

Policy Iteration Approach to the Infinite Horizon Average Optimal Control of Probabilistic Boolean Networks

Yuhu Wu^{ID}, *Member, IEEE*, Yuqian Guo^{ID}, *Member, IEEE*, and Mitsuru Toyoda^{ID}, *Member, IEEE*

Abstract—This article studies the optimal control of probabilistic Boolean control networks (PBCNs) with the infinite horizon average cost criterion. By resorting to the semitensor product (STP) of matrices, a nested optimality equation for the optimal control problem of PBCNs is proposed. The Laurent series expression technique and the Jordan decomposition method derive a novel policy iteration-type algorithm, where finite iteration steps can provide the optimal state feedback law, which is presented. Finally, the intervention problem of the probabilistic Ara operon in *E. coli*, as a biological application, is solved to demonstrate the effectiveness and feasibility of the proposed theoretical approach and algorithms.

Index Terms—Boolean networks (BNs), infinite horizon problem, logical networks, optimal control, probabilistic BNs (PBNs), semitensor product (STP) of matrix.

I. INTRODUCTION

BOOLEAN networks (BNs), as a special kind of discrete (logical) dynamical models with Boolean-valued variables, were first proposed by a theoretical biologist Kauffman [1] in 1969 to model and analyze the complex biological behavior in biological systems, including gene regularity networks [2]–[4]. Since BNs modeling may be the simplest representation of the relevant biological and physical concepts for some finite-state systems, BNs have been also used in various theoretical and practical applications, such as fault detection in logic circuits [5], [6], game theory [7], [8], combustion engines [9], and many other areas.

After introducing the semitensor product (STP) of matrix [10] to BNs, as an effective approach, some fundamental

concepts and properties, such as stability, stabilization, controllability, observability, synchronization, and sampled-data control, pinning control of Boolean or multivalued logical networks have been well discussed and exploited [11]–[18], in recent years.

To deal with randomness of gene regulatory networks [19], probabilistic BNs (PBNs) were proposed [20]. As discussed in [20]–[22], the PBN model can well capture the key qualitative features of gene regularity network with inherent biological uncertainty, where, at each time step, the updating rule of each gene is randomly selected from among several possible regularity rules.

The dynamical model inference and network identification for PBNs have great practical significance in bioinformatics since the inference and reconstruction of gene regulatory networks are key issues for genomic signal processing [23], [24]. An effective method for calculating the Boolean functions, and the corresponding selecting probabilities of a Boolean function in the PBN, based on network structure and steady-state probabilities, was designed in [25]. Recently, a tractable learning algorithm for identification of large-scale PBNs was introduced in [26], in the framework of stochastic conjunctive normal form, an equivalent representation for the PBN.

In recent years, the optimal control and optimization problem for Boolean control networks (BCNs) have received considerable attention [27]–[29]. A special finite horizon Mayer-type optimization problem for BCNs was discussed by Laschov and Margaliot [30]. In addition, they also investigated the minimum-time control of BCNs [31]. Finite horizon optimal control problems for PBNs and stochastic logical networks were investigated in [28] and [32], respectively. Integer programming algorithm [29] and polynomial optimization algorithm for the finite horizon optimal control problem of a PBN were developed by Kobayashi and Hiraishi [33] to reduce the computational complexity.

In general, analyzing the long (infinite) horizon criterion and designing the corresponding optimal controller are more challenging issues, comparing with the finite horizon problem. The basic criteria for infinite horizon problem of BCNs or PBCNs are twofold: the discounted and average cost criteria. Pal *et al.* [34] first investigated the infinite horizon discounted cost problem for PBNs and obtained theoretical results were successfully applied in intervention problem on melanoma gene-expression network. In addition, an improved

Manuscript received August 20, 2019; revised April 14, 2020; accepted July 9, 2020. Date of publication July 23, 2020; date of current version July 7, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61773090, in part by the Liao Ning Revitalization Talents Program under Grant XLYC1907100, and in part by the Japan Society for the Promotion of Science, Grants-in-Aid for Scientific Research, under Grant JP19K15019. (*Corresponding author: Mitsuru Toyoda.*)

Yuhu Wu is with the Key Laboratory of Intelligent Control and Optimization for Industrial Equipment of Ministry of Education, Dalian University of Technology, Dalian 116024, China, and also with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: wuyuhu@dlut.edu.cn).

Yuqian Guo is with the School of Automation, Central South University, Changsha 410083, China (e-mail: gyuqian@csu.edu.cn).

Mitsuru Toyoda is with the Department of Mechanical Systems Engineering, Tokyo Metropolitan University, Tokyo 191-0065, Japan (e-mail: toyoda@tmu.ac.jp).

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.2020.3008960

dynamical programming, which provides a finite time convergence algorithm for the discounted infinite optimization problem of PBCNs, was developed in [35]. Furthermore, a new policy iteration-type algorithm, which solves the discounted infinite horizon problems of PBCNs, was derived in [36], and it has been successfully utilized to design feedback law for the residual gas fraction control in internal combustion engine [37].

Applying topology properties of trajectories and the graph theory, the infinite horizon problem for deterministic BCNs was first addressed by Zhao *et al.* [38]. Using a recursive algorithm, Fornasini and Valcher solved the average infinite horizon optimization problem for deterministic BCNs as the limit of the corresponding finite horizon optimization one in [39]. The policy iteration approach for the infinite horizon optimal control for deterministic BCNs was also given in [40]. By introducing the optimal input-state transfer graph, Zhu *et al.* [41] successfully reduced both computational complexity and space complexity in finding optimal controllers for deterministic BCNs.

Pal *et al.* [34] first investigated the optimal control problem for PCBN, and a policy iteration algorithm was deduced under the assumption that the PBCN is ergodic (or recurrent), which requires that the transition matrix of PBCN for every stationary policy consists of a single recurrent class. As mentioned in [34], the context-sensitive PBN satisfies the ergodic assumption, and accordingly, the policy iteration algorithm given in [34] can work effectively for the context-sensitive PBN. However, it was found that the optimal criteria given by [34, Th. 5] and the corresponding policy iteration algorithm are no longer applicable for the general PBCN (please see Example 13). As far as the authors know, there are still no works reported solving the general nonergodic case, which is the main motivation of this work. The results of this work can be regarded as a generalization to probabilistic BCNs (PBCNs) of the results that we obtained in [40] for deterministic BCNs. This generalization is nontrivial because when designing the optimal controller for PBCNs, one must to carefully address the conditional transition probabilities resulting from selection of update the Boolean functions, especially in the nonergodic case.

The main contributions of this work can be briefly summed up in the following aspects.

- 1) By applying the technique of the STP, a nested type optimality criterion (see Theorem 10) for the infinite horizon problem of PBCNs with average cost is derived. Compared with the optimality criterion given in [34], which requires the ergodic assumption, the nested optimality criterion proposed in this work can applied to arbitrary PBCN without any requirement (please see Example 13).
- 2) By resorting to the Jordan decomposition technique (see Proposition 14) and the Laurent type series expression (see Lemma 17), a policy iteration algorithm (Algorithm 1) is deduced. Compared with the value iteration algorithm as given in [39], the theoretical analysis on the proposed iteration algorithm guarantees that finite

TABLE I
NOTATIONS

Notations	Definitions
$\mathbb{Z}_{>0}$ ($\mathbb{Z}_{\geq 0}$)	Set of positive (nonnegative) integers
\mathcal{D}	Boolean domain $\{\text{True} \equiv 1, \text{False} \equiv 0\}$
I_k	$k \times k$ identity matrix
δ_n^i	i th column of I_n
Δ_n	Set of columns of I_n
$\mathcal{R}^{m \times n}$	Set of $m \times n$ real matrices
$\text{Col}_i(A)$	i th column of matrix A
A^T	Transpose of matrix A
$\delta_n[i_1, i_2, \dots, i_m]$	Matrix A with $\text{Col}_s(A) = \delta_n^{i_s}$
\prod	Cartesian product
\otimes	Kronecker product
\times	Semi-tensor product
$[1, N]$	Set of integers k satisfying $1 \leq k \leq N$
$ A $	Cardinality of set A

step iterations deduce a stationary state feedback optimal policy (see Remark 20).

This article is organized as follows: An equivalent matrix expression of the model considered in this work is presented in Section III, after introducing the problem formulation in Section II. The main results of this article are derived in Sections IV and V. Then, the probabilistic intervention problem of Ara operon in *E. coil*, as a practical biological application, is solved by the proposed algorithm in Section VI. Finally, brief conclusions are provided in Section VII. For clarity of presentation, some technical proofs are presented in the Appendix.

II. PRELIMINARIES AND PROBLEM FORMULATION

This section summarizes basic concepts about the PBNs and the average optimal control problem for PBNs.

A. Definitions and Notations

The notations used in this article are listed in Table I. For statement ease, the following definitions will be used.

- 1) A partial order relation $a \preceq b$ means $[a]_i \leq [b]_i, \forall i \in [1, s]$. Furthermore, $a \not\preceq b$ means $a \preceq b$ and $[a]_{i_0} \neq [b]_{i_0}$ for some i_0 .
- 2) A matrix $A \in \mathcal{R}^{M \times N}$ is called a logical matrix if its columns $\text{Col}(A) \subset \Delta_M$. Then, any logical matrix A has the form $A = [\delta_M^{i_1}, \delta_M^{i_2}, \dots, \delta_M^{i_N}]$, and briefly defined as $A = \delta_M[i_1, i_2, \dots, i_N]$. The set of $M \times N$ logical matrices is denoted by $\mathcal{L}_{M \times N}$.
- 3) The STP [10] of $A \in \mathcal{R}^{m \times n}$ and $B \in \mathcal{R}^{p \times q}$, denoted as $A \times B$, is defined as

$$A \times B := (A \otimes I_{s/n})(B \otimes I_{s/p}) \quad (1)$$

where s is the least common multiple of n and p , and \otimes is the Kronecker product. The symbol \times may be omitted without causing confusion.

- 4) The power-reducing matrix is given by $M_{\text{pr}}^n = \text{diag}[\delta_{2^n}^1, \delta_{2^n}^2, \dots, \delta_{2^n}^{2^n}]$, i.e., a block diagonal matrix with diagonal elements $\delta_{2^n}^1, \dots, \delta_{2^n}^{2^n}$. It deduces an equation $x \times x = M_{\text{pr}}^n x$, for any $x \in \Delta_{2^n}$.

B. Boolean Network

A BN, which is a directed network containing binary (Boolean) logical-valued state nodes, can be represented by a set of nodes $V = \{x_1, x_2, \dots, x_n\}$ and a set of Boolean functions $F = \{f_1, f_2, \dots, f_n\}$, where the state $x_i(t)$ of node $x_i, i \in [1, n]$ belongs to \mathcal{D} at each time t . The update rule of $x_i(t)$ is determined by the Boolean function $f_i(x_{i_1}, x_{i_2}, \dots, x_{i_k})$ with k specified input nodes, and the value of f_i is assigned to next state of node x_i . In general, k could be varying as a function of i , but, without loss of generality, we assume that for each x_i , the corresponding update rule function $f_i : \mathcal{D}^n \rightarrow \mathcal{D}$, by allowing unnecessary nodes in update rule function f_i to be fictitious. For an update rule function f , the logical node x_i is fictitious if $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) = f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ for all $(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \in \mathcal{D}^{n-1}$. Hence, the dynamics of a BN of V and F can be described as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \dots, x_n(t)) \\ x_2(t+1) = f_2(x_1(t), \dots, x_n(t)) \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \dots, x_n(t)). \end{cases}$$

C. Probabilistic Boolean Control Network

A PBCN is a directed network containing Boolean logical-valued state nodes $V = \{x_1, \dots, x_n\}$ and input control nodes $U = \{u_1, \dots, u_m\}$. In a PBCN, the update rule of state node x_i is regulated by one Boolean function, which is randomly selected from a set of Boolean functions. More precisely, define a collection of Boolean function sets \mathcal{F} and the corresponding collection of probability set Υ as follows.

- 1) $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n\}$, with $\mathcal{F}_i = \{f_i^j\}_{j=1}^{c(i)}$, where $c(i)$ is the number of possible update logical rules for node x_i , and $f_i^j : \mathcal{D}^n \times \mathcal{D}^m \rightarrow \mathcal{D} (j = 1, \dots, c(i))$ are possible update logical functions for node x_i .
- 2) $\Upsilon = \{\Upsilon_1, \Upsilon_2, \dots, \Upsilon_n\}$, with $\Upsilon_i = \{r_i^j\}_{j=1}^{c(i)}$, where r_i^j is the probability that the logical function f_i^j will be chosen as the update law for node x_i . Note that

$$\sum_{j=1}^{c(i)} r_i^j = 1.$$

In the summary, the dynamics of a PBCN with n state nodes $V = \{x_1, \dots, x_n\}$ and m input nodes $U = \{u_1, \dots, u_m\}$ can be described as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t)) \\ x_2(t+1) = f_2(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t)) \\ \vdots \\ x_n(t+1) = f_n(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t)) \end{cases} \quad (2)$$

with $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n\}$, $\Upsilon = \{\Upsilon_1, \Upsilon_2, \dots, \Upsilon_n\}$, where, for each $i \in [1, n]$

$$f_i \in \mathcal{F}_i = \{f_i^1, \dots, f_i^{c(i)}\} \quad (3)$$

with the probability of f_i choosing f_i^j is

$$\Pr\{f_i = f_i^j\} = r_i^j \in \Upsilon_i. \quad (4)$$

D. Average Optimal Control Problem

Now, consider the class of policies, which consists of an infinite sequence of control laws $\pi = \{\mu_t : t \in \mathbb{Z}_{\geq 0}\}$, where a control law $\mu_t : \mathcal{D}^n \rightarrow \mathcal{D}^m, t \in \mathbb{Z}_{\geq 0}$ maps logical states $x(t)$ onto control input $u(t) = \mu_t(x(t))$ in such a way that $\mu_t(x(t)) \in \mathcal{D}^m$ for all $x(t) \in \mathcal{D}^n$. For notational brevity, we refer to $\pi_\mu = \{\mu, \mu, \dots\}$ as the stationary policy μ .

