Optimal control problem of Boolean Networksp

Series Seven

Lecturer: Yuhu Wu

(School of Control Science and Engineering, Dalian University of Technology)

Center of STP Theory and Its Applications
August 15-23, 2020

LiaoCheng University, LiaoCheng, Shandong, P.R. China

Outline

1. **Introduction**
2. **Related works on Optimal Control Problem**
3. **Average Optimal control problem for BCNs**
4. **Policy iteration algorithm**
5. **Some applications**
   - Output tracking problem for BCNs
   - Optimal intervention problem of Ara operon in E. coil
6. **Future work**
7. **Reference**

Optimal control problems for Boolean Control Networks (BCNs)

A BCN with n state nodes and m input nodes can be described as

$$\begin{cases} x_1(t+1) = f_1(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t)), \\ \quad\vdots \\ x_n(t+1) = f_n(x_1(t), \cdots, x_n(t), u_1(t), \cdots, u_m(t)), \end{cases} \tag{1}$$

- state variables $x_i \in \mathcal{D} \triangleq \{0, 1\}, i = 1, \cdots, n$
- control inputs $u_j \in \mathcal{D}, j = 1, \cdots, m$
- Boolean update law $f_i : \mathcal{D}^{n+m} \to \mathcal{D}$

Optimal Control Problem for BCN (1) or PBCNs

- Finite horizon case

$$J_F(x_0) = \inf_u \mathop{E}_w \left\{ \sum_{k=0}^{N-1} g(x_k, u_k) + \mathcal{K}(x_N) \right\}, \tag{2}$$

- Infinite horizon case with discounted criteria

$$J_\pi(x_0) = \lim_{N\to\infty} \mathop{E}_{\substack{w_k \\ k=0,1,\cdots}} \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k)). \tag{3}$$

- Infinite horizon case with average criteria

$$J_a(x_0) = \inf_u \lim_{N\to\infty} \frac{1}{N} \mathop{E}_w \sum_{k=0}^{N-1} g(x_k, u_k, k) \tag{4}$$

Optimal Control Problem for BCN (1) or PBCNs

- Finite horizon case

$$J_F(x_0) = \inf_u \mathop{E}_w \left\{ \sum_{k=0}^{N-1} g(x_k, u_k) + \mathcal{K}(x_N) \right\}, \tag{2}$$

- Infinite horizon case with discounted criteria

$$J_\pi(x_0) = \lim_{N\to\infty} \mathop{E}_{\substack{w_k \\ k=0,1,\cdots}} \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k)). \tag{3}$$

- Infinite horizon case with average criteria

$$J_a(x_0) = \inf_u \lim_{N\to\infty} \frac{1}{N} \mathop{E}_w \sum_{k=0}^{N-1} g(x_k, u_k, k) \tag{4}$$

Optimal Control Problem for BCN (1) or PBCNs

- Finite horizon case

$$J_F(x_0) = \inf_u \mathop{E}_w \left\{ \sum_{k=0}^{N-1} g(x_k, u_k) + \mathcal{K}(x_N) \right\}, \tag{2}$$

- Infinite horizon case with discounted criteria

$$J_\pi(x_0) = \lim_{N\to\infty} \mathop{E}_{\substack{w_k \\ k=0,1,\cdots}} \sum_{k=0}^{N-1} \alpha^k g(x_k, \mu_k(x_k)). \tag{3}$$

- Infinite horizon case with average criteria

$$J_a(x_0) = \inf_u \lim_{N\to\infty} \frac{1}{N} \mathop{E}_w \sum_{k=0}^{N-1} g(x_k, u_k, k) \tag{4}$$

Related works on Optimal Control Problem

▷ Minimum-time control for BCNs

Laschov D, Margaliot M., SIAM J Control Optim, 2013

▷ Finite horizon case

☞ Mayer-type criterion: *Laschov D, Margaliot M., IEEE TAC 2011; Toyoda. M, Wu. Y, IEEE Cybernetics, 2020*

☞ Discounted criterion: *Zhu, Liu, Lu, and Cao, SIAM J Control Optim, 2018*

☞ Finite horizon control for SLCs: *Wu, Y, Shen, T. Systems Control Letters, 2015*

Related works on Optimal Control Problem

▷ Minimum-time control for BCNs

Laschov D, Margaliot M., SIAM J Control Optim, 2013

▷ Finite horizon case

☞ Mayer-type criterion: *Laschov D, Margaliot M., IEEE TAC 2011; Toyoda. M, Wu. Y, IEEE Cybernetics, 2020*

☞ Discounted criterion: *Zhu, Liu, Lu, and Cao, SIAM J Control Optim, 2018*

☞ Finite horizon control for SLCs: *Wu, Y, Shen, T. Systems Control Letters, 2015*

Related works on Optimal Control Problem

▷ Minimum-time control for BCNs

Laschov D, Margaliot M., SIAM J Control Optim, 2013

▷ Finite horizon case

☞ Mayer-type criterion: *Laschov D, Margaliot M., IEEE TAC 2011; Toyoda. M, Wu. Y, IEEE Cybernetics, 2020*

☞ Discounted criterion: *Zhu, Liu, Lu, and Cao, SIAM J Control Optim, 2018*

☞ Finite horizon control for SLCs: *Wu, Y, Shen, T. Systems Control Letters, 2015*

Related works on Optimal Control Problem

▷ Infinite horizon case

☞ Discounted criteria: *Pal, Datta, Dougherty, IEEE TSP, 2006; Cheng, Zhao, and Xu, IEEE TAC, 2015; Wu, Shen, IEEE TNNLS, 2018*

☞ Average criteria: *Zhao Y, Li Z Q, Cheng D Z. IEEE TAC 2011; Fornasini E, Valcher M E., IEEE TAC 2014, Wu, Sun, Zhao, Shen, Automatica, 2019*

▷ Applications

☞ Genetic regulatory networks: *Shmulevich, Dougherty, and Zhang, 2009*

☞ Human-Machine Game: *Cheng, Zhao, and Xu, IEEE TAC 2015*

☞ Engine control problem: *Wu, Kumar, Shen, Applied Thermal Engineering, 2015, Wu, Shen, IEEE TCST, 2017*

☞ Fuel efficiency of commuting vehicles: *Kang, Wu, Shen, International J. of Automotive Tech., 2017*

Related works on Optimal Control Problem

▷ Infinite horizon case

☞ Discounted criteria: *Pal, Datta, Dougherty, IEEE TSP, 2006; Cheng, Zhao, and Xu, IEEE TAC, 2015; Wu, Shen, IEEE TNNLS, 2018*

☞ Average criteria: *Zhao Y, Li Z Q, Cheng D Z. IEEE TAC 2011; Fornasini E, Valcher M E., IEEE TAC 2014, Wu, Sun, Zhao, Shen, Automatica, 2019*

▷ Applications

☞ Genetic regulatory networks: *Shmulevich, Dougherty, and Zhang, 2009*

☞ Human-Machine Game: *Cheng, Zhao, and Xu, IEEE TAC 2015*

☞ Engine control problem: *Wu, Kumar, Shen, Applied Thermal Engineering, 2015, Wu, Shen, IEEE TCST, 2017*

☞ Fuel efficiency of commuting vehicles: *Kang, Wu, Shen, International J. of Automotive Tech., 2017*

Average Optimal control problem for BCNs

Based on STP, the algebraic expression of BCN (1) is as

$$x(t+1) = L \ltimes u(t) \ltimes x(t) \tag{5}$$

For BCN (5) with a control sequence $\mathbf{u} = \{u(t) : t \in \mathbb{Z}_{\geq 0}\}$, consider

$$J(x_0, \mathbf{u}) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} g(x(t), u(t))), \tag{6}$$

where $g : \Delta_N \times \Delta_M \to R$ is the per-step cost function.

Then, the optimal cost problem is to find a optimal control sequence $\mathbf{u}^* = \{u^*(t) : t \in \mathbb{Z}_{\geq 0}\}$ such that

$$J(x_0, \mathbf{u}^*) = J^*(x_0) = \inf_{u} \lim_{N \to \infty} \frac{1}{N} \sum_{k=0}^{N-1} g(x_k, u_k, k). \tag{7}$$

The infinite horizon problem for deterministic BCNs with average cost first was addressed by [16]. Based on the graph theory and topology properties of trajectories, they prove that

Theorem

Then there exists a logical matrix K^* such that the optimal control $u*(t)$ of Problem (12) *satisfying*

$$\begin{cases} x^*(t+1) & = L \ltimes u^*(t) \ltimes x^*(t), \\ u^*(t+1) & = K^* \ltimes u^*(t) \ltimes x^*(t). \end{cases} \tag{8}$$

This approach was described as **"This method is very elegant and has an appealing graph theoretic interpretation"** in [17].

[16] Zhao, Y., Cheng, D., (2011). Optimal control of logical control networks, IEEE Transactions on Automatic Control, 55(8), 1766–1776.

