IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 21, 2021, accepted November 3, 2021, date of publication November 23, 2021,
date of current version December 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3130118

Optimal Control of Probabilistic Boolean Networks: An Information-Theoretic Approach

K. SONAM[1], (Member, IEEE), SARANG SUTAVANI[2], S. R. WAGH[1], (Senior Member, IEEE), AND N. M. SINGH[1]

[1]Electrical Engineering Department, Veermata Jijabai Technological Institute, Mumbai 400019, India
[2]Department of Mechanical Engineering, Clemson University, Clemson, SC 29634, USA

Corresponding author: K. Sonam (srkharade_p18@ee.vjti.ac.in)

ABSTRACT The primary challenge with biological sciences is to control gene regulatory networks (GRNs), thereby creating therapeutic intervention methods that alter network dynamics in the desired manner. The optimal control of GRNs with probabilistic Boolean control networks (PBCNs) as the underlying structure is a solution to this challenge. Owing to the exponential growth in network size with the increase in the number of genes, we need an optimal control approach that scales to large systems without imposing any limitations on network dynamics. Furthermore, we are encouraged to use the graphics processing unit (GPU) to reduce time complexity utilizing the easily available and enhanced computational resources. The optimal control of PBCNs in the Markovian framework is developed in this paper employing an information-theoretic approach which includes Kullback-Leibler (KL) divergence. We convert the nonlinear optimal control problem of PBCN to a linear problem by using the exponential transformation of the cost function, also known as the desirability function. The linear formulation enables us to compute an optimal control using the path integral (PI) method. Furthermore, we offer sampling-based methodologies for approximating PI and therefore optimizing PBCN control. The sampling-based method can be implemented in parallel, which solves the optimal control problem for large PBCNs.

INDEX TERMS Information-theoretic control, Markov decision processes (MDPs), optimal control, parallel processing, probabilistic Boolean control networks (PBCNs).

I. INTRODUCTION

The popularity of Boolean networks (BNs) has increased gradually since Kauffman introduced them [1]. BNs are qualitative models that describe the dynamics of the gene activity profile (GAP) characterizing the status of a gene as active/inactive (or 0/1) in gene regulatory networks (GRNs). BNs can be used to investigate GRNs in a restricted environment, however their simplicity, along with their deterministic rigidity, limits the ability to analyze reasonably complex network dynamics. The use of probabilistic switching between several networks substantially improves a model's capacity to represent behaviors that are near to actual observations. The probabilistic Boolean networks (PBNs) introduced by Shmulevich [2] include this stochastic nature, making them a natural choice for modeling the limited information GRNs. PBNs with the addition of Boolean control inputs are called probabilistic Boolean control networks (PBCNs). Since the introduction of semi-tensor product (STP) of matrices by Cheng *et al.* [3], many fundamental properties of switched Boolean control networks, PBNs and PBCNs have been characterized in the literature including but not limited to observability [4]–[6], controllability [7]–[9], reconstructibility [4], fault detection [10], stabilizability [11]–[15], structure identification [16], [17], output tracking control [18], [19] and model checking [20].

GRN activities are closely associated to a certain health-related problems, such as cancer. Naturally, the possibility of controlling GRN behavior in such a way that it avoids its states from adverse configurations attracts a lot of interest. PBCNs are used to design such control strategies. The goal of developing a PBCN control approach is to determine the gene perturbation effect and devise an optimal intervention to alter the network's long-term behavior or modify its dynamics. The former is traditionally referred to as structural intervention, while the latter is known as external control of PBCNs [21].

The authors in [22]–[24] propose the control-theoretic formulations that employ the STP. In particular, the robust event-triggered control of PBCNs is examined in [23], optimal

The associate editor coordinating the review of this manuscript and approving it for publication was Daniel Grosu.

time-varying feedback controllability for PBCNs is discussed in [22], and the authors have proposed a pinning control approach in [24]. Because PBCN states form a Markov chain with a finite state space, it is compatible with Markov decision processes (MDPs). In [25], the authors include discrete-time MDP into PBCN modeling and establish a probability criterion to restrict the induced loss defined by the state cost for finite time steps, before the network encounters the desirable states for the first time. Several more studies use the MDP framework for PBCN optimal control, including [26]–[28]. In [29], the context-sensitive PBNs with perturbations are used to minimize the finite horizon expected cost. Recently, reinforcement learning (RL) based techniques have been proposed in [30]–[36] to solve the optimal control problem of PBCNs.

The curse of dimensionality i.e., the problem of exponentially growing states and controls as the network size grows, affects most of the approaches available to control PBCNs. This problem cannot be solved without making some simplifying assumptions. Due to the necessity of computations on large matrices, the matrix-based techniques, i.e., STP or MDP, worsen the performance. Besides the utility of matrix-based methods is infeasible to even smaller systems than methods which do not involve matrix operations. Furthermore, for large systems, the approximate solution techniques impose a computational load, and convergence to the (sub)optimal solution is only guaranteed in the limited scenario of infinite iterations. Furthermore, adequate error margin performance can only be ensured after several simulations.

The techniques from continuous-time stochastic optimal control (SOC) [37], [38] can be adapted for PBCNs to address most of the above challenges. A family of nonlinear control problems with control affine dynamics and quadratic control cost is examined by Kappen [39] using the path integral (PI) based representation. The PI-based SOC is restated as a problem of minimizing the Kullback-Leibler (KL) divergence between controlled and uncontrolled transition distributions in [40]. The KL divergence is also referred to as an information cost between two distributions. The resulting control formulation is regarded as information-theoretic control [41] with the cumulative sum of the state dependent cost and information cost as free energy [42]. The framework of linearly-solvable MDP (LMDP) [43] achieves a comparable formulation for discrete state space with the restriction of information cost in terms of KL divergence and the transition probabilities denoting continuous inputs.

Inspired by the preceding discussion, we develop a novel information-theoretic strategy to effectively solve the optimal control of PBCN and implement the same using a graphics processing unit (GPU) based parallel processing framework. The optimal control of PBCNs is proposed in the MDP framework, with the cost-to-go consisting of the state cost and the information cost. The following are the key contributions of the proposed framework in this paper:

1) To get the advantage of the inherent stochastic behavior of PBCNs, an information-theoretic formulation utilizing the augmented state space is proposed for optimal control of PBCNs.
2) To obtain the solution to information-theoretic control through approximation and overcome limitations of the Monte Carlo sampling method, an entropy-based improved Monte Carlo sampling technique is proposed.
3) To overcome memory constraints, a matrix-free approach for simulating the PBCN model is developed.
4) To obtain scalability of optimal control in large PBCNs, a GPU-based parallel implementation is introduced.

The paper is organized as follows: Section II introduces the required fundamentals. In Section III, the PBCN classical and information-theoretic optimal control formulation over the proposed augmented state space is investigated. Section IV deals with the solution in terms of desirability function estimation, employing the path integral representation and its improved Monte Carlo sampling-based approximation. In Section V, several algorithms and GPU-based implementation are developed. A couple of illustrative examples are presented to validate the effectiveness proposed method, and the scalability is demonstrated by implementation on a 37-gene T-cell network in Section VI.