Given an initial state x_0 with a policy $\pi = \{\mu_0, \mu_1, \dots\}$, consider the infinite horizon average expected cost

$$J_\pi(x_0) := \lim_{T \rightarrow \infty} \mathbf{E} \frac{1}{T} \sum_{t=0}^{T-1} g(x(t), \mu(x(t))) \quad (5)$$

where the notation \mathbf{E} means the expectation, and $g : \mathcal{D}^n \times \mathcal{D}^m \rightarrow \mathcal{R}$ is the per-step cost function.

The set of all policies π is denoted by Π , that is, the set of all sequences of functions $\pi = \{\mu_0, \mu_1, \dots\}$. Then, the optimal average cost function J^* is given by

$$J^*(x_0) = \inf_{\pi \in \Pi} J_\pi(x_0), \quad x_0 \in \Delta_N. \quad (6)$$

The aim of the infinite horizon average optimal control for PBCNs is to find an optimal policy $\pi^* \in \Pi$, which achieves the optimal cost J^* , that is

$$J_{\pi^*}(x_0) = J^*(x_0), \quad \text{for all } x_0 \in \mathcal{D}^n.$$

Example 1: Consider a PBCN (2)–(4) with three state nodes $V = \{x_1, x_2, x_3\}$, one input nodes $U = \{u_1\}$, and an update dynamics

$$\begin{cases} x_1(t+1) = f_1(x_1(t), x_2(t), x_3(t), u_1(t)) \\ x_2(t+1) = f_2(x_1(t), x_2(t), x_3(t), u_1(t)) \\ x_3(t+1) = f_3(x_1(t), x_2(t), x_3(t), u_1(t)) \end{cases} \quad (7)$$

where $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3\}$, and $\Upsilon = \{\Upsilon_1, \Upsilon_2, \Upsilon_3\}$, with

$$\begin{cases} f_1^1(x_1, x_2, x_3, u_1) = (x_1 \wedge x_3) \vee \neg u_1 \\ f_2^1(x_1, x_2, x_3, u_1) = \neg x_1 \vee u_1 \\ f_2^2(x_1, x_2, x_3, u_1) = x_2 \vee (\neg x_1 \wedge u_1) \\ f_3^1(x_1, x_2, x_3, u_1) = \neg x_1 \\ f_3^2(x_1, x_2, x_3, u_1) = \neg x_1 \vee x_3 \end{cases} \quad (8)$$

and accordingly $c(1) = 1$, and $c(2) = c(3) = 2$, respectively.

As described in Fig. 1, the update rule of state x_2 with (or without) self-looped influence is given by f_2^2 (or f_2^1) assuming that the probability having self-looped influence for x_2 is 0.4, i.e., $r_2^1 = \Pr\{f_2 = f_2^1\} = 0.6$ and $r_2^2 = \Pr\{f_2 = f_2^2\} = 0.4$. Similarly, assume that the probability having self-looped influence for x_3 is 0.3. Hence, $\mathcal{F}_1 = \{f_1^1\}$, $\mathcal{F}_2 = \{f_2^1, f_2^2\}$, $\mathcal{F}_3 = \{f_3^1, f_3^2\}$, $\Upsilon_1 = \{r_1^1 = 1\}$, $\Upsilon_2 = \{r_2^1 = 0.6, r_2^2 = 0.4\}$, $\Upsilon_3 = \{r_3^1 = 0.3, r_3^2 = 0.7\}$.

Consider the optimal average control problem (6) for this PBCN, with the following cost function:

$$g(x_1, x_2, x_3, u_1) = \begin{cases} 2 + x_1, & \text{if } u_1 = 0, \\ 5 + 3x_1 - 4x_3, & \text{if } u_1 = 1. \end{cases} \quad (9)$$

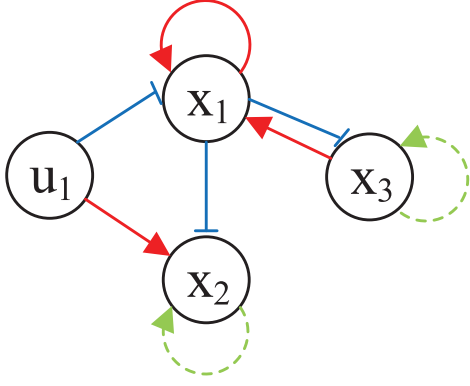


Fig. 1. Network graph of Example 1, where red \rightarrow and blue \rightarrow express the activation and inhibition relationship, respectively, and dotted green \rightarrow indicates that the probability of generating the loop is less than 1.

III. MATRIX EXPRESSION OF MODEL

As illustrated in Section II, the PBCNs are expressed by the combination of logical functions and their stochastic switching. Since the logical functions are originally intractable, this section provides a systematic way to deal with PBCNs by exploiting STP techniques.

Since there are $c(i)$ possible update logical functions f_i^j for each node $x_i \in V$, the number of possible realization of PBCN is

$$C_0 = \prod_{i=1}^n c(i).$$

We give a lexicographically order to possible realizations of PBCN, defining the following special matrix, introduced as [42], [43]

$$\Lambda = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & \cdots & 1 & 2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & c(n) \\ 1 & 1 & \cdots & 2 & 1 \\ 1 & 1 & \cdots & 2 & 2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 2 & c(n) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c(1) & c(2) & \cdots & c(n-1) & c(n) \end{bmatrix} \quad (10)$$

where a row α ($\alpha = 1, \dots, C_0$) of $\Lambda \in \mathcal{R}^{C_0 \times n}$ corresponds to the possible update logical functions

$$\{f_1^{\Lambda_{\alpha 1}}, f_2^{\Lambda_{\alpha 2}}, \dots, f_n^{\Lambda_{\alpha n}}\} \quad (11)$$

with $\Lambda_{\alpha j}$ denoting the (α, j) th entry of the matrix Λ .

Then, according to (3) and (4), the probability that the network α is selected as a realization of the PBCN is

$$\Pr[\alpha] := \Pr(\text{network } \alpha \text{ is selected}) = \prod_{j=1}^n r_j^{\Lambda_{\alpha j}}. \quad (12)$$

Furthermore, a Boolean variable $X \in \mathcal{D}$ is identified with a vector $x \in \Delta_2$ in the following form: $1 \sim \delta_2^1, 0 \sim \delta_2^2$. Under this vector identification, the whole states and inputs are expressed by an STP of elements as $x(t) = \times_{i=1}^n x_i(t) \in \Delta_N$,

with $N := 2^n$, and $u(t) = \times_{j=1}^m u_j(t) \in \Delta_M$, with $M := 2^m$, respectively. Accordingly, logical state space \mathcal{D}^n and input space \mathcal{D}^m can also be rewritten as Δ_N and Δ_M , respectively.

Applying the STP of matrix defined in (1), a logical function can be represented in an algebraic form.

Lemma 2 (10, Th. 3.2): Assume that $f : \mathcal{D}^p \rightarrow \mathcal{D}^q$ is a logical function, and let $y = f(x_1, x_2, \dots, x_p) \in \mathcal{D}^q$, with $(x_1, x_2, \dots, x_p) \in \mathcal{D}^p$. Then, there exists a unique logical matrix $M_f \in \mathcal{L}_{2^q \times 2^p}$, such that f can be rewritten in a multilinear form as

$$f(x_1, x_2, \dots, x_p) = M_f \times_{i=1}^p x_i. \quad (13)$$

The matrix M_f is called the structure matrix of the logical function f , and (13) is called the algebraic expression of f .

Assume that the structure matrix of the logical function f_i^j is M_i^j for each $i \in [1, n]$, $j \in [1, c(i)]$; then, based on [44, Th. 2] and Lemma 2, we can obtain, for each $\alpha \in [1, C_0]$, the following algebraic expression of α possible realizations of PBCN as

$$x(t+1) = L[\alpha] \times u(t) \times x(t) \quad (14)$$

where $L[\alpha] \in \mathcal{L}_{N \times NM}$ is calculated as

$$\text{Col}_k(L[\alpha]) := \times_{i=1}^n \text{Col}_k(M_i^{\Lambda_{\alpha i}}) \quad \forall k \in [1, NM]. \quad (15)$$

Definition 3: For a PBCN (2)–(4), define

$$\Gamma := \sum_{\alpha=1}^{C_0} \Pr[\alpha] L[\alpha] \quad (16)$$

which is called the transition matrix of the PBCN, where $p[\alpha]$ and $L[\alpha]$ are given by (12) and (15), respectively.

For a given $u = \times_{j=1}^m u_j \in \Delta_M$ at a time step t , denote by $p_{ij}(u)$ the transition probability from a logical state δ_N^i to a next logical state δ_N^j under the control $u(t) = u$

$$p_{ij}(u) := \Pr(x(t+1) = \delta_N^j | x(t) = \delta_N^i, u(t) = u) \quad (17)$$

for all $\delta_N^i, \delta_N^j \in \Delta_N$. It is noticed that the transition probabilities $p_{ij}(u)$ satisfy $\sum_{j=1}^N p_{ij}(u) = 1 \quad \forall i \in [1, N], u \in \Delta_M$. The following fact implies that the transition probabilities of a PBCN can be directly calculated by the transition matrix Γ .

Lemma 4: For any $\delta_N^i, \delta_N^j \in \Delta_N$, and $u \in \Delta_M$, we have

$$p_{ij}(u) = (\delta_N^j)^\top \Gamma \times u \times \delta_N^i. \quad (18)$$

Proof: See the Appendix. \blacksquare

Lemma 5: For a given PBCN (2)–(4), the evolution dynamics of expectation of state $x(t)$ can be expressed by the following linear form:

$$\mathbf{E}x(t+1) = \Gamma u(t) \mathbf{E}x(t) \quad (19)$$

where Γ is the transition matrix of PBCN (2)–(4).

Proof: See the Appendix. \blacksquare

The per-step cost function $g : \mathcal{D}^n \times \mathcal{D}^m \rightarrow \mathcal{R}$ in the formulation (5) can be expressed by a matrix expression as¹

$$g(x, u) = x^\top G u \quad \forall x \in \Delta_N, u \in \Delta_M \quad (20)$$

¹As considered in [39], an equivalent linear form of the per-step cost function $g : \Delta_N \times \Delta_M \rightarrow \mathcal{R}$ can be given as $g(x, u) = c^\top \times u \times x$, where $c = (c_1, \dots, c_{MN})^\top \in \mathcal{R}^{MN}$ with $c_{(j-1)N+i} = g(\delta_N^i, \delta_M^j)$, $i = 1, \dots, N$, $j = 1, \dots, M$.

where $G = (G_{i,j})_{N \times M} = (g(\delta_N^i, \delta_M^j))_{N \times M}$ is called the cost matrix.

Example 6: Recall Example 1. Since, in this example, $c(1) = 1, c(2) = c(3) = 2$, we have $C_0 = \prod_{i=1}^3 c(i) = 4$ and

$$\Lambda = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \end{bmatrix} \in \mathcal{R}^{4 \times 3} \quad (21)$$

by Definition (10). Then, according to (12), we get

$$\begin{aligned} \Pr[\alpha = 1] &= \prod_{j=1}^3 r_j^{\Lambda_{1j}} = r_1^{\Lambda_{11}} r_2^{\Lambda_{12}} r_3^{\Lambda_{13}} \\ &= r_1^1 r_2^1 r_3^1 = 1 \times 0.6 \times 0.3 = 0.18 \end{aligned}$$

and similarly, $\Pr[\alpha = 2] = 0.42, \Pr[\alpha = 3] = 0.12, \Pr[\alpha = 4] = 0.28$. Furthermore, applying Lemma 2, one can obtain the structure matrix $M_{f_1^1} \in \mathcal{L}_{2 \times 2^4}$ of f_1^1 as $M_{f_1^1} = \delta_2[2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, 2]$ and, similarly, can obtain $M_{f_2^1}, M_{f_3^1}, M_{f_3^2}$, respectively. As a result, according to (15) and (16), we get the transition matrix Γ of this example, given by (23), as shown at the bottom of the page. Furthermore, according to Lemma 4, the transition probability diagram with fixed control input $u = \delta_2^1$ and $u = \delta_2^2$ is shown in Fig. 2.

In addition, according to (20), the cost matrix of (9) is

$$G = \begin{bmatrix} 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 5 & 1 & 5 & 1 & 8 & 4 & 8 & 4 \end{bmatrix}^\top. \quad (22)$$

IV. AVERAGE OPTIMALITY CRITERION

In this section, an optimality criterion, called a nested condition, for the avarice optimal control is derived without a conventional ergodic assumption.