[17] Fornasini, E., Valcher, M. E. (2014). Optimal control of boolean control networks. IEEE Transactions on Automatic Control, 59(5), 1258 – 1270.

In [17], the average optimal solution J^* is obtained as the limit of the solution of the finite horizon problem

$$J^* = \lim_{T\to\infty} \frac{1}{T}\tilde{J}_T^*$$

with

$$\tilde{J}_T^* = \inf_{\mathbf{u}} \sum_{t=0}^{T-1} g(x(t), u(t)). \tag{9}$$

For each $T \in \mathbb{Z}_{>0}$, the finite optimal cost (9) can be solved by a value iteration algorithm, provided in [17, page 1261].

But the number of convergence steps has no upper bound, this approach may converge to the average optimal solution very slowly.

[17]Fornasini, E., Valcher, M. E. (2014). Optimal control of boolean control networks. IEEE Transactions on Automatic Control, 59(5), 1258 – 1270.

Average Optimal control problem for BCNs

Set $\mathcal{U} = \{\mu \mid \mu : \Delta_N \to \Delta_M\}$.

- If a admissible policy $\pi = \{\mu_0, \mu_1, \cdots\}$, with $\mu_i \in \mathcal{U}$, is given

$$x_{k+1} = L \ltimes \mu_k(x_k) \ltimes x_k, \tag{10}$$

- then

$$J_\pi(x_0) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} g(x_k, \mu_k(x_k)). \tag{11}$$

The per-step cost function $g : \Delta_N \times \Delta_M \to R$ can be expressed in the form [1]

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

with $G = (G_{i,j})_{N\times M} = \left(g(\delta_N^i, \delta_N^j)\right)_{N\times M}$.

[1] The linear form of the per-step cost function $g : \Delta_N \times \Delta_M \to R$ is $g(x,u) = c^\top \ltimes u \ltimes x$, where $c = (c_1 \cdots, c_{MN})^\top \in \mathcal{R}^{MN}$ with $c_{(j-1)N+i} = g(\delta_N^i, \delta_M^j), i = 1, \cdots, N, j = 1, \cdots, M$. This equivalent linear form of cost function g was considered in [17].

Average Optimal control problem for BCNs

Set $\mathcal{U} = \{\mu \mid \mu : \Delta_N \to \Delta_M\}$.

- If a admissible policy $\pi = \{\mu_0, \mu_1, \cdots\}$, with $\mu_i \in \mathcal{U}$, is given

$$x_{k+1} = L \ltimes \mu_k(x_k) \ltimes x_k, \tag{10}$$

- then

$$J_\pi(x_0) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} g(x_k, \mu_k(x_k)). \tag{11}$$

The per-step cost function $g : \Delta_N \times \Delta_M \to R$ can be expressed in the form [1]

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

with $G = (G_{i,j})_{N\times M} = \left(g(\delta_N^i, \delta_N^j)\right)_{N\times M}$.

[1] The linear form of the per-step cost function $g : \Delta_N \times \Delta_M \to R$ is $g(x,u) = c^\top \ltimes u \ltimes x$, where $c = (c_1 \cdots, c_{MN})^\top \in \mathcal{R}^{MN}$ with $c_{(j-1)N+i} = g(\delta_N^i, \delta_M^j), i = 1, \cdots, N, j = 1, \cdots, M$. This equivalent linear form of cost function g was considered in [17].

Then, the optimal cost problem is to find a optimal control sequence $\mathbf{u}^* = \{u^*(t) : t \in \mathbb{Z}_{\geq 0}\}$ s.t.

$$J(x_0, \mathbf{u}^*) = J^*(x_0) = \inf_{\mathbf{u}} \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} x(t)^\top G u(t). \tag{12}$$

Consider a deterministic policy $\pi = \{\mu_0, \mu_1, \cdots, \}$,

$$J_\pi(x_0) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} x(t)^\top G \mu_t(x(t)). \tag{13}$$

Hence, referring to Theorem 3.1 of [1], the following result is fundamental.

Proposition

For any control law $\mu \in \mathcal{U}$*, there exists a unique logical matrix* $K_\mu \in \mathcal{L}_{M\times N}$*, called the structure feedback matrix of* μ*, such that* μ *is expressed in the vector form*

$$\mu(x) = K_\mu x, \qquad \forall\, x \in \Delta_N. \tag{14}$$

Under the state feedback control $u(t) = \mu(x(t)) = K_\mu x(t)$, the BCN (5) becomes a closed-loop system as

$$x(t+1) = L_\mu x(t), \tag{15}$$

where $L_\mu = LK_\mu \Phi_n$.

Vector Expression of Cost Function

For a feedback control $\mu \in \mathcal{U}$, since for any $x \in \Delta_N$, and $\mu \in \mathcal{U}$,

$$g(x, \mu(x)) = xGK_\mu x = x^\top g_\mu, \tag{16}$$

with

$$g_\mu = \big(g(\delta_s^1, \mu(\delta_s^1)), \cdots, g(\delta_s^s, \mu(\delta_s^s))\big)^\top. \tag{17}$$

For any given policy $\pi = \{\mu_0, \mu_1, \cdots\}$, according to matrix expression (15) of closed-loop BCN, we have

$$g(x(t), \mu_t(x(t))) = x(t)^\top g_{\mu_t} = (L_{\mu_{t-1}} \cdots L_{\mu_0} x(0))^\top g_{\mu_t} = x(0)^\top \prod_{k=0}^{t-1} L_{\mu_k}^\top g_{\mu_t}.$$

Hence, if $x(0) = \delta_N^i$, then

$$J_\pi(\delta_N^i) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} g(x(t), \mu_t(x(t))) = (\delta_N^i)^\top \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \prod_{k=0}^{t-1} L_{\mu_k}^\top g_{\mu_t}.$$

Accordingly, we obtain the vector expression of J_π as

$$J_\pi = \big(J_\pi(\delta_N^1), \cdots, J_\pi(\delta_N^N)\big)^\top = \lim_{T\to\infty} \sum_{t=0}^{T-1} \frac{1}{T} \prod_{k=0}^{t-1} L_{\mu_k}^\top g_{\mu_t},$$

Vector Expression of Cost Function

Especially, for a stationary policy $\pi^{\mu} = \{\mu, \mu, \cdots, \}$,

$$J_{\mu} = J_{\pi^{\mu}} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} (L_{\mu}^{\top})^t g_{\mu}.$$

Define the Cesaro limiting matrix $L_{\mu}^{\sharp}$ with respect to μ by

$$L_{\mu}^{\sharp} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} (L_{\mu}^{\top})^t. \tag{18}$$

- $L_{\mu} = LK_{\mu}\Phi_n \in \mathcal{L}_{N\times N}$.
- $L_{\mu}^{\sharp} = L_{\mu}^{\sharp}L_{\mu}^{\top} = L_{\mu}^{\top}L_{\mu}^{\sharp}$.
- $R(I - L_{\mu}^{T}) < N$.

Vector Expression of Cost Function

Especially, for a stationary policy $\pi^{\mu} = \{\mu, \mu, \cdots, \}$,

$$J_{\mu} = J_{\pi^{\mu}} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} (L_{\mu}^{\top})^t g_{\mu}.$$

Define the Cesaro limiting matrix $L_{\mu}^{\sharp}$ with respect to μ by

$$L_{\mu}^{\sharp} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} (L_{\mu}^{\top})^t. \tag{18}$$

- $L_{\mu} = LK_{\mu}\Phi_n \in \mathcal{L}_{N\times N}$.
- $L_{\mu}^{\sharp} = L_{\mu}^{\sharp} L_{\mu}^{\top} = L_{\mu}^{\top} L_{\mu}^{\sharp}$.
- $R(I - L_{\mu}^{T}) < N$.

- $L_\mu^\sharp = L_\mu^\sharp L_\mu^\top = L_\mu^\top L_\mu^\sharp$.
- $R(I - L_\mu^T) < N$.

Proof.