NOTATIONS

$\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{Z}$, and $\mathbb{Z}^+$ denote the sets of real numbers, nonnegative real numbers, integers and nonnegative integers, respectively. The scalar multiplication of two numbers is denoted by $\times$. $\mathbb{E}[\cdot]$ represents the expectation. The set of integers is indicated by $\{m_1, \ldots, m_2\}$ for given any integers $m_1, m_2 \in \mathbb{Z}^+$, such that $m_1 \leq m_2$. The symbol $|P|$ denote the cardinality of any set P. If both of the inputs are the identical, the function $\delta(\cdot, \cdot)$ returns 1, else it returns 0. We use the symbols X (U) and s (a) for states (control inputs) of PBCN and MDP, respectively. We denote the Boolean domain by $\mathcal{B} := \{0, 1\}$. Similarly,the Cartesian product of $\mathcal{B}$ n-times is given by $\mathcal{B}^n := \underbrace{\mathcal{B} \times \cdots \times \mathcal{B}}_{n}$. Logical AND, OR, and NOT operations are denoted by $\wedge$, $\vee$, and $\neg$, respectively.

II. PRELIMINARIES

In the following, we present a brief review of Probabilistic Boolean control networks (PBCNs) in the Markovian framework, Markov decision processes (MDPs), and information-theoretic control framework.

A. PROBABILISTIC BOOLEAN CONTROL NETWORKS

For $i \in \{1, 2, \ldots, n\}$ and $k \in \{1, 2, \ldots, m\}$ consider the nodes $x_i(t) \in \mathcal{B}$ and Boolean control inputs $u_k(t) \in \mathcal{B}$. In this network, the vector representation of the expression levels of all the genes at time t is given by the row vector $x(t) \in \mathcal{B}^n$ defined as $x(t) := [x_1(t)\ x_2(t) \ldots x_n(t)]$. Similarly, the row vector corresponding to inputs is represented by $u(t) \in \mathcal{B}^m$ and defined as $[u_1(t) \ldots u_m(t)]$. For every node x_i consider a set $\mathcal{F}_i = \{f_i^{(j)}\} \forall j \in 1, \ldots, l_i$, where each $f_i^{(j)} : \mathcal{B}^{n+m} \to \mathcal{B}$

represents a possible predictor function (or an update rule) to determine the value of gene x_i and $|\mathcal{F}_i|$ is the number of possible predictor functions of x_i. A PBCN with n genes and m control inputs for $t \in \mathbb{Z}^+$ is described as follows

$$x_i(t+1) = f_i^{(j)}(x_1(t), \ldots, x_n(t), u_1(t), \ldots, u_m(t)). \quad (1)$$

The probability of selection associated with the predictor function $f_i^{(j)}$ is denoted as $c_i^{(j)}$ and it satisfies the following condition

$$\Sigma_{j=1}^{|\mathcal{F}_i|} c_i^{(j)} = 1.$$

For any PBCN, the total number of constituent Boolean control networks (BCNs) is given by

$$\mathcal{N} = \prod_{i=1}^{n} |\mathcal{F}_i|.$$

Let $\tilde{f}_l$ denote the dynamics of the l^{th} constituent BCN and $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{\mathcal{N}}$ be the selection probability associated with each of the $\mathcal{N}$ possible networks. The selection probability of a l^{th} BCN denoted by $\mathcal{P}_l$ is obtained as,

$$\mathcal{P}_l = \prod_{i}^{n} c_i^{(l_i)},$$

with selection of the functional relationships comprising of $(f_1^{(l_1)}, f_2^{(l_2)}, \ldots, f_n^{(l_n)})$ where $l_i \in \{1, \ldots, |\mathcal{F}_i|\}$.

B. MARKOV DECISION PROCESS

An Markov decision process (MDP) is represented by a tuple of a state space, an action space, transition probabilities, and reward, i.e., $\big(\mathcal{S}, \mathcal{A}, P(s'|s,a), R(s'|s,a)\big)$ [44] given as state space and action space as

$$\mathcal{S} := \{s_1, \ldots, s_{|\mathcal{S}|}\}, \quad \mathcal{A} := \{a_1, \cdots, a_{|\mathcal{A}|}\}.$$

It's a Markov process that generates a series of states which follow the Markov property

$$\Pr(s(t+1) \mid s(t), \ldots, s(0)) = \Pr(s(t+1)|s(t)),$$

where $s(t+1), s(t), s(0) \in \mathcal{S}$ and are affected by external interventions. Given a state $s \in \mathcal{S}$ and a control action $a \in \mathcal{A}$, we can calculate the transition probability for possible next state $s' \in \mathcal{S}$ as

$$P(s'|s,a) = \Pr s(t+1) = s'|s(t) = s, a(t) = a.$$

Likewise, there is a cost associated with any current state s, current action a, and the future state s' as follows

$$R(s'|s,a) = \mathbb{E}\{r(t+1)|s(t) = s, a(t) = a, s(t+1) = s'\},$$

which indicates favorability of a state. In this case, $r(t+1)$ represents a real valued function of the state and action.

C. INFORMATION-THEORETIC CONTROL FRAMEWORK

The standard control problem with discrete control inputs and arbitrary control cost varies from the information-theoretic control problem [42] presented here.

Definition 1: Continuous input is regarded as the input across a continuous set of transition probabilities from a given state under the effect of discrete control input a, i.e., $a(s'|s) = P(s'|s,a)$.

Definition 2: At state s, the uncontrolled distribution also referred to as passive dynamics, $P(\cdot|s)$, describes the system behavior in the absence of control inputs.

In continuous stochastic systems, the passive dynamics (or distribution) is clearly defined, but its discrete case equivalent corresponds to the random walk in state space.

Definition 3 [45]: The KL divergence given below is a dissimilarity measure between two distributions $h_1(s)$ and $h_2(s)$

$$\begin{aligned}\mathrm{KL}(h_1 \parallel h_2) &= \mathbb{E}_{h_1}\left[\log\left(\frac{h_1(s)}{h_2(s)}\right)\right] \\ &= \int h_1(s) \log\left(\frac{h_1(s)}{h_2(s)}\right) ds. \end{aligned} \quad (2)$$

The information-theoretic approach can be applied to MDPs where immediate cost representing the one-step running cost, i.e., $l(s,a)$ is calculated as the sum of the state cost and the KL divergence between controlled and passive dynamics, $\mathrm{KL}\big(P(\cdot|s,a) \parallel P(\cdot|s)\big)$ as shown below.

$$\mathrm{KL}\big(P(\cdot|s,a) \parallel P(\cdot|s)\big) = \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[\log \frac{P(s'|s,a)}{P(s'|s)}\right],$$

where $P(s'|s,a)$ and $P(s'|s)$ are transition probabilities under the controlled and passive dynamics respectively.

$$l(s,a) = q(s) + \mathbb{E}_{s' \sim a(\cdot|s)}\big[\log(a(\cdot|s)/P(\cdot|s))\big], \quad (3)$$

The cost of altering the passive distribution can be viewed as the KL divergence.

The following absolute continuity condition must be satisfied ffor well-defined KL divergence.

$$P(s'|s) = 0 \implies P(s'|s,a) = 0. \quad (4)$$

By using the Bellman principle of optimality, the optimal control is determined as the minimizing control a of the cost-to-go function represented by $J(s)$.

$$J(s) = \min_{a \in \mathcal{A}} \left\{ q(s) + \mathbb{E}_{s' \sim a(\cdot|s)}\left[\log \frac{a(\cdot|s)}{P(\cdot|s)}\right] + J(s') \right\}. \quad (5)$$

To get the linear form of the above nonlinear Bellman equation, the desirability function obtained as $z(s) = \exp(-J(s))$. By the introduction of normalization term $G_z(s) = \sum P(s'|s)z(s) = \mathbb{E}_{s' \sim P(\cdot|s)}$ in (5), yields the following optimization problem

$$\min_{a \in \mathcal{A}} \left\{ q(s) - \log G_z(s) + \mathrm{KL}\left(a(s'|s) \parallel \frac{P(s'|s)z(s)}{G_z(s)}\right) \right\}, \quad (6)$$

which achieves minimum when the controls are selected using (7).

$$a^*(s'|s) = \frac{P(s'|s)z(s)}{G_z(s)}. \tag{7}$$

Substituting (7) in (5) results in the following linear equation in z

$$z(s) = \exp(-q(s))G_z(s'). \tag{8}$$

The function $z(s)$ indicates *how desirable a state is*. Since the desirability function is defined as an exponential of the negative cost function, the desirability function has a higher value for trajectories with lower costs. An expectation operator, which is a linear operator, constitutes the normalising term, as a result, the equation (8) can be solved using techniques applicable to the linear system of equations such as the Eigenvalue problem, iterative backward evaluation, and others. The linear equation (8) is solved to find desirability function of all states. The information-theoretic control framework doesn't provide the optimal control input explicitly; instead, it gives the optimal transition probability under optimal control input.

