SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

April 2026, Vol. 69, Iss. 4, 140202:1–140202:17
https://doi.org/10.1007/s11432-025-4787-9

Special Topic: Logical System Control

Optimal control of Boolean control networks: a data-driven perspective

Giorgia DISARÒ & Maria Elena VALCHER*

Dipartimento di Ingegneria dell'Informazione, Università di Padova, Padova 2-35122, Italy

Received 3 October 2025/Revised 28 November 2025/Accepted 13 January 2026

Abstract Data-driven approaches have recently emerged in the analysis of Boolean control networks (BCNs), with the aim of addressing fundamental control problems, such as state-feedback stabilization, safe control, and output regulation, without requiring an explicit model, provided that a sufficiently informative dataset is available. This paper develops a data-based framework for the finite-horizon and infinite-horizon optimal control problems in BCNs. For the finite-horizon problem, the conditions for solvability are relatively mild. In contrast, solving the infinite-horizon problem from data imposes stricter requirements on both the generating BCN and the collected dataset. Our analysis builds on the concept of data informativity, which ensures that any proposed solution is feasible for all BCNs consistent with the data. While the resulting controllers may not be optimal for every such BCN, they represent the best performance attainable given the available information. The degree of sub-optimality is characterized in detail, and the effectiveness of the proposed (both finite- and infinite-horizon) methods is illustrated through an example based on a biological system of practical relevance.

Keywords data-driven methods, Boolean control networks, algebraic representation, optimal control, λ switch

Citation Disarò G, Valcher M E. Optimal control of Boolean control networks: a data-driven perspective. Sci China Inf Sci, 2026, 69(4): 140202, https://doi.org/10.1007/s11432-025-4787-9

1 Introduction

Originally introduced by Kauffman [1] to study genetic regulatory mechanisms, Boolean networks (BNs) have since been applied across a wide range of domains, due to their ability to capture complex dynamics within a simple binary framework. Indeed, application areas include cell regulation [2], game theory [3], smart homes [4], and personalized healthcare management [5]. A BN is a finite-state system in which each variable takes values in the Boolean set. Similar to classical dynamical systems, BNs can be influenced by external inputs, giving rise to the concept of a Boolean control network (BCN). A BCN can be viewed as a parametrized family of BNs, where each configuration of the control inputs determines a distinct logical evolution.

Early analyses of BCNs relied primarily on combinatorial and enumerative representations of states and transitions, which made the formulation and solution of advanced control problems challenging. A breakthrough came with the introduction of the semi-tensor product (STP) by Cheng [6], which provided an algebraic representation of BCNs [7–10]. The key idea of the STP approach is that a Boolean network with n state variables exhibits 2^n possible configurations, each of which can be represented by a canonical vector of dimension 2^n. Consequently, the logical state-update function can be equivalently represented by a $2^n \times 2^n$ logical matrix, by this meaning that its columns are canonical vectors. In this way, a BN can be reformulated as a discrete-time linear system, and analogously, a BCN can be transformed into a discrete-time bilinear system. Despite this linear and bilinear formulation, classical tools and results from linear and bilinear system theory cannot be directly applied to BNs and BCNs, respectively, because in these models all vectors are constrained to be canonical. Nonetheless, this matrix-based framework made it possible to adapt standard tools from discrete-time state-space models to the analysis and control of BCNs [11–14].

Despite these advances, the sheer complexity and high-dimensional behavior of BCNs often render accurate modeling or system identification impractical. To address these challenges, recent research has explored data-driven approaches to BCN control [15–17]. These methods aim to leverage informative datasets to design control strategies without requiring complete knowledge of the underlying system dynamics, thereby mitigating the limitations of model-based techniques.

* Corresponding author (email: meme@dei.unipd.it)

© The Author(s) 2026. This article is published with open access at link.springer.com.

The authors in [16] propose a one-step data-driven approach to solve the output stabilization problem, which is inspired by the results available for the stabilization of probabilistic Boolean networks. The solution relies on an algorithm that first computes an approximation of the output prediction matrix, based on data, and then generates a sequence of concentric annular sets, each containing the output values that lie at a specific "distance" from the desired output value, with respect to the output/input/output transitions captured by the data-based output prediction matrix. Finally, using these sets, the algorithm computes, when possible, an output feedback gain.

An alternative approach was proposed in [17], where the concept of informativity, originally introduced in [18] (see also [19]), was first applied to BCNs. Within this framework, controllability and stabilizability were analyzed, and problems such as state-feedback stabilization and guaranteed-cost control were addressed using solely data. The central idea of informativity is that, when the available data are insufficient to uniquely identify the system, one can instead require that every system consistent with the data (including the true, unknown system) satisfies the desired property or achieves, via the designed control action, the desired behavior.

Recently, the problems of safe control and output regulation of BCNs using only data have been investigated in [15]. In that work, necessary and sufficient conditions were established for the existence of a state-feedback law that ensures either the evolution of the state of the original BCN (and of all the BCNs consistent with the data) towards a designated safe set, or the regulation of its output to a desired value. For both problems, constructive procedures were also proposed to design the corresponding feedback matrix directly from the collected data.

In this paper, we investigate the finite-horizon and infinite-horizon optimal control problems for BCNs using exclusively data generated by the target network. The optimal control of BCNs has previously been studied from a model-based perspective in a number of studies [10, 20–23], while related results for probabilistic Boolean networks have been reported, e.g., in [14, 24–26]. These contributions underline the natural relevance of optimal control in the context of biological systems, such as genetic regulatory networks, and in particular the core network governing the mammalian cell cycle [27], for which logical networks provide a convenient modeling framework.

In [27], both finite-horizon and infinite-horizon optimal control problems for BCNs were addressed. By employing the semi-tensor product, the problem was solved through a recursive algorithm that serves as the analogue of the Riccati difference equation for linear systems. For the infinite-horizon case, the solution was obtained as the limit of the finite-horizon problem over $[0, T]$, and shown to coincide with the finite-horizon solution (with no terminal cost) on sufficiently long horizons.

The infinite-horizon optimal control problem with a time-discounted performance criterion was studied in [20]. The authors showed that the optimal strategy can be found among essentially periodic strategies, meaning those that become periodic after a finite transient.

Another approach was proposed in [28], where the authors addressed the infinite-horizon optimal control problem of BCNs with an average cost criterion. Using the semi-tensor product of matrices and a Jordan decomposition approach, they derived a nested optimality equation for the average-cost problem. Furthermore, by exploiting a Laurent series representation, they developed a novel policy-iteration algorithm capable of computing the optimal state-feedback controller in a finite number of iterations.

Both finite- and infinite-horizon optimal control problems for BCNs were investigated in [23], by introducing an optimal input-state transfer graph. For finite-horizon problems, a sufficient condition is derived and an algorithm based on binary decomposition is proposed. For infinite-horizon discounted problems, the existence of optimal control is established, and a novel algorithm is developed to compute the corresponding optimal controllers.

The authors in [29] investigated the Bolza-type optimal control problem for logical control networks under function perturbations, focusing on the robustness of the resulting optimal control sequence. By leveraging the structural properties of the state transition matrix and the stage cost function, they constructed a novel optimal matrix that offers a new approach to addressing the solvability of the problem. This matrix was further employed to characterize the deviation of the cost function induced by perturbations, through which several necessary and sufficient conditions were established to guarantee the robustness of the optimal control sequence. As a generalization of the robust Bolza-type optimal control problem, the infinite-horizon optimal control problem was also investigated.

More recently, the authors in [30] proposed a Q-learning-based method for solving the finite-horizon optimal control problem for BCNs subject to disturbances. They developed a robust Q-learning algorithm capable of learning the optimal policy through interactions with the environment, even when disturbances are present.

The goal of this article is to explore how the solutions to the finite-horizon and infinite-horizon optimal control problems provided in [27], which are particularly suitable for adaptation to a data-driven framework, can be adjusted to cope with the fact that we do not fully know the model of the BCN, but we have partial knowledge of it based on collected data. To do this, we adopt the strategy proposed in [15, 17], and we solve the problems for all the BCNs compatible with the collected data. However, by proceeding in this way, the set of admissible solutions will generically turn out to be only sub-optimal for the original BCN. Therefore, among all the feasible sub-optimal solutions, we aim to find the one that is optimal based on the available data. Such a solution will be called data-optimal. We derive necessary and sufficient conditions for the solvability of both problems, and provide algorithms to obtain the data-optimal solutions. Moreover, for both the finite-horizon and the infinite-horizon optimal control problems, we also provide a lower bound and an upper bound (the latter coinciding with the data-optimal solution) for the actual optimal solutions. The paper builds on the conference paper [31], where finite-horizon and infinite-horizon optimal control problems were investigated. Compared with [31], here we have extended the analysis of finite-horizon optimal control to the case of time-varying costs, and introduced Algorithms 2 and 3 for the problem's solutions. We have derived lower and upper bounds on the data-optimal solutions, and tested the proposed finite-horizon and infinite-horizon data-optimal solutions on a practical example, borrowed from biology.

The paper is organized as follows. Section 2 contains some preliminary notions and results about BCNs. In Section 3 we recall the statement and solution of the finite-horizon optimal control problem for BCNs, and provide an algorithm to solve this problem by relying only on the information about the admissible transitions contained in the collected data. We also discuss the tightness of the upper bound given by the data-optimal solution by proposing a potential lower bound on the true optimal cost. In Section 4 we recall the infinite-horizon optimal control problem solution, derive necessary and sufficient conditions for the problem solvability based on data, and provide an algorithmic procedure to derive the data-optimal solution. Finally, we demonstrate that the data-based algorithm yields an upper bound on the optimal cost for all BCNs compatible with the observed data, and we additionally establish a (tight) lower bound.

Notation. Given two nonnegative integers k and n, with $k \leqslant n$, let $[k,n]$ denote the integer set $\{k, k+1, \dots, n\}$. We consider Boolean vectors and matrices whose entries take values in $\mathcal{B} \triangleq \{0,1\}$, equipped with the standard logical operations: sum (OR) $\vee$, product (AND) $\wedge$, and negation (NOT) $\neg$. The symbol δ_k^i denotes the k-dimensional i-th canonical vector, $\mathcal{L}_k$ is the set of all k-dimensional canonical vectors, and $\mathcal{L}_{k\times q} \subset \mathcal{B}^{k\times q}$ is the set of all $k \times q$ logical matrices, namely matrices whose q columns are canonical vectors of size k. A logical matrix $L \in \mathcal{L}_{k\times q}$ can be described as $L = [\delta_k^{i_1}\ \delta_k^{i_2}\ \dots\ \delta_k^{i_q}]$, for indices $i_1, i_2, \dots, i_q \in [1,k]$. I_n is the n-dimensional identity matrix. The k-dimensional vector with all unitary (zero) entries is denoted by $\mathbb{1}_k$ (by $\mathbb{0}_k$). The (i,j)-th entry of a matrix M is $[M]_{ij}$, and the j-th entry of a vector $\mathbf{v}$ is $[\mathbf{v}]_j$. A vector sequence $\mathbf{v}(i), \dots, \mathbf{v}(j)$, with $i, j \in \mathbb{Z}_+$ and $i \leqslant j$, is denoted by $\{\mathbf{v}(t)\}_{t=i}^{j}$. Given a set $\mathcal{S}$, the symbol $|\mathcal{S}|$ denotes its cardinality.