By $\|L_\mu\| = \|LK_\mu\| \leq 1$, we have $\|L_\mu^\top\| = \|L_\mu\| \leq 1$. Hence,

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

Then, according to definition (18) of limiting matrix $L_\mu^\sharp$,

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

We have proved $L_\mu^\sharp = L_\mu^\sharp L_\mu^\top$.
It is noticed that $\sum_{j=1}^N \left[I_N - L_\mu^\top\right]_{ij} = 0$, for any $i = 1, 2, \cdots, N$. That implies $\mathbf{1} = [1, 1, \cdots, 1]^\top \in R_N$ is a solution of homogeneous linear equation$(I_N - L_\mu^\top)x = 0$. Hence, $Rank(I_N - L_\mu^\top) < N$. □

Since $r = Rank(I_N - L_\mu^\top) < N$, based on Jordan decomposition, there is a nonsingular matrix $V \in \mathcal{R}^{N\times N}$, and a nonsingular upper triangular matrix $S \in \mathcal{R}^{r\times r}$ such that

$$I_N - L_\mu^\top = V \begin{bmatrix} 0 & 0 \\ 0 & S \end{bmatrix} V^{-1}. \tag{19}$$

Lemma

For any control law $\mu \in \mathcal{U}$, matrix $I_N - L_\mu^\top + L_\mu^\sharp$ is nonsingular. Furthermore, assume that the Jordan decomposition of $I_N - L_\mu^\top$ is given by (19), then, J_μ and $h_\mu = H_\mu^\sharp g_\mu$, with

$$H_\mu^\sharp := (I_N - L_\mu^\top + L_\mu^\sharp)^{-1}(I - L_\mu^\sharp), \tag{20}$$

which can be calculated by

$$\begin{cases} J_\mu = V \begin{bmatrix} I_{N-r} & 0 \\ 0 & 0 \end{bmatrix} V^{-1} g_\mu, \\ h_\mu = V \begin{bmatrix} 0 & 0 \\ 0 & S^{-1} \end{bmatrix} V^{-1} g_\mu, \end{cases} \tag{21}$$

Proof of Lemma: According to Jordan decomposition (19), $L_\mu^\top = V \begin{bmatrix} I_{N-r} & 0 \\ 0 & I_r - S \end{bmatrix}$.

Then, by definition (18) of limit matrix $L_\mu^\sharp$, we have

$$L_\mu^\sharp = V \begin{bmatrix} I_{N-r} & 0 \\ 0 & L_{22}^\sharp \end{bmatrix} V^{-1}, \tag{22}$$

where $L_{22}^\sharp = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} (I_r - S)^\top$. Recalling $L_\mu^\top L_\mu^\sharp = L_\mu^\sharp$ we get $S L_{22}^\sharp = 0$. Since $S \in \mathcal{R}^{r\times r}$ is nonsingular upper triangular matrix, we have $L_{22}^\sharp = 0$. Hence, (22) becomes

$$L_\mu^\sharp = V \begin{bmatrix} I_{N-r} & 0 \\ 0 & 0 \end{bmatrix} V^{-1}. \tag{23}$$

Then, noticing that $J_\mu = L_\mu^\sharp g_\mu$ from (18), we obtain the first equation of (21). In addition, combining Jordan decomposition (19) and (23), we have

$$(I - L_\mu^\top + L_\mu^\sharp) = V \begin{bmatrix} I_{N-r} & 0 \\ 0 & S \end{bmatrix} V^{-1}. \tag{24}$$

That implies matrix $I - L_\mu^\top + L_\mu^\sharp$ is nonsingular, and then

$$(I - L_\mu^\top + L_\mu^\sharp)^{-1}(I - L_\mu^\sharp) = V \begin{bmatrix} 0 & 0 \\ 0 & S^{-1} \end{bmatrix} V^{-1}. \tag{25}$$

Hence, by definition of $H_\mu^\sharp$, we prove the second equation of (21).

Remark

From the proof of Lemma 2, we can observe that J_μ satisfies

$$J_\mu = L_\mu^\top J_\mu,$$

which is a direct consequence of (21).

The following theorem provides an optimality criterion for the average optimal control problem of BCNs.

Theorem

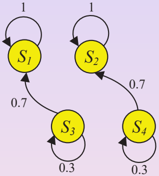
Suppose there exist two vectors $(J, h) \in R^N \times R^N$ which satisfy the following nested optimality condition, for each $i = 1, \cdots, N$,

$$\begin{cases} \min_{\mu \in \mathcal{U}} \left[(L_\mu^\top - I_N)J\right]_i = 0, & \text{(25-a)} \\ \min_{\mu \in \mathcal{U}_i} \left[g_\mu - J + (L_\mu^\top - I_N)h\right]_i = 0, & \text{(25-b)} \\ \quad \text{where } \mathcal{U}_i = \left\{\mu \in \mathcal{U} \,\middle|\, \left[(L_\mu^\top - I_N)J\right]_i = 0\right\} & \end{cases}$$

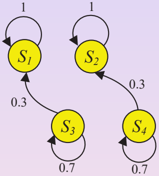
Then, J is the optimal cost of the average optimal problem (12), i.e., $J = J^*$.

Remark

In [12], a policy iteration algorithm for PBCNs was deduced under the assumption that the PBCN is ergodic, which requires that the transition matrix of PBCN for every stationary policy consists of a single recurrent class.
But their approach are no longer applicable for the general PBCN [13].



(a) When $u = \delta_2^1$.



(b) When $u = \delta_2^2$.

Figure 1: The transition probability diagram

[12] Pal, Datta, Dougherty, IEEE TSP, 2006.
[13] Wu, Toyoda, Guo, IEEE TNNLS, 2020.

Proof of Theorem: Condition (25-a) and (25-b) imply there exists a $\mu' \in \mathcal{U}$ s. t., for each $i = 1, \cdots, N$,

$$\begin{cases} \left[(L_{\mu'}^{\top} - I_N)J\right]_i = \min_{\mu \in \mathcal{U}} \left[(L_{\mu}^{\top} - I_N)J\right]_i = 0, & (26) \\ \left[g_{\mu'} - J + (L_{\mu'}^{\top} - I_N)h\right]_i & \\ \qquad = \min_{\mu \in \mathcal{U}} \left[g_{\mu} - J + (L_{\mu}^{\top} - I_N)h\right]_i = 0. & (27) \end{cases}$$

Equation (27) implies

$$J = g_{\mu'} + (L_{\mu'}^{\top} - I_N)h.$$

Multiplying the above equation by $L_{\mu'}^{\top}$ and applying equality (26) yield

$$J = L_{\mu'}^{\top} J = L_{\mu'}^{\top} g_{\mu'} + L_{\mu'}^{\top}(L_{\mu'}^{\top} - I_N)h.$$

Repeating this process with induction, we get, for any $n \in \mathbb{Z}_{\geq 0}$,

$$J = \left(L_{\mu'}^{\top}\right)^n g_{\mu'} + \left(L_{\mu'}^{\top}\right)^n (L_{\mu'}^{\top} - I_N)h. \tag{28}$$

Summing those expression over n, we have

$$nJ = \sum_{t=0}^{n-1} (L_{\mu}^{\top})^t g_{\mu} + \left[\left(L_{\mu'}^{\top}\right)^n - I_N\right] h.$$

Continue to Proof of Theorem: Noticing that $\left\|\left[\left(L_{\mu'}^{\top}\right)^n - I_N\right] h\right\| \leq 2\|h\|$, and applying equation (18), we deduce that, for all $i = 1, \cdots, N$,

$$[J]_i = \lim_{n\to\infty}\left[\frac{1}{n}\sum_{t=0}^{n-1}(L_\mu^\top)^t g_\mu\right]_i \quad = \quad [J_{\pi^{\mu'}}]_i \geq \inf_{\pi\in\Pi}[J_\pi]_i = [J^*]_i\,.$$

Next, we claim that if $(J,h) \in R^N \times R^N$ satisfies the nested optimality condition (25), then there exists a $C \geq 0$ such that J and $\hbar = h + CJ$ satisfy the following modified optimality condition, for each $i = 1, \cdots, N$,

$$\begin{cases} \min\limits_{\mu\in\mathcal{U}}\left[(L_\mu^\top - I_N)J\right]_i = 0, & \text{(30-a)} \\ \min\limits_{\mu\in\mathcal{U}}\left[g_\mu - J + (L_\mu^\top - I_N)\hbar\right]_i = 0, & \text{(30-b)} \end{cases}$$

Notice condition (30-b) is the same as condition (25-a). If (J,h), given in (25), satisfy (30-b), then we just set $\hbar = h$ with $C = 0$. Suppose J and h do not satisfy (30-b), then for some $i_0 \in \{1, \cdots, N\}$, and $\mu_0 \in \mathcal{U}\backslash\mathcal{U}_{i_0}$, we have

$$C_1 = \left[g_{\mu_0} - J + (L_{\mu_0}^\top - I_N)h\right]_{i_0} < 0,$$

Furthermore, $\mu_0 \in \mathcal{U}\backslash\mathcal{U}_{i_0}$ implies

$$C_2 = \left[(L_{\mu_0}^\top - I_N)J\right]_{i_0} > 0$$

Continued to Proof of Theorem: Now, let $\hbar = h + C_3 J$, where $C_3 > 0$ will be given latter. Then

$$\begin{aligned} & \left[g_{\mu_0} - J + (L_{\mu_0}^\top - I_N)\hbar\right]_{i_0} \\ = \quad & \left[g_{\mu_0} - J + (L_{\mu_0}^\top - I_N)h + C_3(L_{\mu_0}^\top - I_N)J\right]_{i_0} = C_1 + C_3 C_2. \end{aligned}$$