The set of all feedback logical laws $\mu : \Delta_N \rightarrow \Delta_M$ is denoted by \mathcal{U} , that is, $\mathcal{U} = \{\mu \mid \mu : \Delta_N \rightarrow \Delta_M\}$. One can easily get that the capacity of \mathcal{U} is $|\mathcal{U}| = M^N$; by noticing, the state space Δ_N and control space Δ_M are both finite. In the framework of the vector formulation, the control law can be regarded as a logical function from Δ_N to Δ_M . Therefore, referring to Lemma 2, we will first present the following fundamental result.

Proposition 7: For an arbitrary control law $\mu \in \mathcal{U}$, there exists a unique logical matrix $K_\mu \in \mathcal{L}_{M \times N}$ that satisfies

$$\mu(x) = K_\mu x \quad \forall x \in \Delta_N \quad (24)$$

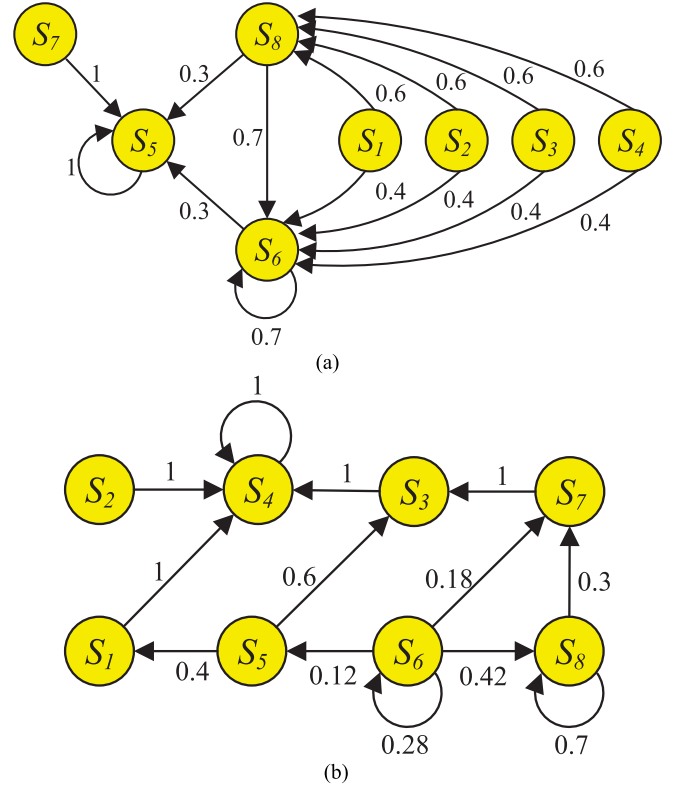


Fig. 2. Transition probability diagram for Example 6, where S_1, \dots, S_8 represent states $\delta_1^1, \dots, \delta_8^8$, respectively. (a) When $u = \delta_2^1$. (b) When $u = \delta_2^2$.

and it is called the structure feedback matrix of μ .

For any given $\mu \in \mathcal{U}$, we define the matrix Γ_μ associated with μ , as

$$\Gamma_\mu = \Gamma \times K_\mu M_{pr}^n \quad (25)$$

which includes all the information about the transition probability of the PBN under feedback control law μ , where K_μ is the structure matrix of μ , and M_{pr}^n is the power-reducing matrix given in Section II.

For any given $\mu \in \mathcal{U}$ with a structure matrix K_μ , since $\mu(\delta_N^i) \in \Delta_M, \forall \delta_N^i \in \Delta_N$, the following equation:

$$\Gamma \times \mu(\delta_N^i) \times \delta_N^i = \Gamma \times K_\mu M_{pr}^n \delta_N^i = \Gamma_\mu \times \delta_N^i \quad (26)$$

holds.

Hence, for a given the state feedback control $u(t) = \mu(x(t)) = K_\mu x(t)$, the evolution dynamics (19) of PBCN

$$\Gamma = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0.3 & 1 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.4 & 0.4 & 0.4 & 0 & 0.7 & 0 & 0.7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.6 & 0.6 & 0.6 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathcal{R}^{2^3 \times 2^4} \quad (23)$$

becomes a closed-loop system as

$$\mathbf{E}x(t+1) = \Gamma_\mu \mathbf{E}x(t) \quad (27)$$

where Γ_μ is given by (25).

Lemma 8: For any policy $\pi = \{\mu_0, \mu_1, \dots\}$, the vector form of the expected cost $J_\pi = [J_\pi(\delta_N^1), \dots, J_\pi(\delta_N^N)]^\top$ can be expressed as

$$J_\pi = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \prod_{k=0}^{t-1} \Gamma_{\mu_k}^\top g_{\mu_t} \quad (28)$$

and especially, for a stationary policy $\pi^\mu = \{\mu, \mu, \dots\}$

$$J_\mu = J_{\pi^\mu} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} (\Gamma_\mu^\top)^t g_\mu \quad (29)$$

where the special cost vector g_μ associated with feedback control law $\mu \in \mathcal{U}$, defined as

$$g_\mu = [g(\delta_N^1, \mu(\delta_N^1)), \dots, g(\delta_N^N, \mu(\delta_N^N))]^\top. \quad (30)$$

Proof: See the Appendix. ■

Lemma 9: For a given $\mu \in \mathcal{U}$, if there exist two vectors $(J, h) \in \mathcal{R}^N \times \mathcal{R}^N$ that satisfy the following equations:

$$\begin{cases} \Gamma_\mu^\top J = J \\ g_\mu + (\Gamma_\mu^\top - I_N)h = J. \end{cases} \quad (31)$$

$$(32)$$

Then, we have $J^* \leq J$.

Proof: Left-multiplying (32) by Γ_μ^\top and applying equality (31), we get

$$J = \Gamma_\mu^\top J = \Gamma_\mu^\top g_\mu + \Gamma_\mu^\top (\Gamma_\mu^\top - I_N)h.$$

Repeating the process above with induction, we deduce the following equation:

$$J = (\Gamma_\mu^\top)^n g_\mu + (\Gamma_\mu^\top)^n (\Gamma_\mu^\top - I_N)h \quad (33)$$

for any $n \in \mathbb{Z}_{\geq 0}$. Summing those expressions from 0 to $n-1$, we have

$$J = \frac{1}{n} \sum_{t=0}^{n-1} (\Gamma_\mu^\top)^t g_\mu + \frac{1}{n} [(\Gamma_\mu^\top)^n - I_N]h.$$

Recalling that Γ_μ is a stochastic matrix, we have $\|\Gamma_\mu\| = \|\Gamma_\mu^\top\| \leq 1$, and accordingly, $\|(\Gamma_\mu^\top)^n - I_N\| \leq \|\Gamma_\mu\|^n + 1 \leq 2$. Hence

$$\lim_{n \rightarrow \infty} \frac{1}{n} \|[(\Gamma_\mu^\top)^n - I_N]h\| \leq \lim_{n \rightarrow \infty} \frac{2\|h\|}{n} = 0$$

and applying (29), we obtain that

$$\begin{aligned} [J]_i &= \lim_{n \rightarrow \infty} \left[\frac{1}{n} \sum_{t=0}^{n-1} (\Gamma_\mu^\top)^t g_\mu \right]_i = [J_{\pi^\mu}]_i \\ &\geq \inf_{\pi \in \Pi} [J_\pi]_i = [J^*]_i \end{aligned}$$

for all $i \in [1, N]$. The proof is complete. ■

Now, it is ready to present an optimality criterion for the average optimal control problem of PBCNs as the following theorem.

Theorem 10: Assume there are two vectors $(J, h) \in \mathcal{R}^N \times \mathcal{R}^N$ such that, for any $i \in [1, N]$, the following condition holds

$$\min \begin{cases} \min_{j \in [1, M]} \left[\Gamma \delta_M^j \delta_N^i \right]^\top J = (\delta_N^i)^\top J, \\ \min_{j \in \Theta_i} G_{ij} - [\Gamma \delta_M^j \delta_N^i]^\top h = (\delta_N^i)^\top (J + h), \\ \text{with } \Theta_i = \{j \in [1, M] \mid [\Gamma \delta_M^j \delta_N^i]^\top J = (\delta_N^i)^\top J\}. \end{cases} \quad (34a)$$

$$(34b)$$

Then, the vector J is the optimal cost vector of the average optimal problem (6), that is, $J = J^*$.

Remark 11: In the optimality conditions (34), the control index set Θ_i , which is the control candidate domain of the minimization problem in the left-hand side of (34a), depends on the index set of control inputs that achieve the minimum in (34a) when J is substituted into it. This is the reason that the optimality conditions (34) are said to be nested.

Proof of Theorem 10: Nested condition (34) implies $\Theta_i \cap \Phi_i \neq \emptyset$, for each $i \in [1, N]$, where the set Θ_i is given in (34a), and Φ_i is defined as

$$\Phi_i = \{j \in [1, M] \mid G_{ij} - [\Gamma \delta_M^j \delta_N^i]^\top h = (\delta_N^i)^\top (J + h)\}. \quad (35)$$

In other words, the existence of a minimizer of the left-hand side of (34a) is guaranteed by the nested condition, and such an index $j_i \in \Theta_i \cap \Phi_i$ is chosen as a feedback law $\mu' \in \mathcal{U}$

$$\mu'(\delta_N^i) = \delta_M^{j_i}, j_i \in \Theta_i \cap \Phi_i$$

for each $i \in [1, N]$. Recalling definition of the sets Θ_i , Φ_i , and the cost vector $g_{\mu'}$ given by (30), we have for each $i \in [1, N]$

$$\begin{cases} (\delta_N^i)^\top [\Gamma_{\mu'}^\top J] = (\delta_N^i)^\top J \\ (\delta_N^i)^\top [g_{\mu'} - J + (\Gamma_{\mu'}^\top - I_N)h] = 0 \end{cases} \quad (36)$$

$$(37)$$

which is equivalent to (31) and (32) for μ' . Thus, according to Lemma 9, we get

$$[J]_i \geq [J^*]_i \quad \forall i \in [1, N]. \quad (38)$$

Next, we will prove that if vectors $(J, h) \in \mathcal{R}^N \times \mathcal{R}^N$ satisfy condition (34), then there is a constant $C \geq 0$ such that vectors J and $\tilde{h} = h + CJ$ satisfy the following condition:

$$\min \begin{cases} \min_{j \in [1, M]} [\Gamma \delta_M^j \delta_N^i]^\top J = (\delta_N^i)^\top J \\ \min_{j \in [1, M]} G_{ij} - [\Gamma \delta_M^j \delta_N^i]^\top \tilde{h} = (\delta_N^i)^\top (J + \tilde{h}) \end{cases} \quad (39a)$$

$$(39b)$$

for each $i \in [1, N]$. We refer condition (39a) as the modified optimality condition of condition (34a). Notice condition (39a) is the same as the condition (34a). If vectors (J, h) , given in (34), satisfy (39b), then we simply chose $\tilde{h} = h$ with $C = 0$. Suppose J and h , given in (34), do not satisfy (39b), then there exist $i_0 \in [1, N]$, and $j_0 \in [1, M] \setminus \Theta_{i_0}$ such that

$$G_{i_0 j_0} - [\Gamma \delta_M^{j_0} \delta_N^{i_0}]^\top h < (\delta_N^{i_0})^\top (J + h)$$

which is equivalent to

$$C_1 := G_{i_0 j_0} - (\delta_N^{i_0})^\top J + (\delta_N^{i_0})^\top [(\Gamma \delta_M^{j_0})^\top - I_N]h < 0.$$

Furthermore, since $j_0 \in [1, M] \setminus \Theta_{i_0}$, recalling the definition of Θ_{i_0} given by (34a), we have

$$C_2 := (\delta_N^{i_0})^\top [((\Gamma \delta_M^{j_0})^\top - I_N)J] > 0.$$

Now, set $\tilde{h} := h + C_3 J$, where $C_3 > 0$ will be given later. Then

$$\begin{aligned} G_{i_0 j_0} - (\delta_N^{i_0})^\top [J - ((\Gamma \delta_M^{j_0})^\top - I_N)\tilde{h}] \\ = G_{i_0 j_0} - (\delta_N^{i_0})^\top [J - ((\Gamma \delta_M^{j_0})^\top - I_N)h] \\ + (\delta_N^{i_0})^\top [C_3((\Gamma \delta_M^{j_0})^\top - I_N)J] = C_1 + C_3 C_2. \end{aligned}$$

Hence, taking C_3 large enough such that $C_3 > (|C_1|/C_2)$, we have

$$(\delta_N^{i_0})^\top [G_{i_0 j_0} - J + ((\Gamma \delta_M^{j_0})^\top - I_N)\tilde{h}] > 0. \quad (40)$$

Because both of the space of state and control input are finite, there exists large enough C_3 for which (39b) holds for all $i \in [1, N]$ and $\mu \in \mathcal{U}$.