There is a one-to-one correspondence between Boolean variables $X \in \mathcal{B}$ and vectors $\mathbf{x} \in \mathcal{L}_2$, given by $\mathbf{x} = [X\ \neg X]^\top$. For two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$, we define their (left) semi-tensor product ($\ltimes$) (see [6]) as

$$A \ltimes B \triangleq (A \otimes I_{l/n})(B \otimes I_{l/p}), \quad l \triangleq \text{l.c.m.}\{n,p\}.$$

Details about the properties of the semi-tensor product can be found in [10]. The semi-tensor product allows us to extend the one-to-one correspondence between $\mathcal{B}$ and $\mathcal{L}_2$ to a one-to-one correspondence between $\mathcal{B}^n$ and $\mathcal{L}_{2^n}$, as detailed in the following. Each vector $X = [X_1\ X_2\ \dots\ X_n]^\top \in \mathcal{B}^n$ is mapped to

$$\mathbf{x} \triangleq \begin{bmatrix} X_1 \\ \neg X_1 \end{bmatrix} \ltimes \begin{bmatrix} X_2 \\ \neg X_2 \end{bmatrix} \ltimes \cdots \ltimes \begin{bmatrix} X_n \\ \neg X_n \end{bmatrix}.$$

Given a Boolean (in particular, a logical) matrix $L \in \mathcal{B}^{k\times k}$, we associate with it the directed graph (digraph), $\mathcal{G}(L) = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [1,k]$ is the set of nodes and $\mathcal{E}$ is the set of edges. An edge (j,ℓ), from j to ℓ, belongs to $\mathcal{E}$ if and only if $[L]_{\ell j} = 1$. A sequence $j_1 \to j_2 \to \cdots \to j_r \to j_{r+1}$ in $\mathcal{G}(L)$ is a path of length r from j_1 to j_{r+1} if $(j_1, j_2), \dots, (j_r, j_{r+1})$ are edges in $\mathcal{E}$. A closed path is a cycle. A node $j^* \in \mathcal{V}$ is said to be globally reachable if there exists a path to j^* from any other node in the digraph.

2 Boolean control networks: preliminaries

A BCN is a logical system described by the following equation:

$$X(t+1) = f(X(t), U(t)), \quad t \in \mathbb{Z}_+, \tag{1}$$

where $X(t) \in \mathcal{B}^n$ is the n-dimensional state variable and $U(t) \in \mathcal{B}^m$ is the m-dimensional input at time t. $f : \mathcal{B}^n \times \mathcal{B}^m \to \mathcal{B}^n$ is a logical function. A BCN whose dynamics is not influenced by external inputs is a Boolean network (BN), and its describing equation is

$$X(t+1) = f(X(t)), \quad t \in \mathbb{Z}_+, \tag{2}$$

where $f : \mathcal{B}^n \to \mathcal{B}^n$ is a logical function. By exploiting the one-to-one correspondence between Boolean and logical vectors, the BN (2) can be equivalently described through its algebraic representation [10]:

$$\mathbf{x}(t+1) = L\mathbf{x}(t), \quad t \in \mathbb{Z}_+, \tag{3}$$

where $\mathbf{x}(t) \in \mathcal{L}_N$ and $L \in \mathcal{L}_{N\times N}$, with $N \triangleq 2^n$. Similarly, the algebraic representation of the BCN (1) is

$$\mathbf{x}(t+1) = L \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \tag{4}$$

where $\mathbf{x}(t) \in \mathcal{L}_N$, $\mathbf{u}(t) \in \mathcal{L}_M$, and $L \in \mathcal{L}_{N\times NM}$, with $N = 2^n$ and $M \triangleq 2^m$. It is worth noting that model (4) can represent any state-space model whose state and input variables take values in finite sets of cardinalities N and M, respectively. Consequently, all subsequent analysis applies to arbitrary nonnegative integer values of N and M, not necessarily powers of 2. The logical matrix $L \in \mathcal{L}_{N\times NM}$ can be partitioned into M square blocks of size N as follows:

$$L = \left[\, L_1 \middle| L_2 \middle| \dots \middle| L_M \,\right], \tag{5}$$

where each logical matrix $L_i \in \mathcal{L}_{N\times N}, i \in [1, M]$, captures the dynamics of the Boolean control network corresponding to the i-th input (i.e., the i-th subsystem of the BCN). In other words,

$$\mathbf{x}(t+1) = L_i\mathbf{x}(t), \quad t \in \mathbb{Z}_+ \tag{6}$$

represents the dynamics of the BCN when $\mathbf{u}(t) = \delta_M^i, \forall t \in \mathbb{Z}_+$.

3 Finite-horizon data-optimal control problem

A complete solution to the optimal control of Boolean control networks, by making use of their algebraic representation, was first proposed in [27]. It has been shown that every finite-horizon optimal control problem for a BCN described as in (4) can be stated as follows:

$$\begin{aligned} &\min\nolimits_{\mathbf{u}(0),\dots,\mathbf{u}(T-1)} J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) \\ &\text{subject to (4) and } \mathbf{x}(0) = \mathbf{x}_0, \end{aligned} \tag{7}$$

where $\mathbf{x}_0 \in \mathcal{L}_N$ is the (arbitrarily chosen) initial condition and

$$J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) \triangleq \mathbf{c}_f^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \mathbf{c}(t)^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \tag{8}$$

with $\mathbf{c}_f \in \mathbb{R}_+^N$ and $\{\mathbf{c}(t)\}_{t=0}^{T-1}, \mathbf{c}(t) \in \mathbb{R}_+^{NM}$, a sequence of time-varying nonnegative cost vectors. At each time $t \in \mathbb{Z}_+$, the vector $\mathbf{c}(t)$ can be partitioned into M blocks of dimension N as follows:

$$\mathbf{c}(t) = \left[\, \mathbf{c}_1(t)^\top \middle| \mathbf{c}_2(t)^\top \middle| \dots \middle| \mathbf{c}_M(t)^\top \,\right]^\top,$$

where $\mathbf{c}_i(t) \in \mathbb{R}_+^N, i \in [1, M]$. The entry $[\mathbf{c}_i(t)]_j$ represents the cost of applying the input δ_M^i when the system is in state δ_N^j at time t. Indeed, this is the most general formulation one can conceive for BCNs described as in (4) (see Section IV-A in [27]), and it encompasses both the case of time-invariant costs (i.e., $\mathbf{c}(t) = \mathbf{c}$, for every $t \in [0, T-1]$) and the case where $\mathbf{c}(t) = \lambda^t\mathbf{c}, \exists\lambda \in (0, 1)$, for every $t \in [0, T-1]$, investigated in [23] (see also [24]). Nonetheless, it requires only minor modifications compared to its time-invariant counterpart, first addressed in a data-driven framework in [31].

In a model-based framework, in order to solve the problem, one can proceed as follows. First, a sequence of (nonnegative) N-dimensional vectors $\{\mathbf{m}(t)\}_{t=0}^T$ is generated through Algorithm 1.

The problem solution, namely the minimum value of the cost function, is

$$J_T^*(\mathbf{x}_0) = \min_{\mathbf{u}(0),\dots,\mathbf{u}(T-1)} J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) = \mathbf{m}(0)^\top\mathbf{x}_0. \tag{9}$$

The corresponding optimal control input can be expressed as a time-varying state-feedback $\mathbf{u}(t) = K(t)\mathbf{x}(t)$, where[1)]

$$K(t) = \left[\delta_M^{i^*(1,t)} \; \delta_M^{i^*(2,t)} \; \dots \; \delta_M^{i^*(N,t)}\right],$$

1) In the paper, let the notation $i^* = \arg\min_i(\cdot)$ denote any index i for which the minimum is attained, even when such index is not unique.

Algorithm 1 Computation of the sequence $\{\mathbf{m}(t)\}_{t=0}^{T}$.

Input: The matrix $L \in \mathcal{L}_{N\times N}$; the vector $\mathbf{c}_f \in \mathbb{R}_+^N$; the sequence of vectors $\{\mathbf{c}(t)\}_{t=0}^{T-1}, \mathbf{c}(t) \in \mathbb{R}_+^{NM}$.
Output: The sequence $\{\mathbf{m}(t)\}_{t=0}^{T}$.
(1) *Initialization:* Set $t = T$ and $\mathbf{m}(t) = \mathbf{c}_f$.
(2) *Iterative procedure:*

```
t ← t − 1;
for j ∈ [1, N], do
    [m(t)]_j = min_{i∈[1,M]}{[c_i(t)]_j + [m(t+1)^⊤ L_i]_j};
end for
if t = 0, then
    Go to step (3);
end if
```

(3) *Conclusion:* Return $\{\mathbf{m}(t)\}_{t=0}^{T}$.

$$i^*(j,t) \triangleq \arg\min_{i\in[1,M]} \left\{[\mathbf{c}_i(t)]_j + [\mathbf{m}(t+1)^\top L_i]_j\right\},\ j \in [1,N].$$

We aim to investigate how the finite-horizon optimal control problem can be addressed when the complete model of the BCN is unavailable, but a large set of data generated by the BCN has been collected.

To address the problem, we assume the same set-up as in [15]; namely we suppose to have performed some offline experiments (say $r \geqslant 1$) during which we have collected state/input data from the BCN (4) on finite time intervals $[0, T_i], T_i \in \mathbb{Z}_+, i \in [1, r]$. We define the vector sequences $\mathbf{x}_d^i \triangleq \{\mathbf{x}_d^i(t)\}_{t=0}^{T_i}$, $\mathbf{u}_d^i \triangleq \{\mathbf{u}_d^i(t)\}_{t=0}^{T_i-1}$, and accordingly $\mathbf{x}_d \triangleq \{\mathbf{x}_d^i\}_{i=1}^r$, $\mathbf{u}_d \triangleq \{\mathbf{u}_d^i\}_{i=1}^r$. We rearrange the data collected during the r experiments into the following logical matrices with $T_d \triangleq \sum_{i=1}^r T_i$ columns:

$$X_p \triangleq \left[\, \mathbf{x}_d^1(0)\ \ldots\ \mathbf{x}_d^1(T_1-1) \middle| \ldots \middle| \mathbf{x}_d^r(0)\ \ldots\ \mathbf{x}_d^r(T_r-1) \,\right], \tag{10a}$$

$$X_f \triangleq \left[\, \mathbf{x}_d^1(1)\ \ldots\ \mathbf{x}_d^1(T_1) \middle| \ldots \middle| \mathbf{x}_d^r(1)\ \ldots\ \mathbf{x}_d^r(T_r) \,\right], \tag{10b}$$

$$U_p \triangleq \left[\, \mathbf{u}_d^1(0)\ \ldots\ \mathbf{u}_d^1(T_1-1) \middle| \ldots \middle| \mathbf{u}_d^r(0)\ \ldots\ \mathbf{u}_d^r(T_r-1) \,\right], \tag{10c}$$

where the subscripts p and f stand for past and future, respectively.

Since the data have been generated by the BCN (4), they are of course compatible with it, by this meaning that $\mathbf{x}_d^i(t+1) = L \ltimes \mathbf{u}_d^i(t) \ltimes \mathbf{x}_d^i(t), \forall t \in [0, T_i-1], i \in [1, r]$[2)]. On the other hand, there may be other BCNs (equivalently, other $N \times NM$ logical matrices) compatible with the data $(\mathbf{x}_d, \mathbf{u}_d)$ and we define the set of all such BCNs as

$$\mathcal{B}_d \triangleq \{\tilde{L} \in \mathcal{L}_{N\times NM} : \mathbf{x}_d^i(t+1) = \tilde{L} \ltimes \mathbf{u}_d^i(t) \ltimes \mathbf{x}_d^i(t), \forall t \in [0, T_i-1], i \in [1, r]\}. \tag{11}$$

Necessary and sufficient conditions for the data to be informative for the identifiability of the original BCN, namely for the set $\mathcal{B}_d$ to coincide with L, have been derived in [15, 17]. An alternative characterization of informativity for identifiability is proposed in the following. Given the data $(\mathbf{x}_d, \mathbf{u}_d)$, we define the set of transitions compatible with the data as

$$\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}} \triangleq \left\{ (\delta_N^{j_1}, \delta_M^i, \delta_N^{j_2}) : \begin{bmatrix} X_p \\ U_p \\ X_f \end{bmatrix} \delta_{T_d}^k = \begin{bmatrix} \delta_N^{j_1} \\ \delta_M^i \\ \delta_N^{j_2} \end{bmatrix}, \exists k \in [1, T_d] \right\}. \tag{12}$$

Also, we define the set of missing transitions as

$$\overline{\mathbf{D}^{\mathbf{x},\mathbf{u}}} \triangleq \left\{ (\delta_N^j, \delta_M^i) : \begin{bmatrix} X_p \\ U_p \end{bmatrix} \delta_{T_d}^k = \begin{bmatrix} \delta_N^j \\ \delta_M^i \end{bmatrix}, \nexists k \in [1, T_d] \right\}. \tag{13}$$

Intuitively, $(\mathbf{x}_d, \mathbf{u}_d)$ are informative for the identifiability of the original BCN if and only if $\overline{\mathbf{D}^{\mathbf{x},\mathbf{u}}} = \emptyset$, which ensures that the collected data provide information about the successor state for every possible input applied to each state, allowing the dynamics of the BCN to be fully inferred. If this is not the case, the set $\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$ enables partial identification of the BCNs in $\mathcal{B}_d$. Specifically, $(\delta_N^{j_1}, \delta_M^i, \delta_N^{j_2}) \in \mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$ if and only if $\tilde{L}_i \delta_N^{j_1} = \delta_N^{j_2}$ for every $\tilde{L} \in \mathcal{B}_d$. In other words, the set $\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$ identifies (all and only) the columns that are shared by every matrix $\tilde{L} \in \mathcal{B}_d$, and hence in particular a subset of the columns of L.

