Delta_{2^n}$. By Lemma 1, one can obtain the structural matrices of f_i $(i = 1, 2, \dots, r)$ as $L_i \in \mathscr{L}_{2^n\times 2^n}$ $(i = 1, 2, \dots, r)$. Then, (2) can be converted into

$$x(t + 1) = Lx(t) \tag{3}$$

where $L \in \mathscr{L}_{2^n\times 2^n}$ is chosen from the set $\{L_1, L_2, \dots, L_r\}$ at every time step with $P\{L = L_i\} = p_i$. Similarly, a PBCN with n state nodes and m controllers can be expressed as

$$X(t + 1) = f(X(t), U(t)) \tag{4}$$

where $U(t) = [u_1(t), u_2(t), \dots, u_m(t)]^\top$ is the state feedback control input of the PBCN with the form of $U(t) = g(X(t))$; g: $\mathscr{D}^n \to \mathscr{D}^m$ is the control function, which will be designed; f: $\mathscr{D}^{n+m} \to \mathscr{D}^n$ is randomly chosen from the set $\{f_1, f_2, \dots, f_r\}$ at every time step, and f_i: $\mathscr{D}^{n+m} \to \mathscr{D}^n$, $i = 1, 2, \dots, r$. Based on Lemma 1, the PBCN (4) can be rewritten as

$$x(t + 1) = Lu(t)x(t) \tag{5}$$

where $u := u_1 \ltimes u_2 \cdots \ltimes u_m \in \Delta_{2^m}$ can be given as $u(t) = Gx(t)$, $G \in \mathscr{L}_{2^m\times 2^n}$ is the structural matrix of the control function g, and $L \in \mathscr{L}_{2^n\times 2^{n+m}}$ is selected from the set $\{L_1, L_2, \dots, L_r\}$ with $P\{L = L_i\} = p_i$.

In this article, the stability and stabilization problems for PBNs (3) and PBCNs (5) will be studied in probability, instead of 100 %determinism. It is an important generalization of traditional stability and stabilization with probability one, which are much more difficult to be satisfied in real systems. More specifically, the concerned definitions are given follows.

Definition 2 (Stability in Probability): For a given target state $x^* \in \Delta_{2^n}$, the PBN (3) is said to be stable in probability, if there exists an integer $T > 0$ such that for any initial state $x_0 \in \Delta_{2^n}$

$$P\{x(t) = x^*|x(0) = x_0\} > 0 \tag{6}$$

holds for $\forall t \geq T$.

Definition 3 (Stabilization in probability): For a given reference state $x^* \in \Delta_{2^n}$, the PBCN (5) can realize stabilization in probability, if there exist an integer $T > 0$ and a controller in the form of $u(t) = g(x(t))$ such that for any initial state $x_0 \in \Delta_{2^n}$

$$P\{x(t) = x^*|x(0) = x_0, u(\tau) = g(x(\tau))\} > 0 \tag{7}$$

holds for $\forall t \geq T$.

Remark 2: The dynamics of PBNs can be studied by using Markov chains, whose steady-state distributions are employed to depict the long-run behavior of networks. Although stability in probability also refers to the long-run behavior of PBNs, there are two main differences between these two problems. First, the steady-state distribution is an equilibrium of the Markov chain. However, for stability in probability, such equilibrium is not required. The probability can arbitrarily vary, as long as it is positive. Second, the steady-state distribution gives long-run probabilities for all the states, which means that we cannot discuss a single state separately. Whereas, the in-probability problem can be studied for distinct states. It is common that some of them are stable in probability, while some others are not.

III. Main Results

In this section, the stability in probability for PBNs (3) and stabilization in probability for PBCNs (5) are studied for the first time. They are meaningful and challenging generalizations of the traditional stability/stabilization with probability one. Several remarks are given to illustrate the difficulties. Some necessary and sufficient conditions are proposed in Sections III-A and III-B, respectively. Furthermore, the design algorithm for controllers is proposed.

A. Stability in Probability

First, we will iteratively define a sequence of sets $\{\mathcal{C}_k(L, x^*)\}_{k=1,2,\dots}$ for any given $x^* \in \Delta_{2^n}$ and $L \in \mathscr{L}_{2^n\times 2^n}$ as follows:

$$\mathcal{C}_1(L, x^*) = \Delta_{2^n}\backslash\mathscr{T}^\top(x^{*\top} L) \tag{8}$$

$$\mathcal{C}_{k+1}(L, x^*) = \Delta_{2^n}\backslash\mathscr{T}^\top((\mathbf{1}_{2^n} - A_k)^\top \ltimes_{\mathscr{B}} L) \tag{9}$$

where $A_k = \sum_{a\in\mathcal{C}_k(L,x^*)} a$. Considering the PBN (3), let $\mathcal{L} = (\mathscr{B}) \sum_{s=1}^r L_s$. The following lemma reveals the relation between $\mathcal{C}_k(\mathcal{L}, x^*)$ and the probabilities of state transferring in the PBNs.

Lemma 3: $P\{x(t) = x^*|x(0) = a\} = 0$ if and only if $a \in \mathcal{C}_t(\mathcal{L}, x^*)$ for any integer $t \geq 1$.

Proof: We will prove the conclusion by induction. For the case $t = 1$, we have

$$\begin{aligned} &P\{x(1) = x^*|x(0) = a\} = 0 \\ &\quad \Leftrightarrow x^* \neq L_s \ltimes a, \quad \text{for } \forall\, s \in \{1, 2, \ldots, r\} \\ &\quad \Leftrightarrow x^{*\top} L_s \ltimes a = 0, \quad \text{for } \forall\, s \in \{1, 2, \ldots, r\} \\ &\quad \Leftrightarrow x^{*\top} \mathcal{L} \ltimes a = 0 \\ &\quad \Leftrightarrow a \in \Delta_{2^n} \backslash \mathscr{T}^\top(x^{*\top}\mathcal{L}) = \mathcal{C}_1(\mathcal{L}, x^*). \quad \text{(by Lemma 2)} \end{aligned}$$

Assume that the conclusion holds for some $t = k$. For the case $k + 1$, we have

$$\begin{aligned} &P\{x(k+1) = x^*|x(0) = a\} = 0 \\ &\quad \Leftrightarrow \sum_{a_1 \in \Delta_{2^n}} P\{x(k+1) = x^*|x(1) = a_1\} \\ &\qquad\qquad \times P\{x(1) = a_1|x(0) = a\} = 0 \\ &\quad \Leftrightarrow \sum_{a_1 \notin \mathcal{C}_k(\mathcal{L}, x^*)} P\{x(k+1) = x^*|x(1) = a_1\} \\ &\qquad\qquad \times P\{x(1) = a_1|x(0) = a\} = 0 \\ &\quad \Leftrightarrow P\{x(1) = a_1|x(0) = a\} = 0 \text{ for } \forall\, a_1 \notin \mathcal{C}_k(\mathcal{L}, x^*) \\ &\quad \Leftrightarrow a \in \Delta_{2^n} \backslash \mathscr{T}^\top\big(a_1^\top \mathcal{L}\big) \text{for } \forall\, a_1 \notin \mathcal{C}_k(\mathcal{L}, x^*) \\ &\quad \Leftrightarrow a \in \bigcap_{a_1 \notin \mathcal{C}_k(\mathcal{L}, x^*)} \Delta_{2^n} \backslash \mathscr{T}^\top\big(a_1^\top \mathcal{L}\big) \\ &\quad \Leftrightarrow a \in \Delta_{2^n} \backslash \bigcup_{a_1 \notin \mathcal{C}_k(\mathcal{L}, x^*)} \mathscr{T}^\top\big(a_1^\top \mathcal{L}\big) \\ &\quad \Leftrightarrow a \in \Delta_{2^n} \backslash \mathscr{T}^\top((\mathbf{1}_{2^n} - A_k)^\top \ltimes_{\mathscr{B}} \mathcal{L}) = \mathcal{C}_{k+1}(\mathcal{L}, x^*). \end{aligned}$$

Thus, the conclusion holds for the case $t = k+1$. By induction, the conclusion holds for all integer $t \geq 1$. □

Lemma 3 shows that $\mathcal{C}_t(\mathcal{L}, x^*)$ is substantially a nonreachable set of the state x^*. If there exists an integer $T > 0$ such that $\mathcal{C}_T(\mathcal{L}, x^*) = \emptyset$, we can declare that the PBN is stable in probability. However, we still need to estimate the bound of T. This work involves another property of the set $\mathcal{C}_t(\mathcal{L}, x^*)$. It is discussed in next lemma.

Lemma 4:

1) If $\mathcal{C}_{k+1}(\mathcal{L}, x^*) \subseteq \mathcal{C}_k(\mathcal{L}, x^*)$, then $\mathcal{C}_{k+2}(\mathcal{L}, x^*) \subseteq \mathcal{C}_{k+1}(\mathcal{L}, x^*)$ holds, for any integer $k \geq 1$.
2) If $\mathcal{C}_k(\mathcal{L}, x^*) = \mathcal{C}_{k+1}(\mathcal{L}, x^*)$, then $\mathcal{C}_k(\mathcal{L}, x^*) = \mathcal{C}_j(\mathcal{L}, x^*)$ holds, for any integer $j \geq k$.