Hence, taking C_3 large enough such that $C_3 > \frac{|C_1|}{C_2}$, we have

$$\left[g_{\mu_0} - J + (L_{\mu_0}^\top - I_N)\hbar\right]_{i_0} > 0. \tag{31}$$

Since there exist only finite states and control inputs, we can choose large enough C_3 for which (30-b) holds for all $i = 1, \cdots, N$ and $\mu \in \mathcal{U}$. For any policy $\pi = \{\mu_0, \mu_1, \cdots,\} \in \Pi$, condition (25-a) implies

$$\begin{cases} [J]_i \leq \left[L_{\mu_0}^\top J\right]_i, & (32) \\ [J]_i \leq \left[g_{\mu_0} + (L_{\mu_0}^\top - I_N)\hbar\right]_i, & (33) \end{cases}$$

for all $i = 1, \cdots, N$, and applying condition (30-b) to μ_1 implies

$$[J]_i \leq \left[g_{\mu_1} + (L_{\mu_1}^\top - I_N)\hbar\right]_i, \ \forall i = 1, \cdots, N. \tag{34}$$

Multiplying above expression by $L_{\mu_0}^\top$ and applying inequality (32) yields, for any $i = 1, \cdots, N$,

$$[J]_i \leq \left[L_{\mu_0}^\top J\right]_i \leq \left[L_{\mu_0}^\top g_{\mu_1} + L_{\mu_0}^\top (L_{\mu_1}^\top - I_N)\hbar\right]_i.$$

Continued to Proof of Theorem: Repeating this process with induction, we get, for any $n \in \mathbb{Z}_{\geq 0}$

$$[J]_i \leq \left[L_{\mu_0}^\top \cdots L_{\mu_{n-1}}^\top g_{\mu_n} + L_{\mu_0}^\top \cdots L_{\mu_{n-1}}^\top (L_{\mu_n}^\top - I_N)\hbar\right]_i,$$

where set $L_{\mu_{-1}} = I_N$, when $n = 0$. Summing those expression over $n+1$, we have, $\forall i = 1, \cdots, N$,

$$[J]_i \quad \leq \quad \frac{1}{n+1}\left[\sum_{t=0}^{n}\prod_{k=-1}^{t-1} L_{\mu_k}^\top g_{\mu_t}\right]_i + \frac{\left[(L_{\mu_0}^\top \cdots L_{\mu_{n-1}}^\top L_{\mu_n}^\top - I_N)\hbar\right]_i}{n+1}.$$

Furthermore, noticing that $\|(L_{\mu_0}^\top \cdots L_{\mu_{n-1}}^\top L_{\mu_n}^\top - I_N)\hbar\| \leq 2\|\hbar\|$, we get that, for all $i = 1 \cdots, N$,

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

In consideration of the arbitrariness of π, we get for all $i = 1 \cdots, N$,

$$[J]_i \leq \inf_{\pi\in\Pi}[J_\pi]_i = [J^*]_i. \tag{35}$$

Finally, combining (29) and (35), we obtain $J = J^*$, and finish the proof.

Algorithm (Policy iteration for optimal problem (12))

Step 0. Initialization: *Given an initial policy* $\mu^0 \in \mathcal{U}$.
Step 1. Policy Evaluation: *for policy* μ^n, *compute* J_{μ^n}, h_{μ^n}
Step 2. Policy Improvement:

2.A *Choose policy* μ^{n+1} *s. t.* $K_{n+1} = L_N[q_1^{n+1}, \cdots, q_N^{n+1}]$ *satisfy,*

$$q_i^{n+1} \in \arg\min_{j=1,\cdots,M} \left\{ (\delta_N^i)^\top \ltimes (\delta_M^j)^\top L^\top J_{\mu^n} \right\}, i = 1, \cdots, N,$$

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 policy* μ^{n+1} *s. t.*

$$q_i^{n+1} \in \arg\min_{j=1,\cdots,M} \left\{ G_{ij} + (\delta_N^i)^\top \ltimes (\delta_M^j)^\top L^\top h_{\mu^n} \right\}, i = 1, \cdots, N,$$

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****.*

Now we provide the Laurent series expansion of $(I_N - \alpha L_\mu^\top)^{-1}$, and a monotonicity criterion.

$$(1-x)^{-1} = \frac{1}{1-x} = \sum_{i=0}^{\infty} x^i = 1 + x + o(x)$$

Lemma

For any feedback control law $\mu \in \mathcal{U}$, we have, $0 < \alpha < 1$,

$$(I_N - \alpha L_\mu^\top)^{-1} = \frac{1}{1-\alpha} L_\mu^\sharp + H_\mu^\sharp + F(\alpha, \mu), \tag{36}$$

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

Proof of Lemma: For $0 < \alpha < 1$, we take $\alpha = \frac{1}{1+\beta}$, $\beta > 0$, then

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

By Jordan decomposition (19),

$$\beta I_N + (I_N - L_\mu^\top) = V \begin{bmatrix} \beta I_{N-r} & 0 \\ 0 & \beta I_r + S \end{bmatrix} V^{-1}.$$

Hence,

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

We now analyze $(\beta I_l + S)^{-1}$. $(\beta I_r + S)^{-1} = [(I_r + \beta S^{-1})S]^{-1} = S^{-1}(I_r + \beta S^{-1})^{-1}$. Notice that, when $0 < \beta\|S^{-1}\| < 1$, then $I_r + \beta S^{-1}$ has inverse, and its inverse can be expressed as $[I_r + \beta S^{-1}]^{-1} = \sum_{i=0}^{\infty}(-\beta)^i S^{-i}$. Hence,

$$(\beta I_r + S)^{-1} = S^{-1}(I_r + \beta S^{-1})^{-1} = S^{-1} - \beta \sum_{i=0}^{\infty} (-\beta)^i S^{-i-2} \qquad (38)$$

Substituting (38) into (37), we get

$$
\begin{aligned}
(I_N - \alpha L_\mu^\top)^{-1} = & \frac{\beta+1}{\beta} V \begin{bmatrix} I_{N-r} & 0 \\ 0 & 0 \end{bmatrix} V^{-1} \\
& -\beta(\beta+1) V \begin{bmatrix} 0 & 0 \\ 0 & \sum_{i=0}^{\infty}(-\beta)^i s^{-i-2} \end{bmatrix} V^{-1} \\
& +(1+\beta) V \begin{bmatrix} 0 & 0 \\ 0 & S^{-1} \end{bmatrix} V^{-1} = \frac{\beta+1}{\beta} L_\mu^\sharp + H_\mu + F(\alpha,\mu),
\end{aligned} \tag{39}
$$

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 we used (22), and (25) in the last step of (39). Finally, by noticing $\frac{\beta+1}{\beta} = \frac{1}{1-\alpha}$, and when $\alpha \to 1$, we have $\beta = \frac{1-\alpha}{\alpha} \to 0$, and $\beta(\beta+1) = \frac{1-\alpha}{\alpha^2} \to 0$. Accordingly, $F(\alpha,\mu) \to 0$, as $\alpha \to 1$. We complete the proof. □

Proposition

For any $\mu, \eta \in \mathcal{U}$, define three special subsets of Δ_N,

$$S_e(\mu,\eta) = \{\delta_N^i | \mu(\delta_N^i) = \eta(\delta_N^i)\}, \tag{40}$$

$$S_1(\mu,\eta) = \left\{\delta_N^i \,\middle|\, \left[L_\eta^\top J_\mu\right]_i < \left[L_\mu^\top J_\mu\right]_i \right\}, \tag{41}$$

$$S_2(\mu,\eta) = \left\{\delta_N^i \,\middle|\, \begin{array}{l} \left[L_\mu^\top J_\mu\right]_i = \left[L_\eta^\top J_\mu\right]_i, \text{ and} \\ \left[g_\eta + L_\eta^\top h_\mu\right]_i < \left[g_\mu + L_\mu^\top h_\mu\right]_i \end{array} \right\} \tag{42}$$

If

$$\emptyset \neq (S_e(\mu,\eta))^C \subset (S_1(\mu,\eta) \cup S_2(\mu,\eta)), \tag{43}$$

then

$$\lim_{\alpha \uparrow 1} J_\eta^\alpha \precneqq \lim_{\alpha \uparrow 1} J_\mu^\alpha, \tag{44}$$

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

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

Algorithm (Policy iteration for optimal problem (12))

Step 0. Initialization: *Given an initial policy* $\mu^0 \in \mathcal{U}$.
Step 1. Policy Evaluation: *for policy* μ^n, *compute* J_{μ^n}, h_{μ^n}
Step 2. Policy Improvement:

2.A *Choose policy* μ^{n+1} *s. t.* $K_{n+1} = L_N[q_1^{n+1}, \cdots, q_N^{n+1}]$ *satisfy,*

$$q_i^{n+1} \in \arg\min_{j=1,\cdots,M} \left\{ (\delta_N^i)^\top \ltimes (\delta_M^j)^\top L^\top J_{\mu^n} \right\}, i = 1, \cdots, N,$$

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 policy* μ^{n+1} *s. t.*

$$q_i^{n+1} \in \arg\min_{j=1,\cdots,M} \left\{ G_{ij} + (\delta_N^i)^\top \ltimes (\delta_M^j)^\top L^\top h_{\mu^n} \right\}, i = 1, \cdots, N,$$

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****.*

Proposition 5.1 guarantees that the policy iteration process terminates in finite steps.