2) We assume, as in [17], that the collected data satisfy the property that any recorded transition starting from the same state under the same input always leads to the same successor state. More formally, since the data are generated by the "true" BCN, for every $i, j \in [1, r]$, and for all $0 \leqslant t \leqslant T_i - 1$ and $0 \leqslant k \leqslant T_j - 1$, if $(\mathbf{x}_d^i(t), \mathbf{u}_d^i(t)) = (\mathbf{x}_d^j(k), \mathbf{u}_d^j(k))$, then $\mathbf{x}_d^i(t+1) = \mathbf{x}_d^j(k+1)$.

In the following, we will assume that the data $(\mathbf{x}_d,\ \mathbf{u}_d)$ are not informative for identifiability. So, in order to solve the finite-horizon optimal control problem (7) and (8) based on $(\mathbf{x}_d, \mathbf{u}_d)$, we need to identify the minimum cost solution among those that are compatible with the data. This leads to the following problem statement.

Problem 1 (Finite-horizon data-optimal control problem)**.** Given the data $(\mathbf{x}_d, \mathbf{u}_d)$, solve the following optimization problem:

$$\begin{aligned} &\hat{J}_T^*(\mathbf{x}_0) \triangleq \min\nolimits_{\mathbf{u}(0),\ldots,\mathbf{u}(T-1)} J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) \\ &\text{subject to } (\mathbf{x}(t), \mathbf{u}(t), \mathbf{x}(t+1)) \in \mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}, \forall t \in [0, T-1], \\ &\qquad\qquad\quad \mathbf{x}(0) = \mathbf{x}_0, \end{aligned} \tag{14}$$

where $J_T(\mathbf{x}_0, \mathbf{u}(\cdot))$ is given in (8) and $\mathbf{x}_0 \in \mathcal{L}_N$ is arbitrary, and determine the optimal control input as a time-varying state-feedback, i.e., as $\mathbf{u}(t) = \hat{K}(t)\mathbf{x}(t), \exists \hat{K}(t) \in \mathcal{L}_{M\times N}, t \in [0, T-1]$.

It is important to note that the optimal solution to Problem 1, derived from the available data, constitutes only a sub-optimal solution to the original finite-horizon optimal control problem (7) and (8). Specifically, for every $\mathbf{x}_0 \in \mathcal{L}_N$, it holds that $\hat{J}_T^*(\mathbf{x}_0) \geqslant J_T^*(\mathbf{x}_0)$. For this reason, we refer to it as data-optimal.

Definition 1. Given a cost function to be minimized and a sequence of data $(\mathbf{x}_d,\ \mathbf{u}_d)$ generated by a BCN of the form (4), a solution to the minimization problem is said to be data-optimal if it is the minimum value that the cost function can attain subject to the available data information, or, equivalently, considering only the transitions in $\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$.

To solve Problem 1, we introduce the following assumption.

Assumption 1. The data matrix X_p has full row rank[3)], i.e., N.

Proposition 1. Problem 1 is solvable if and only if Assumption 1 holds.

Proof. *Only if.* In order to solve Problem 1 for every $\mathbf{x}_0 \in \mathcal{L}_N$, we need to ensure that for every state in $\mathcal{L}_N$ we know at least one of its successors (corresponding to some input value). This is equivalent to requiring that for every $j_1 \in [1, N]$ there exist $i \in [1, M]$ and $j_2 \in [1, N]$ such that $(\delta_N^{j_1}, \delta_M^i, \delta_N^{j_2}) \in \mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$. Since the previous condition must hold for every $j_1 \in [1, N]$, this implies that X_p includes each vector $\delta_N^{j_1}, j_1 \in [1, N]$, among its columns and hence has rank N.

If. If Assumption 1 holds, it means that for every T and every $\mathbf{x}_0 \in \mathcal{L}_N$ there exists at least one trajectory $\{\mathbf{x}(t)\}_{t=0}^T$ starting from $\mathbf{x}_0$ (and obtained corresponding to some input sequence $\{\mathbf{u}(t)\}_{t=0}^{T-1}$), and hence the (finite) set on which the cost function is minimized is non-empty.

We aim to revise the solution proposed in [27] to account for the fact that only partial knowledge of the BCN dynamics is available, so that the cost function can be optimized using only transitions that are admissible according to the observed data.

Consider an arbitrary family of N-dimensional real vectors $\hat{\mathbf{m}}(t), t \in [0, T]$, and a state trajectory $\mathbf{x}(t), t \in [0, T]$, of the BCN (4). Clearly,

$$0 = \sum_{t=0}^{T-1} [\hat{\mathbf{m}}(t+1)^\top \mathbf{x}(t+1) - \hat{\mathbf{m}}(t)^\top \mathbf{x}(t)] + \hat{\mathbf{m}}(0)^\top \mathbf{x}(0) - \hat{\mathbf{m}}(T)^\top \mathbf{x}(T).$$

Therefore, the cost function (8) can be rewritten as

$$\begin{aligned} J_T(\mathbf{x}_0, \mathbf{u}(\cdot)) &= \hat{\mathbf{m}}(0)^\top \mathbf{x}_0 + [\mathbf{c}_f - \hat{\mathbf{m}}(T)]^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \mathbf{c}(t)^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) \\ &\quad + \sum_{t=0}^{T-1} [\hat{\mathbf{m}}(t+1)^\top \mathbf{x}(t+1) - \hat{\mathbf{m}}(t)^\top \mathbf{x}(t)] \\ &= \hat{\mathbf{m}}(0)^\top \mathbf{x}_0 + [\mathbf{c}_f - \hat{\mathbf{m}}(T)]^\top \mathbf{x}(T) + \sum_{t=0}^{T-1} \begin{bmatrix} \hat{\mathbf{w}}_1(t)^\top & \hat{\mathbf{w}}_2(t)^\top & \ldots & \hat{\mathbf{w}}_M(t)^\top \end{bmatrix} \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t), \end{aligned}$$

where (see [27] for details)

$$\hat{\mathbf{w}}_i(t)^\top \triangleq \mathbf{c}_i(t)^\top + \hat{\mathbf{m}}(t+1)^\top L_i - \hat{\mathbf{m}}(t)^\top,\ i \in [1, M]. \tag{15}$$

It is immediate to observe that the first term in the cost function $J_T(\mathbf{x}_0, \mathbf{u}(\cdot))$ is unavoidable, whereas the remaining two terms could in principle be eliminated by properly selecting the vectors of the sequence $\{\hat{\mathbf{m}}(t)\}_{t=0}^T$ and the input sequence $\{\mathbf{u}(t)\}_{t=0}^{T-1}$, thereby yielding the minimum value of the cost function.

3) Assumption 1 imposes $T_d \geqslant N$.

In Algorithm 2, below, we outline the procedure for computing the solution to the finite-horizon data-optimal control problem defined in (14). The solution is expressed as a pair: the first component is the vector of data-optimal costs corresponding to each initial state, while the second component is the sequence of data-optimal feedback matrices.

Algorithm 2 Finite-horizon data-optimal solution.

Input: The time horizon T;the matrices X_p, U_p and X_f; the vector $\mathbf{c}_f \in \mathbb{R}_+^N$; the sequence of vectors $\{\mathbf{c}(t)\}_{t=0}^{T-1}, \mathbf{c}(t) \in \mathbb{R}_+^{NM}$.
Output: The pair $\left(\hat{\mathbf{m}}(0), \{\hat{K}(t)\}_{t=0}^{T-1}\right)$, i.e., the data-optimal solution.
(1) *Initialization:* Set $t = T$, $\hat{\mathbf{m}}(t) = \mathbf{c}_f$, and $\mathcal{I}_j = \emptyset, \forall j \in [1, N]$.
(2) `for` $k \in [1, T_d]$, `do`
`if` $\begin{bmatrix} X_p \\ U_p \end{bmatrix} \delta_{T_d}^k = \begin{bmatrix} \delta_N^j \\ \delta_M^i \end{bmatrix}$, `then`
$\mathcal{I}_j \leftarrow \mathcal{I}_j \cup \{i\}$;
$L_i \delta_N^j = X_f \delta_{T_d}^k$;
`end if`
`end for`
(3) *Iterative procedure:*
$t \leftarrow t - 1$;
`for` $j \in [1, N]$, `do`

$$[\hat{\mathbf{m}}(t)]_j \triangleq \min_{i \in \mathcal{I}_j} \left\{ [\mathbf{c}_i(t)]_j + [\hat{\mathbf{m}}(t+1)^\top L_i]_j \right\} = \min_{i \in \mathcal{I}_j} \left\{ [\mathbf{c}_i(t)]_j + \hat{\mathbf{m}}(t+1)^\top L_i \delta_N^j \right\}; \tag{16}$$

$$\hat{i}^*(j,t) \triangleq \arg\min_{i \in \mathcal{I}_j} \left\{ [\mathbf{c}_i(t)]_j + [\hat{\mathbf{m}}(t+1)^\top L_i]_j \right\} = \arg\min_{i \in \mathcal{I}_j} \left\{ [\mathbf{c}_i(t)]_j + \hat{\mathbf{m}}(t+1)^\top L_i \delta_N^j \right\}; \tag{17}$$

`end for`
Set

$$\hat{K}(t) = \begin{bmatrix} \delta_M^{\hat{i}^*(1,t)} & \delta_M^{\hat{i}^*(2,t)} & \dots & \delta_M^{\hat{i}^*(N,t)} \end{bmatrix}; \tag{18}$$

`if` $t \leqslant 0$, `then`
Go to step (4);
`end if`
(4) *Conclusion:* Return $\left(\hat{\mathbf{m}}(0), \{\hat{K}(t)\}_{t=0}^{T-1}\right)$.

In step (1) of Algorithm 2, we initialize the sequence of vectors $\{\hat{\mathbf{m}}(t)\}_{t=0}^T$ in the same way as in the model-based setting. However, the procedure used to generate the vectors must be adapted to incorporate only the transitions that are available from the data. To this end, in step (2) we introduce the sets $\mathcal{I}_j$, $j \in [1, N]$, that collect the indices of the inputs for which the successor of state δ_N^j is known from the data. Moreover, when $i \in \mathcal{I}_j$, the data allow us to identify the j-th column of L_i, i.e., $L_i \delta_N^j$. Therefore, in step (3), by restricting the search to the set $\mathcal{I}_j$, we guarantee that the minimization in (16) is well-posed, even if the full matrix L_i is not known. By relying solely on the state/input/state transitions in $\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$ and hence selecting $\hat{\mathbf{m}}(t)$, $t \in [0, T-1]$, according to (16), we cannot guarantee that the vectors $\hat{\mathbf{w}}_i(t)$ in (15) are nonnegative. However, we can guarantee that for each $j \in [1, N]$ there exists an index $\hat{i}^*(j,t) \in \mathcal{I}_j$ (see (17) in Algorithm 2) such that $[\hat{\mathbf{w}}_{\hat{i}^*(j,t)}(t)]_j = 0$ and for all the other indices $i \in \mathcal{I}_j$ one has $[\hat{\mathbf{w}}_i(t)]_j \geqslant 0$. By choosing $\mathbf{u}(t) = \delta_M^{\hat{i}^*(j,t)}$ when $\mathbf{x}(t) = \delta_N^j$ we can impose that $0 = \begin{bmatrix} \hat{\mathbf{w}}_1(t)^\top & \hat{\mathbf{w}}_2(t)^\top & \dots & \hat{\mathbf{w}}_M(t)^\top \end{bmatrix} \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t)$ and hence minimize the cost function. As a consequence, the data-optimal cost in (14) is given by

$$\hat{J}_T^*(\mathbf{x}_0) = \hat{\mathbf{m}}(0)^\top \mathbf{x}_0,$$

where $\hat{\mathbf{m}}(0)$ is obtained as described in Algorithm 2. The data-optimal control input is given by the time-varying state-feedback law $\mathbf{u}(t) = \hat{K}(t)\mathbf{x}(t), t \in [0, T-1]$, with the sequence of feedback matrices $\{\hat{K}(t)\}_{t=0}^{T-1}$ computed in step (3) of Algorithm 2 (see (18)).