For any given $\mu \in \mathcal{U}$, if $\mu(\delta_N^i) = \delta_M^j$, then based on (26) and (30), we get $\Gamma \delta_M^j \delta_N^i = \Gamma_\mu \delta_N^i$ and $G_{ij} = (\delta_N^i)^\top g_\mu$, which implies that

$$\min_{j \in [1, M]} [\Gamma \delta_M^j \delta_N^i]^\top J = \min_{\mu \in \mathcal{U}} [(\delta_N^i)^\top]^\top \Gamma_\mu^\top J \quad (41)$$

$$\min_{j \in [1, M]} G_{ij} = \min_{\mu \in \mathcal{U}} [(\delta_N^i)^\top]^\top g_\mu. \quad (42)$$

Hence, condition (39) is equal to

$$\min \begin{cases} \mu \in \mathcal{U} [\Gamma_\mu \delta_N^i]^\top J = (\delta_N^i)^\top J, & (43a) \\ \min_{\mu \in \mathcal{U}} (\delta_N^i)^\top [g_\mu + (\Gamma_\mu - I_N)^\top \tilde{h}] = (\delta_N^i)^\top J. & (43b) \end{cases}$$

By noticing $(\delta_N^i)^\top J = J_i$, condition (43a) implies, for all $i \in [1, N]$, that

$$\begin{cases} [J]_i \leq [\Gamma_{\mu_0}^\top J]_i & (44) \\ [J]_i \leq [g_{\mu_0} + (\Gamma_{\mu_0}^\top - I_N)\tilde{h}]_i & (45) \end{cases}$$

and applying condition (43b) to μ_1 implies that

$$[J]_i \leq [g_{\mu_1} + (\Gamma_{\mu_1}^\top - I_N)\tilde{h}]_i \quad \forall i = 1, \dots, N. \quad (46)$$

Left-multiplying (46) by $\Gamma_{\mu_0}^\top$ and using inequality (44) imply that we obtain

$$[J]_i \leq [\Gamma_{\mu_0}^\top J]_i \leq [\Gamma_{\mu_0}^\top g_{\mu_1} + \Gamma_{\mu_0}^\top (\Gamma_{\mu_1}^\top - I_N)\tilde{h}]_i$$

for any $i \in [1, N]$. Then, repeating the abovementioned process with induction, we get for any $n \in \mathbb{Z}_{\geq 0}$

$$[J]_i \leq [\Gamma_{\mu_0}^\top \cdots \Gamma_{\mu_{n-1}}^\top g_{\mu_n} + \Gamma_{\mu_0}^\top \cdots \Gamma_{\mu_{n-1}}^\top (\Gamma_{\mu_n}^\top - I_N)\tilde{h}]_i$$

where set $\Gamma_{\mu_{-1}} = I_N$, if $n = 0$. Therefore, summing up those expressions from 1 to $n + 1$, we obtain $\forall i \in [1, N]$

$$[J]_i \leq \frac{1}{n+1} \left[\sum_{t=0}^n \prod_{k=-1}^{t-1} \Gamma_{\mu_k}^\top g_{\mu_t} \right]_i + \frac{[\Gamma_{\mu_0}^\top \cdots \Gamma_{\mu_n}^\top - I_N]\tilde{h}]_i}{n+1}.$$

In addition, applying $\|(\Gamma_{\mu_0}^\top \cdots \Gamma_{\mu_{n-1}}^\top \Gamma_{\mu_n}^\top - I_N)\tilde{h}\| \leq 2\|\tilde{h}\|$, we have, for all $i \in [1, N]$

$$[J]_i \leq \lim_{n \rightarrow \infty} \left[\frac{1}{n+1} \sum_{t=0}^n \prod_{k=0}^{t-1} \Gamma_{\mu_k}^\top g_{\mu_t} \right]_i = [J_\pi(x_0)]_i.$$

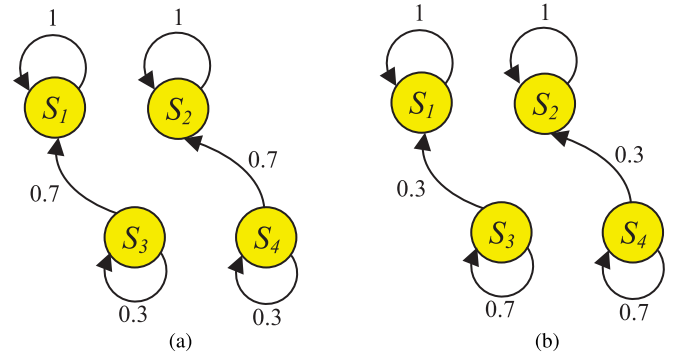


Fig. 3. Transition probability diagram for Example 13, where S_1, \dots, S_4 represent state $\delta_1^1, \dots, \delta_4^1$, respectively. (a) When $u = \delta_2^1$. (b) When $u = \delta_2^2$.

Then, due to the arbitrariness of π , we deduce that

$$[J]_i \leq \inf_{\pi \in \Pi} [J_\pi]_i = [J^*]_i \quad (47)$$

for all $i \in [1, N]$. Finally, we get $J = J^*$, combining (34) and (47), and finish the proof. ■

Remark 12: As mentioned in Section I, Pal *et al.* [34, Th. 5] have given an optimal criterion for the average optimal control for PBNs, and it has been successfully applied in the case of context-sensitive PBCNs. However, it requires the ergodic assumption and cannot be applied in the general case, as illustrated in the following example.

Example 13: Consider the PBCN with two state nodes $V = \{x_1, x_2\}$, one input nodes $U = \{u_1\}$, and update dynamics

$$\begin{cases} x_1(t+1) = f_1(x_1(t), x_2(t), u_1(t)) \\ x_2(t+1) = f_2(x_1(t), x_2(t), u_1(t)). \end{cases} \quad (48)$$

Here, $\mathcal{F} = \{\mathcal{F}_1, \mathcal{F}_2\}$, and $\Upsilon = \{\Upsilon_1, \Upsilon_2\}$, with

$$\begin{aligned} \mathcal{F}_1 &= \{f_1^1, f_1^2\}, \quad \mathcal{F}_2 = \{f_2^1\} \\ \Upsilon_1 &= \{r_1^1 = 0.7, r_1^2 = 0.3\}, \quad \Upsilon_2 = \{r_2^1 = 1\}, \text{ and} \\ \begin{cases} f_1^1(x_1, x_2, u_1) = x_1 \wedge u_1, \\ f_1^2(x_1, x_2, u_1) = x_1 \wedge \neg u_1, \\ f_2^1(x_1, x_2, u_1) = x_2. \end{cases} \end{aligned}$$

Then, according to the analysis given in Section IV, one easily obtains

$$\Lambda = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \in \mathcal{R}^{C_0 \times n}$$

with $C_0 = \prod_{i=1}^2 c(i) = 2, c(1) = 2$, and $c(2) = 1$. Furthermore, the transition matrix of this PBCN is calculated as

$$\Gamma = \begin{bmatrix} 1 & 0 & 0.7 & 0 & 1 & 0 & 0.3 & 0 \\ 0 & 1 & 0 & 0.7 & 0 & 1 & 0 & 0.3 \\ 0 & 0 & 0.3 & 0 & 0 & 0 & 0.7 & 0 \\ 0 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0.7 \end{bmatrix}. \quad (49)$$

The corresponding transition probability diagram is shown in Fig. 3.

We consider the optimal average control problem (6) for this PBCN, where the per-step cost function g is given as

$$G = \begin{bmatrix} 0 & 3 & 1 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix}^\top. \quad (50)$$

If one tries to solve this optimal control problem based on the result of [34], it needs to solve the following optimal criteria given in [34, Th. 5] for all $i \in [1, 4]$:

$$\lambda + h(i) = \min_{u \in \Delta_2} \left[g(\delta_4^i, u) + \sum_{j=1}^4 p_{ij}(u)h(j) \right]. \quad (51)$$

According to transition matrix Γ in (49) and cost matrix G in (50), the first and second equations of the abovementioned equations are

$$\begin{cases} \lambda + h(1) = \min\{0 + h(1), 1 + h(1)\} \\ \lambda + h(2) = \min\{3 + h(2), 2 + h(2)\}. \end{cases} \quad (52)$$

The first equation of (52) indicates that $\lambda = 0$, which, on substitution into the second equation of (52), shows that the system is inconsistent. The abovementioned analysis implies that the criterion given in [34, Th. 5] cannot solve this optimal control problem. The main reason is that the algorithm in [34] requires the ergodicity of the PBCN (please see [34, Th. 4]). However, it is easily observed that the PBCN considered in this example is not ergodic. Indeed, it is observed that there is no possible way to go from the state δ_4^1 to state δ_4^2 , under both of control inputs $u = \delta_2^1$ and $u = \delta_2^2$, as shown in Fig. 3, implying this PBCN is not ergodic since the ergodicity of a PBCN requires that it is possible to go from each state to all the states under each stationary policy.

In the following, we will see that the policy iteration algorithm based on Theorem 10 can successfully solve the problem in this example.

V. POLICY ITERATION ALGORITHM

By using matrix analysis techniques, including the Jordan decomposition and the Laurent series expansion, a policy iteration algorithm for the optimal control formulation given in Section II is proposed in this section. At the end of the section, the convergence of the algorithm with finite iteration steps is examined with the example shown in Sections II–IV.

Several preliminary results, which will be used to prove the correctness of the policy iteration algorithm, are introduced first.

Proposition 14: For any control law $\mu \in \mathcal{U}$, the rank of $I_N - \Gamma_\mu^\top \in \mathcal{R}^{N \times N}$ is less than N , that is

$$r := \text{Rank}(I_N - \Gamma_\mu^\top) < N.$$

In addition, there exist a nonsingular upper triangular matrix $S_\mu \in \mathcal{R}^{r \times r}$ and a nonsingular matrix $V_\mu \in \mathcal{R}^{N \times N}$ such that

$$I_N - \Gamma_\mu^\top = V_\mu \begin{bmatrix} 0 & 0 \\ 0 & S_\mu \end{bmatrix} V_\mu^{-1}. \quad (53)$$

Proof: By definition of Γ_μ , it is obvious that $\sum_{j=1}^N [I_N - \Gamma_\mu^\top]_{ij} = 0$, for any $i = 1, 2, \dots, N$, which implies that $\lambda = 1$ is an eigenvalue of Γ_μ^\top , and $\mathbf{1} = [1, 1, \dots, 1]^\top \in \mathcal{R}^N$ is a

solution of homogeneous linear equation $(I_N - \Gamma_\mu^\top)x = 0$. Hence, $r < N$.

In addition, since $r = \text{Rank}(I_N - \Gamma_\mu^\top) < N$, applying the Jordan decomposition theorem (see [45, Th. 3.1.11]), there are a nonsingular upper triangular matrix $S_\mu \in \mathcal{R}^{r \times r}$ and a nonsingular matrix $V_\mu \in \mathcal{R}^{N \times N}$ such that (53) holds. ■

According to the Jordan decomposition form of the matrix $I_N - \Gamma_\mu^\top$ given in Proposition 14, we will deduce a computational formula for J_μ as follows.

Lemma 15: For any $\mu \in \mathcal{U}$, define the corresponding limiting matrix Γ_μ^\sharp as

$$\Gamma_\mu^\sharp = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} (\Gamma_\mu^\top)^t. \quad (54)$$

Then the following hold:

- 1) The Cesaro type limit, defined by the right-hand side of (54), always exists, and accordingly, the limiting matrix Γ_μ^\sharp satisfies the following properties:

$$\Gamma_\mu^\sharp = V_\mu \begin{bmatrix} I_{N-r} & 0 \\ 0 & 0 \end{bmatrix} V_\mu^{-1} \quad (55)$$

$$\Gamma_\mu^\sharp \Gamma_\mu^\top = \Gamma_\mu^\sharp = \Gamma_\mu^\top \Gamma_\mu^\sharp. \quad (56)$$

- 2) The vector J_μ defined in (29) can be calculated by

$$J_\mu = V_\mu \begin{bmatrix} I_{N-r} & 0 \\ 0 & 0 \end{bmatrix} V_\mu^{-1} g_\mu. \quad (57)$$

Proof: Indeed, by $\|\Gamma_\mu\| = \|\Gamma K_\mu\| \leq 1$, we have $\|\Gamma_\mu^\top\| = \|\Gamma_\mu\| \leq 1$. Hence

$$\lim_{T \rightarrow \infty} \frac{\|(\Gamma_\mu^\top)^T - I_N\|}{T} \leq \lim_{T \rightarrow \infty} \frac{\|\Gamma_\mu\|^T + 1}{T} = \lim_{T \rightarrow \infty} \frac{2}{T} = 0.$$

Then, from the definition (54) of limiting matrix Γ_μ^\sharp , we get

$$\Gamma_\mu^\sharp \Gamma_\mu^\top = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T (\Gamma_\mu^\top)^t = \Gamma_\mu^\sharp + \lim_{T \rightarrow \infty} \frac{(\Gamma_\mu^\top)^T - I_N}{T} = \Gamma_\mu^\sharp$$

that means the first equation of (56) holds, and similarly, we also deduce that $\Gamma_\mu^\sharp = \Gamma_\mu^\top \Gamma_\mu^\sharp$.

According to the Jordan decomposition (53), we have

$$\Gamma_\mu^\top = V_\mu \begin{bmatrix} I_{N-r} & 0 \\ 0 & I_r - S_\mu \end{bmatrix} V_\mu^{-1}. \quad (58)$$

Then, using again the definition (54) of the limit matrix Γ_μ^\sharp , we get

$$\Gamma_\mu^\sharp = V_\mu \begin{bmatrix} I_{N-r} & 0 \\ 0 & L_{22}^\sharp \end{bmatrix} V_\mu^{-1} \quad (59)$$

where $L_{22}^\sharp = \lim_{T \rightarrow \infty} (1/T) \sum_{t=0}^{T-1} (I_r - S_\mu)^t$. Furthermore, recalling $\Gamma_\mu^\top \Gamma_\mu^\sharp = \Gamma_\mu^\sharp$ of (56), we get $S_\mu L_{22}^\sharp = 0$, from (58) and (59). Then, from the fact that the upper triangular matrix $S_\mu \in \mathcal{R}^{r \times r}$ is nonsingular, we obtain $L_{22}^\sharp = 0$. Hence, from (59), we obtain (55), which also implies that the Cesaro type limit (54) exists.