Remark

In [17], the average optimal solution J^* is obtained as the limit of the solution of the finite horizon problem

$$J^* = \lim_{T\to\infty} \frac{1}{T}\tilde{J}_T^*$$

with

$$\tilde{J}_T^* = \inf_{\mathbf{u}} \sum_{t=0}^{T-1} g(x(t), u(t)). \tag{45}$$

For each $T \in \mathbb{Z}_{>0}$, the finite optimal cost (45) can be solved by a value iteration algorithm, provided in [17, page 1261].

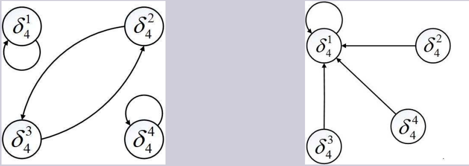
[17] Fornasini, E., Valcher, M. E. (2014). Optimal control of boolean control networks. IEEE Transactions on Automatic Control, 59(5), 1258 – 1270.

Example

Consider the following BNC

$$\begin{cases} x_1(t+1) = (x_2(t) \vee u_1(t)) \wedge \neg u_1(t), \\ x_2(t+1) = (x_1(t) \vee u_1(t)) \wedge \neg u_1(t) \end{cases} \tag{46}$$

The corresponding state transition diagram is shown in Fig. 2.



(a) When $u = \delta_2^1$. (b) When $u = \delta_2^2$.

Figure 2: State transition diagram.

Based on STP techniques, the algebraic form of (46) is

$$x(t+1) = L \ltimes u(t) \ltimes x(t)$$

with $x(t) = x_1(t) \ltimes x_2(t)$, and

$$L = \quad \delta_4[1\ 3\ 2\ 4\ 1\ 1\ 1\ 1]$$

Assume that the cost function g is given by following cost matrix

$$G_\varepsilon = \begin{pmatrix} 0 & 1 & 1 & 1 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{pmatrix}^T.$$

with parameter $\varepsilon > 0$.

Then, applying the value iteration algorithm given in [17, Sec. III] it is obtained that

$$\frac{1}{T}\tilde{J}_T^* = \begin{cases} [0, \varepsilon, \varepsilon, \varepsilon]^\top, & \text{for } T \leq \left\lfloor \frac{1}{\varepsilon} \right\rfloor, \\ \left[0, \frac{\varepsilon}{T} \left\lfloor \frac{1}{\varepsilon} \right\rfloor, \frac{\varepsilon}{T} \left\lfloor \frac{1}{\varepsilon} \right\rfloor, \frac{\varepsilon}{T} \left\lfloor \frac{1}{\varepsilon} \right\rfloor\right]^\top, & \text{for } T > \left\lfloor \frac{1}{\varepsilon} \right\rfloor, \end{cases}$$

the optimal controller has the time-varying state feedback form $\mu_t^*(x) = K_{\mu_t}^* x$, for all $x \in \Delta_N$, with structure matrix

$$K_{\mu_t}^* = \begin{cases} \delta_4[2, 1, 1, 1], & \text{for } t \leq \left\lfloor \frac{1}{\varepsilon} \right\rfloor, \\ \delta_4[2, 2, 2, 2], & \text{for } t > \left\lfloor \frac{1}{\varepsilon} \right\rfloor. \end{cases}$$

- Accordingly, the convergence depends on the choice of the cost function G_ε.
- For every $\varepsilon \in (0,1)$, the $\frac{\varepsilon}{2}$-tolerance approximate optimal cost require $2\left\lfloor \frac{1}{\varepsilon} \right\rfloor + 1$ steps in this value iteration approach.
- The number of iteration steps is no upper bound

$$2\lfloor 1/\varepsilon \rfloor + 1 \to \infty$$

as $\varepsilon \to 0$,
- The convergence of this approach is very slow.

- Accordingly, the convergence depends on the choice of the cost function G_ε.
- For every $\varepsilon \in (0,1)$, the $\frac{\varepsilon}{2}$-tolerance approximate optimal cost require $2\left\lfloor\frac{1}{\varepsilon}\right\rfloor + 1$ steps in this value iteration approach.
- The number of iteration steps is no upper bound

$$2\lfloor 1/\varepsilon\rfloor + 1 \to \infty$$

as $\varepsilon \to 0$,
- The convergence of this approach is very slow.

- Accordingly, the convergence depends on the choice of the cost function G_ε.
- For every $\varepsilon \in (0,1)$, the $\frac{\varepsilon}{2}$-tolerance approximate optimal cost require $2\left\lfloor \frac{1}{\varepsilon} \right\rfloor + 1$ steps in this value iteration approach.
- The number of iteration steps is no upper bound

$$2\lfloor 1/\varepsilon \rfloor + 1 \to \infty$$

as $\varepsilon \to 0$,

- The convergence of this approach is very slow.

- Accordingly, the convergence depends on the choice of the cost function G_ε.
- For every $\varepsilon \in (0,1)$, the $\frac{\varepsilon}{2}$-tolerance approximate optimal cost require $2\left\lfloor \frac{1}{\varepsilon} \right\rfloor + 1$ steps in this value iteration approach.
- The number of iteration steps is no upper bound

  $$2\left\lfloor 1/\varepsilon \right\rfloor + 1 \to \infty$$

  as $\varepsilon \to 0$,
- The convergence of this approach is very slow.

- **Initialization**: The initial policy μ^0 is selected as $\mu^0(x) = L_4[1, 1, 1, 1]x, \forall x$ Δ_{12}.
- **Policy Evaluation**:
  Applying Lemma 2, obtain $J_{\mu^0} = [1, 1, 1, 1]^T$, $h_{\mu^0} = [0, 0, 0, 0]^T$.
- l**Policy Improvement**:
  Substep (2.A), obtain μ^1 with $K_1 = L_4[1, 1, 1, 1]$;
  Substep (2.B), since $\mu^1 = \mu^0$, go to (2.C);
  Substep (2.C), renew policy μ^1 with $K_1 = L_4[2, 1, 1, 1]$;
  Substep (2.D), since $\mu^1 \neq \mu^0$, return to the **Step 1**.

$$\vdots$$

- Substep (2.D) of the third iteration $\mu^3 = \mu^2$.

Hence, μ^2 is optimal with $K_2 = L_4[2, 2, 2, 2]$ and the corresponding optimal performance is

$$J^* = J_{\mu^2} = [0, 0, 0, 0]^\top.$$

- **Initialization**: The initial policy μ^0 is selected as $\mu^0(x) = L_4[1, 1, 1, 1]x, \forall x$ Δ_{12}.
- **Policy Evaluation**:
  Applying Lemma 2, obtain $J_{\mu^0} = [1, 1, 1, 1]^T$, $h_{\mu^0} = [0, 0, 0, 0]^T$.
- l**Policy Improvement**:
  Substep (2.A), obtain μ^1 with $K_1 = L_4[1, 1, 1, 1]$;
  Substep (2.B), since $\mu^1 = \mu^0$, go to (2.C);
  Substep (2.C), renew policy μ^1 with $K_1 = L_4[2, 1, 1, 1]$;
  Substep (2.D), since $\mu^1 \neq \mu^0$, return to the **Step 1**.

$$\vdots$$

- Substep (2.D) of the third iteration $\mu^3 = \mu^2$.

  Hence, μ^2 is optimal with $K_2 = L_4[2, 2, 2, 2]$ and the corresponding optimal performance is

$$J^* = J_{\mu^2} = [0, 0, 0, 0]^\top.$$

- **Initialization**: The initial policy μ^0 is selected as $\mu^0(x) = L_4[1,1,1,1]x, \forall x$ Δ_{12}.
- **Policy Evaluation**:
  Applying Lemma 2, obtain $J_{\mu^0} = [1,1,1,1]^T$, $h_{\mu^0} = [0,0,0,0]^T$.
- l**Policy Improvement**:
  Substep (2.A), obtain μ^1 with $K_1 = L_4[1,1,1,1]$;
  Substep (2.B), since $\mu^1 = \mu^0$, go to (2.C);
  Substep (2.C), renew policy μ^1 with $K_1 = L_4[2,1,1,1]$;
  Substep (2.D), since $\mu^1 \neq \mu^0$, return to the **Step 1**.

$$\vdots$$

- Substep (2.D) of the third iteration $\mu^3 = \mu^2$.

  Hence, μ^2 is optimal with $K_2 = L_4[2,2,2,2]$ and the corresponding optimal performance is

$$J^* = J_{\mu^2} = [0,0,0,0]^\top.$$

- **Initialization**: The initial policy μ^0 is selected as $\mu^0(x) = L_4[1, 1, 1, 1]x, \forall x$ Δ_{12}.
- **Policy Evaluation**:
  Applying Lemma 2, obtain $J_{\mu^0} = [1, 1, 1, 1]^T$, $h_{\mu^0} = [0, 0, 0, 0]^T$.
- l**Policy Improvement**:
  Substep (2.A), obtain μ^1 with $K_1 = L_4[1, 1, 1, 1]$;
  Substep (2.B), since $\mu^1 = \mu^0$, go to (2.C);
  Substep (2.C), renew policy μ^1 with $K_1 = L_4[2, 1, 1, 1]$;
  Substep (2.D), since $\mu^1 \neq \mu^0$, return to the **Step 1**.

$$\vdots$$

- Substep (2.D) of the third iteration $\mu^3 = \mu^2$.