Remark 1. The data-optimal solution $\hat{J}_T^*(\mathbf{x}_0)$ to the finite-horizon problem provides an upper bound on the true optimal cost $J_T^*(\mathbf{x}_0)$, namely $\hat{J}_T^*(\mathbf{x}_0) \geqslant J_T^*(\mathbf{x}_0)$. To better understand the uncertainty associated with the true optimal value of the cost function for the original BCN, one may seek a lower bound on $J_T^*(\mathbf{x}_0)$. This requires identifying, among all BCNs compatible with the data (i.e., in $\mathcal{B}_d$), the one whose optimal cost is minimal. This can be achieved by iteratively selecting, at each time step $t = T-1, T-2, \dots, 0$, a successor for each state/input pair in $\overline{\mathbf{D}^{\mathbf{x},\mathbf{u}}}$ that minimizes the cost-to-go, namely the cost from t onward. However, such choices may need to be revised during the backward recursion, requiring all tentative assignments and paths to be stored and potentially updated. Unless one adopts a greedy strategy, leading to an over-estimation of the minimum, this process demands tracking all assignment possibilities. Another option would be to consider all logical matrices $\tilde{L}$ compatible with the data, amounting to $N^{|\overline{\mathbf{D}^{\mathbf{x},\mathbf{u}}}|}$ possibilities. This exhaustive analysis is computationally prohibitive. Nevertheless, we can derive a (generally non-tight) lower bound on $J_T^*(\mathbf{x}_0)$ as follows:

- At $t=0$, select the minimum transition cost from the initial state $\mathbf{x}_0 = \delta_N^{j_0}$ among all applicable inputs, i.e., $\min_{i\in[1,M]}[\mathbf{c}_i(0)]_{j_0}$;
- At each subsequent stage $t=1,2,\ldots,T-1$, select the minimum transition cost over all state/input pairs, i.e., $\min_{i\in[1,M],j\in[1,N]}[\mathbf{c}_i(t)]_j$;
- At the final stage $t=T$, choose the minimum terminal cost among all states, i.e., $\min_{j\in[1,N]}[\mathbf{c}_f]_j$.

Clearly, the quantity

$$\underline{J}_T^*(\mathbf{x}_0) \triangleq \min_{i\in[1,M]}[\mathbf{c}_i(0)]_{j_0} + \sum_{t=1}^{T-1} \min_{i\in[1,M],j\in[1,N]}[\mathbf{c}_i(t)]_j + \min_{j\in[1,N]}[\mathbf{c}_f]_j$$

is a valid lower bound on $J_T^*(\mathbf{x}_0)$.

Remark 2. To the best of our knowledge, all solutions to the finite-horizon optimal control problem for BCNs currently available in the literature rely on model-based methods. A first data-driven solution was presented in the conference version of this paper [31], where the case of a constant cost vector was investigated. In this paper, instead, we address the most general form of finite-horizon optimal control problem, assuming a time-varying cost vector with no structural constraints. This formulation also encompasses the discounted-cost case, previously studied in a model-based setting in [23], where the authors solve the problem using a Floyd-like algorithm based on the iterative computation of two matrices, $C \in (\mathcal{L}_M)^{N\times N}$ and $D \in \mathbb{R}^{N\times N}$. These matrices store the optimal input value and the associated transition cost for every existing transition in the BCN, respectively. The strategy consists of updating these two matrices by optimizing over intermediate paths while incorporating the discount factor λ. To address the more general formulation considered in this paper, the algorithm from [23] would need to be adapted to a non-discounted yet time-varying cost setting. However, the approach proposed in [27], that we have followed in this paper, is more suitable for our purposes, as it can be more readily extended to a generic time-varying cost structure and to data-driven implementations. Moreover, it is computationally more efficient, since it only requires computing and updating a vector of dimension N, rather than two matrices of size $N \times N$.

We test our data-optimal control strategy using a biological system known as λ switch [32]. The following example is taken from [23, Example 2] (see also [33, Section 4]), where it was used as a benchmark to assess their model-based approach to finite-horizon optimal control.

Example 1 (λ switch: finite-horizon scenario)**.** Consider the mechanism of infection of a bacterium by a λ phage, a virus that grows on the bacterium. Once inside, the viral genome can follow one of two alternative developmental pathways, depending on environmental conditions: lysogeny or lysis, which are the two equilibrium points [10] of the system. If environmental conditions are favorable, the phage enters the lysogenic pathway, in which its DNA integrates into the bacterial genome and is replicated passively as the bacterium divides. Under unfavorable conditions, the phage switches to the lytic pathway, in which it replicates actively inside the host and ultimately causes the cell to burst, releasing many new viral particles. This switching mechanism primarily depends on five phage genes, cI, cro, cII, $cIII$, N, as well as on the environmental conditions, and its dynamics can be described as follows:

$$\begin{cases} N(t+1) = \neg cI(t) \wedge \neg cro(t), \\ cI(t+1) = \neg cro(t) \wedge (cI(t) \vee cII(t)), \\ cII(t+1) = \neg cI(t) \wedge N(t) \wedge cIII(t) \wedge U(t), \\ cIII(t+1) = \neg cI(t) \wedge N(t) \wedge U(t), \\ cro(t+1) = \neg cI(t) \wedge \neg cII(t), \end{cases} \tag{19}$$

where $U(t) \in \mathcal{B}$ is a binary input that specifies whether the environmental conditions are favorable (for the genes cII and $cIII$) or not. Figure 1 presents a schematic of the gene interaction network, in which edges marked with "$\rightarrow$" represent activating interactions and edges marked with "$\dashv$" represent inhibitory interactions.

If let $\mathbf{u}(t) \in \mathcal{L}_2$ denote the logic vector representing $U(t) \in \mathcal{B}$, we define $\mathbf{x}(t) \triangleq N(t) \ltimes cI(t) \ltimes cII(t) \ltimes cIII(t) \ltimes cro(t)$ and use the algebraic representation of a BCN. The dynamics of the system in (19) can be expressed as in (4), with $N = 2^5 = 32$, $M = 2^1 = 2$, and $L = \left[\, L_1 \middle| L_2 \,\right]$, where

$$\begin{aligned} L_1 = \Big[\, & \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{26}\ \delta_{32}^{2}\ \delta_{32}^{26}\ \delta_{32}^{2}\ \delta_{32}^{25}\ \delta_{32}^{9}\ \delta_{32}^{25}\ \delta_{32}^{9} \\ & \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{28}\ \delta_{32}^{4}\ \delta_{32}^{32}\ \delta_{32}^{8}\ \delta_{32}^{27}\ \delta_{32}^{11}\ \delta_{32}^{31}\ \delta_{32}^{15} \,\Big], \end{aligned} \tag{20a}$$

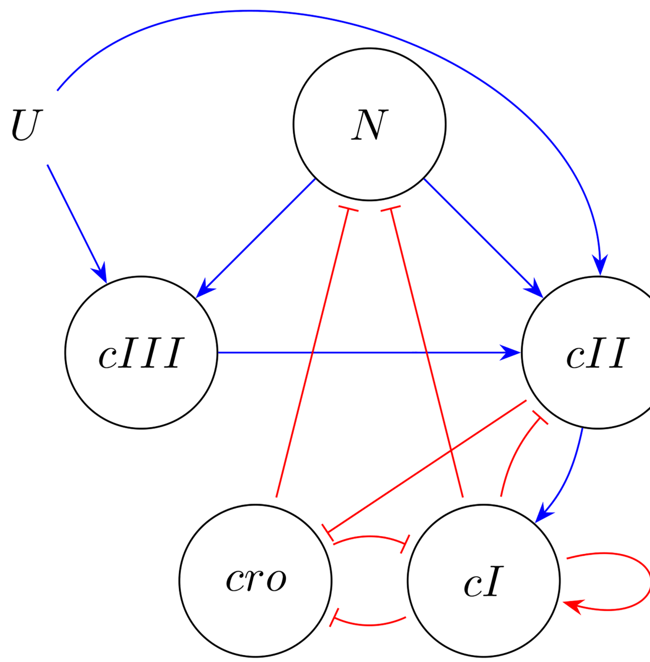

Figure 1 (Color online) Gene interactions for the λ switch.

$$L_2 = \Big[\delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{8}\ \delta_{32}^{32}\ \delta_{32}^{8}\ \delta_{32}^{31}\ \delta_{32}^{15}\ \delta_{32}^{31}\ \delta_{32}^{15}$$
$$\delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{24}\ \delta_{32}^{32}\ \delta_{32}^{8}\ \delta_{32}^{32}\ \delta_{32}^{8}\ \delta_{32}^{31}\ \delta_{32}^{15}\ \delta_{32}^{31}\ \delta_{32}^{15} \Big]. \tag{20b}$$

We consider the optimal control problem (7) with cost function (8), where
- $\mathbf{c}_f = \mathbb{0}$;
- for every $t \in \mathbb{Z}_+$, $\mathbf{c}(t) = \lambda^t \mathbf{c}$, where $\lambda = 0.95$ is the discount factor [23, 33], and $\mathbf{c}^\top = [\, \mathbf{c}_1^\top \,|\, \mathbf{c}_2^\top \,]$, with

$$\mathbf{c}_1^\top = [\,1\ 7\ 4\ 0\ 9\ 4\ 8\ 8\ 2\ 4\ 5\ 5\ 1\ 7\ 1\ 1\ 5\ 2\ 7\ 6\ 1\ 4\ 2\ 3\ 2\ 2\ 1\ 6\ 8\ 5\ 7\ 6\,], \tag{21a}$$
$$\mathbf{c}_2^\top = [\,1\ 8\ 9\ 2\ 7\ 9\ 5\ 4\ 3\ 1\ 2\ 3\ 3\ 4\ 1\ 1\ 3\ 8\ 7\ 4\ 2\ 7\ 7\ 9\ 3\ 1\ 9\ 8\ 6\ 5\ 0\ 2\,]. \tag{21b}$$

We assume as in [23, Example 2] that the length of the time horizon is $T = 6$. The optimal solution to this problem has been computed in [23] using a model-based approach for all possible initial conditions.

We performed $r = 21$ offline experiments on different time intervals $[0, T_i], i \in [1, r]$, and collected the corresponding data sequences (see Table A1 in Appendix A).

Remark 3. The choice of the discount factor $\lambda = 0.95$ and of the time horizon $T = 6$ was primarily meant to enable a direct comparison with the model-based results derived in [23]. The discount factor λ determines the relative weighting of future costs or rewards compared to the present: a lower λ emphasizes short-term dynamics, such as quickly switching the lambda network, while a higher λ emphasizes long-term outcomes, such as reaching the lysogenic or lytic state. Regarding the time horizon T, it should be chosen to capture the relevant biological dynamics—specifically, the number of discrete steps required for the λ switch to settle into either the lysogenic or lytic state. Accordingly, T must be long enough to cover all meaningful state transitions, yet not so long as to introduce unnecessary computational complexity. Examining the system evolution from various initial states, we can observe that the final states after $T = 6$ steps are consistently either the lysogenic state δ_{32}^{24} or the lytic state δ_{32}^{31} (see [33, Section 4]).

The number of offline experiments, instead, is closely related to the data-driven framework, and in particular to the need for collecting informative data. Indeed, given the inherent structure of the network (which is assumed to be unknown), it can be shown that the minimum number of experiments required is $r = 19$, as it represents the number of states in $\mathcal{L}_{32}$ that are not successors of any other state and thus must necessarily be chosen as initial states for at least one of the experiments. However, in a data-driven setting, where the network model is not available, we can only perform a number of experiments that appear reasonable based on approximate knowledge of the system. Clearly, the more experiments we perform, the greater the likelihood of obtaining informative data, if possible. For practical reasons and due to space constraints, we chose a number slightly above this theoretical minimum (i.e., $r = 21$). In a realistic scenario, the number of experiments needed to obtain informative data could be higher, since we cannot know in advance which initial states and control sequences will be the most effective in achieving the desired objective.