Proof: For the Conclusion 1), it follows from $\mathcal{C}_{k+1}(\mathcal{L}, x^*) \subseteq \mathcal{C}_k(\mathcal{L}, x^*)$ that $A_{k+1} = A_k \wedge A_{k+1}$. Thus, $\mathbf{1}_{2^n} - A_k = (\mathbf{1}_{2^n} - A_k) \wedge (\mathbf{1}_{2^n} - A_{k+1})$. It implies that $\mathscr{T}((\mathbf{1}_{2^n} - A_k)^\top \ltimes_{\mathscr{B}} \mathcal{L}) \subseteq \mathscr{T}((\mathbf{1}_{2^n} - A_{k+1})^\top \ltimes_{\mathscr{B}} \mathcal{L})$, which further means that $\mathcal{C}_{k+2}(\mathcal{L}, x^*) \subseteq \mathcal{C}_{k+1}(\mathcal{L}, x^*)$.

We will prove the Conclusion 2) by induction. When $j = k$, it is trivial. Assume that $\mathcal{C}_k(\mathcal{L}, x^*) = \mathcal{C}_j(\mathcal{L}, x^*)$ holds for some fixed $j > k$. Thus, we have $A_k = A_j$. It yields that

$$\begin{aligned} \mathcal{C}_{j+1}(\mathcal{L}, x^*) &= \Delta_{2^n} \backslash \mathscr{T}^\top((\mathbf{1}_{2^n} - A_j)^\top \ltimes_{\mathscr{B}} \mathcal{L}) \\ &= \Delta_{2^n} \backslash \mathscr{T}^\top((\mathbf{1}_{2^n} - A_k)^\top \ltimes_{\mathscr{B}} \mathcal{L}) \\ &= \mathcal{C}_{k+1}(\mathcal{L}, x^*). \end{aligned}$$

That is, $\mathcal{C}_k(\mathcal{L}, x^*) = \mathcal{C}_{k+1}(\mathcal{L}, x^*) = \mathcal{C}_{j+1}(\mathcal{L}, x^*)$. By induction, the Conclusion 2) holds for all $j \geq k$. □

Combined with the above-mentioned lemmas, a necessary and sufficient condition is proposed in the following theorem to guarantee the stability in probability of PBNs.

Theorem 1: Given a reference state $x^* \in \Delta_{2^n}$, the PBN (3) can realize stability in probability at x^*, if and only if there exist two minimum integers S and T ($S \leq T$), such that $\mathcal{C}_{S+1}(\mathcal{L}, x^*) \subseteq \mathcal{C}_S(\mathcal{L}, x^*)$, $\mathcal{C}_T(\mathcal{L}, x^*) = \emptyset$ and $T \leq |\mathcal{C}_S(\mathcal{L}, x^*)| + S$.

Proof: From Lemmas 3 and 4, the PBN (3) can stabilize at x^* in probability, if and only if there exists an integer T such that $\mathcal{C}_T(\mathcal{L}, x^*) = \emptyset$.

Next, we will show the existence of the minimum integer S. If $\mathcal{C}_T(\mathcal{L}, x^*) = \emptyset$, we have $\mathcal{C}_T(\mathcal{L}, x^*) \subseteq \mathcal{C}_{T-1}(\mathcal{L}, x^*)$. Thus, a minimum integer $S(\leq T)$ always exists to satisfy $\mathcal{C}_{S+1}(\mathcal{L}, x^*) \subseteq \mathcal{C}_S(\mathcal{L}, x^*)$. On the other hand, if the PBN (3) can stabilize at x^* in probability, we will show $\mathcal{C}_{S+1}(\mathcal{L}, x^*) \subseteq \mathcal{C}_S(\mathcal{L}, x^*)$ by contradiction. Assume that $\mathcal{C}_t(\mathcal{L}, x^*) \nsubseteq \mathcal{C}_{t-1}(\mathcal{L}, x^*)$ for all integer $t > 1$, which is equivalent to $\exists a_t \in \mathcal{C}_t(\mathcal{L}, x^*) \backslash \mathcal{C}_{t-1}(\mathcal{L}, x^*)$. It contradicts to the fact that there exists an integer T, such that $\mathcal{C}_T(\mathcal{L}, x^*) = \emptyset$. Thus, $\mathcal{C}_{S+1}(\mathcal{L}, x^*) \subseteq \mathcal{C}_S(\mathcal{L}, x^*)$ must hold for some integer $S \leq T$.

To prove $T \leq |\mathcal{C}_S(\mathcal{L}, x^*)| + S$, it is enough to show that $|\mathcal{C}_t(\mathcal{L}, x^*)| \geq |\mathcal{C}_{t+1}(\mathcal{L}, x^*)| + 1$, for all $S \leq t \leq T - 1$. Actually, if $S = T$, it is trivial; if $S < T$, adding these $T - S$ inequalities together, it follows that $|\mathcal{C}_S(\mathcal{L}, x^*)| \geq |\mathcal{C}_T(\mathcal{L}, x^*)| + T - S$, which further implies $T \leq |\mathcal{C}_S(\mathcal{L}, x^*)| + S$. Assume $|\mathcal{C}_t(\mathcal{L}, x^*)| < |\mathcal{C}_{t+1}(\mathcal{L}, x^*)| + 1$, i.e., $|\mathcal{C}_t(\mathcal{L}, x^*)| = |\mathcal{C}_{t+1}(\mathcal{L}, x^*)|$. Due to $\mathcal{C}_t(\mathcal{L}, x^*) \subseteq \mathcal{C}_{t+1}(\mathcal{L}, x^*)$, we have $\mathcal{C}_t(\mathcal{L}, x^*) = \mathcal{C}_{t+1}(\mathcal{L}, x^*)$. Based on the Conclusion 2) of Lemma 4, it implies that $\mathcal{C}_S(\mathcal{L}, x^*) = \mathcal{C}_T(\mathcal{L}, x^*) = \emptyset$, which contradicts that T is the minimum integer such that $\mathcal{C}_T(\mathcal{L}, x^*) = \emptyset$. Thus, we have $|\mathcal{C}_t(\mathcal{L}, x^*)| \geq |\mathcal{C}_{t+1}(\mathcal{L}, x^*)| + 1$. □

It should be pointed out that $P\{x(t+1) = x^*|x(t) = x^*\} > 0$ is not a necessary condition for the in-probability problem. That is, despite $P\{x(t+1) = x^*|x(t) = x^*\} = 0$, the PBN may still realize stability in probability. For example, let $L_1 = \delta_4[4, 4, 1, 3]$, $L_2 = \delta_4[4, 3, 4, 3]$ and $p_1 = p_2 = 0.5$. Thus

$$\mathcal{L} = L_1 +_{\mathscr{B}} L_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

Assume $x^* = \delta_4^1$. Based on (8) and (9), we can have that $\mathcal{C}_1(\mathcal{L}, x^*) = \{\delta_4^1, \delta_4^2, \delta_4^4\}$, $\mathcal{C}_2(\mathcal{L}, x^*) = \{\delta_4^1, \delta_4^3\}$, $\mathcal{C}_3(\mathcal{L}, x^*) = \{\delta_4^4\}$, $\mathcal{C}_4(\mathcal{L}, x^*) = \{\delta_4^1\}$ and $\mathcal{C}_5(\mathcal{L}, x^*) = \emptyset$. From Theorem 1, the PBN (3) is stable at x^* in probability. However, it can be easily observed that $P\{x(t+1) = x^*|x(t) = x^*\} = 0$. It is quite a different feature compared to the stability of PBNs with probability one, in which $P\{x(t+1) = x^*|x(t) = x^*\} = 1$ is a necessary condition [30], [32]. This feature makes the stability in probability of PBNs a much more complicated problem. Without $P\{x(t+1) = x^*|x(t) = x^*\} > 0$, which is equivalent to $x^* \notin \mathcal{C}_1(\mathcal{L}, x^*)$, it is impossible to build a series of inclusion relation $\mathcal{C}_T(\mathcal{L}, x^*) \subseteq \cdots \subseteq \mathcal{C}_2(\mathcal{L}, x^*) \subseteq \mathcal{C}_1(\mathcal{L}, x^*)$, such as

Algorithm 1 Checking Stability in Probability of a PBN

Step 1	Set $A = \mathbf{1}_{2^n} - x^*$ and $\mathcal{L} = (\mathscr{B})\sum_{s=1}^{r} L_s$.
Step 2	Compute $B = \mathbf{1}_{2^n} - \mathcal{L}^\top \ltimes_{\mathscr{B}} (\mathbf{1}_{2^n} - A)$.
Step 3	If $B = \mathbf{0}_{2^n}$, set $flag = 1$ and go to End; elseif $B = A$, set $flag = 0$ and go to End; else, set $A = B$ and jump to Step 2.
End	If $flag = 1$, the PBN is stable in probability at x^*; else, PBN is not stable in probability at x^*.

the probability-one problem (see [30, Th. 1]). The size of $\mathcal{C}_k(\mathcal{L}, x^*)$ fluctuates, until some S can be found such that $\mathcal{C}_{S+1}(\mathcal{L}, x^*) \subseteq \mathcal{C}_S(\mathcal{L}, x^*)$.

If the PBN is assumed to satisfy $P\{x(t+1) = x^*|x(t) = x^*\} > 0$, some more easily verifiable conditions for stability in probability can be proposed as follows.

Corollary 1: Given a target state $x^* \in \Delta_{2^n}$, under the assumption that $P\{x(t+1) = x^*|x(t) = x^*\} > 0$, the PBN (3) can realize stability in probability at x^*, if and only if there exists an integer $T \le |\mathcal{C}_1(\mathcal{L}, x^*)| + 1$, such that $\mathcal{C}_T(\mathcal{L}, x^*) = \emptyset$.