  Hence, μ^2 is optimal with $K_2 = L_4[2, 2, 2, 2]$ and the corresponding optimal performance is

$$J^* = J_{\mu^2} = [0, 0, 0, 0]^\top.$$

- **Initialization**: The initial policy μ^0 is selected as $\mu^0(x) = L_4[1, 1, 1, 1]x, \forall x$ Δ_{12}.
- **Policy Evaluation**:
  Applying Lemma 2, obtain $J_{\mu^0} = [1, 1, 1, 1]^T$, $h_{\mu^0} = [0, 0, 0, 0]^T$.
- l**Policy Improvement**:
  Substep (2.A), obtain μ^1 with $K_1 = L_4[1, 1, 1, 1]$;
  Substep (2.B), since $\mu^1 = \mu^0$, go to (2.C);
  Substep (2.C), renew policy μ^1 with $K_1 = L_4[2, 1, 1, 1]$;
  Substep (2.D), since $\mu^1 \neq \mu^0$, return to the **Step 1**.

$$\vdots$$

- Substep (2.D) of the third iteration $\mu^3 = \mu^2$.

  Hence, μ^2 is optimal with $K_2 = L_4[2, 2, 2, 2]$ and the corresponding optimal performance is

$$J^* = J_{\mu^2} = [0, 0, 0, 0]^\top.$$

Complexity analysis.

- In Step 1 of Algorithm 5.2, since for each $\mu \in \mathcal{U}$, $I_N - L_\mu^\top$ is a special sparse matrix with $\tau(I_N - L_\mu^\top) \leq 2N$. Hence, according to [11], the complexity of Jordan decomposition (19) in Step 1 is $O(N^2)$.
- Furthermore, in the computation of J_{μ_n}, and h_{μ_n}, the matrix-vector multiplication performs $3N^2$ scalar multiplication and $3N(N-1)$ additions.
- Thus, in each loop, the complexity of Step 1 (Policy evolution) is $O(N^2)$.

Complexity analysis.

- In Step 1 of Algorithm 5.2, since for each $\mu \in \mathcal{U}$, $I_N - L_\mu^\top$ is a special sparse matrix with $\tau(I_N - L_\mu^\top) \leq 2N$. Hence, according to [11], the complexity of Jordan decomposition (19) in Step 1 is $O(N^2)$.
- Furthermore, in the computation of J_{μ_n}, and h_{μ_n}, the matrix-vector multiplication performs $3N^2$ scalar multiplication and $3N(N-1)$ additions.
- Thus, in each loop, the complexity of Step 1 (Policy evolution) is $O(N^2)$.

Complexity analysis.

- In Step 1 of Algorithm 5.2, since for each $\mu \in \mathcal{U}$, $I_N - L_\mu^\top$ is a special sparse matrix with $\tau(I_N - L_\mu^\top) \leq 2N$. Hence, according to [11], the complexity of Jordan decomposition (19) in Step 1 is $O(N^2)$.
- Furthermore, in the computation of J_{μ_n}, and h_{μ_n}, the matrix-vector multiplication performs $3N^2$ scalar multiplication and $3N(N-1)$ additions.
- Thus, in each loop, the complexity of Step 1 (Policy evolution) is $O(N^2)$.

- Since Substep 2.B and 2.D in Algorithm 1 are decision making statements, Policy improvement has two main part as: Substep 2.A and Substep 2.C.
- The argmin process in Substep 2.A is accomplished with $M-1$ comparisons. Furthermore, recalling each column of L_μ has a unique nonzero entry, Substep 2.A need $N(2M-1)$ operations.
- Similarly, Substep 2.C of Policy improvement need $N(3M-1)$ operations.
- Thus, in each loop, the complexity of Step 2 (Policy improvement) is $O(NM)$.
- As a result, the complexity of each iteration loop of Algorithm 5.2 is

$$O(N^2+NM).$$

- Since Substep 2.B and 2.D in Algorithm 1 are decision making statements, Policy improvement has two main part as: Substep 2.A and Substep 2.C.
- The argmin process in Substep 2.A is accomplished with $M - 1$ comparisons. Furthermore, recalling each column of L_μ has a unique nonzero entry, Substep 2.A need $N(2M - 1)$ operations.
- Similarly, Substep 2.C of Policy improvement need $N(3M - 1)$ operations.
- Thus, in each loop, the complexity of Step 2 (Policy improvement) is $O(NM)$.
- As a result, the complexity of each iteration loop of Algorithm 5.2 is

$$O(N^2 + NM).$$

- Since Substep 2.B and 2.D in Algorithm 1 are decision making statements, Policy improvement has two main part as: Substep 2.A and Substep 2.C.
- The argmin process in Substep 2.A is accomplished with $M - 1$ comparisons. Furthermore, recalling each column of L_μ has a unique nonzero entry, Substep 2.A need $N(2M - 1)$ operations.
- Similarly, Substep 2.C of Policy improvement need $N(3M - 1)$ operations.
- Thus, in each loop, the complexity of Step 2 (Policy improvement) is $O(NM)$.
- As a result, the complexity of each iteration loop of Algorithm 5.2 is

$$O(N^2 + NM).$$

- Since Substep 2.B and 2.D in Algorithm 1 are decision making statements, Policy improvement has two main part as: Substep 2.A and Substep 2.C.
- The argmin process in Substep 2.A is accomplished with $M-1$ comparisons. Furthermore, recalling each column of L_μ has a unique nonzero entry, Substep 2.A need $N(2M-1)$ operations.
- Similarly, Substep 2.C of Policy improvement need $N(3M-1)$ operations.
- Thus, in each loop, the complexity of Step 2 (Policy improvement) is $O(NM)$.
- As a result, the complexity of each iteration loop of Algorithm 5.2 is

$$O(N^2+NM).$$

- Since Substep 2.B and 2.D in Algorithm 1 are decision making statements, Policy improvement has two main part as: Substep 2.A and Substep 2.C.
- The argmin process in Substep 2.A is accomplished with $M-1$ comparisons. Furthermore, recalling each column of L_μ has a unique nonzero entry, Substep 2.A need $N(2M-1)$ operations.
- Similarly, Substep 2.C of Policy improvement need $N(3M-1)$ operations.
- Thus, in each loop, the complexity of Step 2 (Policy improvement) is $O(NM)$.
- As a result, the complexity of each iteration loop of Algorithm 5.2 is

$$O(N^2+NM).$$

- The worst case possibility of iteration number is $M^N - 1$.
- Hence, the total computational complexity of Algorithm 5.2 is

$$O(M^N \cdot (N^2 + NM)).$$

- The value iteration approach [17] is a ε-suboptimal approximation process, given error tolerance ε.
- Notice that the complexity of each value iteration loop is $O(NM)$.
- Hence, the total complexity of the VI algorithm [17] is

$$O(\tilde{N}(\varepsilon) \cdot NM),$$

with iteration number $\tilde{N}(\varepsilon)$, which depends on error tolerance ε.
- The iteration numbers is not upper bounded, i.e.,

$$\lim_{\varepsilon \to 0} \tilde{N}(\varepsilon) = +\infty.$$

[17] Fornasini, E., Valcher, M. E. (2014). Optimal control of boolean control networks. IEEE Transactions on Automatic Control, 59(5), 1258 – 1270.

- The worst case possibility of iteration number is $M^N - 1$.
- Hence, the total computational complexity of Algorithm 5.2 is

$$O(M^N \cdot (N^2 + NM)).$$

- The value iteration approach [17] is a ε-suboptimal approximation process, given error tolerance ε.
- Notice that the complexity of each value iteration loop is $O(NM)$.
- Hence, the total complexity of the VI algorithm [17] is

$$O(\tilde{N}(\varepsilon) \cdot NM),$$

  with iteration number $\tilde{N}(\varepsilon)$, which depends on error tolerance ε.
- The iteration numbers is not upper bounded, i.e.,

$$\lim_{\varepsilon \to 0} \tilde{N}(\varepsilon) = +\infty.$$

[17] Fornasini, E., Valcher, M. E. (2014). Optimal control of boolean control networks. IEEE Transactions on Automatic Control, 59(5), 1258 – 1270.