The set $\mathcal{B}_d$ of BCNs compatible with the collected data is described as the set of matrices $\tilde{L} = \left[\, \tilde{L}_1 \middle| \tilde{L}_2 \,\right]$, with

$$
\begin{aligned}
\tilde{L}_1 = & \Big[* \; * \; \delta_{32}^{32} \; * \; * \; * \; * \; \delta_{32}^{24} \; \delta_{32}^{26} \; \delta_{32}^{2} \; \delta_{32}^{26} \; \delta_{32}^{2} \; * \; \delta_{32}^{9} \; \delta_{32}^{25} \; \delta_{32}^{9} \\
& \quad \delta_{32}^{32} \; * \; \delta_{32}^{32} \; * \; \delta_{32}^{32} \; \delta_{32}^{24} \; \delta_{32}^{32} \; \delta_{32}^{24} \; \delta_{32}^{28} \; \delta_{32}^{4} \; \delta_{32}^{32} \; * \; \delta_{32}^{27} \; \delta_{32}^{11} \; * \; \delta_{32}^{15} \Big], \\
\tilde{L}_2 = & \Big[\delta_{32}^{32} \; \delta_{32}^{24} \; * \; \delta_{32}^{24} \; \delta_{32}^{32} \; \delta_{32}^{24} \; \delta_{32}^{32} \; \delta_{32}^{24} \; * \; * \; * \; * \; \delta_{32}^{31} \; * \; \delta_{32}^{31} \; * \\
& \quad * \; \delta_{32}^{24} \; * \; \delta_{32}^{24} \; * \; * \; * \; \delta_{32}^{24} \; * \; \delta_{32}^{8} \; * \; \delta_{32}^{8} \; * \; * \; \delta_{32}^{31} \; \delta_{32}^{15} \Big],
\end{aligned}
$$

where "$*$" stands for an arbitrary vector in $\mathcal{L}_{32}$. We then apply Algorithm 2 to compute the data-optimal solution for each possible initial condition, namely the solution to Problem 1. The results are in Table A2 of Appendix A.

By comparing the data-driven results with the model-based ones, we observe that, despite the presence of numerous missing transitions not captured in the data, the data-optimal solutions often coincide with the true optimal solutions. Moreover, in the remaining cases, the data-optimal cost remains close to the model-based optimal value.

4 Infinite-horizon data-optimal control problem

The infinite-horizon optimal control problem considered in [27] for a fully known BCN (4) is formulated as

$$
\begin{array}{l}
\min_{\mathbf{u}(\cdot)} J(\mathbf{x}_0, \mathbf{u}(\cdot)) \\
\text{subject to (4) and } \mathbf{x}(0) = \mathbf{x}_0,
\end{array}
\tag{22}
$$

where $\mathbf{x}_0 \in \mathcal{L}_N$ is an (arbitrarily chosen) initial condition and

$$
J(\mathbf{x}_0, \mathbf{u}(\cdot)) \triangleq \sum_{t=0}^{+\infty} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) \tag{23}
$$

with $\mathbf{c} \in \mathbb{R}_+^{NM}$ a constant vector with nonnegative entries.

It has been shown that the infinite-horizon optimal control problem (22) and (23) is well-posed, meaning that the minimum value $J^*(\mathbf{x}_0)$ of the cost function (23) is finite for every initial state $\mathbf{x}_0 \in \mathcal{L}_N$, under rather strong assumptions on the BCN dynamics and on the cost assigned to its state/input trajectories.

Proposition 2 ([27])**.** The minimum value $J^*(\mathbf{x}_0)$ of the infinite-horizon cost function (23) is finite for every initial state $\mathbf{x}_0 \in \mathcal{L}_N$ if and only if for every $\mathbf{x}_0$ there exists an input sequence $\mathbf{u}(t), t \in \mathbb{Z}_+$, with $\mathbf{u}(t) \in \mathcal{L}_M$, that makes the resulting state/input trajectory $\{(\mathbf{x}(t), \mathbf{u}(t))\}_{t \in \mathbb{Z}_+}$ both periodic and zero-cost starting from some time instant $\tau \geqslant 0$.

For well-posed infinite-horizon optimal control problems, the solution takes the form of a static, time-invariant state-feedback.

Theorem 1 ([27])**.** Assume that for every initial condition $\mathbf{x}_0 \in \mathcal{L}_N$ there exists an input sequence that makes the resulting state/input trajectory both periodic and zero-cost starting from some time instant.

(1) There exists $\bar{T} \geqslant 0$ and $\mathbf{m}^* \in \mathbb{R}_+^N$ such that, for every $\mathbf{x}_0$,

$$
J_T^*(\mathbf{x}_0) = J_{\bar{T}}^*(\mathbf{x}_0) = (\mathbf{m}^*)^\top \mathbf{x}_0, \qquad \forall\, T \geqslant \bar{T},
$$

and therefore

$$
J^*(\mathbf{x}_0) = \min_{\mathbf{u}(\cdot)} \sum_{t=0}^{+\infty} \mathbf{c}^\top \ltimes \mathbf{u}(t) \ltimes \mathbf{x}(t) = (\mathbf{m}^*)^\top \mathbf{x}_0.
$$

(2) $\mathbf{m}^* = \mathbf{m}(0)$ when Algorithm 1 in Section 3 is run[4] for $T \geqslant \bar{T}$, by assuming $\mathbf{m}(T) = \mathbf{c}_f = \mathbb{0}$ and $\mathbf{c}_i(t) = \mathbf{c}_i$ for every $t \in [0, T-1]$ and $i \in [1, M]$. Moreover, $\mathbf{m}^*$ is a fixed point of the algorithm, namely a solution of the family

4) From a practical perspective, in order to solve the infinite-horizon optimal control problem, it is convenient to apply Algorithm 1 backward, for $t \in \mathbb{Z}, t \leqslant 0$, starting from the final condition $\mathbf{m}(0) = \mathbb{0}$, and to stop the procedure when the same vector is obtained at two consecutive time steps, namely when $\mathbf{m}(t) = \mathbf{m}(t+1)$ for some $t < 0$. In this way we do not need to know a priori the value of $\bar{T}$.

of equations:

$$[\mathbf{m}^*]_j = \min_{i\in[1,M]} \left\{[\mathbf{c}_i]_j + [(\mathbf{m}^*)^\top L_i]_j\right\}, \ \forall j \in [1,N]. \tag{24}$$

(3) Upon defining

$$i^*(j) \triangleq \arg\min_{i\in[1,M]} \Big\{[\mathbf{c}_i]_j + [(\mathbf{m}^*)^\top L_i]_j\Big\},$$

the optimal control input can be expressed as a static state-feedback law $\mathbf{u}(t) = K\mathbf{x}(t)$, with

$$K = \left[\delta_M^{i^*(1)} \ \delta_M^{i^*(2)} \ \dots \ \delta_M^{i^*(N)}\right].$$

Remark 4. The j-th entry of the vector $\mathbf{m}^*$, i.e., $[\mathbf{m}^*]_j$, quantifies the minimal cost required for a trajectory starting at state δ_N^j to reach a periodic state/input sequence with zero cost. In other words, it represents the lowest possible cumulative cost over all trajectories $\{(\mathbf{x}(t),\mathbf{u}(t))\}_{t\in\mathbb{Z}_+}$ stemming from $\mathbf{x}(0) = \delta_N^j$.

We aim to address the infinite-horizon optimal control problem (22) and (23) for an unknown BCN (4) using only state/input sequences $(\mathbf{x}_d, \mathbf{u}_d)$ collected from the offline experiments, or equivalently, the corresponding data matrices (X_p, U_p, X_f) introduced in Section 3.

Problem 2 (Infinite-horizon data-optimal control problem)**.** Given the data $(\mathbf{x}_d, \mathbf{u}_d)$, solve the following optimization problem:

$$\begin{aligned} &\hat{J}^*(\mathbf{x}_0) \triangleq \min\nolimits_{\mathbf{u}(\cdot)} J(\mathbf{x}_0, \mathbf{u}(\cdot)) \\ &\text{subject to } (\mathbf{x}(t), \mathbf{u}(t), \mathbf{x}(t+1)) \in \mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}, \forall t \in \mathbb{Z}_+, \\ &\qquad\qquad \mathbf{x}(0) = \mathbf{x}_0, \end{aligned} \tag{25}$$

where $J(\mathbf{x}_0, \mathbf{u}(\cdot))$ and $\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$ are given in (23) and (12), respectively, while $\mathbf{x}_0 \in \mathcal{L}_N$ is any assigned initial condition. Also, determine the optimal control input as a time-invariant state-feedback, i.e., $\mathbf{u}(t) = \hat{K}\mathbf{x}(t), t \in \mathbb{Z}_+, \exists \hat{K} \in \mathcal{L}_{M\times N}$.

As for Problem 1, the optimal solution to Problem 2 (if it exists) is data-optimal for the original infinite-horizon optimal control problem in (22) and (23).

Assumption 1, despite being still necessary (for the same reasons provided in the proof of Proposition 1) is no longer sufficient to guarantee the existence of a solution to Problem 2. Consequently, the first step is to determine whether the data are informative for the solvability of Problem 2, meaning that they allow us to conclude that problem (25) is solvable for every BCN in $\mathcal{B}_d$. This involves checking, based on (X_p, U_p, X_f), if there exists (at least) one periodic state/input trajectory with zero cost common to all BCNs in $\mathcal{B}_d$, and if every state can reach one of these zero-cost periodic trajectories using only transitions compatible with the data (and hence belonging to $\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$).

Algorithm 3 returns the set $\mathbf{C}_d$, with cardinality $q \geqslant 0$, of the indices of the zero-cost periodic state/input trajectories common to all BCNs in $\mathcal{B}_d$, by constructing a directed graph $\mathcal{G}_d^{\mathbf{x},\mathbf{u}} = (\mathcal{V}^{\mathbf{x},\mathbf{u}}, \mathcal{E}^{\mathbf{x},\mathbf{u}})$, where $\mathcal{V}^{\mathbf{x},\mathbf{u}}$ is the set

Algorithm 3 Data-based zero-cost periodic state/input trajectories.

Input: The matrices X_p, U_p and X_f; the vector $\mathbf{c} \in \mathbb{R}_+^N$.
Output: The set $\mathbf{C}_d$ of the index pairs of the zero-cost periodic state/input trajectories compatible with all BCNs in $\mathcal{B}_d$.
(1) *Initialization:* $\mathcal{G}_d^{\mathbf{x},\mathbf{u}} \triangleq (\mathcal{V}^{\mathbf{x},\mathbf{u}}, \mathcal{E}^{\mathbf{x},\mathbf{u}})$, with $\mathcal{V}^{\mathbf{x},\mathbf{u}} = \emptyset$ and $\mathcal{E}^{\mathbf{x},\mathbf{u}} = \emptyset$; $\mathbf{C}_d = \emptyset$;
(2) `for` $k \in [1, T_d]$, `do`

`if` $\begin{bmatrix} X_p \\ U_p \end{bmatrix} \delta_{T_d}^k = \begin{bmatrix} \delta_N^j \\ \delta_M^i \end{bmatrix}$ `and` $\mathbf{c}^\top \ltimes \delta_M^i \ltimes \delta_N^j = [\mathbf{c}_i]_j = 0$, `then`
$\mathcal{V}^{\mathbf{x},\mathbf{u}} \leftarrow \mathcal{V}^{\mathbf{x},\mathbf{u}} \cup (j,i)$;
`end if`
`end for`
(3) `for` $k \in [1, T_d]$, `do`

`if` $\begin{bmatrix} X_p \\ U_p \\ X_f \end{bmatrix} \delta_{T_d}^k = \begin{bmatrix} \delta_N^j \\ \delta_M^i \\ \delta_N^r \end{bmatrix}$, $(j,i) \in \mathcal{V}^{\mathbf{x},\mathbf{u}}$, `and` $\exists \ell \in [1,M]$ s.t. $(r,\ell) \in \mathcal{V}^{\mathbf{x},\mathbf{u}}$, `then`
$\mathcal{E}^{\mathbf{x},\mathbf{u}} \leftarrow \mathcal{E}^{\mathbf{x},\mathbf{u}} \cup ((j,i),(r,\ell))$;
`end if`
`end for`
(4) Apply Johnson's Algorithm [34] to $\mathcal{G}_d^{\mathbf{x},\mathbf{u}}$ to find all cycles, $\mathcal{C}_1, \dots, \mathcal{C}_q$, with $q \in \mathbb{Z}_+$, and $\mathcal{C}_\ell = ((j_1^{(\ell)}, i_1^{(\ell)}), \dots, (j_{p_\ell}^{(\ell)}, i_{p_\ell}^{(\ell)})), p_\ell \in \mathbb{Z}_+, \forall \ell \in [1,q]$;
(5) *Conclusion:* Return $\mathbf{C}_d = \{\mathcal{C}_1, \dots, \mathcal{C}_q\}$.

of index pairs (j,i) corresponding to the state/input pairs $(\mathbf{x},\mathbf{u}) = (\delta_N^j, \delta_M^i)$ that are compatible with the data and yield zero cost. Given (j,i) and (r,ℓ) in $\mathcal{V}^{\mathbf{x},\mathbf{u}}$, there is an edge from (j,i) to (r,ℓ) if and only if $(\delta_N^j, \delta_M^i, \delta_N^r) \in \mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$.