Proof: We only need to prove $T \le |\mathcal{C}_1(\mathcal{L}, x^*)| + 1$. Based on Lemma 3, $P\{x(t+1) = x^*|x(t) = x^*\} > 0$ is equivalent to $x^* \notin \mathcal{C}_1(\mathcal{L}, x^*)$. For any $a \in \mathcal{C}_2(\mathcal{L}, x^*)$, it follows from Lemma 2 and (9) that $((\mathbf{1}_{2^n} - A_1)^\top \ltimes_{\mathscr{B}} \mathcal{L}) \ltimes a = 0$. Due to $x^* \notin \mathcal{C}_1(\mathcal{L}, x^*)$, it yields that $x^* \in \mathscr{T}(\mathbf{1}_{2^n} - A_1)$. Thus, $(x^*\mathcal{L}) \ltimes a = 0$, which further means that $a \in \mathcal{C}_1(\mathcal{L}, x^*)$. It can be obtained $\mathcal{C}_2(\mathcal{L}, x^*) \subseteq \mathcal{C}_1(\mathcal{L}, x^*)$. Thus, $S = 1$ in Theorem 1, i.e., $T \le |\mathcal{C}_1(\mathcal{L}, x^*)| + 1$. □

Remark 3: The integer T is the number of the sequence of sets $\{\mathcal{C}_k(\mathcal{L}, x^*)\}$, which is needed to be estimated. In Corollary 1, an upper bound of T is given. Thus, the theoretical results can be easily verified by checking whether the set $\mathcal{C}_k(\mathcal{L}, x^*)$ is empty or not at most $|\mathcal{C}_1(\mathcal{L}, x^*)| + 1$ times. However, in Theorem 1, the upper bound of T cannot be obtained without knowing S. It is still a lack of feasible methods to estimate S. Furthermore, the value range of S is very wide (up to $2^{2^n} - 1$). In many works [30], the estimation depends on the inclusion relation of reachable sets, which may not be satisfied in Theorem 1. Thus, some new methods, which are not based on the reachable sets, are needed. Our future work will try to solve it. Note that the inclusion property is satisfied in Corollary 1. Thus, the upper bound of T can be given. It is much easier to verify Corollary 1 than Theorem 1.

Based on the above-mentioned discussions, we can design Algorithm 1 a feasible algorithm to check stability in probability according to the condition proposed in Corollary 1, assuming that $P\{x(t+1) = x^*|x(t) = x^*\} > 0$ holds for PBNs.

B. Stabilization in Probability

In this section, stabilization in probability is investigated for PBCNs. Based on the study of stability in probability, a necessary and sufficient condition is given. However, the result is hard to verify, since it is lack of efficient methods to build a controller. Then, the time-varying controller is designed as an alternative to overcome it. Finally, the problem is simplified by assuming that $P\{x(t+1) = x^*|x(t) = x^*, u\} > 0$ holds for some u.

Theorem 2: Let $\mathcal{L} = (\mathscr{B})\sum_{s=1}^{r} L_s \in \mathscr{B}_{2^n \times 2^{n+m}}$. The PBCN (5) can realize stabilization at $x^* \in \Delta_{2^n}$ in probability with the state feedback controller $u(t) = Gx(t)$, if and only if for $\forall i \in \{1, 2, \ldots, 2^n\}$, there exists $d_i \in \{1, 2, \ldots, 2^m\}$, such that to construct a matrix $\mathbf{L} \in \mathscr{L}_{2^n \times 2^n}$ as

$$Col_i(\mathbf{L}) = Col_i(Blk_{d_i}(\mathcal{L}))$$

and build a sequence of sets $\mathcal{C}_k(\mathbf{L}, x^*)$ as (8) and (9), we have $\mathcal{C}_T(\mathbf{L}, x^*) = \emptyset$ for some integer $T \ge 1$. The state feedback control matrix can be given as $G = \delta_{2^m}[d_1, d_2, \ldots, d_{2^n}]$.

Proof: For $\forall \delta_{2^n}^i \in \Delta_{2^n}$, let $u = G\delta_{2^n}^i$. According to the definition of the matrix $\mathbf{L}$, it follows that:

$$\mathcal{L}u\delta_{2^n}^i = \mathcal{L}\delta_{2^m}^{d_i}\delta_{2^n}^i = Col_i(Blk_{d_i}(\mathcal{L})) = Col_i(\mathbf{L}) = \mathbf{L}\delta_{2^n}^i.$$

We claim that $\delta_{2^n}^i \in \mathcal{C}_t(\mathbf{L}, x^*)$ if and only if for any integer $t \ge 1$, $P\{x(t) = x^*|x(0) = \delta_{2^n}^i, u(\tau) = Gx(\tau)\} = 0$. We will prove it by induction.

When $t = 1$, it follows from Lemma 2 that:

$$\begin{aligned} &P\{x(1) = x^*|x(0) = \delta_{2^n}^i, u(0) = G\delta_{2^n}^i\} = 0 \\ &\Leftrightarrow x^{*\top}\mathcal{L}\delta_{2^m}^{d_i}\delta_{2^n}^i = 0 \\ &\Leftrightarrow x^{*\top}\mathbf{L}\delta_{2^n}^i = 0 \\ &\Leftrightarrow \delta_{2^n}^i \in \Delta_{2^n}\backslash\mathscr{T}^\top(x^{*\top}\mathbf{L}) = \mathcal{C}_1(\mathbf{L}, x^*). \end{aligned}$$

Assume the conclusion holds for the case $t = k$. When $t = k+1$, we have

$$\begin{aligned} &P\{x(k+1) = x^*|x(0) = \delta_{2^n}^i, u(\tau) = Gx(\tau)\} = 0 \\ &\Leftrightarrow \sum_{a_1 \in \Delta_{2^n}} P\{x(k+1) = x^*|x(1) = a_1, u(\tau) = Gx(\tau)\} \\ &\qquad \times P\{x(1) = a_1|x(0) = \delta_{2^n}^i, u(0) = G\delta_{2^n}^i\} = 0 \\ &\Leftrightarrow \sum_{a_1 \notin \mathcal{C}_t(\mathbf{L}, x^*)} P\{x(k+1) = x^*|x(1) = a_1, u(\tau) = Gx(\tau)\} \\ &\qquad \times P\{x(1) = a_1|x(0) = \delta_{2^n}^i, u(0) = G\delta_{2^n}^i\} = 0 \\ &\Leftrightarrow P\{x(1) = a_1|x(0) = \delta_{2^n}^i, u(0) = G\delta_{2^n}^i\} = 0 \\ &\qquad \forall a_1 \notin \mathcal{C}_t(\mathbf{L}, x^*) \\ &\Leftrightarrow a_1^\top \mathbf{L}\delta_{2^n}^i = 0 \ \forall a_1 \notin \mathcal{C}_t(\mathbf{L}, x^*) \\ &\Leftrightarrow (\mathbf{1}_{2^n} - A_t)^\top \ltimes_{\mathscr{B}} \mathbf{L}\delta_{2^n}^i = 0 \ (\text{since } A_t = \sum_{a \in \mathcal{C}_t(\mathbf{L}, x^*)} a) \\ &\Leftrightarrow \delta_{2^n}^i \in \Delta_{2^n}\backslash\mathscr{T}^\top((\mathbf{1}_{2^n} - A_t)^\top \ltimes_{\mathscr{B}} \mathbf{L}) \ (\text{by Lemma 2}) \\ &\Leftrightarrow \delta_{2^n}^i \in \mathcal{C}_{t+1}(\mathbf{L}, x^*). \end{aligned}$$

Thus, from the claim, we can see that the PBCN (5) can realize stabilization at x in probability with the controller $u(t) = Gx(t)$, if and only if there exists an integer T such that $\mathcal{C}_T(\mathbf{L}, x^*) = \emptyset$. This completes the proof. □

Remark 4: To construct the controller, the matrix $\mathbf{L}$ should be found with $\mathcal{C}_T(\mathbf{L}, x^*) = \emptyset$ for some T. The total number of all the possible combinations to build $\mathbf{L}$ is $(2^m)^{2^n}$, since for each $i \in \{1, 2, \ldots, 2^n\}$, d_i has 2^m choices. Unfortunately, efficient algorithms are still rare to find $\mathbf{L}$. The reason is also that the reachable sets may not have the inclusion property, as discussed in Remark 3. When n or m is large, it is impossible to investigate the stabilizability of the PBCN. To overcome this problem, we will consider a time-varying controller to stabilize the PBCN as an alternative.

Considering the PBCN (5), let $\mathcal{L} = (\mathscr{B})\sum_{s=1}^{r} L_s \in \mathscr{B}_{2^n \times 2^{n+m}}$. We define a sequence of sets $\{\mathcal{H}_k(\mathcal{L}, x^*)\}$ for each $x^* \in \Delta_{2^n}$ as follows:

$$\mathcal{H}_0(\mathcal{L}, x^*) = \{x^*\} \tag{10}$$

$$\mathcal{H}_{k+1}(\mathcal{L}, x^*) = \bigcup_{s=1}^{2^m} \mathscr{T}^\top\left(B_k^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L})\right) \tag{11}$$

where $k = 0, 1, \ldots$ and $B_k = \sum_{a\in\mathcal{H}_k(\mathcal{L},x^*)} a$, and $Blk_s(\mathcal{L})$ is the sth square block of the matrix $\mathcal{L}$.