- The worst case possibility of iteration number is $M^N - 1$.
- Hence, the total computational complexity of Algorithm 5.2 is

$$O(M^N \cdot (N^2 + NM)).$$

- The value iteration approach [17] is a ε-suboptimal approximation process, given error tolerance ε.
- Notice that the complexity of each value iteration loop is $O(NM)$.
- Hence, the total complexity of the VI algorithm [17] is

$$O(\tilde{N}(\varepsilon) \cdot NM),$$

with iteration number $\tilde{N}(\varepsilon)$, which depends on error tolerance ε.
- The iteration numbers is not upper bounded, i.e.,

$$\lim_{\varepsilon \to 0} \tilde{N}(\varepsilon) = +\infty.$$

[17] Fornasini, E., Valcher, M. E. (2014). Optimal control of boolean control networks. IEEE Transactions on Automatic Control, 59(5), 1258 - 1270.

- The worst case possibility of iteration number is $M^N - 1$.
- Hence, the total computational complexity of Algorithm 5.2 is

$$O(M^N \cdot (N^2 + NM)).$$

- The value iteration approach [17] is a ε-suboptimal approximation process, given error tolerance ε.
- Notice that the complexity of each value iteration loop is $O(NM)$.
- Hence, the total complexity of the VI algorithm [17] is

$$O(\tilde{N}(\varepsilon) \cdot NM),$$

with iteration number $\tilde{N}(\varepsilon)$, which depends on error tolerance ε.

- The iteration numbers is not upper bounded, i.e.,

$$\lim_{\varepsilon \to 0} \tilde{N}(\varepsilon) = +\infty.$$

[17] Fornasini, E., Valcher, M. E. (2014). Optimal control of boolean control networks. IEEE Transactions on Automatic Control, 59(5), 1258 – 1270.

- The worst case possibility of iteration number is $M^N - 1$.
- Hence, the total computational complexity of Algorithm 5.2 is

$$O(M^N \cdot (N^2 + NM)).$$

- The value iteration approach [17] is a ε-suboptimal approximation process, given error tolerance ε.
- Notice that the complexity of each value iteration loop is $O(NM)$.
- Hence, the total complexity of the VI algorithm [17] is

$$O(\tilde{N}(\varepsilon) \cdot NM),$$

  with iteration number $\tilde{N}(\varepsilon)$, which depends on error tolerance ε.
- The iteration numbers is not upper bounded, i.e.,

$$\lim_{\varepsilon \to 0} \tilde{N}(\varepsilon) = +\infty.$$

[17] Fornasini, E., Valcher, M. E. (2014). Optimal control of boolean control networks. IEEE Transactions on Automatic Control, 59(5), 1258 – 1270.

- The worst case possibility of iteration number is $M^N - 1$.
- Hence, the total computational complexity of Algorithm 5.2 is

$$O(M^N \cdot (N^2 + NM)).$$

- The value iteration approach [17] is a ε-suboptimal approximation process, given error tolerance ε.
- Notice that the complexity of each value iteration loop is $O(NM)$.
- Hence, the total complexity of the VI algorithm [17] is

$$O(\tilde{N}(\varepsilon) \cdot NM),$$

  with iteration number $\tilde{N}(\varepsilon)$, which depends on error tolerance ε.
- The iteration numbers is not upper bounded, i.e.,

$$\lim_{\varepsilon \to 0} \tilde{N}(\varepsilon) = +\infty.$$

[17] Fornasini, E., Valcher, M. E. (2014). Optimal control of boolean control networks. IEEE Transactions on Automatic Control, 59(5), 1258 – 1270.

Output tracking problem for BCNs

Consider the following BCN with output

$$\begin{cases} x(t+1) = L \ltimes u(t) \ltimes x(t), \\ y(t) = Cx(t), \end{cases} \tag{47}$$

The output tracking problem for network (47) with $x(0) = x_0$ is to design a control input $\mathbf{u} = \{u(t) : t \in \mathbb{Z}_{\geq 0}\}$, s.t. the output $y(t; x_0, \mathbf{u})$ tracks a given reference $y_r \in \Delta_P$, that is, there exists an integer $\tau > 0$ such that $y(t; x_0, \mathbf{u}) = y_r$, for all $t \geq \tau$.

A constructive procedure was designed in [13] to obtain output tracking state feedback controllers for BCNs.

[13] Li, H., Wang, Y., Xie, L. Output tracking control of boolean control networks via state feedback: constant reference signal case. Automatica, 2015.

For the reference signal $y_r = \delta_P^\alpha$, define a set, denoted by $\mathcal{S}(\alpha) \subset \Delta_N$, as $\mathcal{S}(\alpha) = \{\delta_N^r : Col_r(C) = \delta_P^\alpha, 1 \leq r \leq N\}$.

Now define a special per-step cost function g associate with δ_P^α as

$$g(\delta_N^i, \delta_M^j) = \begin{cases} 0, & \text{if } \delta_N^i \in \mathcal{S}(\alpha), \\ 1, & \text{if } \delta_N^i \notin \mathcal{S}(\alpha). \end{cases} \tag{48}$$

Theorem

The output of network (47) tracks the reference signal $y_r = \delta_P^\alpha$ by a control sequence $\mathbf{u}$ if and only if $\mathbf{u}$ can solve the optimal control problem (12) with per-step cost g given by (48), and $J^* = 0$.

Optimal intervention of Ara operon in E. coil

We consider an optimal intervention problem of Ara operon in *E. coil* . [12], shown in Fig. 3, and the update logics is

$$\begin{cases} f_A = A_e \wedge T, \\ f_{A_m} = (A_{em} \wedge T) \vee A_e, \\ f_{A_{ra_+}} = (A_m \vee A) \wedge A_{ra_-}, \\ f_C = \neg G_e \\ f_E = M_S \\ f_D = \neg A_{ra_+} \wedge A_{ra_-}, \\ f_{M_S} = A_{ra_+} \wedge C \wedge \neg D, \\ f_{M_T} = A_{ra_+} \wedge C, \\ f_T = M_T. \end{cases} \quad (49)$$

Here, four Boolean control parameters are A_e, A_m, Ara_-, and G_e, respectively.

Optimal intervention of Ara operon in E. coil

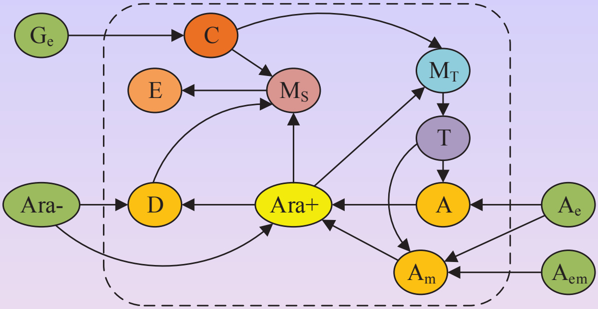


Figure 3: A Boolean model of Ara operon in *E. coil*. M_S denotes the $mRNA$ of the structural genes (araBAD), MT is the $mRNA$ of the transport genes (araE-FGH), E is the enzymes $AraA, AraB$, and $AraD$, coded for by the structural genes, T is the transport protein, coded for by the transport genes, A is the intracellular arabinose (high levels), A_m 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.

Optimal intervention of Ara operon in E. coil

According to Th. 5. 2 of [1], Monostability and Bistability of this network was considered in [7].

Figure 4: The state transition graph of Ara operonp.

[1] D. Cheng, H. Qi, and Z. Li, Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach, Springer, 2011.

[7] S. Chen, Y. Wu, M. Macauley, X. Sun, Monostability and Bistability of Boolean Networks Using Semitensor Products, IEEE TCNS, 2019

Optimal intervention of Ara operon in E. coil

Set

- $(A, A_m, A_{ra_+}, C, E, D, M_S, M_T, T) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$
- $(A_e, A_{em}, A_{ra_-}, G_e) = (u_1, u_2, u_3, u_4)$

Then, based on STP, the vector expression of Boolean network (49) is obtained as

$$x(t+1) = Lu(t)x(t),$$

with a structure matrix

$$L \in \mathcal{L}_{2^9 \times 2^{13}}.$$

Consider the average cost problem, with the cost function $g : \Delta_{2^9} \times \Delta_{2^4} \to \mathcal{R}$ as

$$g(x,u) = g(\ltimes_{i=1}^{9} x_i, \ltimes_{j=1}^{4} u_j) = \mathcal{A}X + \mathcal{B}U. \quad (50)$$

Optimal intervention of Ara operon in E. coil

Set

- $(A, A_m, A_{ra_+}, C, E, D, M_S, M_T, T) = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$
- $(A_e, A_{em}, A_{ra_-}, G_e) = (u_1, u_2, u_3, u_4)$

Then, based on STP, the vector expression of Boolean network (49) is obtained as

$$x(t+1) = Lu(t)x(t),$$

with a structure matrix

$$L \in \mathcal{L}_{2^9 \times 2^{13}}.$$

Consider the average cost problem, with the cost function $g : \Delta_{2^9} \times \Delta_{2^4} \to \mathcal{R}$ as

$$g(x,u) = g(\ltimes_{i=1}^{9} x_i, \ltimes_{j=1}^{4} u_j) = \mathcal{A}X + \mathcal{B}U. \tag{50}$$

Optimal intervention of Ara operon in E. coil

According to discussion for the lac operon in [18], weight vectors are

$$\mathcal{A} = [-28, -12, 12, 16, 0, 0, 0, 20, 16], \quad \mathcal{B} = [-8, 40, 20, 40].$$

Then, applying Algorithm 5.2

- the optimal performance $J^*(x) \equiv -4$, for all $x \in \Delta_{512}$,
- optimal feedback control law $\mu^*(x) = \delta_{16}^9$, for all $x \in \Delta_{512}$,
- optimal stationery control parameters are $(A_e, A_m, Ara_-, G_e) = (1, 0, 0, 0)$.