Clearly, if $\mathbf{C}_d = \emptyset$, Problem 2 is not solvable. On the other hand, if $\mathbf{C}_d \neq \emptyset$, let $\mathcal{V}^{\mathbf{C}_d}$ denote the set of indices of the state components of the pairs that belong to at least one of the q cycles in $\mathbf{C}_d$, by this meaning the set of all indices j such that $(j,i) \in \mathcal{C}_\ell, \exists \ell \in [1,q], \exists i \in [1,M]$. Then we need to verify that the set of states indexed in $\mathcal{V}^{\mathbf{C}_d}$ is globally reachable in all the BCNs in $\mathcal{B}_d$, by this meaning [10, 15] that for every $\mathbf{x}_0 = \delta_N^j \in \mathcal{L}_N$ there exist (a) δ_N^r, $r \in \mathcal{V}^{\mathbf{C}_d}$, (b) $\tau \in \mathbb{Z}_+$ and (c) an input $\mathbf{u}(t), t \in [0, \tau-1]$, that leads the state trajectory from $\mathbf{x}(0) = \mathbf{x}_0$ to $\mathbf{x}(\tau) = \delta_N^r$. To this end, we can resort to Algorithm 1 in [15].

In summary, the essential steps of the procedure are listed below.

(1) Construct the digraph $\mathcal{G}_d^{\mathbf{x},\mathbf{u}}$ as described in steps (2) and (3) of Algorithm 3.

(2) Detect all cycles in $\mathcal{G}_d^{\mathbf{x},\mathbf{u}}$ in step (4) using the well-known Johnson's algorithm [34].

(3) Verify the global reachability of the set

$$\mathcal{X}^{\mathbf{C}_d} \triangleq \{\delta_N^j, j \in \mathcal{V}^{\mathbf{C}_d}\} \tag{26}$$

by applying Algorithm 1 from [15].

The above procedure allows us to verify whether Proposition 2 holds for all BCNs in $\mathcal{B}_d$, and thereby whether the data are informative for the infinite-horizon data-optimal control problem. This is formalized in Proposition 3.

Proposition 3. The data $(\mathbf{x}_d, \mathbf{u}_d)$ are informative for the solvability of the infinite-horizon data-optimal control problem, i.e., of Problem 2, if and only if

(1) the set $\mathbf{C}_d$ is non-empty;

(2) the set $\mathcal{X}^{\mathbf{C}_d}$ is globally reachable.

If the conditions of Proposition 3 hold, i.e., if the data are informative for the solvability of Problem 2, we can compute a solution by once again adapting the model-based reasoning to account for the fact that only partial information on the state/input/state transitions of the original BCN is available. More specifically, we can apply a slightly amended version of Algorithm 2 in Section 3 to derive an estimate $\hat{\mathbf{m}}^*$ of the vector $\mathbf{m}^*$.

Similarly to the model-based scenario, the j-th component of the vector $\hat{\mathbf{m}}^*$, i.e., $[\hat{\mathbf{m}}^*]_j$, represents the minimum cost required to reach a zero-cost periodic state/input trajectory identified from the data (i.e., corresponding to some $\mathcal{C}_\ell \in \mathbf{C}_d$), starting from the state δ_N^j and using only the transitions that can be inferred from the data (i.e., those in $\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$). Equivalently, $[\hat{\mathbf{m}}^*]_j$ is the minimum cost attainable by any state/input trajectory $\{(\mathbf{x}(t), \mathbf{u}(t))\}_{t \in \mathbb{Z}_+}$ starting from $\mathbf{x}(0) = \delta_N^j$ and satisfying $(\mathbf{x}(t), \mathbf{u}(t), \mathbf{x}(t+1)) \in \mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$ for all $t \in \mathbb{Z}_+$. Constraining the transitions to those compatible with the data requires that, in step (2) of Algorithm 4, below, the minimization for each $j \in [1,N]$ is restricted to the indices $i \in \mathcal{I}_j$ (see Algorithm 2). Note that, consistent with footnote 4), we evaluate $\hat{\mathbf{m}}(t)$ for $t \leqslant 0$.

Algorithm 4 Computation of the vector $\hat{\mathbf{m}}^*$.

Input: The data matrices X_p, U_p and X_f; the vector $\mathbf{c} \in \mathbb{R}_+^{NM}$.
Output: The vector $\hat{\mathbf{m}}^* \in \mathbb{R}_+^N$.
(1) *Initialization:* Set $t = 0$ and $\hat{\mathbf{m}}(t) = 0$;
(2) *Iterative procedure:*
  $t \leftarrow t-1$;
  `for` $j \in [1,N]$, `do`
    $[\hat{\mathbf{m}}(t)]_j = \min_{i \in \mathcal{I}_j} \left\{ [\mathbf{c}_i]_j + [(\hat{\mathbf{m}}(t+1))^\top L_i]_j \right\}$;
  `end for`
  `if` $\hat{\mathbf{m}}(t) = \hat{\mathbf{m}}(t+1)$, `then`
    $\hat{\mathbf{m}}^* = \hat{\mathbf{m}}(t)$;
    Go to step (3);
  `end if`
(3) *Conclusion:* Return $\hat{\mathbf{m}}^*$.

Remark 5. (i) It is worth noting that the comment regarding the knowledge of the j-th column of L_i made in connection with (16) also applies to the minimization performed within the `for`-loop in step (2) of Algorithm 4.

(ii) Algorithm 4 applies only when the data are informative for the solvability of Problem 2, that is, when conditions (1) and (2) of Proposition 3 hold. If this is not the case, some components of the vector $\hat{\mathbf{m}}(t)$ necessarily diverge to infinity as t goes to $-\infty$, and hence the termination condition in step (2) is never satisfied.

(iii) The formulation of the infinite-horizon optimal control problem in (22) and (23), and consequently of its data-driven counterpart in (23)–(25), is predicated on the complete arbitrariness of the initial condition. Accordingly, in order for a solution to exist, the value of the cost function must be finite for every initial state in $\mathcal{L}_N$. On the other hand, the two necessary and sufficient conditions for the solvability of Problem 2, stated in Proposition 3, have distinct implications. If condition (1) is not satisfied, the minimization problem in (25) admits no solution for any initial condition, and therefore running Algorithm 4 is meaningless. By contrast, if condition (1) holds but condition (2) fails, then Problem 2 is solvable only for the states from which it is possible to reach the set $\mathcal{X}^{\mathbf{C}_d}$ in (26). To evaluate the reachability of this set, we employ Algorithm 1 from [15], which also returns the basin of attraction of $\mathcal{X}^{\mathbf{C}_d}$, denoted by $\mathcal{S}^*$, namely the set of states from which it is possible to reach some state in $\mathcal{X}^{\mathbf{C}_d}$. Therefore, the problem can still be solved for all states in $\mathcal{S}^*$, for which the existence of a solution is guaranteed by a broader interpretation of Proposition 3, which considers each initial condition individually. To compute such solutions, Algorithm 4 can be applied with the same termination condition, but evaluated only for the entries that are indexed in $\mathcal{S}^*$. An illustration of this scenario is provided in Example 2.

We are now ready to state the main result of this section.

Theorem 2. Assume that the data $(\mathbf{x}_d, \mathbf{u}_d)$ are informative for the solvability of Problem 2.

(1) Algorithm 4 converges in a finite number of steps to some nonnegative vector $\hat{\mathbf{m}}^*$.

(2) The solution to Problem 2 is given by

$$\hat{J}^*(\mathbf{x}_0) = (\hat{\mathbf{m}}^*)^\top \mathbf{x}_0. \tag{27}$$

Moreover, since $\hat{J}^*(\mathbf{x}_0)$ is a data-optimal solution, for every BCN in $\mathcal{B}_d$ the optimal cost function $J^*(\mathbf{x}_0)$ satisfies

$$J^*(\mathbf{x}_0) \leqslant \hat{J}^*(\mathbf{x}_0).$$

(3) Upon defining

$$\hat{i}^*(j) \triangleq \arg\min_{i\in\mathcal{I}_j}\Big\{[\mathbf{c}_i]_j + [(\hat{\mathbf{m}}^*)^\top L_i]_j\Big\},$$

the data-optimal control input can be implemented by means of the static state-feedback law $\mathbf{u}(t) = \hat{K}\mathbf{x}(t)$, where

$$\hat{K} = \left[\delta_M^{\hat{i}^*(1)}\ \delta_M^{\hat{i}^*(2)}\ \ldots\ \delta_M^{\hat{i}^*(N)}\right].$$

Remark 6. The infinite-horizon optimal control problem for BCNs has been formulated in several ways in the literature, depending on the chosen cost criterion. Existing approaches include the use of a constant cost vector [27], a discounted cost function [20, 23], and an average cost formulation [28]. Focusing on the case of a constant cost vector, which is the framework considered in this paper, a solution exists if and only if, from every initial state, there exists a path that reaches a state/input trajectory of zero cost. Obviously, while in a model-based framework we have complete knowledge of the BCNs transitions, in a data-driven context we only have partial knowledge of the network. Thus, there may be cases in which the problem is solvable with a model-based approach, but if the collected data are not sufficiently representative of the system dynamics, the existence of state trajectories ending in zero-cost cycles (or even the existence of such cycles) cannot be deduced from the available data. The results obtained in this paper align with those obtained in [17], where a problem related to Problem 2 was studied[5)]. More specifically, the authors employed the data informativity framework to address guaranteed-cost control. Their objective was to design a feedback controller that ensures that a given cost function J does not exceed a prescribed bound c, starting from any initial state in the network. The minimum cost $\hat{\mathbf{m}}^*$ obtained in the present paper thus corresponds to the smallest cost bound c in [17] for which the data are informative for guaranteed-cost control. The proposed solution relies on solving an equation analogous to the Bellman equation in dynamic programming. Alternatively, an iterative strategy is applied only to the states in $\mathcal{L}_N$ that can reach a zero-cost cycle, i.e., the states belonging to the set V^* (referring to the notation in [17]). For each such state, the optimal cost, computed from data, corresponds to the final term of the generated sequence.

We have already highlighted the fact that the solution to Problem 2 is data-optimal, and hence represents a sub-optimal solution to the infinite-horizon optimal control problem in (22) and (23). In the following proposition we aim at exploring the conservativeness of the bound provided by the data-optimal solution.

Proposition 4. Assume that the data $(\mathbf{x}_d, \mathbf{u}_d)$ are informative for the solvability of the infinite-horizon data-optimal control problem. For every BCN in $\mathcal{B}_d$ (and hence, in particular, for the one that generated the data), the solution $J^*(\mathbf{x}_0)$ to the optimization problem in (22) and (23) satisfies

$$\underline{J}^*(\mathbf{x}_0) \leqslant J^*(\mathbf{x}_0) \leqslant \hat{J}^*(\mathbf{x}_0), \tag{28}$$

5) To the best of our knowledge, this is the only other reference where an infinite-horizon optimal control problem is addressed in a data-driven scenario.

where $\hat{J}^*(\mathbf{x}_0)$ is given in (27), and $\underline{J}^*(\mathbf{x}_0)$ is the minimum value that the cost function $J(\mathbf{x}_0, \mathbf{u}(\cdot))$ in (23) takes, for a(ny) BCN in $\mathcal{B}_d$ in which all the transitions in $\overline{\mathbf{D}^{\mathbf{x},\mathbf{u}}}$ lead to some state $\delta_N^r, r \in \mathcal{V}^{\mathbf{C}_d}$.