III. PBCN OPTIMAL CONTROL PROBLEM FORMULATION

The control dependent counterpart of the probability distribution vector $\mathrm{w}(t)$ for one-step evolution is given by

$$\mathrm{w}(t+1) = \mathrm{w}(t)P(U(t)), \tag{9}$$

where $P(U(t))$, represented by $P(U)$ in sequel for $U \in \mathcal{U}$, is a controlled transition probability matrix in $\mathbb{R}^{2^n \times 2^n}$. For detailed description one can refer [28]. Let $\tilde{f}_l(X_I, U_K)$ be the dynamics of the l^{th} constituent BCN. For notational simplicity, we have suppressed the dependence of states (X), control inputs (U) over time t.

The controlled Markov chain can be utilised to describe the dynamical behavior of PBCN, the theory of MDP can be employed to find an optimal intervention policy [46]. The PBCN can be defined as an MDP comprising of the set of states $\mathcal{X} := \{X_1, \ldots, X_{2^n}\}$ with $X_I = 1 + \sum_{i=1}^{n} 2^{n-i}x_i$ the set of controls $\mathcal{U} := \{U_1, \ldots, U_{2^m}\}$ with the control vector $U_K = 1 + \sum_{k=1}^{m} 2^{m-k}u_k$, and the probability of transition from $X = X_I$ to $X' = X_J$ under input $U \in \mathcal{U}$ is obtained from the element in I^{th} row and J^{th} column of the controlled transition distribution matrix $P(U)$ in (9). In the following, we discuss the classical approach, its limitations, and our proposed framework for PBCN optimal control.

A. CLASSICAL APPROACH TO OPTIMAL CONTROL

Under the probability distributions specified by (9), optimal control problem is to minimizing the expected cost (or cost-to-go function) of trajectories beginning from any state. To regulate PBCN behavior, control inputs are applied for a finite horizon t_f and the states are divided into two categories: favorable (low penalty) and undesired (high penalty). The optimal control seeks a policy $\pi = \{\mu_0, \mu_1, \cdots, \mu_{t_f-1}\}$, where $\mu_t : \mathcal{X} \to \mathcal{U}$ mapping the state space to control space. A one step cost $l(X, U) : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^+$ comprises of the state dependent cost $q(X) : \mathcal{X} \to \mathbb{R}^+$ and control cost $g(U) : \mathcal{U} \to \mathbb{R}^+$ i.e., $l(X, U) = q(X) + g(U)$. For X, the cost-to-go for a trajectory generated by the policy is calculated as follows:

$$J_{\pi,t}(X) = \mathbb{E}_{X' \sim P(\cdot|X,U)}\Big[\sum_{\tau=t}^{t_f-1} l(X(\tau), \pi_\tau(X(\tau))) + \phi(X(t_f))\Big],$$

and the corresponding Bellman equation as

$$\begin{aligned} J_t(X) &= \min_U\{l(X, U) + \mathbb{E}_{X' \sim P(\cdot|X,U)}\big[J_{t+1}(X')\big]\}, \\ J_{t_f}(X(t_f)) &= \phi(X(t_f)). \end{aligned} \tag{10}$$

The control policy $\pi^* = \mu_0^*, \mu_1^*, \cdots, \mu_{t_f-1}^*$ is optimum and $J_t(X) = J_{\pi^*,t}(X)$ is the optimal cost-to-go, if $U^*(t) = \mu_t^*(X)$ minimizes the right hand side of (10) for each state X and time t. Because the cost remains bounded, a solution to the optimal control problem is always exists. Hence, in this case, the resultant control policy is a time-dependent short-term policy that alters PBCN's dynamic behavior.

Despite the fact that the MDP framework enables dynamic programming, the established Bellman equation (10) is nonlinear and ineffective in several cases as

i) it rarely offers analytic solutions and is computationally expensive to solve using approximation methods in general,
ii) each iteration in iterative approaches requires an exhaustive search over all feasible inputs,
iii) in terms of computational cost and memory, an exact solution for large PBCN suffers from the curse of dimensionality.

In comparison to traditional MDP-based approaches that approximate the solution of a nonlinear Bellman equation, the information-theoretic framework that can overcome the said hurdles is more powerful computationally because it approximates the solution of a linear equation. However, in posing the PBCN optimal control in this context, the following difficulties arise:

i) The classical PBCN possesses discrete and finite inputs, however we require continuous inputs that represent the transition probabilities.
ii) Because the PBCN is control action non-affine, a possible passive dynamics can coincide to a network with the equally likely occurrence of actions. However, out of $P(U_K)\ \forall U_K \in \mathcal{U}$, the passive dynamics $P(\cdot|X)$ have no obvious choice, and switching between controlled dynamics $P(U_K)$ may not be feasible if the absolute continuity requirement (4) is not fulfilled.
iii) Information-theoretic cost cannot be incorporated into the context-specific control cost specified for given PBCN.

In the next section, we develop a novel state space architecture to overcome these difficulties.

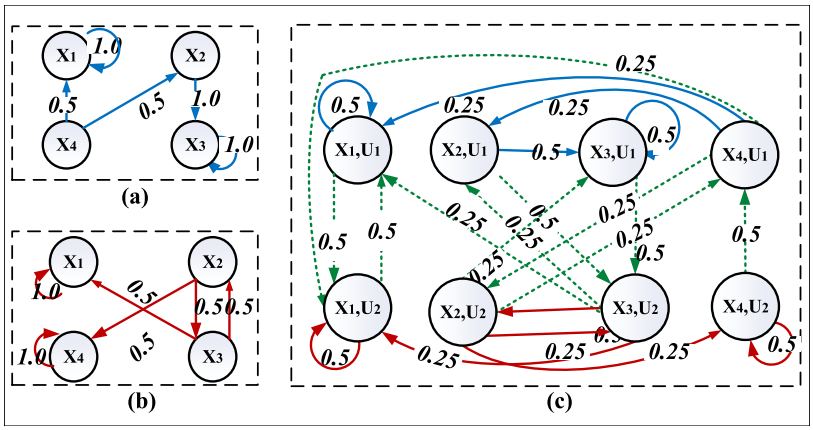

FIGURE 1. Example 1 state transition (a) under input U_1 (b) under input U_2 (c) in augmented state space. Solid lines represent transitions determined from state space for inputs U_1 (in blue) and U_2 (in red). Dotted lines indicate the new transitions as a result of augmentation.

B. CONSTRUCTION OF AUGMENTED STATE SPACE

To allow information-theoretic control of PBCN, we start with augmented state space generation and provide the resolution of the difficulties described in the preceding section.