Theorem 3: For the PBCN (5) and any given target state $x^* \in \Delta_{2^n}$, $P\{x(T) = x^* | x(0) = x_0, u(\tau) = G(\tau)x(\tau)\} > 0$ holds for any initial state $x_0 \in \Delta_{2^n}$, if and only if there exists an integer $T > 0$, such that $\mathcal{H}_T(\mathcal{L}, x^*) = \Delta_{2^n}$. The time-varying control matrix is constructed as $G(\tau) = \delta_{2^m}[d_1(\tau), d_2(\tau), \ldots, d_{2^n}(\tau)]$ $(\tau = 0, 1, \ldots, T-1)$, where

$$d_i(\tau) \in \{s | B_{T-\tau-1}^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_{2^n}^i = 1\} \tag{12}$$

if the rhs set is nonempty; otherwise, $d_i(\tau)$ can be arbitrarily chosen from $\{1, 2, \ldots, 2^m\}$.

Proof: First, we will prove that for any integer $t \geq 0$, $\delta_{2^n}^i \notin \mathcal{H}_t(\mathcal{L}, x^*)$ if and only if $P\{x(t) = x^* | x(0) = \delta_{2^n}^i, u(\tau) = G(\tau)x(\tau)\} = 0$ for $\forall G(\tau) \in \mathscr{L}_{2^m \times 2^n}$ by induction.

When $t = 1$, we have

$$\begin{aligned}
&\delta_{2^n}^i \notin \mathcal{H}_1(\mathcal{L}, x^*)\\
&\Leftrightarrow \delta_{2^n}^i \in \Delta_{2^n} \backslash \mathscr{T}^\top(x^{*\top} Blk_s(\mathcal{L})), \quad \text{for } \forall s \in \{1, \ldots, 2^m\}\\
&\Leftrightarrow x^{*\top} Blk_s(\mathcal{L})\delta_{2^n}^i = 0, \quad \text{for } \forall s \in \{1, \ldots, 2^m\}\\
&\Leftrightarrow P\left\{x(1) = x^* \middle| x(0) = \delta_{2^n}^i, u(0) = G(0)\delta_{2^n}^i\right\} = 0,\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{for } \forall G(0) \in \mathscr{L}_{2^m \times 2^n}.
\end{aligned}$$

Supposing that the conclusion holds for the case $t = k$, we consider the case $t = k+1$ as

$$\begin{aligned}
&\delta_{2^n}^i \notin \mathcal{H}_{k+1}(\mathcal{L}, x^*)\\
&\Leftrightarrow \delta_{2^n}^i \in \Delta_{2^n} \backslash \mathscr{T}^\top\left(B_k^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L})\right),\\
&\qquad\qquad\qquad\qquad\qquad \text{for } \forall s \in \{1, \ldots, 2^m\}\\
&\Leftrightarrow B_k^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L})\delta_{2^n}^i = 0, \quad \text{for } \forall s \in \{1, \ldots, 2^m\}\\
&\Leftrightarrow a^\top Blk_s(\mathcal{L})\delta_{2^n}^i = 0, \quad \text{for } \forall s \in \{1, \ldots, 2^m\}\\
&\qquad\qquad\qquad\qquad\qquad\qquad \forall a \in \mathcal{H}_k(\mathcal{L}, x^*)\\
&\Leftrightarrow P\{x(1) = a | x(0) = \delta_{2^n}^i, u(0) = G(0)\delta_{2^n}^i\} = 0,\\
&\qquad\qquad \text{for } \forall a \in \mathcal{H}_k(\mathcal{L}, x^*) \text{ and } \forall G(0) \in \mathscr{L}_{2^m \times 2^n}\\
&\Leftrightarrow P\{x(k+1) = x^* | x(0) = \delta_{2^n}^i, u(\tau) = G(\tau)x(\tau)\}\\
&= \sum_{a \notin \mathcal{H}_k(\mathcal{L}, x^*)} P\left\{x(1) = a | x(0) = \delta_{2^n}^i, u(0) = G(0)\delta_{2^n}^i\right\}\\
&\quad \times P\{x(k+1) = x^* | x(1) = a, u(\tau) = G(\tau)x(\tau)\},\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{for } \forall G(0) \in \mathscr{L}_{2^m \times 2^n}\\
&\Leftrightarrow P\left\{x(k+1) = x^* | x(0) = \delta_{2^n}^i, u(\tau) = G(\tau)x(\tau)\right\} = 0,\\
&\qquad\qquad \text{for } \forall G(\tau) \in \mathscr{L}_{2^m \times 2^n} \text{ and } \tau = 0, 1, \ldots, k.
\end{aligned}$$

By induction, we can conclude that for any integer $t \geq 0$, $\delta_{2^n}^i \notin \mathcal{H}_t(\mathcal{L}, x^*)$ if and only if $P\{x(t) = x^* | x(0) = \delta_{2^n}^i, u(\tau) = G(\tau)x(\tau)\} = 0$ for $\forall G(\tau) \in \mathscr{L}_{2^m \times 2^n}$. In other words, $\mathcal{H}_T(\mathcal{L}, x^*) = \Delta_{2^n}$ if and only if there exists a time-varying matrix $G(\tau) \in \mathscr{L}_{2^m \times 2^n}$ $(0 \leq \tau < T)$ such that $P\{x(T) = x^* | x(0) = x_0, u(\tau) = G(\tau)x(\tau)\} > 0$ holds for any initial state $x_0 \in \Delta_{2^n}$.

Secondly, we will show that the proposed time-varying controller can guide any initial state to x^* in probability at time T. According to the definition of $G(\tau)$, we have

$$\begin{aligned}
&B_{T-t-1}^\top \ltimes_{\mathscr{B}} Blk_{d_i(t)}(\mathcal{L}) \ltimes \delta_{2^n}^i = 1\\
&\quad \Leftrightarrow a_{t+1}^\top \ltimes \mathcal{L} \ltimes \left(G(t) \ltimes \delta_{2^n}^i\right) \ltimes \delta_{2^n}^i = 1\\
&\quad \Leftrightarrow P\left\{x(t+1) = a_{t+1} | x(t) = \delta_{2^n}^i, u(t) = G(t)\delta_{2^n}^i\right\} > 0
\end{aligned}$$

for $\forall \delta_{2^n}^i \in \mathcal{H}_{T-t}(\mathcal{L}, x^*)$ and $\forall a_{t+1} \in \mathcal{H}_{T-t-1}(\mathcal{L}, x^*)$. Note that $\mathcal{H}_T(\mathcal{L}, x^*) = \Delta_{2^n}$ guarantees $\mathcal{H}_t(\mathcal{L}, x^*) \neq \emptyset$ for any integer $0 \leq t < T$. Thus, for any initial state $x_0 \in \Delta_{2^n}$, it can be got that

$$\begin{aligned}
&P\{x(T) = x^* | x(0) = x_0, u(\tau) = G(\tau)x(\tau)\}\\
&> \sum_{a_{T-1} \in \mathcal{H}_1(\mathcal{L}, x^*)} P\{x(T) = x^* | x(T-1) = a_{T-1}\\
&\qquad\qquad\qquad\qquad u(T-1) = G(T-1)a_{T-1}\}\\
&\quad \times \cdots \times \sum_{a_1 \in \mathcal{H}_{T-1}(\mathcal{L}, x^*)} P\{x(1) = a_1 | x(0) = x_0, u(0) = G(0)x_0\}\\
&> 0.
\end{aligned}$$

□

Algorithm 2 Designing a Time-varying Controller for a PBCN

Step 1 Set $\tau = T - 1$, $B = x^*$ and $\mathcal{L} = (\mathscr{B})\sum_{s=1}^{r} L_s$.
Step 2 Set $i = 1$.
Step 3 Set $d_i = 1$ and $s = 1$.
Step 4 Compute $flag = B^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_{2^n}^i$.
Step 5 If $flag = 1$, set $d_i = s$ and go to Step 6;
elseif $s = 2^m$, go to Step 6;
else set $s = s + 1$ and jump to Step 4.
Step 6 If $i = 2^n$, set $G(\tau) = \delta_{2^m}[d_1, d_2, \ldots, d_{2^n}]$
and continue to Step 7;
else set $i = i + 1$ and jump to Step 3.
Step 7 If $\tau = 0$, continue to End;
else set $\tau = \tau - 1$, $B = (\mathcal{L} \ltimes_{\mathscr{B}} \mathbf{1}_{2^m})^\top \ltimes_{\mathscr{B}} B$
and jump to Step 2.
End

Providing that $\mathcal{H}_T(\mathcal{L}, x^*) = \Delta_{2^n}$ holds for some integer $T > 0$, the time-varying controller can be designed by (12). The algorithm to implement the controller is given in Algorithm 2.

Remark 5: The proposed condition of Theorem 3 ensures that the whole network can arrive at x^* in probability at time T. However, it does not ensure the network stabilized at x^* in probability. It is still very hard to design the controller for the time $t \geq T$. If we assume that $P\{x(t+1) = x^* | x(t) = x^*, u\} > 0$ holds for some $u \in \Delta_{2^m}$, i.e., $x^* \in \mathcal{H}_1(\mathcal{L}, x^*)$, the stabilization condition can be derived. In addition, the controller can be time invariant.