Proof. Assuming that the data are informative for the problem solvability automatically implies that there exist some zero-cost periodic state/input trajectories that can be identified from the collected data, i.e., $\mathbf{C}_d \neq \emptyset$, and that the set of states indexed in $\mathcal{V}^{\mathbf{C}_d}$ is globally reachable.

The rightmost inequality of (28) trivially follows from point (2) of Theorem 2. So, we only need to prove the leftmost inequality, i.e., $\underline{J}^*(\mathbf{x}_0) \leqslant J^*(\mathbf{x}_0)$. To this end, we first notice that since the vector $\mathbf{c} \in \mathbb{R}_+^{NM}$ is known, we also know for every $i \in [1, M]$ and every $j \in [1, N]$ the cost $[\mathbf{c}_i]_j$ of applying the input δ_M^i to the state δ_N^j. However, by having access only to the collected data to solve the problem, we do not necessarily know the ending state of such transitions, and thus the subsequent costs. This is the case for all (and only) the pairs $(\delta_N^j, \delta_M^i) \in \overline{\mathbf{D}^{\mathbf{x},\mathbf{u}}}$.

We now want to construct a specific BCN belonging to $\mathcal{B}_d$ (i.e., to select for each missing transition $(\delta_N^j, \delta_M^i) \in \overline{\mathbf{D}^{\mathbf{x},\mathbf{u}}}$ a successor state, namely some logical vector $\delta_N^r, r \in [1, N]$, such that $\delta_N^r = L_i \delta_N^j$) in such a way that the optimal value of the cost function is the minimum one that any BCN in $\mathcal{B}_d$ can attain starting from $\mathbf{x}_0$. Let $\underline{L} \in \mathcal{L}_{N \times NM}$ denote the matrix describing such BCN. Clearly, for every $j \in [1, N]$ and every $i \in \mathcal{I}_j$ the value of $\underline{L}_i \delta_N^j$ is known from data, while for every $(\delta_N^j, \delta_M^i) \in \overline{\mathbf{D}^{\mathbf{x},\mathbf{u}}}$ we assume that $\underline{L}_i \delta_N^j$ coincides with δ_N^r, where r is an arbitrary index in $\mathcal{V}^{\mathbf{C}_d}$.

Let $\{\underline{\mathbf{m}}(t)\}_{t \in \mathbb{Z}, t \leqslant 0}$ and $\{\mathbf{m}(t)\}_{t \in \mathbb{Z}, t \leqslant 0}$ be the sequences of vectors obtained for the BCN described by $\underline{L}$ and for an arbitrary BCN $\tilde{L} \in \mathcal{B}_d$, respectively, when we run Algorithm 1 backward (see footnote 4)). We want to prove that $\forall t \leqslant 0$ we have $\underline{\mathbf{m}}(t) \leqslant \mathbf{m}(t)$, by this meaning that $[\underline{\mathbf{m}}(t)]_j \leqslant [\mathbf{m}(t)]_j, \forall j \in [1, N]$. This will clearly ensure that $\underline{\mathbf{m}}^* \leqslant \mathbf{m}^*$. We proceed by induction on t.

- For $t = 0$, we have $\underline{\mathbf{m}}(0) = \mathbb{0}_n = \mathbf{m}(0)$, and hence the result trivially holds. We also note that if $j \in \mathcal{V}^{\mathbf{C}_d}$, then $[\underline{\mathbf{m}}(t)]_j = [\mathbf{m}(t)]_j = 0, \forall t \leqslant 0$.
- We now suppose that the result is true for some $t \in \mathbb{Z}, t \leqslant 0$, i.e., $\underline{\mathbf{m}}(t) \leqslant \mathbf{m}(t)$, and we prove it for $t - 1$, i.e., $\underline{\mathbf{m}}(t - 1) \leqslant \mathbf{m}(t - 1)$. For every $j \in [1, N]$, it holds

$$
\begin{aligned}
[\underline{\mathbf{m}}(t-1)]_j &= \min_{i \in [1,M]} \left\{ [\mathbf{c}_i^\top + (\underline{\mathbf{m}}(t))^\top \underline{L}_i]_j \right\} \\
&= \min \left\{ \min_{i \in \mathcal{I}_j} \left\{ [\mathbf{c}_i^\top + (\underline{\mathbf{m}}(t))^\top \underline{L}_i]_j \right\}, \min_{i \notin \mathcal{I}_j} \left\{ [\mathbf{c}_i^\top + (\underline{\mathbf{m}}(t))^\top \underline{L}_i]_j \right\} \right\} \\
&\overset{\text{(a)}}{=} \min \left\{ \min_{i \in \mathcal{I}_j} \left\{ [\mathbf{c}_i^\top + (\underline{\mathbf{m}}(t))^\top \tilde{L}_i]_j \right\}, \min_{i \notin \mathcal{I}_j} [\mathbf{c}_i]_j \right\} \\
&\overset{\text{(b)}}{\leqslant} \min \left\{ \min_{i \in \mathcal{I}_j} \left\{ [\mathbf{c}_i^\top + (\mathbf{m}(t))^\top \tilde{L}_i]_j \right\}, \min_{i \notin \mathcal{I}_j} \left\{ [\mathbf{c}_i^\top + (\mathbf{m}(t))^\top \tilde{L}_i]_j \right\} \right\} \\
&= \min_{i \in [1,M]} \left\{ [\mathbf{c}_i^\top + (\mathbf{m}(t))^\top \tilde{L}_i]_j \right\} = [\mathbf{m}(t-1)]_j, \quad \forall \tilde{L} \in \mathcal{B}_d.
\end{aligned}
$$

The equality (a) follows from the fact that for every $j \in [1, N]$, if $i \in \mathcal{I}_j$, then $\underline{L}_i \delta_N^j = \tilde{L}_i \delta_N^j = L_i \delta_N^j$, since such transitions can be deduced from data, while if $i \notin \mathcal{I}_j$, we have $\underline{L}_i \delta_N^j = \delta_N^r, r \in \mathcal{V}^{\mathbf{C}_d}$, by construction, and hence $(\underline{\mathbf{m}}(t))^\top \underline{L}_i \delta_N^j = [\underline{\mathbf{m}}(t)]_r = 0$. The inequality (b), instead, follows from the inductive hypothesis and the nonnegativity of the vector sequence $\{\mathbf{m}(t)\}_{t \in \mathbb{Z}, t \leqslant 0}$.

Therefore, for every BCN in $\mathcal{B}_d$, it holds

$$
\underline{J}^*(\mathbf{x}_0) = (\underline{\mathbf{m}}^*)^\top \mathbf{x}_0 \leqslant (\mathbf{m}^*)^\top \mathbf{x}_0 = J^*(\mathbf{x}_0).
$$

Remark 7. Note that if the lower- and upper-bounds in (28) coincide, we obtain that the data-optimal solution is actually optimal.

As a benchmark for the infinite-horizon data-optimal control problem, we revisit Example 1 from the finite-horizon scenario, but now with a constant cost vector (i.e., discount factor $\lambda = 1$) and an infinite time horizon.

Example 2 (λ switch: infinite-horizon scenario)**.** Consider the same system as in Example 1, described by (19), or equivalently by (4) with L given in (20). The two equilibrium points (equivalently, cycles of unit length) of the BCN correspond to the lysogenic state and the lytic state, which, according to [33, Section 4], are associated with δ_{32}^{24} and δ_{32}^{31}, respectively.

Our goal is to solve the infinite-horizon minimization problem in (22) and (23) with the objective of avoiding the lytic state and stabilizing the system at the equilibrium corresponding to the lysogenic state. To this end, we

Table 1 Solutions to the infinite-horizon optimal and data-optimal control problems for valid initial states.

Initial state	δ_{32}^{1}	δ_{32}^{2}	δ_{32}^{3}	δ_{32}^{4}	δ_{32}^{5}	δ_{32}^{6}	δ_{32}^{7}	$\boldsymbol{\delta_{32}^{8}}$	δ_{32}^{9}	δ_{32}^{10}	δ_{32}^{11}
Optimal cost	16	7	19	0	22	4	20	**4**	4	5	7
Data-optimal cost	18	8	21	2	24	9	22	**4**	6	12	9
Initial state	δ_{32}^{12}	$\underline{\delta_{32}^{13}}$	δ_{32}^{14}	δ_{32}^{15}	δ_{32}^{16}	δ_{32}^{17}	δ_{32}^{18}	δ_{32}^{19}	$\boldsymbol{\delta_{32}^{20}}$	δ_{32}^{21}	$\boldsymbol{\delta_{32}^{22}}$
Optimal cost	7	13	11	13	5	18	2	22	**4**	16	**4**
Data-optimal cost	13	–	13	15	7	22	8	24	**4**	18	**4**
Initial state	δ_{32}^{23}	$\boldsymbol{\delta_{32}^{24}}$	δ_{32}^{25}	δ_{32}^{26}	δ_{32}^{27}	δ_{32}^{28}	δ_{32}^{29}	δ_{32}^{30}	$\underline{\delta_{32}^{31}}$	δ_{32}^{32}	
Optimal cost	17	**0**	12	2	16	10	24	12	–	15	
Data-optimal cost	19	**0**	14	4	18	12	26	14	–	17	

modify the cost vector $\mathbf{c}$ in (21) as follows:

$$\mathbf{c}_1^\top = [1\ 7\ 4\ 0\ 9\ 4\ 8\ 8\ 2\ 4\ 5\ 5\ 1\ 7\ 1\ 1\ 5\ 2\ 7\ 6\ 1\ 4\ 2\ \color{red}{0}\ 2\ 2\ 1\ 6\ 8\ 5\ 7\ 6],$$

$$\mathbf{c}_2^\top = [1\ 8\ 9\ 2\ 7\ 9\ 5\ 4\ 3\ 1\ 2\ 3\ 3\ 4\ 1\ 1\ 3\ 8\ 7\ 4\ 2\ 7\ 7\ 9\ 3\ 1\ 9\ 8\ 6\ 5\ \color{red}{10}\ 2].$$

Specifically, we set to zero the cost of applying input δ_2^1 to the lysogenic state δ_{32}^{24}, while increasing to 10 the cost of applying input δ_2^2 to the lytic state δ_{32}^{31} (see the two values in red). By doing so, it is easy to see that the only zero-cost periodic state/input trajectory in the system is the one generated by applying input δ_2^1 to the equilibrium point δ_{32}^{24}. We use the same data sequences as in Example 1 (see Table A1 in Appendix A).

To facilitate comparison, we first present the model-based solution, i.e., the true optimal solution, obtained by applying the procedure proposed in [27] and reported in Theorem 1. The only zero-cost periodic trajectory of the BCN (i.e., the equilibrium point δ_{32}^{24}) cannot be reached from every initial state $\mathbf{x}(0) \in \mathcal{L}_{32}$; consequently, the infinite-horizon control problem, as stated (i.e., for arbitrary initial conditions), is not solvable. Nonetheless, as already highlighted in Remark 5, there exist initial states for which the minimum value of the infinite-horizon cost function in (23) is finite; in particular, this holds for all states except the other equilibrium point δ_{32}^{31}. By restricting the computation of the solution to the states for which the problem is well-posed, we obtain the results appearing in the second row of Table 1. The feedback matrix K that realizes the optimal control input $\mathbf{u}(t) = K\mathbf{x}(t)$ is given by

$$K = \left[\, \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^2\ \delta_2^1\ \delta_2^2\ \delta_2^2\ \delta_2^2\ \delta_2^2\ \delta_2^2\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^2\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ *\ \delta_2^2 \,\right],$$

where the entry denoted by "$*$" cannot be determined, since it corresponds to the state δ_{32}^{31} which, if taken as initial condition, leads to trajectories that never reach the zero-cost periodic trajectory. Consequently, the value of $i^*(31)$ in Theorem 1 is not defined. However, the value of this entry is irrelevant, since trajectories originating from valid initial states never pass through the states corresponding to an undetermined value in K; otherwise, the zero-cost periodic trajectory would also be reachable from those states. An analogous observation holds in the data-driven setting.