Definition 4: For a PBCN (1), the augmented state space $\tilde{\mathcal{X}}$ is constructed as the set of all the tuples generated by the Cartesian product of $\mathcal{X}$ and the set of available control inputs $\mathcal{U}$ i.e.,

$$\tilde{\mathcal{X}} := \mathcal{X} \times \mathcal{U} := \{(X_I, U_K) | \, X_I \in \mathcal{X}, U_K \in \mathcal{U}\}. \quad (11)$$

Definition 5: The transition probability of the augmented state $\tilde{X}$ to $\tilde{X}'$ is defined as

$$\tilde{P}_{\tilde{X},\tilde{X}'} = \sum_{l=1}^{\mathcal{L}} \delta(X_I', \tilde{f}_l(X_I, U_K)) \mathcal{P}_l \frac{1}{2^m}. \quad (12)$$

Remark 1: The transition probability for augmented states is the average value of all underlying BCN transition probabilities. Whereas any alternative choice is likely to experience bias, the impact of continuous control inputs is considered to be equally probable for an unspecified natural intervention rate.

Example 1 [28]: Consider a two-gene PBCN with single control input with the system evolution according to $(c_i^{(j)})$. The gene x_1 has a single Boolean function $f_1^{(1)}$ with probability $c_1^{(1)} = 1$ while gene x_2 has two Boolean predictor functions $f_2^{(1)}$ and $f_2^{(2)}$ with probability of selection $c_2^{(1)} = 0.5$ and $c_2^{(2)} = 0.5$ respectively.

$$x_1(t+1) = f_1^{(1)} = x_1(t) \vee u(t)$$
$$x_2(t+1) = \begin{cases} f_2^{(1)} = x_2(t) \vee x_1(t) \wedge u(t) \\ f_2^{(2)} = x_1(t) \wedge u(t) \end{cases}$$

FIGURE 1 (a) and (b) depicts the state transition diagrams corresponding to different values of control input. The augmented state space is obtained in FIGURE 1 (c) as

$$\tilde{\mathcal{X}} = \{(X_1, U_1), (X_2, U_1), (X_3, U_1), (X_4, U_1), (X_1, U_2), (X_2, U_2), (X_3, U_2), (X_4, U_2)\}.$$

$$\tilde{P} = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0 & 0 & 0.25 & 0.25 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0.25 & 0 & 0 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 \end{bmatrix}$$

As shown below, the augmented state space resolves the challenges in restructuring PBCN optimal control in the paradigm of information-theoretic control.

1) the input A of the augmented state space can now be considered equivalent to the continuous transition probabilities i.e., $P(\tilde{X}'|\tilde{X}, A) = A(\tilde{X}'|\tilde{X})$ as required by the information-theoretic framework,
2) All of the controlled transition probabilities $P(U_K)$ can be integrated into a single state transition matrix $\tilde{P}$, which is $2^{m+n} \times 2^{m+n}$ in size. The probability matrix $\tilde{P}$ is used to represent the state transition matrix for the system's passive dynamics, which addresses problem (ii) from the preceding section. Furthermore, meeting the criterion of absolute continuity is no longer required in switching between controlled dynamics.
3) The penalty to be incurred for reshaping the passive dynamics can be viewed as the KL divergence between continuous input A and passive dynamics $\tilde{P}$.

Remark 2: The aforementioned transition probability matrix $2^{n+m} \times 2^{n+m}$) in size over augmented state space is not the same as the STP-based matrix 2^{n+m} in size [34]. The developed approach, in this paper, does not involve a matrix representation of the system dynamics. The additional dimensions show that the state space has grown, yet they do not cause problems when used in large systems. This is due to the fact that this methodology is based on a trajectory dependent approach. The trajectories can be obtained from recorded data or produced through simulations using a known dynamic model.

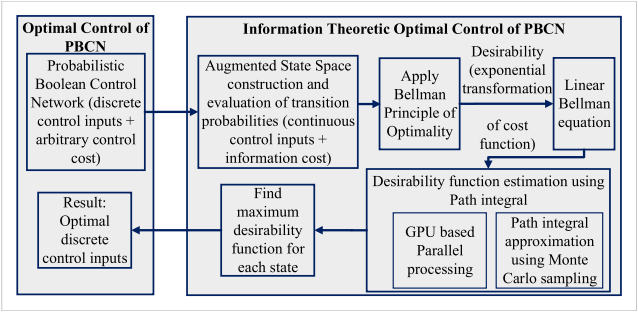

FIGURE 2. PBCN information-theoretic optimal control in nutshell.

C. PBCN INFORMATION-THEORETIC CONTROL

The proposed information-theoretic optimal control PBCN is presented in the FIGURE 2. In augmented state space, the Bellman equation is reformulated with the immediate cost $\tilde{l}(\tilde{X}, A)$ and the terminal cost $J_{t_f}(\tilde{X}(t_f)) = \phi(\tilde{X}(t_f))$ as follows

$$J_t(\tilde{X}) = \min_A \{\tilde{l}(\tilde{X}, A) + \mathbb{E}_{\tilde{X}' \sim A(\cdot|\tilde{X})}[J_{t+1}(\tilde{X}')]\}, \tag{13}$$

In this case, $\tilde{q}(\tilde{X})$ is the augmented state cost obtained as $q(X_I) + g(U_K)$. The immediate cost is given as

$$\tilde{l}(\tilde{X}, A) = q(X_I) + g(U_K) + \mathbb{E}_{\tilde{X}' \sim A(\cdot|\tilde{X})}\left[\log \frac{A(\cdot|\tilde{X})}{\tilde{P}(\cdot|\tilde{X})}\right].$$

The minimizing control A of the optimum control is generated from (13).

$$J_t(\tilde{X}) = \min_A \Big\{ q(X_I) + g(U_K) + \mathbb{E}_{\tilde{X}' \sim A(\cdot|\tilde{X})}\left[\ln \frac{A(\cdot|\tilde{X})}{\tilde{P}(\cdot|\tilde{X}) \exp\left(-J_{t+1}(\tilde{X}')\right)}\right]\Big\}. \tag{14}$$

Taking into account the desirability function $z_t(\tilde{X}) = \exp(-J_t(\tilde{X}))$ and $z_{t+1}(\tilde{X}) = \exp(-J_{t+1}(\tilde{X}))$. Incorporating z_t and z_{t+1} in (14) provides

$$-\ln z_t(\tilde{X}) = \min_A \Big\{ q(X_I) + g(U_K) + \mathbb{E}_{\tilde{X}' \sim A(\cdot|\tilde{X})}\left[\ln \frac{A(\cdot|\tilde{X})}{\tilde{P}(\cdot|\tilde{X}) z_{t+1}(\tilde{X}')}\right]\Big\}, \tag{15}$$

Consider the normalizing term for the denominator of expectation as $\sum_{\tilde{X}'} \tilde{P}(\tilde{X}'|\tilde{X}) z_{t+1}(\tilde{X}') = G[z_{t+1}](\tilde{X})$. The normalizing term $G[z_{t+1}](\tilde{X})$ is divided by the term inside expectation results in

$$-\ln(z_t(\tilde{X})) = \min_A \left\{ q(X_I) + g(U_K) + \mathbb{E}_{A(\cdot|\tilde{X})}\left[\ln\left(\frac{\frac{A(\tilde{X}'|\tilde{X})}{G[z_{t+1}](\tilde{X})}}{\frac{\tilde{P}(\tilde{X}'|\tilde{X}) z_{t+1}(\tilde{X})}{G[z_{t+1}](\tilde{X})}}\right)\right]\right\},$$