Finally, noticing that $J_\mu = \Gamma_\mu^\sharp g_\mu$ from (29), we obtain (57) by (55). ■

Lemma 16: For any control law $\mu \in \mathcal{U}$, define $h_\mu := H_\mu^\sharp g_\mu$, with

$$H_\mu^\sharp := (I_N - \Gamma_\mu^\top + \Gamma_\mu^\sharp)^{-1} (I - \Gamma_\mu^\sharp). \quad (60)$$

Then, h_μ can be calculated by

$$h_\mu = V_\mu \begin{bmatrix} 0 & 0 \\ 0 & S_\mu^{-1} \end{bmatrix} V_\mu^{-1} g_\mu. \quad (61)$$

Proof: From the Jordan decomposition (53) and (55), we obtain

$$(I - \Gamma_\mu^\top + \Gamma_\mu^\sharp) = V_\mu \begin{bmatrix} I_{N-r} & 0 \\ 0 & S_\mu \end{bmatrix} V_\mu^{-1}. \quad (62)$$

This implies that the matrix $I - \Gamma_\mu^\top + \Gamma_\mu^\sharp$ is nonsingular. Then, according to $\Gamma_{22}^\sharp = 0$, we get

$$(I - \Gamma_\mu^\top + \Gamma_\mu^\sharp)^{-1} (I - \Gamma_\mu^\sharp) = V_\mu \begin{bmatrix} 0 & 0 \\ 0 & S_\mu^{-1} \end{bmatrix} V_\mu^{-1}. \quad (63)$$

Therefore, noticing definition (60) of H_μ^\sharp , we obtain (61). ■

We also present the Laurent series expansion of $(I_N - \alpha \Gamma_\mu^\top)^{-1}$, which plays an important role in the proof of the correction of the policy iteration algorithm given in the following.

Lemma 17: For any given feedback logical control law $\mu \in \mathcal{U}$ and $0 < \alpha < 1$, we have

$$(I_N - \alpha \Gamma_\mu^\top)^{-1} = \frac{1}{1 - \alpha} \Gamma_\mu^\sharp + H_\mu^\sharp + F(\alpha, \mu) \quad (64)$$

where $F(\alpha, \mu) \in \mathcal{R}^{N \times N}$ denotes a matrix that will converges to zero when $\alpha \rightarrow 1$.

Proof: See the Appendix. ■

Based on Theorem 10, we now present a policy iteration algorithm for PBCN.

Remark 18: In Algorithm 1, the initial policy μ_0 is a preset variable given by a user. As discussed in proof of correctness of Algorithm 1 afterward, an arbitrary initial policy μ_0 converges to an optimal policy. Indeed, the initial policy of Example 5.1 is simply chosen as $\mu(x) = \delta_2^i, \forall x$.

The next proposition characterizes the monotonicity property between two control laws in \mathcal{U} .

Proposition 19: For any two control laws $\mu, \eta \in \mathcal{U}$, we define the following three special subsets of Δ_N :

$$\mathcal{A}_0(\mu, \eta) = \{\delta_N^i \mid \mu(\delta_N^i) = \eta(\delta_N^i)\} \quad (65)$$

$$\mathcal{A}_1(\mu, \eta) = \{\delta_N^i \mid [\Gamma_\eta^\top J_\mu]_i < [\Gamma_\mu^\top J_\mu]_i\} \quad (66)$$

$$\mathcal{A}_2(\mu, \eta) = \left\{ \delta_N^i \mid \begin{array}{l} [\Gamma_\mu^\top J_\mu]_i = [\Gamma_\eta^\top J_\mu]_i, \text{ and} \\ [g_\eta + \Gamma_\eta^\top h_\mu]_i < [g_\mu + \Gamma_\mu^\top h_\mu]_i \end{array} \right\}. \quad (67)$$

If two different feedback laws $\mu, \eta \in \mathcal{U}$ satisfy the following condition

$$\mathcal{A}_0(\mu, \eta) \cup \mathcal{A}_1(\mu, \eta) \cup \mathcal{A}_2(\mu, \eta) = \Delta_N \quad (68)$$

then

$$\lim_{\alpha \uparrow 1} J_\eta^\alpha \not\asymp \lim_{\alpha \uparrow 1} J_\mu^\alpha \quad (69)$$

where, for all $0 < \alpha < 1$

$$J_\eta^\alpha := (I_N - \alpha \Gamma_\eta^\top)^{-1} g_\eta. \quad (70)$$

Algorithm 1 Policy Iteration for Optimal Control Problem (6) of PBCN (2)–(4)

Step 0. Initialization:

- 1) Compute Γ and G based on (16) and (20).
- 2) Guess an initial policy $\mu_0 \in \mathcal{U}$.

Step 1. Policy Evaluation:

For a given stationary policy μ_n , compute the corresponding J_{μ_n}, h_{μ_n} based on (57), (61).

Step 2. Policy Improvement:

2.A Choose stationary policy μ_{n+1} such that its structure matrix $K_{n+1} = L_N[q_1^{n+1}, \dots, q_N^{n+1}]$ satisfies

$$q_i^{n+1} \in \operatorname{argmin}_{j=1, \dots, M} \left\{ [\Gamma^\top J_{\mu_n}]_{(j-1)N+i} \right\}, \quad \forall i \in [1, N]$$

and set $q_i^{n+1} = q_i^n$, if possible.

2.B If $\mu_{n+1} = \mu_n$, go to **(2.C)**; else return to **Step 1**.

2.C Choose stationary policy μ_{n+1} such that its structure matrix $K_{n+1} = L_N[q_1^{n+1}, \dots, q_N^{n+1}]$ satisfies

$$q_i^{n+1} \in \operatorname{argmin}_{j \in [1, M]} \{ G_{ij} + h_{\mu_n}^\top \operatorname{Col}_{(j-1)N+i}(\Gamma) \}, \quad \forall i \in [1, N]$$

and set $q_i^{n+1} = q_i^n$, if possible.

2.D If $\mu_{n+1} = \mu_n$, stop and set $\mu^* = \mu_n$; else return to **Step 1** and repeat the process.

Proof: See the Appendix. ■

The condition (68), which can guarantee the monotonicity relation (69) between two different control laws η and μ , will help to prove that the policy improvement process given in Step 2 of Algorithm 1 is greedy.

Proof of Correctness of Algorithm 1: Given an initial stationary policy μ_0 , Algorithm 1 generates a sequence of stationary policy $\{\mu_0, \mu_1, \dots\}$. For any μ_n and μ_{n+1} , if $\mathcal{A}_0(\mu_n, \mu_{n+1}) = \Delta_N$, then by definition (65), we get $\mu_n = \mu_{n+1}$. Then, based on substep (2.D), Algorithm 1 terminates.

By selecting the structure matrix K_{n+1} of μ_{n+1} at substep (2.A) of Algorithm 1, and using STP properties, we deduce that, for any $i \in [1, N]$

$$\begin{aligned} & [\Gamma_{\mu_{n+1}}^\top J_{\mu_n}]_i \\ &= (\delta_N^i)^\top \Gamma_{\mu_{n+1}}^\top J_{\mu_n} = (\delta_N^i)^\top \Phi_n^\top K_{n+1}^\top \Gamma^\top J_{\mu_n} \\ &= (\delta_N^i)^\top \times (\delta_N^i)^\top K_{n+1}^\top \Gamma^\top J_{\mu_n} = (\delta_N^i)^\top \times (K_{n+1} \delta_N^i)^\top \Gamma^\top J_{\mu_n} \\ &= (\delta_N^i)^\top \times (\delta_M^{q_i^{n+1}})^\top \Gamma^\top J_{\mu_n} = \min_{\delta_M^j \in \Delta_M} (\delta_N^i)^\top \times (\delta_M^j)^\top \Gamma^\top J_{\mu_n} \\ &= \min_{\delta_M^j \in \Delta_M} J_{\mu_n}^\top \operatorname{Col}_{(j-1)N+i}(\Gamma) = \min_{\mu \in \mathcal{U}} (\delta_N^i)^\top \Gamma_\mu^\top J_{\mu_n} \\ &= \min_{\mu \in \mathcal{U}} [\Gamma_\mu^\top J_{\mu_n}]_i. \end{aligned}$$

Here, the simple fact $\operatorname{Col}_{(j-1)N+i}(\Gamma) = \Gamma \times (\delta_M^j)^\top \times (\delta_N^i)^\top$ is used. In addition, from substep (2.C) of Algorithm 1, we get, for any $i \in [1, N]$

$$[h_{\mu_{n+1}} + J_{\mu_{n+1}}]_i = \min_{\mu \in \mathcal{U}_i} [g_\mu + \Gamma_\mu^\top h_{\mu_n}]_i.$$

On the other hand, if $\mathcal{A}_0(\mu_n, \mu_{n+1}) \neq \Delta_N$, then substeps (2.A) and (2.C) of Algorithm 1 guarantee that

$\Delta_N \setminus \mathcal{A}_0(\mu_n, \mu_{n+1}) \subset \mathcal{A}_1(\mu_n, \mu_{n+1})$ or $\Delta_N \setminus \mathcal{A}_0(\mu_n, \mu_{n+1}) \subset \mathcal{A}_2(\mu_n, \mu_{n+1})$, respectively, which implies that $\mathcal{A}_0(\mu, \eta) \cup \mathcal{A}_1(\mu, \eta) \cup \mathcal{A}_2(\mu, \eta) = \Delta_N$. Thus, based on Proposition 19, we obtain the following fact:

$$\lim_{\alpha \uparrow 1} J_{\mu_{n+1}}^\alpha \preceq \lim_{\alpha \uparrow 1} J_{\mu_n}^\alpha.$$

Since the set of stationary policies is finite as $|\mathcal{U}| = M^N$, Algorithm 1 must terminate in a finite iterations.

In addition, $\mu_{n+1} = \mu_n$, at termination step μ_n , and it implies that

$$\begin{cases} 0 = [(\Gamma_{\mu_n}^\top - I_N)J_{\mu_n}]_i = [(\Gamma_{\mu_{n+1}}^\top - I_N)J_{\mu_n}]_i \\ = \min_{\mu \in \mathcal{U}} \{[(\Gamma_\mu^\top - I_N)J_{\mu_n}]_i\} \\ [h_{\mu_n} + J_{\mu_n}]_i = [g_{\mu_n} + \Gamma_{\mu_n}^\top h_{\mu_n}]_i \\ = \min_{\mu \in \mathcal{U}} [g_\mu + \Gamma_\mu^\top h_{\mu_n}]_i \end{cases}$$

for any $i \in [1, N]$. Hence, applying Theorem 10, we obtain $J_{\mu_n} = J^*$, which implies that the stationary policy $\{\mu_n\}$ is optimal. ■

Remark 20: The abovementioned correctness analysis for Algorithm 1 also implies that finite iterations can deduce an optimal stationary policy since the number of stationary policies, which are considered as control candidates in the optimization formulation, is finite. At termination step n , an equation $J_{\mu_n} = J^*$ holds, and $\{\mu_n\}$ is a stationary optimal policy. As a result, the following fact holds.

Corollary 21: For the PBCN (2)–(4), there exists a stationary optimal policy $\pi^* = \{\mu^*, \mu^*, \dots\}$, which solves the optimal control problem (6).

Example 22: Now, we apply Algorithm 1 to solve the problem given in Example 1, following from Example 6.

Initialization:

- 1) In this example, Γ and G are given by (23) and (22), respectively.
- 2) Choose the initial stationary policy μ_0 as $\mu_0(x) = \delta_2[1, 1, 1, 1, 1, 1, 1]x$, for all $x \in \Delta_8$.

Policy Evaluation: Applying Lemma 16 to get V_{μ_0} , S_{μ_0} , J_{μ_0} , and h_{μ_0} , respectively, as

$$\begin{aligned} V_{\mu_0} &= \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1.4286 & 0 & 0.4286 & 0 & 0 & -1 & 0 \\ 0 & 1.4286 & 0 & -1.4286 & 0 & 0 & -1.4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.6 & 0 & 0 & 0 & 1.4 & 0 \end{bmatrix} \\ S_{\mu_0} &= \begin{bmatrix} 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

$J_{\mu_0} = [0, 0, 0, 0, 20, 0, 0, 0]^\top$, and $h_{\mu_0} = [2, 2, 2, 2, -173/3, 116/3, 3, 7.8]^\top$.

Policy Improvement:

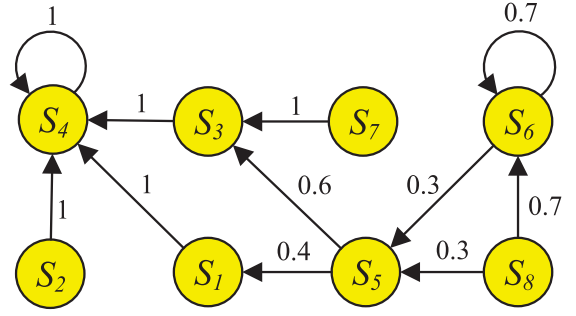


Fig. 4. Transition probability diagram under the optimal policy μ^* .