We now compute the data-optimal solution to the same problem. Needless to say, using only data, the problem, if solvable at all, is expected to be solvable for an even smaller set of initial states, and therefore the data will not be informative for the solvability of Problem 2. Nonetheless, by applying the proposed data-driven procedure, we can first identify the zero-cost periodic state/input trajectories compatible with the data (in this case, the unique zero-cost periodic state/input trajectory) using Algorithm 3. Subsequently, we can use Algorithm 1 from [15] to determine the set of states from which this zero-cost periodic trajectory is reachable, leveraging only the transitions recorded in the data, i.e., those in $\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$, and thereby to identify the states for which the data-optimal value of the cost function is finite.

The digraph $\mathcal{G}_d^{\mathbf{x},\mathbf{u}}$, obtained through Algorithm 3, consists of a single node, $(24, 1)$, with a self-loop, since the state/input/state transition $(\delta_{32}^{24}, \delta_2^1, \delta_{32}^{24})$ occurs during both the first and third experiments and is thus captured by the available data, thereby forming the set $\mathbf{C}_d$. Now it remains to compute the basin of attraction $\mathcal{S}^*$ of the state δ_{32}^{24} via Algorithm 1 from [15]. In the third row of Table 1, we report the data-optimal solutions for the states belonging to $\mathcal{S}^*$. The feedback matrix $\hat{K}$ that realizes the data-optimal control input $\mathbf{u}(t) = \hat{K}\mathbf{x}(t)$ is given by

$$\hat{K} = \left[\, \delta_2^2\ \delta_2^2\ \delta_2^1\ \delta_2^2\ \delta_2^2\ \delta_2^2\ \delta_2^2\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ *\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^2\ \delta_2^1\ \delta_2^2\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^1\ \delta_2^2\ \delta_2^1\ \delta_2^1\ *\ \delta_2^2 \,\right].$$

As it can be seen by comparing the second and third rows of Table 1, for the initial states for which the infinite-horizon control problem is well-posed both in the model-based and data-driven contexts, the data-optimal cost provides an upper estimate of the true optimal cost (although the gap is generally small), except for four initial states for which the two costs coincide. These cases are in bold in Table 1. Instead, the states for which the problem is not solvable are underlined.

Remark 8 (Computational complexity). The computational complexity of the algorithms for BCNs has long posed a major challenge for researchers in the field. Due to the inherent structural complexity of these systems, significant computational difficulties arise whenever algorithmic procedures are applied. In particular, many problems involving BCNs are NP-hard with respect to the state, input, and output dimensions n, m and p, or equivalently polynomial in the corresponding quantities $N = 2^n$, $M = 2^m$ and $P = 2^p$ [35–37]. This complexity barrier is intrinsic and independent of the specific representation used. Consequently, applying such algorithms to large-scale networks becomes extremely demanding.

The same comments apply to the computational complexities of the algorithms presented in this paper, particularly Algorithms 2 and 4 for solving the finite- and infinite-horizon data-driven optimal control problems. Indeed, they are comparable to those of the corresponding model-based algorithms, discussed in Remarks 4 and 7 of [27], respectively. Specifically, in [27], it has been shown that the model-based finite-horizon optimal control algorithm has complexity $O(NMT)$, where T is the length of the time horizon, while the model-based infinite-horizon algorithm (with constant cost vector) has complexity $O(N^2M)$. In a data-driven framework, the main difference is that the computational complexities of the algorithms also depend on the amount of collected data, i.e., T_d, which is, however, lower-bounded by N. On the other hand, since the data provide only partial knowledge of the system dynamics, the number of identified transitions is typically smaller than in the full model. Consequently, if their complete enumeration is required, it imposes a lower computational burden. More specifically, in Algorithm 2, step (2) has computational complexity $O(T_d)$, while the iterative procedure has complexity $O(|\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}|\ T)$. The cardinality of the set $\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}$, which is smaller than NM, can be evaluated as the rank of the matrix $X_p * U_p$, where $*$ denotes the Khatri-Rao product. For Algorithm 4, as discussed in Remark 7 of [27], the iterative procedure terminates in at most N steps. At each iteration, only the state/input transitions deducible from the data are processed, resulting in a computational complexity of $O(|\mathbf{D}^{\mathbf{x},\mathbf{u},\mathbf{x}}|\ N)$, which is lower than that of the model-based counterpart. It should be noted, however, that the algorithm relies on the sets $\mathcal{I}_j$ computed in step (2) of Algorithm 2, whose cost is $O(T_d)$, as previously remarked.

As for Algorithm 3, the computational cost of steps (2) and (3), devoted to constructing the directed graph $\mathcal{G}_d^{\mathbf{x},\mathbf{u}}$ of state/input pairs compatible with the data and associated with zero cost, is $O(T_d)$. However, the overall computational complexity is dominated by step (4), which employs Johnson's algorithm [34] to enumerate all simple cycles in the digraph. The complexity of this procedure is $O((|\mathcal{V}^{\mathbf{x},\mathbf{u}}|+|\mathcal{E}^{\mathbf{x},\mathbf{u}}|)(q+1))$, where $|\mathcal{V}^{\mathbf{x},\mathbf{u}}|$ and $|\mathcal{E}^{\mathbf{x},\mathbf{u}}|$ denote the cardinalities of the node and edge sets of $\mathcal{G}_d^{\mathbf{x},\mathbf{u}}$, respectively, and q is the number of identified cycles.

Therefore, despite its data-driven nature, the proposed methodology still exhibits scalability limitations. Scalability is, in fact, an intrinsic challenge in any control problem involving BCNs, as these systems are inherently complex and affected by the curse of dimensionality, independently of the chosen methodology. Nevertheless, a data-driven approach can help mitigate this issue, as it relies only on a subset of the full set of transitions.

5 Conclusion

In this paper we have investigated how the informativity approach introduced in [18] and recently applied to BCNs in [15, 17] can be extended to deal with optimal control problems for BCNs using only some previously collected data. This approach is based on the idea that, when data do not allow for system identification, the best we can do is to solve the problem of interest for all systems that are compatible with these data. More specifically, we have considered both the finite-horizon and the infinite-horizon optimal control problems, and we have provided necessary and sufficient conditions for the solvability of such problems in all BCNs compatible with the available data. Then, for each of the two problems we have proposed an algorithm to compute the optimal solution based on the data (i.e., the data-optimal solution). In addition, we have discussed the sub-optimality of the data-optimal solutions to the finite-horizon and the infinite-horizon optimal control problems, and provided a lower bound on the true (i.e., data-independent) optimal solutions. Finally, we have tested the proposed finite-horizon and infinite-horizon data-optimal solutions on a biological system of practical relevance.

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

Open Access funding enabled and organized by CAUL and its Member Institutions.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1 Kauffman S A. Metabolic stability and epigenesis in randomly constructed genetic nets. J Theor Biol, 1969, 22: 437–467
2 Bornholdt S. Boolean network models of cellular regulation: prospects and limitations. J R Soc Interface, 2008, 5: S85
3 Cheng D Z. On finite potential games. Automatica, 2014, 50: 1793–1801
4 Kabir M, Hoque M, Koo B J, et al. Mathematical modelling of a context-aware system based on Boolean control networks for smart home. In: Proceedings of ISCE 2014 Conference, Jeju Island, 2014
5 Schreiber F, Valcher M. Feedback context-aware pervasive systems in healthcare management: a Boolean network approach. In: Edge-of-Things in Personalized Healthcare Support Systems. Pittsburgh: Academic Press, 2022. 237–260
6 Cheng D Z. Semi-tensor product of matrices and its application to Morgens problem. Sci China Ser F-Inf Sci, 2001, 44: 195–212
7 Cheng D Z. Input-state approach to Boolean networks. IEEE Trans Neural Netw, 2009, 20: 512–521
8 Cheng D, Qi H. A linear representation of dynamics of Boolean networks. IEEE Trans Automat Contr, 2010, 55: 2251–2258
9 Cheng D Z, Qi H. State-space analysis of Boolean networks. IEEE Trans Neural Netw, 2010, 21: 584–594
10 Cheng D, Qi H, Li Z. Analysis and Control of Boolean Networks. London: Springer-Verlag, 2011
11 Cheng D Z. Disturbance decoupling of Boolean control networks. IEEE Trans Automat Contr, 2011, 56: 2–10
12 Fornasini E, Valcher M E. Feedback stabilization, regulation and optimal control of Boolean control networks. In: Proceedings of 2014 American Control Conference, Portland, 2014. 1993–1998
13 Li R, Yang M, Chu T. State feedback stabilization for Boolean control networks. IEEE Trans Automat Contr, 2013, 58: 1853–1857
14 Liu Q, Guo X, Zhou T. Optimal control for probabilistic Boolean networks. IET Syst Biol, 2010, 4: 99–107
15 Disarò G, Valcher M E. Data-based control of logical networks. 2025. arXiv:2505.11667
16 Leifeld T, Zhang Z, Zhang P. Data-driven controller design for Boolean control networks. In: Proceedings of 2018 Annual American Control Conference (ACC), 2018. 3044–3049
17 Li R, Shi R, Zhang Q, et al. Data informativity for analysis and control design of Boolean control networks. SIAM J Control Optim, 2025, 63: 151–174
18 van Waarde H J, Eising J, Trentelman H L, et al. Data informativity: a new perspective on data-driven analysis and control. IEEE Trans Automat Contr, 2020, 65: 4753–4768
19 Van Waarde H J, Eising J, Camlibel M K, et al. The informativity approach: to data-driven analysis and control. IEEE Control Syst, 2023, 43: 32–66
20 Zhao Y, Li Z, Cheng D. Optimal control of logical control networks. IEEE Trans Automat Contr, 2011, 56: 1766–1776
21 Laschov D, Margaliot M. A Pontryagin maximum principle for multi-input Boolean control networks. In: Recent Advances in Dynamics and Control of Neural Networks. Cambridge: Cambridge Scientific Publishers, 2011
22 Laschov D, Margaliot M. A maximum principle for single-input boolean control networks. IEEE Trans Automat Contr, 2011, 56: 913–917
23 Zhu Q, Liu Y, Lu J, et al. On the optimal control of Boolean control networks. SIAM J Control Optim, 2018, 56: 1321–1341
24 Pal R, Datta A, Dougherty E R. Optimal infinite-horizon control for probabilistic Boolean networks. IEEE Trans Signal Process, 2006, 54: 2375–2387
25 Ching W K, Zhang S Q, Jiao Y, et al. Optimal control policy for probabilistic Boolean networks with hard constraints. IET Syst Biol, 2009, 3: 90–99
26 Liu Q. Optimal finite horizon control in gene regulatory networks. Eur Phys J B, 2013, 86: 245
27 Fornasini E, Valcher M E. Optimal control of Boolean control networks. IEEE Trans Automat Contr, 2014, 59: 1258–1270
28 Wu Y, Sun X M, Zhao X, et al. Optimal control of Boolean control networks with average cost: a policy iteration approach. Automatica, 2019, 100: 378–387
29 Li H, Yang X. Robust optimal control of logical control networks with function perturbation. Automatica, 2023, 152: 110970
30 Li Y, Wu J, Feng J, et al. Robust finite-horizon optimal control of Boolean control networks under disturbances via Q-learning. In: Proceedings of the 44th Chinese Control Conference (CCC 2025), 2025. 5585–5591
31 Disarò G, Valcher M E. Data-based optimal control of logical networks. In: Proceedings of the 29th International Conference on System Theory, Control and Computing (ICSTCC), Cluj-Napoca, 2025. 468–473
32 Ptashne M. A Genetic Switch. New York: Cold Spring Harbor Laboratory Press, 2004
33 Laschov D, Margaliot M. Minimum-time control of Boolean networks. SIAM J Control Optim, 2013, 51: 2869–2892
34 Johnson D B. Finding all the elementary circuits of a directed graph. SIAM J Comput, 1975, 4: 77–84
35 Ghaffari N, Ivanov I, Qian X, et al. A CoD-based reduction algorithm for designing stationary control policies on Boolean networks. Bioinformatics, 2010, 26: 1556–1563
36 Zhao Y, Ghosh B K, Cheng D. Control of large-scale Boolean networks via network aggregation. IEEE Trans Neural Netw Learn Syst, 2016, 27: 1527–1536
37 Akutsu T, Hayashida M, Ching W K, et al. Control of Boolean networks: hardness results and algorithms for tree structured networks. J Theor Biol, 2007, 244: 670–679