Algorithm 3 Designing a Time-invariant Controller for a PBCN

Step 1	Initialize $t = 1$, $A = \mathbf{0}_{2^n}$, $B_0 = \mathbf{0}_{2^n}$, $B = x^*$ and $\mathcal{L} = (\mathscr{B}) \sum_{s=1}^{r} L_s$.
Step 2	Set $i = 1$, $A = A + B_0$, $B_0 = B$ and $B = (\mathcal{L} \ltimes_{\mathscr{B}} \mathbf{1}_{2^m})^\top \ltimes_{\mathscr{B}} B_0$.
Step 3	If $(B - A)^\top \times \delta_{2^n}^i = 1$, set $s = 1$ and continue to Step 4; elseif $i = 2^n$, jump to Step 5; else, set $i = i + 1$ and go to Step 3.
Step 4	Compute $flag = B_0^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_{2^n}^i$. If $flag = 1$, set $d_i = s$ and go to Step 5; else, set $s = s + 1$ and go to Step 4.
Step 5	If $t = T$, go to End; else, set $t = t + 1$ and jump to Step 2.
End	Set $G = \delta_{2^m}[d_1, d_2, \dots, d_{2^n}]$.

By using almost the same method as that of Lemma 4, the following results can be obtained. The proof is omitted here.

Lemma 5: If $\mathcal{H}_k(\mathcal{L}, x^*) \subseteq \mathcal{H}_{k+1}(\mathcal{L}, x^*)$, then $\mathcal{H}_{k+1}(\mathcal{L}, x^*) \subseteq \mathcal{H}_{k+2}(\mathcal{L}, x^*)$ holds, for any integer $k \geq 0$. Especially, if $\mathcal{H}_k(\mathcal{L}, x^*) = \mathcal{H}_{k+1}(\mathcal{L}, x^*)$, then $\mathcal{H}_k(\mathcal{L}, x^*) = \cdots = \mathcal{H}_j(\mathcal{L}, x^*)$ holds, for any integer $j \geq k$.

Theorem 4: Given a target state $x^* \in \Delta_{2^n}$, under the assumption that $x^* \in \mathcal{H}_1(\mathcal{L}, x^*)$, the PBCN (5) can stabilize at x^* in probability, if and only if there exists an integer $T \leq 2^n - 1$, such that $\mathcal{H}_T(\mathcal{L}, x^*) = \Delta_{2^n}$. The control matrix is constructed as $G = \delta_{2^m}[d_1, d_2, \dots, d_{2^n}]$, where

$$d_i \in \{s | B^\top_{\min\{t \geq 1 | \delta_{2^n}^i \in \mathcal{H}_t(\mathcal{L}, x^*)\} - 1} \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_{2^n}^i = 1\}$$

for $i = 1, 2, \dots, 2^n$.

Proof: Based on Theorem 3, just letting $G(t) \equiv G$, we can obtain that $P\{x(T) = x^* | x(0) = x_0, u(\tau) = Gx(\tau)\} > 0$ holds for any initial state $x_0 \in \Delta_{2^n}$ if and only if $\mathcal{H}_T(\mathcal{L}, x^*) = \Delta_{2^n}$. The assumption $x^* \in \mathcal{H}_1(\mathcal{L}, x^*)$ is equivalent to $P\{x(t+1) = x^* | x(t) = x^*, u(t) = Gx^*\} > 0$. Thus, it yields that the PBCN (5) can stabilize at x^* in probability.

Next, we will show $T \leq 2^n - 1$. By using almost the same method of Theorem 1, it follows from Lemma 5 that $|\mathcal{H}_{k+1}(\mathcal{L}, x^*)| \geq |\mathcal{H}_k(\mathcal{L}, x^*)| + 1$ $(0 \leq k < T)$. Considering that $\mathcal{H}_0(\mathcal{L}, x^*) = 1$ and $|\mathcal{H}_T(\mathcal{L}, x^*)| = 2^n$, it follows that $T \leq 2^n - 1$. □

If the PBCNs can be stabilized in probability with the assumption that $P\{x(t+1) = x^* | x(t) = x^*, u\} > 0$ holds for some $u \in \Delta_{2^m}$, Theorem 4 gives a method to build the controller. Based on it, a feasible algorithm to implement the controller is proposed as Algorithm 3.

Remark 6: The computational complexity of Algorithm 3 mainly lies at Step 4, the Boolean matrix multiplication $B_0^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_{2^n}^i$. It needs 2^{n+m} multiplications. Algorithm 3 can be easily conducted for moderate n and m. However, the number of multiplication is $T \times 2^{n+m}$ in Algorithm 2, since it has to construct a feedback controller at each time. Furthermore, the bound of T is hard to be estimated. It may be quite large. Thus, Algorithm 3 is much more feasible, whereas Algorithm 2 is a compromise. Despite of not being as good as Algorithm 2, it is still an acceptable method, when $P\{x(t+1) = x^* | x(t) = x^*, u\} = 0$ for any u.

Remark 7: In this article, the probability p_i corresponding to each structural function f_i is time invariant. It is interesting to study the PBNs with time-varying transition probabilities, while this issue is difficult. The reasons are twofold. First, the dynamics of such kind of PBNs can be handled in the probabilistic context of a nonhomogeneous Markov chain, which is much harder than the homogeneous one. Second, the probabilities $\{p_i\}_{i=1,\dots,r}$ are obtained from experimental data by employing the coefficient of determination [20]. If the probabilities are time dependent, more training data are needed. However, the insufficient amount of accessible data is still a limitation of the research of BNs. Nevertheless, PBNs with time-varying transition probabilities can represent more realistic biological systems. Our future work will focus on it.

Remark 8: In very recent years, the stabilization problem has been studied for PBCNs in many works, for example, set stabilization [48]–[50], output feedback stabilization [51], output feedback set stabilization [52] and sampled-data stabilization [53]. In these articles, the reachable sets of the target state (or set) are iteratively established for proposing stabilization conditions. Since they all investigated the probability-one problem, the target state (or set) must belong to the first reachable set. It makes that the sequence of reachable sets has the inclusion property. Then, the controller can be built according to the difference in the set of two adjacent reachable sets. As discussed earlier, the in-probability problem faces a different situation, in which the inclusion relation may not exist for reachable sets. Thus, it is difficult to design the controller (see Remark 4). To overcome it, the time-varying controller is constructed in Theorem 3 by employing reachable sets directly. Furthermore, if the target state (or set) has the probability of transferring to itself, we can also build a time-invariant controller.

Remark 9: The in-probability stability/stabilization requires the long-term probability of a target state is positive. It may happen that a state with an extremely small probability can also be referred to as a stable state. This is seemingly opposed to common sense. Moreover, it would be practical to know whether the long-term probability is greater than a given threshold or not. Thus, more meaningful definitions of the in-probability problems could be

$$P\{x(t) = x^* | x(0) = x_0\} \geq p$$
$$P\{x(t) = x^* | x(0) = x_0, u(\tau) = g(x(\tau))\} \geq p$$

where $p > 0$ is the given threshold probability. The probability distributions may not be convergent, but the probability of the target state must vary in a fixed range, i.e., the interval $[p, 1)$. It is not a trivial work. Our future work will try to discuss it.

IV. Numerical Examples

In this section, two numerical examples (simulated by a PC with an Intel Core-i7 1.80-GHz processor and 8-GB memory) will be given to show the effectiveness of our theoretical results. Algorithms 1–3 provide principles for programming codes written in MATLAB.

Example 1: In this example, we will verify our results about the stability in probability via a real financial case. In [54], the PBN is used to study the credit default data of different industrial sectors. A series of 88 quarterly default data of three industrial sectors (transport x_1, energy x_2 and consumer x_3) are extracted, where $x_i = \delta_2^1$ represents no default in a quarter; $x_i = \delta_2^2$ represents default observed ($i = 1, 2, 3$). A PBN can be built as follows, by solving an entrop optimization problem:

$$L_1 = \delta_8\{1, 6, 5, 2, 7, 6, 3, 8\}, \quad L_2 = \delta_8\{1, 2, 6, 2, 3, 8, 2, 6\}$$
$$L_3 = \delta_8\{2, 5, 3, 3, 6, 8, 3, 8\}, \quad L_4 = \delta_8\{6, 4, 3, 8, 2, 5, 3, 8\}$$
$$L_5 = \delta_8\{6, 2, 3, 8, 6, 2, 2, 6\}, \quad L_6 = \delta_8\{6, 3, 1, 3, 3, 4, 3, 6\}$$

with $p_1 = 0.3295$, $p_2 = 0.2273$, $p_3 = 0.1477$, $p_4 = 0.1364$, $p_5 = 0.1023$, and $p_6 = 0.0568$. By some easy computations, it can be found that $P\{x(t+1) = \delta_8^i | x(t) = \delta_8^i\} < 1$ for all $i = 1, 2, \ldots, 8$, based on [30, Th. 1]. None of the PBN's states can be stabilized with probability one, which conforms to the common sense that no industrial sector can be expected to boom or shrink with 100% certainty. Not to mention that they can keep boom or shrink forever. A more reasonable perspective is to investigate the stability in probability. Considering $x^* = \delta_8^7$, let

$$\mathcal{L} = (\mathscr{B}) \sum_{i=1}^{6} L_i = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

The sequence of sets $\{\mathcal{C}_k(\mathcal{L}, \delta_8^7)\}$ can be obtained as

$$\mathcal{C}_1(\mathcal{L}, \delta_8^7) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^4, \delta_8^6, \delta_8^7, \delta_8^8\}$$
$$\mathcal{C}_2(\mathcal{L}, \delta_8^7) = \{\delta_8^1, \delta_8^4, \delta_8^5, \delta_8^7, \delta_8^8\}$$
$$\mathcal{C}_3(\mathcal{L}, \delta_8^7) = \emptyset.$$

Note that $\mathcal{C}_2(\mathcal{L}, \delta_8^7) \nsubseteq \mathcal{C}_1(\mathcal{L}, \delta_8^7)$, due to $P\{x(t+1) = \delta_8^7 | x(t) = \delta_8^7\} = 0$. According to Theorem 1, the PBN is stable at x^* in probability with $S = 2$ and $T = 3$, although x^* has zero probability to transfer to itself. It means that no matter what the initial state is, a possible economic situation is that only the consumer sector has no credit default since the third quarter. The numerical results of $P\{x(t) = x^* | x(0)\}$ are given in Table I.