The expectation term is expressed as follows

$$\mathbb{E}_{A(\cdot|\tilde{X})}\left[\ln\left(\frac{A(\tilde{X}'|\tilde{X})}{G[z_{t+1}](\tilde{X})}\right) - \ln\left(\frac{\tilde{P}(\tilde{X}'|\tilde{X}) z_{t+1}(\tilde{X})}{G[z_{t+1}](\tilde{X})}\right)\right]\Big\}, \tag{16}$$

$$= \mathbb{E}_{A(\cdot|\tilde{X})}\left[\ln\left(A(\tilde{X}'|\tilde{X})\right) - \ln\left(G[z_{t+1}](\tilde{X})\right) - \ln\left(\frac{\tilde{P}(\tilde{X}'|\tilde{X}) z_{t+1}(\tilde{X})}{G[z_{t+1}](\tilde{X})}\right)\right]\Big\}, \tag{17}$$

Since the term $-\ln(G[z_{t+1}](\tilde{X}))$ is constant with respect to distribution $A(\cdot|\tilde{X})$, the expectation $\mathbb{E}_{A(\cdot|\tilde{X})}[-\ln(G[z_{t+1}](\tilde{X}))]$ will be independent and equal to $-\ln(G[z_{t+1}](\tilde{X}))$. Substituting the expectation in (16) results in

$$-\ln(z_t(\tilde{X})) = \min_A \left\{ q(X_I) + g(U_K) - \ln\left(G[z_{t+1}](\tilde{X})\right) + \mathbb{E}_{A(\cdot|\tilde{X})}\left[\ln\left(A(\tilde{X}'|\tilde{X})\right) - \ln\left(\frac{\tilde{P}(\tilde{X}'|\tilde{X}) z_{t+1}(\tilde{X})}{G[z_{t+1}](\tilde{X})}\right)\right]\right\},$$

The expectation $\mathbb{E}_{A(\cdot|\tilde{X})}\left[\ln\left(\frac{A(\tilde{X}'|\tilde{X})}{\frac{\tilde{P}(\tilde{X}'|\tilde{X}) z_{t+1}(\tilde{X})}{G[z_{t+1}](\tilde{X})}}\right)\right]$ is represents the KL divergence between $A(\tilde{X}'|\tilde{X})$ and $\frac{\tilde{P}(\tilde{X}'|\tilde{X}) z_{t+1}(\tilde{X})}{G[z_{t+1}](\tilde{X})}$. Therefore,

$$-\ln(z_t(\tilde{X})) = \min_A \left\{ q(X_I) + g(U_K) - \ln G[z_{t+1}](\tilde{X}) + \mathrm{KL}\left(A(\cdot|\tilde{X}) \parallel \frac{\tilde{P}(\tilde{X}'|\tilde{X}) z(\tilde{X}')}{G[z_{t+1}](\tilde{X})}\right)\right\}. \tag{18}$$

Because the normalizing term $(-\ln G[z_{t+1}](\tilde{X}))$, state costs $q(X_I)$, and control costs $g(U_K)$ are independent of A (optimizing variable), if the contribution in (18) of the KL divergence is zero, the minimum is achieved. As a result, by setting the KL divergence component to zero, the optimal continuous inputs can be found as

$$A^*(\tilde{X}'|\tilde{X}) = (\tilde{P}(\tilde{X}'|\tilde{X}) z(\tilde{X}'))/G[z_{t+1}](\tilde{X}). \tag{19}$$

Replacing $A(\cdot|\tilde{X})$ with the optimal continuous control input $A^*(\cdot|\tilde{X})$ and $q(X_I) + g(U_K)$ by $\tilde{q}(\tilde{X})$ results in

$$\begin{aligned} -\ln(z_t(\tilde{X})) &= \tilde{q}(\tilde{X}) - \ln G[z_{t+1}](\tilde{X}), \\ \ln(z_t(\tilde{X})) &= -\left(\tilde{q}(\tilde{X}) - \ln G[z_{t+1}](\tilde{X})\right), \\ z_t(\tilde{X}) &= \exp\left(-\tilde{q}(\tilde{X})\right) + \exp\left(\ln G[z_{t+1}](\tilde{X})\right), \\ z_t(\tilde{X}) &= \exp\left(-\tilde{q}(\tilde{X})\right) G[z_{t+1}](\tilde{X}), \end{aligned} \tag{20}$$

where $z(\tilde{X})$ is referred to as the optimal desirability function of the augmented state $\tilde{X}$. The normalizing term $G[z_{t+1}]$ is replaced with $\sum_{\tilde{X}'} \tilde{P}(\tilde{X}'|\tilde{X}) z_{t+1}(\tilde{X}')$ in (20) results in a linear Bellman equation in the desirability function for optimal control at time t as follows

$$\begin{aligned} z_t(\tilde{X}) &= \exp\left(-\tilde{q}(\tilde{X})\right) G[z_{t+1}](\tilde{X}), \\ &= \exp\left(-\tilde{q}(\tilde{X})\right) \mathbb{E}_{\tilde{X}' \sim \tilde{P}(\cdot|\tilde{X})}\left[z_{t+1}(\tilde{X}')\right], \\ z_{t_f}(\tilde{X}) &= \exp\left(-\phi(\tilde{X}(t_f))\right). \end{aligned} \tag{21}$$

In the augmented settings the desirability function is evaluated for all augmented states by solving the linear equation (21). In this case, the state $\tilde{X}(t) \in \tilde{\mathcal{X}}$ will have the maximum desirability if the trajectories starting from the

same state result in the lower values of the cost function. Besides the desirability function here comprises both costs, i.e., the state and control costs. This scenario helps in finding the optimal control input for the problem under consideration. The desirability For small size systems, the desirability function (21) can be obtained through matrix operation.

IV. PATH INTEGRAL SOLUTION TO INFORMATION-THEORETIC CONTROL OF PBCN

For large systems, the PI-based method is advantageous since it can be adapted to work without explicitly computing the state transition matrix.

A. DESIRABILITY FUNCTION ESTIMATION USING PATH INTEGRAL

Using the Feynman–Kac lemma and diffusion process [39], it is possible to transform the backward in time calculation for the optimal control solution to a forward in time computation. Similar rationale the discrete system can be used to establish PI formulation with continuous inputs and information [47]. The desirability function is determined using PI, in which the expected cost of a particular state is estimated by taking into account all of the system's possible paths. By substituting $\mathcal{G}[z_t](\tilde{X})$ over augmented states, the desirability function $z_t(\tilde{X})$ is expressed in terms of PI as follows:

$$\begin{aligned} z_t(\tilde{X}) &= \exp\big(-\tilde{q}(\tilde{X})\big) G_{z_{t+1}}(\tilde{X}), \\ &= \exp\big(-q(\tilde{X})\big) \mathbb{E}_{\tilde{X}'\sim\tilde{P}(\cdot|\tilde{X})}\big[z_{t+1}(\tilde{X}')\big]. \end{aligned} \tag{22}$$

Similarly, we have,

$$z_{t+1}(\tilde{X}') = \exp\big(-\tilde{q}(\tilde{X}')\big) \mathbb{E}_{\tilde{X}''\sim\tilde{P}(\cdot|\tilde{X})}\big[z_{t+2}(\tilde{X}'')\big],$$

with $\tilde{X}''$ denoting the augmented state at time $t+2$. Substituting $z_{t+1}(\tilde{X}')$ in (22) we get the following form in terms of probability independent cost $\tilde{q}(\cdot)$ for $z_t(\tilde{X})$,

$$\mathbb{E}_{\tilde{X}'\sim\tilde{P}(\cdot|\tilde{X})}\Big[\exp-\big(\tilde{q}(\tilde{X}) + \tilde{q}(\tilde{X}')\big) \mathbb{E}_{\tilde{X}''\sim\tilde{P}(\cdot|\tilde{X})}\big[z_{t+2}(\tilde{X}'')\big]\Big].$$