Optimal intervention of Ara operon in E. coil

According to discussion for the lac operon in [18], weight vectors are

$$\mathcal{A} = [-28, -12, 12, 16, 0, 0, 0, 20, 16], \quad \mathcal{B} = [-8, 40, 20, 40].$$

Then, applying Algorithm 5.2

- the optimal performance $J^*(x) \equiv -4$, for all $x \in \Delta_{512}$,
- optimal feedback control law $\mu^*(x) = \delta_{16}^{9}$, for all $x \in \Delta_{512}$,
- optimal stationery control parameters are $(A_e, A_m, Ara_-, G_e) = (1, 0, 0, 0)$.

Optimal intervention of Ara operon in E. coil

According to discussion for the lac operon in [18], weight vectors are

$$\mathcal{A} = [-28, -12, 12, 16, 0, 0, 0, 20, 16], \quad \mathcal{B} = [-8, 40, 20, 40].$$

Then, applying Algorithm 5.2

- the optimal performance $J^*(x) \equiv -4$, for all $x \in \Delta_{512}$,
- optimal feedback control law $\mu^*(x) = \delta_{16}^9$, for all $x \in \Delta_{512}$,
- optimal stationery control parameters are $(A_e, A_m, Ara_-, G_e) = (1, 0, 0, 0)$.

Optimal intervention of Ara operon in E. coil

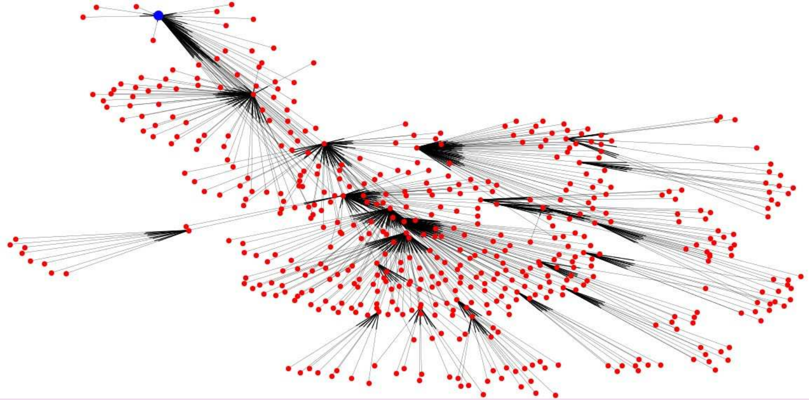


Figure 5: The state transition graph of the lac operon with control parameters $(A_e, A_m, Ara_-, G_e) = (1, 0, 0, 0)$. The unique steady state $(0, 1, 0, 1, 0, 0, 0, 0, 0)$, correspond to δ_{512}^{161}, is represented by a blue dot, and all transient states are denoted by red dots.

Optimal intervention of Ara operon in E. coil

The optimal approximation cost $\frac{1}{T}\hat{J}_T^*(x_0)$ of the value iteration approach [17] with six different initial states are shown in Fig 6.

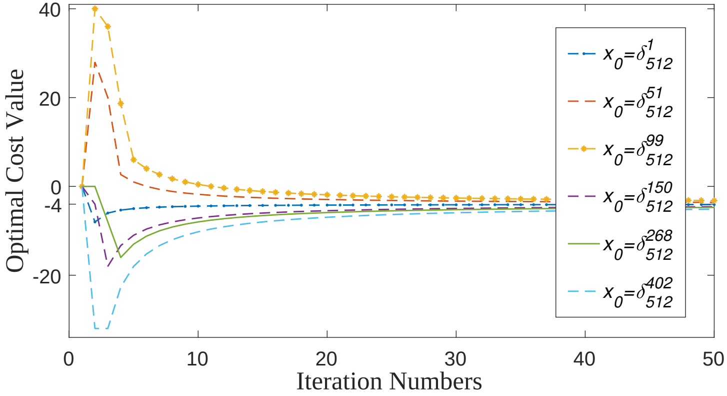


Figure 6: Value iteration approximation result for the Ara operon Network with different initial states.

Optimal intervention of Ara operon in E. coil

As both algorithms ran on the same computer, iteration numbers are collected in Table 1.

A computer with Quad-Core 3.2 GHz processor and 8 GB RAM memory.

Table 1: Comparison of iteration numbers and running times

	Policy Iteration	Value Iteration		
		$\varepsilon = 0.5$	$\varepsilon = 0.1$	$\varepsilon = 0.005$
Iteration Numbers	3	113	561	11187
Running Time (Sec)	8.53771	1.97353	9.17410	556.41600

Future work or challenge

- Data Driven Identification and Control
- Reinforcement Learning, such as Q-Learning
- Computational Complexity

Future work or challenge

- Data Driven Identification and Control
- Reinforcement Learning, such as Q-Learning
- Computational Complexity

Future work or challenge

- Data Driven Identification and Control
- Reinforcement Learning, such as Q-Learning
- Computational Complexity

Reference I

[1] Cheng, D., Qi, H., Li, Z., 2011. Analysis and control of Boolean networks: a semi-tensor product approach. London, U.K.: Springer.

[2] D. Cheng, “On finite potential games,” *Automatica*, vol. 50, no. 7, pp. 1793–1801, 2014.

[3] F. Ettore and V. M. Elena, “Fault detection problems for boolean networks and boolean control networks,” in *Control Conference (CCC), 2015 34th Chinese*. IEEE, 2015, pp. 1–8.

[4] E. Fornasini and M. E. Valcher, “Fault detection analysis of boolean control networks,” *Automatic Control, IEEE Transactions on*, vol. 60, no. 10, pp. 2374–2379, 2015.

[5] 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. Therm. Eng.*, vol. 93, pp. 251–259, 2016.

[6] Y. Wu and T. Shen, “Policy iteration approach to control of residual gas fraction in ic engines under the framework of stochastic logical dynamics,” *IEEE Trans. Control Syst. Technol.*, vol. 25, pp. 1100–1107, 2017.
I

[7] D. Cheng and H. Qi, “Controllability and observability of boolean control networks,” *Automatica*, vol. 45, no. 7, pp. 1659–1667, 2009.

[8] D. Laschov, M. Margaliot, and G. Even, “Observability of boolean networks: A graph-theoretic approach,” *Automatica*, vol. 49, no. 8, pp. 2351–2362, 2013.

[9] D. Laschov and M. Margaliot, “Minimum-time control of boolean networks,” *SIAM Journal on Control and Optimization*, vol. 51, no. 4, pp. 2869–2892, 2013.

[10] Q. Liu, X. Guo, and T. Zhou, “Optimal control for probabilistic boolean networks,” *IET systems biology*, vol. 4, no. 2, pp. 99–107, 2010.

[11] 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, 2015.

[12] R. Pal, A. Datta, and E. R. Dougherty, “Optimal infinite-horizon control for probabilistic boolean networks,” *IEEE Trans. Signal Proces.*, vol. 54, no. 6, pp. 2375–2387, 2006.

[13] D. Cheng, Y. Zhao, and T. Xu, “Receding horizon based feedback optimization for mix-valued logical networks,” *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3362–3366, 2015.

[14] Y. Wu and T. Shen, “A finite convergence criterion for the discounted optimal control of stochastic logical networks,” under second round review.

[15] Y. Wu and T. Shen, “Policy iteration algorithm for optimal control of stochastic logical dynamical systems,” *IEEE Trans. Neural. Netw. Learn. Syst.*, 2017, accepted, DOI: 10.1109/TNNLS.2017.2661863.

Reference II

[16] Y. Zhao, Z. Li, and D. Cheng, "Optimal control of logical control networks," *IEEE Trans. Autom. Control*, vol. 56, no. 8, pp. 1766–1776, 2011.

[17] E. Fornasini and M. E. Valcher, "Optimal control of boolean control networks," *Automatic Control, IEEE Transactions on*, vol. 59, no. 5, pp. 1258–1270, 2014.

[18] H. Li, Y. Wang, and L. Xie, "Output tracking control of boolean control networks via state feedback: Constant reference signal case," *Automatica*, vol. 59, pp. 54–59, 2015.

[19] J. Heidel, J. Maloney, C. Farrow, and J. A. Rogers, "Finding cycles in synchronous boolean networks with applications to biochemical systems," *International Journal of Bifurcation and Chaos*, vol. 13, no. 03, pp. 535–552, 2003.

谢　谢　！

Any Question?