Providing $x^* = \delta_8^8$, the sequence of $\{\mathcal{C}_i(\mathcal{L}, x^*)\}$ can be built as: $\mathcal{C}_1(\mathcal{L}, \delta_8^8) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^7\}$, $\mathcal{C}_2(\mathcal{L}, \delta_8^8) = \{\delta_8^7\}$, $\mathcal{C}_3(\mathcal{L}, \delta_8^8) = \emptyset$. It can be observed that $\mathcal{C}_3(\mathcal{L}, \delta_8^8) \subseteq \mathcal{C}_2(\mathcal{L}, \delta_8^8) \subseteq \mathcal{C}_1(\mathcal{L}, \delta_8^8)$, since $P\{x(t+1) = \delta_8^8 | x(t) = \delta_8^8\} = 0.6136 > 0$. By employing Corollary 1, we know that the state $x^* = \delta_8^8$ ($x_1 = 1$, $x_2 = 1$ and $x_3 = 1$) can be stable in probability with $T = 3$ from any initial states. It coincides with the fact that the quarter with credit default in all three sectors will appear within a short period from any starting quarter [55].

Example 2: We now consider the following PBCN of an apoptosis network [56] as an example for the stabilization in probability

$$f_1 = (\neg x_2 \wedge u, \neg x_1 \wedge x_3, x_2 \vee u), \quad f_2 = (x_1, x_2, x_3)$$
$$f_3 = (x_1, \neg x_1 \wedge x_3, x_2 \vee u), \quad f_4 = (\neg x_2 \wedge u, x_2, x_3)$$
$$f_5 = (\neg x_2 \wedge u, x_2, x_2 \vee u), \quad f_6 = (x_1, \neg x_1 \wedge x_3, x_3)$$
$$f_7 = (x_1, x_2, x_2 \vee u), \quad f_8 = (\neg x_2 \wedge u, \neg x_1 \wedge x_3, x_3)$$

with the probabilities $p_1 = 0.336$, $p_2 = 0.024$, $p_3 = 0.224$, $p_4 = 0.036$, $p_5 = 0.144$, $p_6 = 0.056$, $p_7 = 0.096$, and $p_8 = 0.084$. Here, x_1, x_2, and x_3 denote the concentration levels of inhibitor of apoptosis proteins (IAP), active caspase 3 (C3a), and active caspase 8 (C8a), respectively. The concentration level of the TNF (a stimulus) is denoted by u, and is regarded as the control input. By utilizing Lemma 1, we can have

$$L_1 = \delta_8[7, 7, 3, 3, 5, 7, 1, 3, 7, 7, 8, 8, 5, 7, 6, 8]$$
$$L_2 = \delta_8[1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8]$$
$$L_3 = \delta_8[3, 3, 3, 3, 5, 7, 5, 7, 3, 3, 4, 4, 5, 7, 6, 8]$$
$$L_4 = \delta_8[5, 6, 3, 4, 5, 6, 3, 4, 5, 6, 7, 8, 5, 6, 7, 8]$$
$$L_5 = \delta_8[5, 5, 3, 3, 5, 5, 3, 3, 5, 5, 8, 8, 5, 5, 8, 8]$$
$$L_6 = \delta_8[3, 4, 3, 4, 5, 8, 5, 8, 3, 4, 3, 4, 5, 8, 5, 8]$$
$$L_7 = \delta_8[1, 1, 3, 3, 5, 5, 7, 7, 1, 1, 4, 4, 5, 5, 8, 8]$$
$$L_8 = \delta_8[7, 8, 3, 4, 5, 8, 1, 4, 7, 8, 7, 8, 5, 8, 5, 8]$$

and $\mathcal{L} = (\mathscr{B}) \sum_{i=1}^{8} L_i$.

This PBCN has revealed that the system cannot be globally stabilized with probability one to any state [30]. None of the states can be definitely predicted with the stimulus TNF. From the biological point of view, a cell may fall into either the survival or the apoptotic in probability. However, let $x^* = \delta_8^5$ and $G = \delta_2[*, *, 2, *, *, *, *, 1]$, where $*$ can be arbitrarily selected as 1 or 2. Applying Theorem 2, the PBCN can be stabilized at x^* in probability with state feedback control. For example, let $G = \delta_2[1, 1, 2, 1, 1, 1, 1, 1]$ (i.e. $u = x_1 \rightarrow (x_3 \rightarrow x_2)$). The probabilities of any initial states transferring to x^* are listed in Table II.

It is worthy to point out that the network can only stabilize in probability at the state δ_8^5, which corresponds to $x_1 = 0$, $x_2 = 1$, and $x_3 = 1$. It indicates the system will head toward a complex attractor (SCC 49) with activation of the caspases and absence of IAP. Actually, it is known as the only attractor that all trajectories starting from any state will eventually reach

TABLE I
STABILITY IN PROBABILITY OF THE PBN BUILT IN EXAMPLE 1

$P\{x(t) = \delta_8^7 \| x(0)\}$ \ t ; $x(0)$	$t = 1$	$t = 2$	$t = 3$
$x(0) = \delta_8^1$	0	0	0.0205
$x(0) = \delta_8^2$	0	0.0487	0.0370
$x(0) = \delta_8^3$	0	0.1086	0.0522
$x(0) = \delta_8^4$	0	0	0.0493
$x(0) = \delta_8^5$	0.3295	0	0.0487
$x(0) = \delta_8^6$	0	0.0449	0.0198
$x(0) = \delta_8^7$	0	0	0.0888
$x(0) = \delta_8^8$	0	0	0.0174

TABLE II
STABILIZATION IN PROBABILITY OF THE PBCN GIVEN IN EXAMPLE 2

$P\{x(t)=\delta_8^5 \mid x(0),u\}$ \ t / $x(0)$	$t=1$	$t=2$	$t=3$
$x(0)=\delta_8^1$	0.18	0.3192	0.3912
$x(0)=\delta_8^2$	0.144	0.2675	0.3417
$x(0)=\delta_8^3$	0	0.0336	0.0924
$x(0)=\delta_8^4$	0	0	0.0269
$x(0)=\delta_8^5$	1	1	1
$x(0)=\delta_8^6$	0.24	0.4112	0.4952
$x(0)=\delta_8^7$	0.28	0.3892	0.4668
$x(0)=\delta_8^8$	0	0.0896	0.1478

it and remain in it for all subsequent time [57]. Within this attractor, the apoptotic decision is highly related to the varying stimulation of TNF. Both survival and apoptosis are possible, depending on the cellular state.

Note that $\delta_8^5 \in \mathcal{H}_1(\mathcal{L}, \delta_8^5)$. The stabilization in probability of the PBCN can be checked by Theorem 4, which is more easily verifiable than Theorem 2. By some easy computations, we can obtain that

$$\begin{aligned}
H_0 &= \{\delta_8^5\} \quad B_0 = [0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^\top \\
H_1 &= \{\delta_8^1, \delta_8^2, \delta_8^5, \delta_8^6, \delta_8^7\} \quad B_1 = [1\ 1\ 0\ 0\ 1\ 1\ 1\ 0]^\top \\
H_2 &= \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^6, \delta_8^7, \delta_8^8\} \quad B_2 = [1\ 1\ 1\ 0\ 1\ 1\ 1\ 1]^\top \\
H_3 &= \Delta_{2^3} \quad B_3 = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^\top.
\end{aligned}$$

It also indicates that the PBCN can stabilize at $x^* = \delta_8^5$ in probability from $T = 3$. According to Algorithm 3, the control matrix $G = \delta_2[d_1, d_2, \ldots, d_8]$ can be constructed as

$$\begin{aligned}
d_1 &\in \{s \mid B_0^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_8^1 = 1\} = \{1, 2\} \\
d_2 &\in \{s \mid B_0^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_8^2 = 1\} = \{1, 2\} \\
d_3 &\in \{s \mid B_1^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_8^3 = 1\} = \{2\} \\
d_4 &\in \{s \mid B_2^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_8^4 = 1\} = \{1, 2\} \\
d_5 &\in \{s \mid B_0^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_8^5 = 1\} = \{1, 2\} \\
d_6 &\in \{s \mid B_0^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_8^6 = 1\} = \{1, 2\} \\
d_7 &\in \{s \mid B_0^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_8^7 = 1\} = \{1, 2\} \\
d_8 &\in \{s \mid B_1^\top \ltimes_{\mathscr{B}} Blk_s(\mathcal{L}) \ltimes \delta_8^8 = 1\} = \{1\}.
\end{aligned}$$

It is the same as the one solved from Theorem 2. However, Theorem 4 gives a more feasible algorithm. It is an advantage, provided the condition $P\{x(t+1) = x^* \mid x(t) = x^*, u\} > 0$ satisfying for some u.