- 1) In substep (2.A), get $K_1 = \delta_2[1, 1, 1, 1, 1, 1, 1, 1]$; as a structure matrix μ_1 .
- 2) In substep (2.B), since $\mu_1 = \mu_0$, go to (2.C).
- 3) In substep (2.C), the policy μ_1 is renewed with structure matrix $K_1 = \delta_2[1, 2, 1, 2, 1, 1, 1, 1]$.
- 4) In substep (2.D), since $\mu_1 \neq \mu_0$, return to Step 1, and proceed to the next iteration.

In substep (2.D) of the fourth iteration, it is easily checked that $\mu_4 = \mu_3$. Thus, we obtain the optimal stationary policy $\pi^* = \pi^{\mu_3}$ with $\bar{J}_{\mu_3} = Q\bar{J}_{\mu_3}$. Hence, μ_3 is optimal with $K_3 = \delta_2[2, 2, 2, 2, 2, 1, 2, 1] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$, and the corresponding optimal performance is $J^* = J_{\mu_2} = [1, 1, 1, 1, 1, 1, 1, 1]^\top$. The transition probability diagram under this optimal policy μ^* is shown in Fig. 4.

Example 23: Reconsider the problem given in Example 13. Here γ and G are given in (49) and (50), respectively. We choose an initial policy μ_0 as $\mu_0(x) = \delta_2[2, 2, 2, 2]x$. Then, according to Algorithm 1, similar to the abovementioned example, we easily obtain that the optimal cost of this problem is $J^* = [0, 1, 0, 1]^\top$, and the corresponding optimal stationary policy is $\mu^* = \mu_1$ with structure matrix $K_{\mu^*} = \delta_2[1, 2, 1, 2]$, after one iteration step.

VI. APPLICATION TO INTERVENTION PROBLEM OF ARA OPERON NETWORK

In this section, the optimal control on intervention of arabinose (Ara) operon in *E. coil* is performed as an application to practical biological networks. The contributions [46] and [47] have studied the regulation of Ara operon in *E. coil* and given an observation that the regulatory protein is determined in the presence and absence of arabinose. Fig. 5 shows a graphical interpretation of Ara Network. As investigated in [48], the update logics of Ara operon network can be discredited by the following Boolean equations (71):

$$\begin{cases} f_A = A_e \wedge T \\ f_{A_{em}} = (A_{em} \wedge T) \vee A_e \\ f_{A_{ra+}} = (A_{em} \vee A) \wedge A_{ra-} \\ f_C = \neg G_e \\ f_E = M_S \\ f_D = \neg A_{ra+} \wedge A_{ra-} \\ f_{M_S} = A_{ra+} \wedge C \wedge \neg D \\ f_{M_T} = A_{ra+} \wedge C \\ f_T = M_T \end{cases} \quad (71)$$

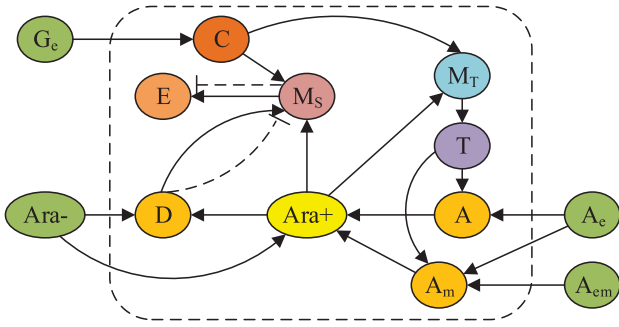


Fig. 5. Boolean model of Ara operon in *E. coli*.

where M_S denotes the mRNA of the structural genes (araBAD), M_T is the mRNA of the transport genes (araEFGH), E is the enzymes AraA, AraB, and AraD, coded for by the structural genes, T is the transport protein, coded for by the transport genes, A is the intracellular arabinose (high levels), A_{em} is the intracellular arabinose (at least medium levels), C is the cAMP-CAP protein complex, D is the DNA loop, and Ara_+ is the arabinose-bound AraC protein. For more details of the biological justification of each update function of (71), please see [48]. There exist four Boolean control variables: the AraC protein (unbound to arabinose), the extracellular glucose, the extracellular arabinose (at least medium levels), and the extracellular arabinose (high levels); those variables are denoted by Ara_- , G_e , A_{em} , and A_e , respectively. Furthermore, the variable D is 1 if the DNA is looped and 0 if it is not looped. All the other variables represent the concentration levels of the corresponding gene products: 1 denotes “present” or “high concentration” and 0 denote “absent” or “low (basal) concentration.”

Consider the context-sensitive case, as discussed in [42], with perturbation on the influence of node M_S to logical function f_E , and the influence of node D to the logical function f_{M_S} , as shown by dotted line in Fig. 5. More precisely, assume that, at each time step, f_E and f_{M_S} are switched randomly to $\hat{f}_E = \neg M_S$ and $\hat{f}_{M_S} = Ara_+ \wedge C \wedge D$, respectively, with probability 0.5, by flipping the state of node M_S and D . Then, the BN of Ara operon becomes a PBCN, and there are four possible realizations of this network.

- 1) No perturbation in network (71).
- 2) Only f_E is switched to \hat{f}_E in network (71).
- 3) Only f_{M_S} is switched to \hat{f}_{M_S} in network (71).
- 4) Both f_E and f_{M_S} are switched to \hat{f}_E and \hat{f}_{M_S} , respectively, in network (71).

Corresponding to the abovementioned four realization, the state transition graphs of the lac operon with fixed control variables $((A_e, A_{em}, Ara_-, G_e) = (0, 1, 1, 0))$ are presented in Figs. 6–9. To give the vector expressions of the nodes, set $(A, A_{em}, Ara_+, C, E, D, M_S, M_T, T) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$ and $(A_e, A_{em}, Ara_-, G_e) = (u_1, u_2, u_3, u_4)$. Then, as presented in Section II, based on STP of matrix, we obtain the linear form BN expression of Ara operon (71) as $x(t+1) = Lu(t)x(t)$, with a structure matrix $L \in \mathcal{L}_{2^9 \times 2^{13}}$.

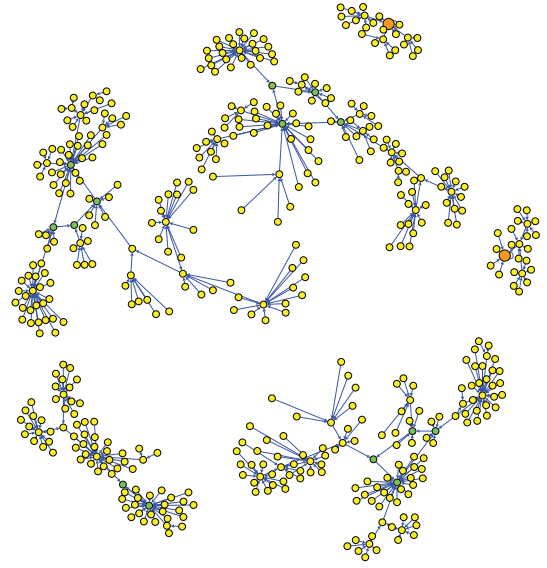


Fig. 6. State transition graph of the lac operon with fixed control variables $((A_e, A_{em}, Ara_-, G_e) = (0, 1, 1, 0))$ in Case 1: no perturbation in network (71). Steady states are represented by red dots, limit cycles are represented by green dots, and all transient states are denoted by yellow dots.

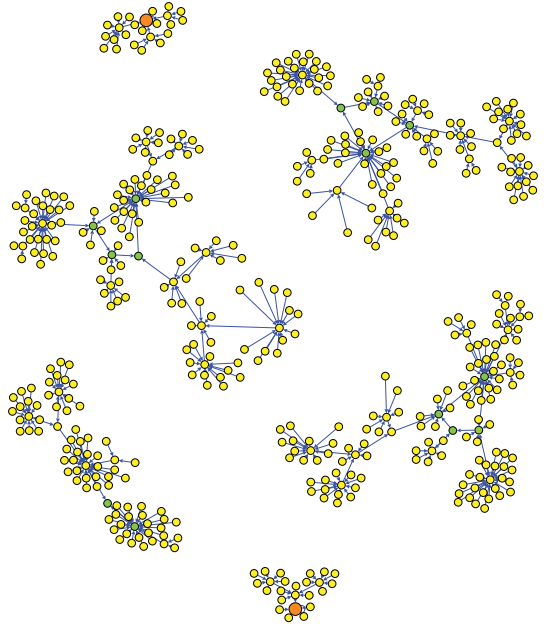


Fig. 7. State transition graph of the lac operon with fixed control variables $((A_e, A_{em}, Ara_-, G_e) = (0, 1, 1, 0))$ in Case 2: only f_E is switched to \hat{f}_E in network (71).

We consider a special intervention problem, that requires to design an optimal feedback law of four parameters (the extracellular arabinose, the AraC protein, and the extracellular glucose) to maximize the “present” level of the mRNA of the structural genes (araBAD). Under the abovementioned PBN expression of Ara operon network, this intervention problem can convert into an average minimum cost problem, with the cost function g as

$$g(x, u) = -x_7 \quad (72)$$

noticing that $x_7 = M_S$ denotes the mRNA of the structural genes (araBAD).

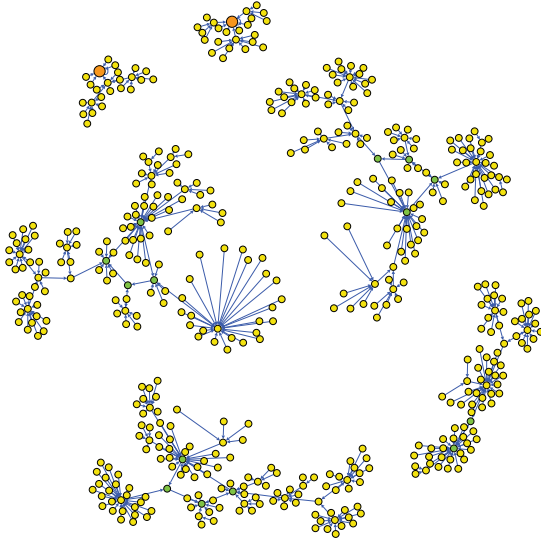


Fig. 8. State transition graph of the lac operon with fixed control variables $((A_e, A_{em}, Ara_-, G_e) = (0, 1, 1, 0))$ in Case 3: only f_{M_S} is switched to \hat{f}_{M_S} in network (71).

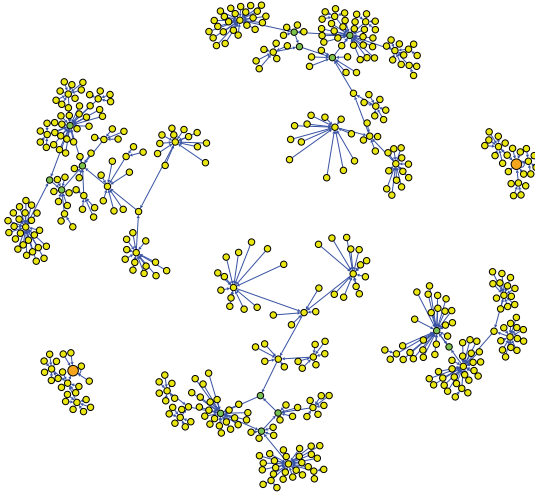


Fig. 9. State transition graph of the lac operon with fixed control variables $((A_e, A_{em}, Ara_-, G_e) = (0, 1, 1, 0))$ in Case 4: both f_E and f_{M_S} are switched to \hat{f}_E and \hat{f}_{M_S} in network (71).

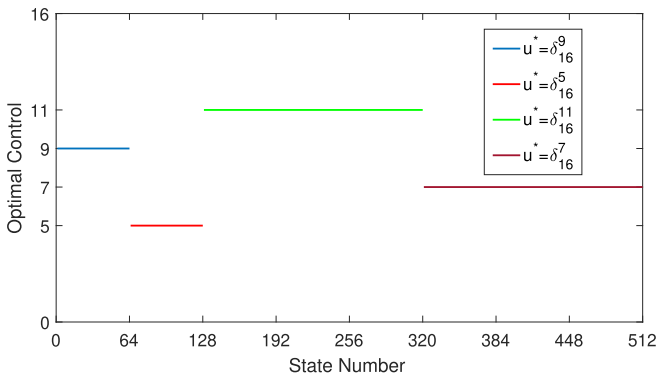


Fig. 10. Stationary optimal policy.

Now, using Algorithm 1, we easily get the optimal performance for this problem, as $J^*(x) \equiv -1$ for all $x \in \Delta_{512}$, and also obtain the corresponding optimal feedback law, as given

in the following form:

$$\mu^*(x) = \begin{cases} (1, 0, 0, 0), & \text{if } x_1 \vee x_2 = 0, \text{ and } x_9 = 0 \\ (0, 1, 0, 0), & \text{if } x_1 \vee x_2 = 0, \text{ and } x_9 = 1 \\ (1, 0, 1, 0), & \text{if } x_1 \vee x_2 = 1, \text{ and } x_9 = 0 \\ (0, 1, 1, 0), & \text{if } x_1 \vee x_2 = 1, \text{ and } x_9 = 1. \end{cases}$$

Furthermore, under new state representation $y = (y_1, \dots, y_9)$ with a special permutation

$$y_i = \begin{cases} x_9, & \text{if } i = 3 \\ x_3, & \text{if } i = 9 \\ x_i, & \text{otherwise.} \end{cases}$$

Then, the optimal stationary policy is given in Fig. 10. The policy iteration takes 5084.525867 s to obtain the exact optimal law on a computer with 8-GB RAM memory and Quad-Core 3.2-GHz processor.