With the recursion of $z(\cdot)$ in the inner expectation the PI based desirability function is derived as follows

$$z_t(\tilde{X}) = \mathbb{E}_{\tilde{X}'\sim\tilde{P}(\cdot|\tilde{X})}\Big[\exp-\big(\tilde{\phi}(\tilde{X}_{t_f}) + \sum_{\tau=t+1}^{t_f-1} \tilde{q}(\tilde{X}_\tau)\big)\Big]. \tag{23}$$

B. SAMPLING BASED APPROXIMATION

In most cases, analytic assessment of PI is impractical, causing the use of numerical approximation. The PI is calculated using a random sampled trajectories from a distribution $\tilde{P}$ using the Monte Carlo (MC) sampling method which yields the desirability function approximation shown below

$$\tilde{z}(\tilde{X}) \approx \frac{1}{\mathrm{S}} \sum_s \exp(-\tilde{q}_{s,t\to t_f}), \tag{24}$$

where S is total number of samples. The cost over the sampled path s, i.e., $\tilde{q}_{s,t\to t_f}$, is defined as $\big(\tilde{\phi}(\tilde{X}_{t_f}) + \sum_{\tau=t+1}^{t_f-1} \tilde{q}(\tilde{X}_\tau)\big)$.

MC sampling, on the other hand, has the following drawbacks:

i) When generating augmented passive dynamics $\tilde{\mathcal{P}}$, an arbitrary starting distribution for selection of the control action must be considered.
ii) The optimal desirability generated corresponds to arbitrarily assumed initial control action distribution. Order of desirability may be different for different initial control action distributions.
iii) The contribution of every randomly generated path is weighted equally in desirability estimation. Therefore, the non-optimal paths could introduce significant bias in the estimation.
iv) There is no way to indirectly comment on other distributions under the selection of different control actions.
v) Uncertainty in system dynamics not taken into account.

C. ENTROPY-BASED IMPROVED MC SAMPLING

The limitations as mentioned above can be overcome by introducing modifications in MC sampling as follows:

i) A weighting function is introduced to modify the expected cost of trajectories starting from the required augmented state.
ii) Cost-dependent weightage is assigned to each path. The weights assigned could be either monotonically increasing or decreasing over the range of costs.
iii) Uncertainty in system dynamics is accounted for by introducing probability dependence in the weighting function.

We describe the weighting function in terms of average entropy, relative cost, and an application-specific parameter for an extra degree of freedom. The entropy for a gene measures the randomness associated with that gene. Furthermore, entropy is affected by the probability distribution of gene selection. This distribution has the highest entropy if it is uniform, while any other distribution has low entropy. For example, in the *Example 1*, because gene x_2 has higher randomness, its entropy will be higher than gene x_1, which has 0 entropy. Having followed these principles, we define the average entropy for a PBCN, which measures the randomness depending on the distribution of selection probabilities across the overall network.

Definition 6: The average entropy $H(x)$ of PBCN, with entropy of genes $h(x_i) = -\sum_{l_i=1}^{|\mathcal{F}_i|} c_i^{(l_i)} \log_{|\mathcal{F}_i|}(c_i^{(l_i)})$, is defined as

$$H = \frac{1}{n} \sum_i h(x_i).$$

Definition 7: The relative error R_e is defined as

$$R_e = \left(\frac{\tilde{q}_{s,t\to t_f} - \min_s \tilde{q}_{s,t\to t_f}}{\max_s \tilde{q}_{s,t\to t_f} - \min_s \tilde{q}_{s,t\to t_f}} \right). \tag{25}$$

Definition 8: Index denoted by I_d which can be used as exponent utilizing entropy is defined as

$$I_d = \frac{1-H}{H}. \tag{26}$$

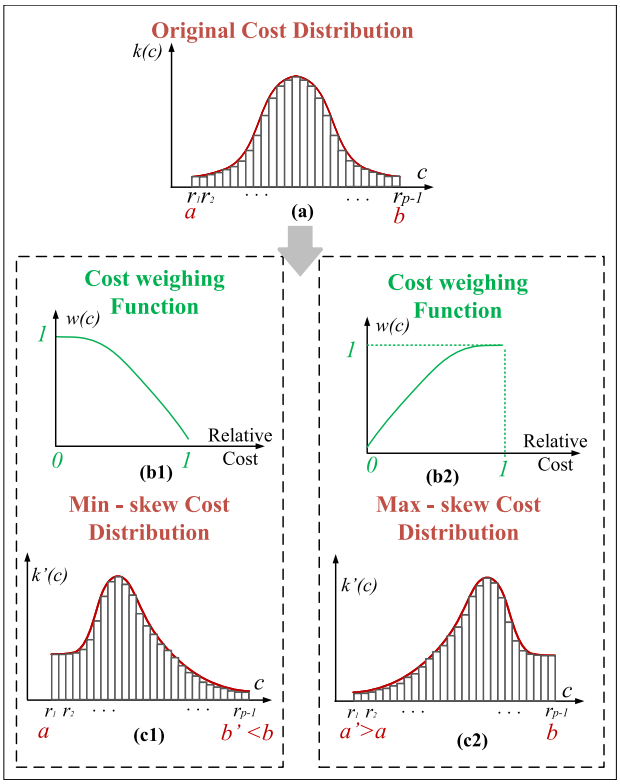

FIGURE 3. Entropy-based improved MC Sampling with min-skew and max-skew.

The weighting function $w_{s,t}(\tilde{X})$ for s^{th} sample is determined using relative error R_e (25) and power factor (26) considering the path costs either in pessimistic or optimistic manner. We propose two approaches in the subsequent section termed min-skew in optimistic way and max-skew in pessimistic way.

1) MIN-SKEW SAMPLING

The optimistic approach reinforces the greedy estimation of costs by skewing the expected path cost in the direction of minimum (min-skew). The paths with lower costs are accommodated with higher weightage and vice-versa in min-skew approach. The control action is selected that pertains to the minimum expected cost after min-skew. FIGURE 3 represents the functioning of min-skew in essence which is summarized as follows:

i) Evaluate the original cost distribution $k(c)$ (FIGURE 3 (a)) using sampled trajectories.
ii) Generate the weighting function $w(c)$ that acts as the non-linear scaling function. (FIGURE 3 (b2))
iii) Apply the nonlinear scaling $w(c)$ on x-axis (cost) depending upon relative distance form minimum cost (point a in (FIGURE 3 (a))) to determine the inwards shift (towards a). Point a remains stationary and point b experiences the maximum shift therefore termed min-skew.
iv) The expected cost after min-skew is acquired by averaging over $k'(c)$ normalized by total shift.

Some candidate weighting function $w(c)$ to be considered for min-skew case are as follows:

$$w_{s,t}(\tilde{X}) = \begin{cases} 1 & \forall\, \tilde{q}_{s,t\to t_f} = \min_s \tilde{q}_{s,t\to t_f} \\ (1-R_e)^{I_d} & \text{Otherwise} \end{cases}$$

$$w_{s,t}(\tilde{X}) = \begin{cases} 1 & \forall\, \tilde{q}_{s,t\to t_f} = \min_s \tilde{q}_{s,t\to t_f} \\ \alpha^{(-R_e)\times I_d} & \text{Otherwise} \end{cases}$$

$$w_{s,t}(\tilde{X}) = \begin{cases} 1 & \forall\, \tilde{q}_{s,t\to t_f} = \min_s \tilde{q}_{s,t\to t_f} \\ (\dfrac{1-R_e+\epsilon}{1+\epsilon})^{I_d} & \text{Otherwise} \end{cases} \tag{27}$$

where α and ϵ are application-specific parameters.