As discussed in Remark 3, for each given target state $x^* \in \Delta_{2^3}$, the total number of possible controllers is $(2^m)^{2^n} = 256$. For each candidate controller, we further need to check the condition derived in Theorem 1. Thus, it is a huge computational cost. In this example, n and m are relatively small. It still takes 3.14 s to obtain the result that the only stabilizable state is δ_8^5, and 64 feedback controllers can stabilize it. If Algorithm 3 is applicable, the computation burden will be reduced greatly (see Remark 6). Due to $\delta_8^5 \in \mathcal{H}_1(\mathcal{L}, \delta_8^5)$, Algorithm 3 can be employed in this example. As displayed earlier, only three sets (H_1, H_2, and H_3) are needed to construct. The 64 controllers can also be obtained, since d_1, d_2, and d_4–d_7 have two choices.

V. CONCLUSION

The stability in probability and stabilization in probability have been respectively discussed for the PBNs and PBCNs in this article. Compared with the traditional probability-one problems, the network is not required to converge to a predefined state in a determinate manner. Thus, it can be used to simulate a more realistic cellular system. Resorting to the STP technique, some necessary and sufficient conditions have been derived, which ensure the stability/stabilization in probability. Numerical examples have been built to illustrate the effectiveness and efficiency of the theoretical results.

Constructing reachable sets is critical to analyze the stability/stabilization of PBNs/PBCNs. Since the inclusion relation may not be satisfied for the reachable sets, it makes the lack of efficient algorithms to estimate the upper bound of S in Theorem 1 and to build the matrix $\mathbf{L}$ in Theorem 2 (see Remarks 3 and 4). One of our future works is to propose some new methods, which are not dependent on the reachable sets, to solve these problems. The transition probability matrix considering in this article is time invariant. It is interesting to investigate PBNs with time-varying transition probability matrices, which are more general and can reflect more realistic systems. However, it is a difficult problem, since the nonhomogeneous Markov chain is involved. This will be another future work.

REFERENCES

[1] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Comput. Biol.*, vol. 7, nos. 3–4, pp. 601–620, Aug. 2000.

[2] I. Shmulevich and W. Zhang, "Binary analysis and optimization-based normalization of gene expression data," *Bioinformatics*, vol. 18, no. 4, pp. 555–565, Apr. 2002.

[3] X. Zhou, X. Wang, and E. R. Dougherty, "Binarization of microarray data on the basis of a mixture model[1]," *Mol. Cancer Therapeutics*, vol. 2, no. 7, pp. 679–684, 2003.

[4] E. Lee, A. Salic, R. Krüger, R. Heinrich, and M. W. Kirschner, "The roles of APC and axin derived from experimental and theoretical analysis of the wnt pathway," *PLoS Biol.*, vol. 1, no. 1, p. e10, Oct. 2003.

[5] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, Mar. 1969.

[6] S. Kauffman, C. Peterson, B. Samuelsson, and C. Troein, "Random Boolean network models and the yeast transcriptional network," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 25, pp. 14796–14799, Dec. 2003.

[7] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*. London, U.K.: Springer, 2010.

[8] D. Cheng and H. Qi, "A linear representation of dynamics of Boolean networks," *IEEE Trans. Autom. Control*, vol. 55, no. 10, pp. 2251–2258, Oct. 2010.

[9] J. Lu, J. Zhong, D. W. C. Ho, Y. Tang, and J. Cao, "On controllability of delayed Boolean control networks," *SIAM J. Control Optim.*, vol. 54, no. 2, pp. 475–494, Jan. 2016.

[10] J. Lu, J. Zhong, C. Huang, and J. Cao, "On pinning controllability of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 61, no. 6, pp. 1658–1663, Jun. 2016.

[11] J. Zhong, J. Lu, T. Huang, and D. W. C. Ho, "Controllability and synchronization analysis of identical-hierarchy mixed-valued logical control networks," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3482–3493, Nov. 2017.

[12] Y. Li, J. Zhong, J. Lu, Z. Wang, and F. E. Alssadi, "On robust synchronization of drive-response Boolean control networks with disturbances," *Math. Problems Eng.*, vol. 2018, pp. 1–9, Aug. 2018.

[13] D. Cheng, H. Qi, Z. Li, and J. B. Liu, "Stability and stabilization of Boolean networks," *Int. J. Robust Nonlinear Control*, vol. 21, no. 2, pp. 134–156, Jan. 2011.

[14] R. Liu, J. Lu, Y. Liu, J. Cao, and Z.-G. Wu, "Delayed feedback control for stabilization of Boolean control networks with state delay," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3283–3288, 2018.

[15] H. Li and Y. Wang, "Output feedback stabilization control design for Boolean control networks," *Automatica*, vol. 49, no. 12, pp. 3641–3645, Dec. 2013.

[16] 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.

[17] Y. Li, B. Li, Y. Liu, J. Lu, Z. Wang, and F. E. Alsaadi, "Set stability and stabilization of switched Boolean networks with state-based switching," *IEEE Access*, vol. 6, pp. 35624–35630, 2018.

[18] H. Li and Y. Wang, "Lyapunov-based stability and construction of Lyapunov functions for Boolean networks," *SIAM J. Control Optim.*, vol. 55, no. 6, pp. 3437–3457, Jan. 2017.

[19] H. Li and X. Ding, "A control Lyapunov function approach to feedback stabilization of logical control networks," *SIAM J. Control Optim.*, vol. 57, no. 2, pp. 810–831, Mar. 2019.

[20] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, Feb. 2002.

[21] C. Huang, J. Lu, D. W. C. Ho, G. Zhai, and J. Cao, "Stabilization of probabilistic Boolean networks via pinning control strategy," *Inf. Sci.*, vol. 510, pp. 205–217, Feb. 2020.

[22] M. Xu, Y. Liu, J. Lou, Z.-G. Wu, and J. Zhong, "Set stabilization of probabilistic Boolean control networks: A sampled-data control approach," *IEEE Trans. Cybern.*, to be published.

[23] L. Lin, J. Cao, and L. Rutkowski, "Robust event-triggered control invariance of probabilistic Boolean control networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 1060–1065, Mar. 2020.

[24] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite-horizon control for probabilistic Boolean networks," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2375–2387, Jun. 2006.

[25] Y. Wu and T. Shen, "Policy iteration algorithm for optimal control of stochastic logical dynamical systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 2031–2036, May 2018.

[26] H. Chen, J. Liang, J. Lu, and J. Qiu, "Synchronization for the realization-dependent probabilistic Boolean networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 819–831, Apr. 2018.

[27] H. Chen, J. Liang, T. Huang, and J. Cao, "Synchronization of arbitrarily switched Boolean networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 612–619, Mar. 2017.

[28] P. Trairatphisan, A. Mizera, J. Pang, A. Tantar, J. Schneider, and T. Sauter, "Recent development and biomedical applications of probabilistic Boolean networks," *Cell Commun. Signaling*, vol. 11, no. 1, p. 46, 2013.

[29] X. Cheng, T. Mori, Y. Qiu, W.-K. Ching, and T. Akutsu, "Exact identification of the structure of a probabilistic Boolean network from samples," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 13, no. 6, pp. 1107–1116, Nov. 2016.

[30] R. Li, M. Yang, and T. Chu, "State feedback stabilization for probabilistic Boolean networks," *Automatica*, vol. 50, no. 4, pp. 1272–1278, Apr. 2014.

[31] Y. Liu, H. Chen, J. Lu, and B. Wu, "Controllability of probabilistic Boolean control networks based on transition probability matrices," *Automatica*, vol. 52, pp. 340–345, Feb. 2015.

[32] H. Li, Y. Wang, and P. Guo, "State feedback based output tracking control of probabilistic Boolean networks," *Inf. Sci.*, vols. 349–350, pp. 1–11, Jul. 2016.

[33] I. Golding, J. Paulsson, S. M. Zawilski, and E. C. Cox, "Real-time kinetics of gene activity in individual bacteria," *Cell*, vol. 123, no. 6, pp. 1025–1036, Dec. 2005.

[34] P. D. Williams, D. D. Pollock, and R. A. Goldstein, "Functionality and the evolution of marginal stability in proteins: Inferences from lattice simulations," *Evol. Bioinf.*, vol. 2, pp. 91–101, Nov. 2017.

[35] J. Paulsson, "Models of stochastic gene expression," *Phys. Life Rev.*, vol. 2, no. 2, pp. 157–175, Jun. 2005.

[36] M. Chaves, T. Eissing, and F. Allgower, "Bistable biological systems: A characterization through local compact input-to-state stability," *IEEE Trans. Autom. Control*, vol. 53, pp. 87–100, Jan. 2008.

[37] M. Chaves, T. Eissing, and F. Allgöwer, "Regulation of apoptosis via the NFκB pathway: Modeling and analysis," in *Dynamics on and of Complex Networks* (Modeling and Simulation in Science, Engineering and Technology). Boston, MA, USA: Birkhäuser, 2009, pp. 19–33.

[38] E. A. Komarova, K. Christov, A. I. Faerman, and A. V. Gudkov, "Different impact of p53 and p21 on the radiation response of mouse tissues," *Oncogene*, vol. 19, no. 33, pp. 3791–3798, Jul. 2000.

[39] S. Huang, "Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery," *J. Mol. Med.*, vol. 77, no. 6, pp. 469–480, Aug. 1999.

[40] M. A. L. Caetano and T. Yoneyama, "Boolean network representation of contagion dynamics during a financial crisis," *Phys. A, Stat. Mech. Appl.*, vol. 417, pp. 1–6, Jan. 2015.