VII. CONCLUSION

This article deal with the infinite horizon optimization problem for general PBCNs with an average cost. Based on STP, an equivalent matrix expression of the model was presented. Then, combining the techniques of the Laurent series expression and the Jordan decomposition, a novel nested policy iteration-type algorithm, which solves the probabilistic optimization problem for arbitrary PBCN without any requirement, was deduced. Finally, some practical applications, including optimal intervention problem of ARA operon, are solved to illustrate the benefit of the proposed algorithm for solving optimal control problems on PBCNs.

APPENDIX

Proof: [Proof of Lemma 4] Define $\Omega_{i,j}(u) = \{\alpha \in [1, C_0] \mid \delta_N^j = L[\alpha] \times u \times \delta_N^i\}$. Then, according to algebraic expression (14) and definition (17) of the conditional transition probability $p_{ij}(u)$, we have

$$p_{ij}(u) = \sum_{\alpha \in \Omega_{i,j}(u)} P[\alpha]. \quad (73)$$

In addition, by observing the relationship

$$\begin{cases} \delta_N^j = L[\alpha] \times u \times \delta_N^i \iff (\delta_N^j)^\top L[\alpha] \times u \times \delta_N^i = 1 \\ \delta_N^j \neq L[\alpha] \times u \times \delta_N^i \iff (\delta_N^j)^\top L[\alpha] \times u \times \delta_N^i = 0 \end{cases}$$

we have

$$\begin{aligned} \sum_{\alpha \in \Omega_{i,j}(u)} P[\alpha] &= \sum_{\alpha=1}^{C_0} P[\alpha] (\delta_N^j)^\top L[\alpha] \times u \times \delta_N^i \\ &= (\delta_N^j)^\top \sum_{\alpha=1}^{C_0} P[\alpha] L[\alpha] \times u \times \delta_N^i \\ &= (\delta_N^j)^\top \Gamma \times u \times \delta_N^i \end{aligned} \quad (74)$$

by recalling the definition (16) of transition matrix Γ . Then, by combining (73) and (74), we complete the proof of this lemma. \blacksquare

Proof of Lemma 5: According to full probability formula

$$\begin{aligned} \mathbf{E}x(t+1) &= \sum_{j=1}^N \Pr(x(t+1) = \delta_N^j) \delta_N^j \\ &= \sum_{j=1}^N \left[\sum_{i=1}^N \Pr(x(t) = \delta_N^i) p_{ij}(u = \delta_M^k) \right] \delta_N^j \\ &= \sum_{i=1}^N \Pr(x(t) = \delta_N^i) \left[\sum_{j=1}^N p_{ij}(u = \delta_M^k) \delta_N^j \right] \end{aligned} \quad (75)$$

In addition, using Lemma 4 and the fact that $\sum_{j=1}^N \delta_N^j (\delta_N^j)^\top = I_N$, we also get

$$\begin{aligned} \sum_{j=1}^N p_{ij}(u = \delta_M^k) \delta_N^j &= \sum_{j=1}^N \delta_N^j (\delta_N^j)^\top \Gamma \times \delta_M^k \times \delta_N^i \\ &= \Gamma \times \delta_M^k \times \delta_N^i. \end{aligned} \quad (76)$$

Combining (75) and (76), we have

$$\begin{aligned} \mathbf{E}x(t+1) &= \sum_{i=1}^N \Gamma \times \delta_M^k \times \delta_N^i \Pr(x(t) = \delta_N^i) \\ &= \Gamma \times \delta_M^k \sum_{i=1}^N \Pr(x(t) = \delta_N^i) \times \delta_N^i = \Gamma \times \delta_M^k \mathbf{E}x(t). \end{aligned}$$

We complete the proof. \blacksquare

Proof of Lemma 8: Since (29) is immediately obtained from (28), we just prove (28).

Based on the definition of g_μ given in (30), it is easily checked that $g(\delta_N^i, \mu(\delta_N^i)) = (\delta_N^i)^\top g_\mu$, for any $\delta_N^i \in \Delta_N$ and $\mu \in \mathcal{U}$. This implies that the expectation of cost g can be calculated as

$$\mathbf{E}g(x, \mu(x)) = \sum_{i=1}^N \Pr(x = \delta_N^i) (\delta_N^i)^\top g_\mu = [\mathbf{E}x]^\top g_\mu. \quad (77)$$

Hence, for the given policy $\pi = \{\mu_0, \mu_1, \dots\}$ and an initial state $x(0) \in \Delta_N$, based on closed-loop matrix expression (27) of PBCN evolution dynamics, we get

$$\begin{aligned} \mathbf{E}g(x(t), \mu_t(x(t))) &= [\mathbf{E}x(t)]^\top g_{\mu_t} = [\mathbf{E}\Gamma_{\mu_{t-1}} x(t-1)]^\top g_{\mu_t} \\ &= [\mathbf{E}\Gamma_{\mu_{t-1}} \cdots \Gamma_{\mu_0} x(0)]^\top g_{\mu_t} = x(0)^\top \prod_{k=0}^{t-1} \Gamma_{\mu_k}^\top g_{\mu_t} \end{aligned}$$

for all $t \geq 1$. Hence, if $x(0) = \delta_N^i$, then

$$\begin{aligned} J_\pi(\delta_N^i) &= \lim_{T \rightarrow \infty} \mathbf{E} \left[\frac{1}{T} \sum_{t=0}^{T-1} g(x(t), \mu_t(t)) \right] \\ &= \lim_{T \rightarrow \infty} \left[\frac{1}{T} \sum_{t=0}^{T-1} \mathbf{E}g(x(t), \mu_t(t)) \right] \\ &= (\delta_N^i)^\top \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \prod_{k=0}^{t-1} \Gamma_{\mu_k}^\top g_{\mu_t} \end{aligned}$$

which implies (28), by recalling $J_\pi = [J_\pi(\delta_N^1), \dots, J_\pi(\delta_N^N)]^\top$. \blacksquare

Proof of Lemma 17: It is noted that if $\Gamma_\mu = I_N$, then (64) automatically holds with $\Gamma_\mu^\# = I_N$ and $H_\mu^\# = F(\alpha, \mu) = 0$. Hence, it is enough to consider the case of $\Gamma_\mu \neq I_N$. For $0 < \alpha < 1$, we take $\alpha = (1/1 + \beta)$, $\beta > 0$, and then

$$I_N - \alpha \Gamma_\mu^\top = \frac{1}{1 + \beta} [\beta I_N + (I_N - \Gamma_\mu^\top)].$$

By the Jordan decomposition (53), we have $\beta I_N + (I_N - \Gamma_\mu^\top) = V \begin{bmatrix} \beta I_{N-r} & 0 \\ 0 & \beta I_r + S \end{bmatrix} V^{-1}$. Hence

$$\begin{aligned} (I_N - \alpha \Gamma_\mu^\top)^{-1} &= (\beta + 1) V \begin{bmatrix} \beta^{-1} I_{N-r} & 0 \\ 0 & (\beta I_r + S)^{-1} \end{bmatrix} V^{-1} \\ &= \frac{\beta + 1}{\beta} V \begin{bmatrix} I_{N-r} & 0 \\ 0 & 0 \end{bmatrix} V^{-1} \\ &\quad + (\beta + 1) V \begin{bmatrix} 0 & 0 \\ 0 & (\beta I_r + S)^{-1} \end{bmatrix} V^{-1}. \end{aligned} \quad (78)$$

We now analyze $(\beta I_r + S)^{-1}$. $(\beta I_r + S)^{-1} = [(I_r + \beta S^{-1})S]^{-1} = S^{-1}(I_r + \beta S^{-1})^{-1}$. Notice that if $0 < \beta \|S^{-1}\| < 1$, which equivalent to $1 - (1/1 + \beta \|S^{-1}\|) < \alpha < 1$, then, based on [45, Corollary 5.6.16], we deduce that $I_r + \beta S^{-1}$ in a nonsingular matrix, and its inverse have an expression as $[I_r + \beta S^{-1}]^{-1} = \sum_{i=0}^{\infty} (-\beta)^i S^{-i}$. Hence

$$\begin{aligned} (\beta I_r + S)^{-1} &= S^{-1}(I_r + \beta S^{-1})^{-1} \\ &= S^{-1} - \beta \sum_{i=0}^{\infty} (-\beta)^i S^{-i-2}. \end{aligned} \quad (79)$$

Substituting (79) into (78), we get

$$\begin{aligned} (I_N - \alpha \Gamma_\mu^\top)^{-1} &= \frac{\beta + 1}{\beta} V \begin{bmatrix} I_{N-r} & 0 \\ 0 & 0 \end{bmatrix} V^{-1} \\ &\quad - \beta(\beta + 1) V \begin{bmatrix} 0 & 0 \\ 0 & \sum_{i=0}^{\infty} (-\beta)^i S^{-i-2} \end{bmatrix} V^{-1} \\ &\quad + (1 + \beta) V \begin{bmatrix} 0 & 0 \\ 0 & S^{-1} \end{bmatrix} V^{-1} \\ &= \frac{\beta + 1}{\beta} \Gamma_\mu^\# + H_\mu + F(\alpha, \mu) \end{aligned} \quad (80)$$

with

$$F(\alpha, \mu) := \beta H_\mu - \beta(\beta + 1) V \begin{bmatrix} 0 & 0 \\ 0 & \sum_{i=0}^{\infty} (-\beta)^i S^{-i-2} \end{bmatrix} V^{-1}$$

where (59) and (63) are used in the last step of (80). Finally, by noticing $(\beta + 1/\beta) = (1/1 - \alpha)$, and when $\alpha \rightarrow 1$, we have $\beta = (1 - \alpha/\alpha) \rightarrow 0$, and $\beta(\beta + 1) = (1 - \alpha/\alpha^2) \rightarrow 0$. As a result $F(\alpha, \mu) \rightarrow 0$, as $\alpha \rightarrow 1$. We finish the proof. \blacksquare

Proof of Proposition 19: Set

$$e_\alpha(\eta, \mu) := (I_N - \alpha \Gamma_\eta^\top) (J_\eta^\alpha - J_\mu^\alpha). \quad (81)$$

Then, by definition (70) of J_μ^α , we obtain

$$e_\alpha(\eta, \mu) = g_\eta + \alpha \Gamma_\eta^\top J_\mu^\alpha - J_\mu^\alpha. \quad (82)$$

For any $x \in S_e(\mu, \eta)$, we get

$$x^\top g_\eta = g(x, \eta(x)) = g(x, \mu(x)) = x^\top g_\mu \quad (83)$$

$$\begin{aligned} x^\top \Gamma_\eta^\top &= (\Gamma_\eta x)^\top = (L \times \eta(x) \times x)^\top \\ &= (L \times \mu(x) \times x)^\top = (\Gamma_\mu x)^\top = x^\top \Gamma_\mu^\top. \end{aligned} \quad (84)$$

Hence, if $\delta_N^i \in \mathcal{A}_0(\mu, \eta)$, using (83), (84), and $(I_N - \alpha \Gamma_\mu^\top) J_\mu^\alpha = g_\mu$

$$\begin{aligned} e_\alpha(\eta, \mu)(\delta_N^i) &= (\delta_N^i)^\top (g_\eta + \alpha \Gamma_\eta^\top J_\mu^\alpha - J_\mu^\alpha) \\ &= (\delta_N^i)^\top (g_\mu + \alpha \Gamma_\mu^\top J_\mu^\alpha) - (\delta_N^i)^\top J_\mu^\alpha = 0. \end{aligned} \quad (85)$$

Now, we analyze the case $x \notin \mathcal{A}_0(\mu, \eta)$. Applying the inverse matrix expression of $(I_N - \alpha \Gamma_\mu^\top)$, given by the Laurent series expansion in Lemma 17

$$\begin{aligned} J_\eta^\alpha &= (I_N - \alpha \Gamma_\eta^\top)^{-1} g_\eta \\ &= \frac{1}{1-\alpha} \Gamma_\mu^\# g_\eta + H_\mu^\# g_\eta + F(\alpha, \mu) g_\eta \\ &= \frac{1}{1-\alpha} J_\mu + h_\mu + F(\alpha, \mu) g_\mu. \end{aligned} \quad (86)$$

Then, by rearranging terms in (82)

$$\begin{aligned} e_\alpha(\eta, \mu) &= g_\eta - (I_N - \alpha \Gamma_\eta^\top) \left[\frac{1}{1-\alpha} J_\mu + h_\mu + F(\alpha, \mu) g_\mu \right] \\ &= D_1 + D_2 + (1-\alpha) \Gamma_\alpha^\top h_\mu + (I_N - \alpha \Gamma_\mu^\top) F(\alpha, \mu) g_\mu \end{aligned} \quad (87)$$

with

$$D_1 = \frac{\alpha}{1-\alpha} (\Gamma_\eta^\top - I_N) J_\mu \quad (88)$$

$$D_2 = g_\eta - J_\mu + (\Gamma_\eta^\top - I_N) h_\mu. \quad (89)$$

Noticing $\lim_{\alpha \rightarrow 1} F(\alpha, \mu) = 0$, we have

$$\lim_{\alpha \uparrow 1} [(1-\alpha) \Gamma_\alpha^\top h_\mu + (I - \alpha \Gamma_\mu^\top) f(\alpha, \mu) g_\mu] = 0. \quad (90)$$