2) MAX-SKEW SAMPLING

In contrast to min-skew, in pessimistic consideration, the preference is given to limit the worst-case scenarios by skewing the expected path cost in the direction of maximum (max-skew). In max-skew paths with higher cost are assigned higher weightage and vice-versa. The control action is chosen that pertains to the minimum expected cost after max-skew. The effect of max-skew is portrayed in FIGURE 3 (b2) and (c2).

Some candidate weighting function $w'(c)$ to be considered for max-skew case are as follows:

$$w_{s,t}(\tilde{X}) = \begin{cases} 0 & \forall\, \tilde{q}_{s,t\to t_f} = \min_s \tilde{q}_{s,t\to t_f} \\ (R_e)^{I_d} & \text{Otherwise} \end{cases}$$

$$w_{s,t}(\tilde{X}) = \begin{cases} 0 & \forall\, \tilde{q}_{s,t\to t_f} = \min_s \tilde{q}_{s,t\to t_f} \\ \alpha^{(1-R_e)\times I_d} & \text{Otherwise} \end{cases}$$

$$w_{s,t}(\tilde{X}) = \begin{cases} 0 & \forall\, \tilde{q}_{s,t\to t_f} = \min_s \tilde{q}_{s,t\to t_f} \\ (\dfrac{R_e+\epsilon}{1+\epsilon})^{I_d} & \text{Otherwise} \end{cases}$$

$$w_{s,t}(\tilde{X}) = \begin{cases} 0 & \forall\, \tilde{q}_{s,t\to t_f} = \min_s \tilde{q}_{s,t\to t_f} \\ \alpha^{(1-R_e)\times I_d} & \text{Otherwise} \end{cases} \tag{28}$$

Use of (28) as the time dependent weighting function can be contemplated by considering two extreme cases.

1) When PBCN contains no uncertainty (i.e., it is nothing but a BCN), it has zero entropy ($H = 0$). The weighting function in (28) comes out to be 1 for the path corresponding to the minimum cost and 0 for all the remaining paths. This is precisely the behavior desired as for deterministic systems, the path of minimum cost can be traversed with certainty.
2) The PBCN has extreme uncertainty embedded in its structure when all the network selection probabilities are equal, i.e. $c_i^{(j)} = c_i^{(k)}\ \forall\, j,k$. For this case the entropy assumes largest possible value $H = 1$ and all paths are weighted equally as expected.

The approximation of desirability over S samples using entropy-based improved MC sampling that introduces a bias towards minimum is specified as,

$$\tilde{z}(\tilde{X}) \approx \frac{1}{\sum_s w_{s,t}(\tilde{X})} \sum_s w_{s,t}(\tilde{X}) \exp\left(-\tilde{q}_{s,t\to t_f}\right). \tag{29}$$

Once the desirability of an augmented state is estimated using (29), we propose a method to determine the corresponding discrete optimal action in the next section.

V. ALGORITHMS AND IMPLEMENTATION

A. OPTIMUM DISCRETE INPUT FROM OPTIMUM DESIRABILITY FUNCTION

The augmented state space desirability function is obtained in (24) which corresponds to the optimal continuous control input. As a result, the continuous input must be linked to the discrete inputs from the original PBCN. For each state, the optimal discrete input is determined by calculating the maximum desirability across the set of augmented states $\tilde{X} \in \tilde{\mathcal{X}}$ for a given state ($X_I \in \mathcal{X}$). Following are the steps in the procedure:

1) The vector for optimal desirability function of a given state $\tilde{X}_I$ with the passive dynamics $\tilde{P}$ is computed as

$$\mathbf{z}_t(\tilde{X}_I) = \exp(-q(\tilde{X}_I))\mathbf{G}_z(\tilde{X}_I), \tag{30}$$

where $\mathbf{z}_t(\tilde{X}_I) = [z(X_I, U_1), \cdots, z(X_I, U_{2^m})]^T$, $\mathbf{G}_z(\tilde{X}_I) = [G_z(X_I, U_1), \cdots, G_z(X_I, U_{2^m}),]^T$ and $\exp(-q(\tilde{X}_I))$ is a diagonal matrix with 2^m elements $\exp(-q(X_I, U_K))$

2) The optimum discrete input $U^* \in \mathcal{U}$ for state X_I is

$$U^* = \arg\max_{U_K}\{z(X_I, U_K) \mid K \in \{1, \cdots, 2^m\}\}. \tag{31}$$

In a summary, the augmented dynamics incorporate the corresponding distributions for all possible state transitions under the feasible control inputs.

An augmented state's desirability that gives the optimal desirability for state X_I is obtained by iterating over all possible future augmented states. This automatically includes state transitions from state X_I under all inputs along with the induced control costs. Therefore, the state with optimal desirability of X_I corresponds to the optimal discrete input U^*.

B. SCALABLE CONTROL ALGORITHM

To avoid the problems associated with matrix-based implementation, a couple of algorithms is suggested to run in conjunction that provide yields the optimal control solution of large PBCN. The matrix-free evaluation of the next state is obtained from Algorithm 1. In this algorithm, we use the predictor function selection probability of each gene and compare it with a randomly generated probability. Based on the probabilities comparison the expression level of gene is determined.

Algorithm 1 is used to generate the trajectories starting from the given initial state X_I for the terminal time t_f while evaluating the PBCN optimal control input U^* using Algorithm 2. In this case, $r_u(arg1, arg2)$ generates a random sequence of length $arg1$ by sampling from an uniform distribution 0 to $(arg2-1)$. We translate the objective and initialize the cost vectors and state in Algorithm 2. For the S number of samples the path cost corresponding to each trajectory is calculate using the function presented in Algorithm 3. Even

Algorithm 1 Matrix Free State Evaluation

function BooleanDynamics (state - $\{x_1(t), x_2(t), \ldots, x_n(t)\}$, input - $\{u_1(t), u_2(t), \ldots, u_m(t)\}$)
1. **for** ($i = 0$ to n)
2. Generate η_i from uniform distribution in [0, 1]
3. Set $b = 1$ and $\bar{c}_i = c_i^{(1)}$
4. **while** ($\eta_i <= \bar{c}_i$)
5. $\bar{c}_i \leftarrow \bar{c}_i + c_i^{(b)}$
6. $b \leftarrow b + 1$
7. $x_i(t+1) \leftarrow f_i^{(b)}(x_i)$

end function
Return $\{x_1(t+1), \cdots, x_n(t+1)\}$

Algorithm 2 Information-Theoretic Control of PBCNs

Result: Optimal control input U^*
Initialization: m, n, Number of samples = S, Cost vector for all genes and control inputs, t_f, initial state $X_I(0)$
1. **for** ($t = 0$ to $t_f - 1$)
2. $t' = t_f - t$
3. **for**($K = 1$ to 2^m)
4. Calculate path cost $\tilde{\mathbf{q}}_{t' \to t_f}$ for S samples by: Sequential_Cost_Compute(·) Algorithm 3 or Parallel_Cost_Compute(·) FIGURE 5
5. Evaluate desirability function $\tilde{z}(X_I(t), U_K(t))$ using: (24) for MC sampling and (29) for entropy-based improved MC sampling.
6. Calculate $U^*(X_I(t))$ using (31)

Algorithm 3 Sequential Cost Computation Over S Samples

function Sequential_Cost_Compute(S, U_K, t')
1. **for**($s = 1$ to S)
2. Generate control sequence $\mathbf{U}_s \leftarrow \{U_K, r_u(t'-1, 2^m)\}$
3. Generate network sequence $\mathbf{X}_s$ as:
4. **for** (U_r in $\mathbf{U}_s$)
5. Calculate $\mathbf{X}_s(r)$ for U_r using Algorithm 1
6. Evaluate path cost $\tilde{\mathbf{q}}_{t' \to t_f}(s) \leftarrow \sum_{\tau=t}^{t_f-1} q(X_{s\tau})$

end function
Return Cost vector $\tilde{\mathbf{q}}_{t' \to t_f}$

though Algorithm 2 along with 3 and 1 allows optimal control inputs evaluation for large PBCN, the sample size required for a reasonable approximation of the desirability function grows rapidly. Consequently, the time taken by the calculation of the sampled trajectories costs tends to grow significantly. This problem is resolved by incorporating a variation of Algorithm 3 in the parallel computation framework, as discussed in the following section.