[41] P. J. R. Torres, E. I. S. Mercado, and L. A. Rifón, "Probabilistic Boolean network modeling of an industrial machine," *J. Intell. Manuf.*, vol. 29, no. 4, pp. 875–890, 2018.

[42] R. Khasminskii, *Stochastic Stability of Differential Equations*, vol. 66. Berlin, Germany: Springer, 2011.

[43] L. E. Shaikhet, "Stability in probability of nonlinear stochastic systems with delay," *Math. Notes*, vol. 57, no. 1, pp. 103–106, Jan. 1995.

[44] D. Ding, Z. Wang, B. Shen, and G. Wei, "Event-triggered consensus control for discrete-time stochastic multi-agent systems: The input-to-state stability in probability," *Automatica*, vol. 62, pp. 284–291, Dec. 2015.

[45] C. Possieri and A. R. Teel, "Asymptotic stability in probability for stochastic Boolean networks," *Automatica*, vol. 83, pp. 1–9, Sep. 2017.

[46] J. Lu, J. Zhong, L. Li, D. W. C. Ho, and J. Cao, "Synchronization analysis of master-slave probabilistic Boolean networks," *Sci. Rep.*, vol. 5, no. 1, Aug. 2015, Art. no. 13437.

[47] Z. Hao, W. Xing-Yuan, and L. Rui, "Synchronization of asynchronous probabilistic Boolean network," *Chin. J. Phys.*, vol. 56, no. 5, pp. 2146–2154, Oct. 2018.

[48] S. Zhu, J. Lou, Y. Liu, Y. Li, and Z. Wang, "Event-triggered control for the stabilization of probabilistic Boolean control networks," *Complexity*, vol. 2018, pp. 1–7, Sep. 2018.

[49] F. Li and L. Xie, "Set stabilization of probabilistic Boolean networks using pinning control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 8, pp. 2555–2561, Aug. 2019, doi: 10.1109/TNNLS.2018.2881279.

[50] H. Tian and Y. Hou, "State feedback design for set stabilization of probabilistic Boolean control networks," *J. Franklin Inst.*, vol. 356, no. 8, pp. 4358–4377, May 2019.

[51] S. Zhu, J. Lu, Y. Liu, T. Huang, and J. Kurths, "Output tracking of probabilistic Boolean networks by output feedback control," *Inf. Sci.*, vol. 483, pp. 96–105, May 2019.

[52] L. Tong, Y. Liu, J. Lou, J. Lu, and F. E. Alsaadi, "Static output feedback set stabilization for context-sensitive probabilistic Boolean control networks," *Appl. Math. Comput.*, vol. 332, pp. 263–275, Sep. 2018.

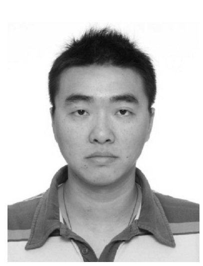
[53] Y. Liu, L. Wang, J. Lu, and J. Cao, "Sampled-data stabilization of probabilistic Boolean control networks," *Syst. Control Lett.*, vol. 124, pp. 106–111, Feb. 2019.

[54] J.-W. Gu, W.-K. Ching, T.-K. Siu, and H. Zheng, "On modeling credit defaults: A probabilistic Boolean network approach," *Risk Decis. Anal.*, vol. 4, no. 2, pp. 119–129, 2013.

[55] G. Giampieri, M. Davis, and M. Crowder, "Analysis of default data using hidden Markov models," *Quant. Finance*, vol. 5, no. 1, pp. 27–34, Feb. 2005.

[56] K. Kobayashi and K. Hiraishi, "An integer programming approach to optimal control problems in context-sensitive probabilistic Boolean networks," *Automatica*, vol. 47, no. 6, pp. 1260–1264, Jun. 2011.

[57] L. Tournier and M. Chaves, "Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics," *J. Theor. Biol.*, vol. 260, no. 2, pp. 196–209, Sep. 2009.



Chi Huang received the M.S. and Ph.D. degrees in mathematics from the City University of Hong Kong, Kowloon, Hong Kong, in 2009 and 2012, respectively.

He is currently an Associate Professor with the School of Economic Information and Engineering, Southwestern University of Finance and Economics, Chengdu, China. His research interests include Boolean networks, multiagent systems, and complex dynamical networks.

Dr. Huang was a recipient of the Best Paper Award from the Eighth Asian Control Conference in 2011.

Jianquan Lu (Senior Member, IEEE) received the B.S. degree in mathematics from Zhejiang Normal University, Zhejiang, China, in 2003, the M.S. degree in mathematics from Southeast University, Nanjing, China, in 2006, and the Ph.D. degree in applied mathematics from the City University of Hong Kong, Hong Kong, in 2009.

From 2010 to 2012, he was an Alexander von Humboldt Research Fellow with the Potsdam Institute for Climate Impact Research, Potsdam, Germany. He is currently a Professor with the School of Mathematics, Southeast University, Nanjing, China. He has published over 90 articles in refereed international journals. His current research interests include collective behavior in complex dynamical networks and multi-agent systems, logical networks, and hybrid systems.

Dr. Lu recipient of an Alexander von Humboldt Fellowship in 2010, the Program for New Century Excellent Talents in University at the The Ministry of Education, China, in 2010, and the Second Award of Jiangsu Provincial Progress in Science and Technology in 2016 as the First Project Member, and the First Award of the Jiangsu Provincial Progress in Science and Technology in 2010 as the Second Project Member. He is an Associate Editor of *Neural Processing Letters*, the *Journal of Franklin Institute*, and *Neural Computing and Applications*, and a Guest Editor of *Science China: Information Sciences*, *Mathematics and Computers in Simulation*, and the *IET Control Theory & Applications*. He was named a Highly Cited Researcher by Clarivate Analytics in 2018. He was elected the Most Cited Chinese Researchers by Elsevier from 2014 to 2018.

Guisheng Zhai (Senior Member, IEEE) received the B.S. degree from Fudan University, China, in 1988, and the M.E. and Ph.D. degrees from Kobe University, Japan, in 1993 and 1996, respectively.

After two years of industrial experience, he moved to Wakayama University, Wakayama, Japan, in 1998, and then to Osaka Prefecture University, Sakai, Japan, in 2004. He held visiting professor position at the University of Notre Dame, Notre Dame, IN, USA, Purdue University, West Lafayette, IN, USA, the Taiyuan University of Technology, Taiyuan, China, the Hubei University of Technology, Wuhan, China, and Fujian Normal University, Fuzhou, China. In 2010, he joined the Shibaura Institute of Technology, Saitama, Japan, where he currently is a Full Professor of mathematical sciences. His research interests include large-scale and decentralized control systems, robust control and applications, switched systems and switching control, networked control and multi-agent systems, neural networks, and signal processing.

Dr. Zhai is a member of the Institute of Systems, Control and Information Engineers (ISCIE), the Society of Instrument and Control Engineers (SICE), Japan Society for Simulation and Technology (JSST), and the Japan Society of Mechanical Engineers (JSME). He is on the Editorial Board of several academic journals, including the *IET Control Theory and Applications*, the *International Journal of Applied Mathematics and Computer Science*, the *Journal of Control and Decision*, the *Frontiers of Mechanical Engineering*, and *Science Nature*.

Jinde Cao (Fellow, IEEE) received the B.S. degree from Anhui Normal University, Wuhu, China, the M.S. degree from Yunnan University, Kunming, China, and the Ph.D. degree from Sichuan University, Chengdu, China, all in mathematics/applied mathematics, in 1986, 1989, and 1998, respectively.

He is currently an Endowed Chair Professor, the Dean of the School of Mathematics, and the Director of the Jiangsu Provincial Key Laboratory of Networked Collective Intelligence of China, Southeast University, Nanjing, China, and where he is also the Director of the Research Center for Complex Systems and Network Sciences.

Dr. Cao is a member of the Academy of Europe and the European Academy of Sciences and Arts, a fellow of the Pakistan Academy of Sciences, and the International Academy for Systems and Cybernetic Sciences (IASCYS) academician. He was a recipient of the National Innovation Award of China, the Obada Prize, and the Highly Cited Researcher Award in Engineering, Computer Science, and Mathematics by Thomson Reuters/Clarivate Analytics.

Guoping Lu received the B.S. degree from the Department of Applied Mathematics, Chengdu University of Science and Technology, Chengdu, China, in 1984, and the M.S. and Ph.D. degrees in applied mathematics from the Department of Mathematics, East China Normal University, Shanghai, China, in 1989 and 1998, respectively.

He is currently a Professor with the School of Electrical Engineering, Nantong University, Nantong, China. His current research interests include singular systems, multiagent systems, networked control, and nonlinear signal processing.

Matjaž Perc is Professor of physics with the University of Maribor, Maribor, Slovenia.

Dr. Perc is a member of Academia Europaea and the European Academy of Sciences and Arts. He was a recipient of the Young Scientist Award for Socio and Econophysics from the German Physical Society in 2015 and the 2017 USERN Laureate. In 2018, he received the Zois Award, which is the highest national research award in Slovenia. He is on the Editorial Board of the *New Journal of Physics*, the *Proceedings of the Royal Society A*, the *Journal of Complex Networks*, the *European Physical Journal B*, *Scientific Reports*, and the *Royal Society Open Science*. He is currently an Editor of *Physics Letters A*, *Chaos, Solitons & Fractals*, *EPL*, and *Frontiers in Physics*. He is among top 1% most cited physicists according to Clarivate Analytics.