Recalling definition (29) and (56), we get $\Gamma_\mu^\top J_\mu = J_\mu$. Hence, if $\delta_N^i \in \mathcal{A}_1(\mu, \eta)$, then we have

$$\lim_{\alpha \uparrow 1} [D_1]_i = \lim_{\alpha \uparrow 1} \frac{\alpha}{1-\alpha} [(\Gamma_\eta^\top - \Gamma_\mu^\top) J_\mu]_i = -\infty. \quad (91)$$

Furthermore, according to the definition (89) of D_2 , we get

$$|[D_2]_i| \leq \|g_\eta\| + \|J_\mu\| + 2\|h_\mu\|. \quad (92)$$

Hence, combining (90)–(92), we deduce that

$$\lim_{\alpha \uparrow 1} [e_\alpha(\eta, \mu)]_i = -\infty, \quad \text{if } \delta_N^i \in \mathcal{A}_1(\mu, \eta). \quad (93)$$

If $\delta_N^i \in S_2(\mu, \eta)$, then we easily obtain

$$[D_1]_i = \frac{\alpha}{1-\alpha} [(\Gamma_\eta^\top - \Gamma_\mu^\top) J_\mu]_i = 0 \quad (94)$$

and the second condition in Definition (66) of $\mathcal{A}_2(\mu, \eta)$ implies that

$$[D_2]_i < [g_\mu + \Gamma_\mu h_\mu]_i - [J_\mu + h_\mu]_i = 0. \quad (95)$$

Thus, from (90), (94), and (95), we deduce that

$$\lim_{\alpha \rightarrow 1} [e_\alpha(\eta, \mu)]_i < 0, \quad \text{if } \delta_N^i \in \mathcal{A}_2(\mu, \eta). \quad (96)$$

Now summing (85), (93), and (96), we obtain

$$\begin{cases} \lim_{\alpha \uparrow 1} [e_\alpha(\eta, \mu)]_i = 0, & \text{if } \delta_N^i \in \mathcal{A}_0(\mu, \eta) \\ \lim_{\alpha \uparrow 1} [e_\alpha(\eta, \mu)]_i < 0, & \text{if } \delta_N^i \in \mathcal{A}_1(\mu, \eta) \cup \mathcal{A}_2(\mu, \eta). \end{cases}$$

In addition, condition (68) implies that there is an i_0 satisfying $\delta_N^{i_0} \in \mathcal{A}_1(\mu, \eta) \cup \mathcal{A}_2(\mu, \eta)$. Hence, $\lim_{\alpha \uparrow 1} e_\alpha(\eta, \mu) \not\geq 0$.

Using again definition (81) of $e_\alpha(\eta, \mu)$, we get

$$J_\eta^\alpha - J_\mu^\alpha = (I_N - \alpha \Gamma_\eta^\top)^{-1} e_\alpha(\eta, \mu).$$

Then, applying again [45, Corollary 5.6.16], we get

$$(I_N - \alpha \Gamma_\eta^\top)^{-1} = \sum_{k=0}^{+\infty} (\alpha \Gamma_\eta^\top)^k = I + \alpha \Gamma_\eta^\top + \dots$$

by noticing $0 < \alpha < 1$. Thus, finally, we have $\lim_{\alpha \uparrow 1} J_\eta^\alpha - J_\mu^\alpha \not\geq \lim_{\alpha \uparrow 1} e_\alpha(\eta, \mu) \not\geq 0$, from which one easily deduces that $\lim_{\alpha \uparrow 1} J_\eta^\alpha \not\geq \lim_{\alpha \uparrow 1} J_\mu^\alpha$. ■

REFERENCES

- [1] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, Mar. 1969.
- [2] H. Kitano, "Computational systems biology," *Nature*, vol. 420, no. 6912, p. 206, 2002.
- [3] B. Faryabi, E. R. Dougherty, and A. Datta, "On approximate stochastic control in genetic regulatory networks," *IET Syst. Biol.*, vol. 1, no. 6, pp. 361–368, Nov. 2007.
- [4] T. Akutsu, S. Kosub, A. A. Melkman, and T. Tamura, "Finding a periodic attractor of a Boolean network," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 9, no. 5, pp. 1410–1421, Sep. 2012.
- [5] F. Ettore and V. M. Elena, "Fault detection problems for Boolean networks and Boolean control networks," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 1–8.
- [6] E. Fornasini and M. E. Valcher, "Fault detection analysis of Boolean control networks," *IEEE Trans. Autom. Contr.*, vol. 60, no. 10, pp. 2374–2379, Oct. 2015.
- [7] D. Cheng, "On finite potential games," *Automatica*, vol. 50, no. 7, pp. 1793–1801, Jul. 2014.
- [8] X. Ding and H. Li, "Optimal control of random evolutionary Boolean games," *Int. J. Control*, early access, Feb. 17, 2019, doi: [10.1080/00207179.2019.1585957](https://doi.org/10.1080/00207179.2019.1585957).
- [9] Y. Wu, M. Kumar, and T. Shen, "A stochastic logical system approach to model and optimal control of cyclic variation of residual gas fraction in combustion engines," *Appl. Thermal Eng.*, vol. 93, pp. 251–259, Jan. 2016.
- [10] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*. London, U.K.: Springer, 2011.
- [11] F. Li, H. Yan, and H. R. Karimi, "Single-input pinning controller design for reachability of Boolean networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3264–3269, Jul. 2018.
- [12] H. Li, Y. Wang, and Z. Liu, "Stability analysis for switched Boolean networks under arbitrary switching signals," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1978–1982, Jul. 2014.
- [13] 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, Jul. 2018.
- [14] 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.
- [15] Q. Zhu, Y. Liu, J. Lu, and J. Cao, "Further results on the controllability of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 64, no. 1, pp. 440–442, Jan. 2019.
- [16] 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, Jan. 2019.
- [17] Y. Yu, M. Meng, J.-E. Feng, and P. Wang, "Stabilizability analysis and switching signals design of switched Boolean networks," *Nonlinear Anal., Hybrid Syst.*, vol. 30, pp. 31–44, Nov. 2018.

- [18] A. Yerudkar, C. Del Vecchio, and L. Glielmo, "Feedback stabilization control design for switched Boolean control networks," *Automatica*, vol. 116, Jun. 2020, Art. no. 108934.
- [19] P. Zhu and J. Han, "Stochastic multiple-valued gene networks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 1, pp. 42–53, Feb. 2014.
- [20] I. Shmulevich and E. R. Dougherty, *Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks*, vol. 118. Philadelphia, PA, USA: SIAM, 2010.
- [21] 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.
- [22] M. Meng, J. Lam, J.-E. Feng, and K. C. Cheung, "Stability and stabilization of Boolean networks with stochastic delays," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 790–796, Feb. 2019.
- [23] W.-K. Ching, M. M. Ng, E. S. Fung, and T. Akutsu, "On construction of stochastic genetic networks based on gene expression sequences," *Int. J. Neural Syst.*, vol. 15, no. 4, pp. 297–310, Aug. 2005.
- [24] S. Marshall, L. Yu, Y. Xiao, and E. R. Dougherty, "Inference of a probabilistic Boolean network from a single observed temporal sequence," *EURASIP J. Bioinf. Syst. Biol.*, vol. 2007, pp. 1–15, 2007.
- [25] K. Kobayashi and K. Hiraishi, "Design of probabilistic Boolean networks based on network structure and steady-state probabilities," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1966–1971, Aug. 2017.
- [26] I. Apostolopoulou and D. Marculescu, "Tractable learning and inference for large-scale probabilistic Boolean networks," 2018, *arXiv:1801.07693*. [Online]. Available: <http://arxiv.org/abs/1801.07693>
- [27] D. Cheng, Y. Zhao, and T. Xu, "Receding horizon based feedback optimization for mix-valued logical networks," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3362–3366, Dec. 2015.
- [28] Q. Liu, X. Guo, and T. Zhou, "Optimal control for probabilistic Boolean networks," *IET Syst. Biol.*, vol. 4, no. 2, pp. 99–107, Mar. 2010.
- [29] 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.
- [30] D. Laschov and M. Margaliot, "A maximum principle for single-input Boolean control networks," *IEEE Trans. Autom. Control*, vol. 56, no. 4, pp. 913–917, Apr. 2011.
- [31] D. Laschov and M. Margaliot, "Minimum-time control of Boolean networks," *SIAM J. Control Optim.*, vol. 51, no. 4, pp. 2869–2892, Jan. 2013.
- [32] Y. Wu and T. Shen, "An algebraic expression of finite horizon optimal control algorithm for stochastic logical dynamical systems," *Syst. Control Lett.*, vol. 82, pp. 108–114, Aug. 2015.
- [33] K. Kobayashi and K. Hiraishi, "Optimization-based approaches to control of probabilistic Boolean networks," *Algorithms*, vol. 10, no. 1, p. 31, Feb. 2017.
- [34] 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.
- [35] Y. Wu and T. Shen, "A finite convergence criterion for the discounted optimal control of stochastic logical networks," *IEEE Trans. Autom. Control*, vol. 63, no. 1, pp. 262–268, Jan. 2018.
- [36] 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.
- [37] Y. Wu and T. Shen, "Policy iteration approach to control residual gas fraction in IC engines under the framework of stochastic logical dynamics," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 3, pp. 1100–1107, May 2017.
- [38] Y. Zhao, Z. Li, and D. Cheng, "Optimal control of logical control networks," *IEEE Trans. Autom. Control*, vol. 56, no. 8, pp. 1766–1776, Aug. 2011.
- [39] E. Fornasini and M. Elena Valcher, "Optimal control of Boolean control networks," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1258–1270, May 2014.
- [40] Y. Wu, X.-M. Sun, X. Zhao, and T. Shen, "Optimal control of Boolean control networks with average cost: A policy iteration approach," *Automatica*, vol. 100, pp. 378–387, Feb. 2019.
- [41] Q. Zhu, Y. Liu, J. Lu, and J. Cao, "On the optimal control of Boolean control networks," *SIAM J. Control Optim.*, vol. 56, no. 2, pp. 1321–1341, Jan. 2018.
- [42] R. Pal, A. Datta, M. L. Bittner, and E. R. Dougherty, "Intervention in context-sensitive probabilistic Boolean networks," *Bioinformatics*, vol. 21, no. 7, pp. 1211–1218, Apr. 2005.
- [43] 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.
- [44] D. Cheng, C. Li, X. Zhang, and F. He, "Controllability of Boolean networks via mixed controls," *IEEE Control Syst. Lett.*, vol. 2, no. 2, pp. 254–259, Apr. 2018.
- [45] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.
- [46] S. M. Soisson, B. Macdougallshackleton, R. Schleif, and C. Wolberger, "Structural basis for ligand-regulated oligomerization of AraC," *Science*, vol. 276, no. 5311, pp. 421–425, Apr. 1997.
- [47] R. Schleif, "Regulation of the l-arabinose operon of escherichia coli," *J. Mol. Biol.*, vol. 16, no. 12, pp. 559–565, Dec. 2000.
- [48] A. Jenkins and M. Macauley, "Bistability and asynchrony in a Boolean model of the L-arabinose operon in Escherichia coli," *Bull. Math. Biol.*, vol. 79, no. 8, pp. 1778–1795, Aug. 2017.



Yuhu Wu (Member, IEEE) received the Ph.D. degree in mathematics from the Harbin Institute of Technology, Harbin, China, in January 2012.

Since September 2012, he has been an Assistant Professor with the Harbin University of Science and Technology, Harbin. He held a post-doctoral research position at Sophia University, Tokyo, Japan, from April 2012 to September 2015. In October 2015, he joined the School of Control Science and Engineering, Dalian University of Technology, Dalian, China, where he is currently a Full Professor.

His research interests are related to the Boolean networks, nonlinear control theory and applications of control to automotive powertrain systems, and unmanned aerial vehicles.



Yuqian Guo (Member, IEEE) graduated from the Department of Physics, Changsha University, Changsha, China, in 1995. He received the M.S. degree in mathematics from Hunan Normal University, Changsha, in 2003, and the Ph.D. degree in systems science from the Institute of Systems Science, Chinese Academy of Sciences, Beijing, China, in 2006.

He was a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from August 2006 to January 2009 and February 2011 to December 2012. Since 2009, he has been with the School of Automation, Central South University, Changsha, where he is currently a Full Professor. His research interests include nonlinear control, hybrid control systems, and logical dynamical networks.



Mitsuru Toyoda (Member, IEEE) received the Ph.D. degree in mechanical engineering from Sophia University, Tokyo, Japan, in March 2018.

Since April 2018, he has been a Project Assistant Professor with the Institute of Statistical Mathematics, Tokyo. In April 2019, he joined the Department of Mechanical Systems Engineering, Tokyo Metropolitan University, Tokyo, where he is currently an Assistant Professor. His research interests are related to optimal control and stochastic system control theory.