C. GPU BASED PARALLEL IMPLEMENTATION

Because of its programmable improvements [48], the Graphics Processing Unit (GPU) had also found utility in domains other than gaming. NVIDIA, one of the most popular GPU manufacturers, offers a compute unified device architecture environment (CUDA) which is a collection of resources for

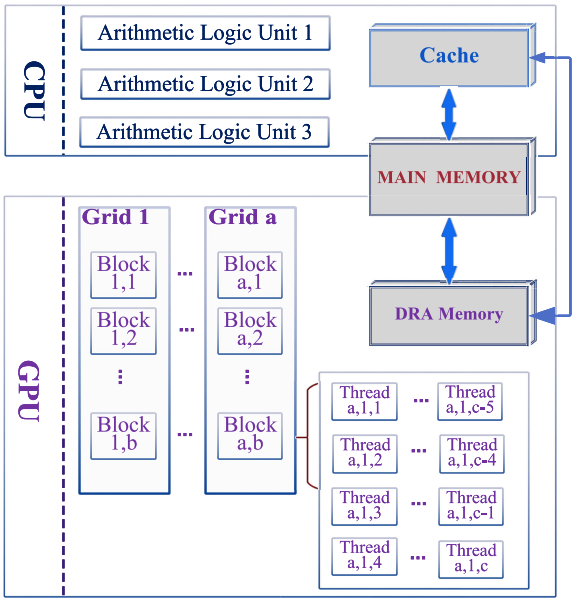

FIGURE 4. Architecture of Parallel processing using CUDA.

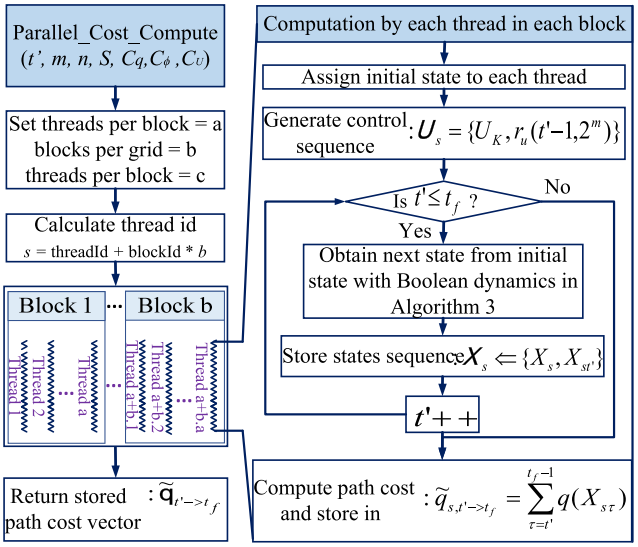

FIGURE 5. Flowchart for Parallel_Cost_Compute(·) with state cost vector C_q, terminal state cost vector C_ϕ and control cost vector C_U.

multi-threaded applications that can perform multiple tasks in parallel. FIGURE 4 shows the generalized architecture for GPU-based parallel processing. The GPU is a co-processor for the CPU with its own dynamic random access memory (DRAM) implementation [49]. In addition to implementing a large number of tasks in parallel, CUDA organizes threads into logical blocks, each of which maps onto a multiprocessor. Because the number of threads a block can handle is limited, the blocks are organized into grids in order to run a large number of threads at the same time without communicating.

Through PI, the desirability function is evaluated by estimating the costs of the paths. The parallel computing architecture is facilitated by the fact that each path's cost is calculated independently of the others. The number of threads, blocks, and grids used in the CUDA architecture is decided by the number of samples needed to approximate the desirability function. The GPU efficiently implements multiple instances of the cost calculation task, greatly reducing the computational time. Threads are the basic component of computations that are executed in the cores of GPU in the CUDA architecture. As shown in the FIGURE 5, each thread, identified by the thread id s, independently computes the cost $\tilde{q}_{s,t\to t_f}$ for one sample. Each thread starting from an initial state evaluates the trajectory and its path cost as show in the right part of FIGURE 5. This resulted path cost is then transferred to the CPU by each thread, which further executes instructions to obtain a desirability function estimate based on the sample costs available. The task division between the GPU and the CPU allows the system memory to be refreshed after each cycle of path cost calculation and desirability function estimation. As a result, for accurate estimation of the desirability function, a much larger number of samples than the system's memory handling capacity can be used.

VI. RESULTS AND DISCUSSION

Some of the existing results from the literature are compared with the solution obtained using proposed technique in this paper. The first is a two-gene artificial PBCN from [28] that was solved for optimal control using the conventional dynamic programming method. In the second illustrated case, the polynomial optimization-based approximation solution approach for a three-gene example from [50] is compared. Furthermore, we demonstrate the efficacy of entropy-based improved sampling in this scenario. After establishing the validity of the results, the technique is used to biological network optimal control problems. The WNT5A biological network, which has seven genes and one control gene, is the first biological network to be considered. A T-cell signaling network with 37 genes and three inputs is used to show the method's effectiveness for large systems. Furthermore, for this example, a GPU-based parallel implementation is used. The Python programming language is used to perform all of the simulations. The parallel implementation algorithm is executed on a Google colab GPU with a maximum virtual RAM of 12.72GB and a maximum disk space of 68.40GB.

A. ILLUSTRATIVE EXAMPLES

1) ARTIFICIAL 2-GENE NETWORK

We match the solution of our proposed method with results from [28] where the problem is solved using the classical dynamic programming approach for two genes artificial PBCN given in Example 1 previously. In this case, the terminal penalties associated with states X_1, X_2, X_3 and X_4 are assumed to be 0, 1, 2 and 3 respectively. For any intermediate time, no cost is associated with states. The control cost of 1 is

levied whenever control U_2 is applied. The control objective with $t_f = 5$ is:

$$J_t(X(t)) = \min_{U(t)} \Big[U(t) + \sum_{\tau=t}^{4} p(X(\tau+1)|X(\tau), U(\tau)) \\ J_{t+1}(X(t+1)) \Big], \quad \forall t \in \{0,1,2,3,4\}$$
$$J_5(X(5)) = \phi(X(5)).$$

The terminal costs and intermediate costs for augmented states are

$$X_I(t) = 0 \quad \forall I \in \{1,2,3,4\}$$
$$X_I(t) = 1 \quad \forall I \in \{5,6,7,8\}$$
$$X_I(t_f) = 0 \quad \forall I \in \{1,5\}\ X_I(t_f) = 1\ \forall I \in \{2,6\}$$
$$X_I(t_f) = 3 \quad \forall I \in \{3,7\}\ X_I(t_f) = 4\ \forall I \in \{4,8\}.$$

For the augmented state space, the passive dynamics are derived in Example 1, and the optimal control problem is solved using (23) and (31). The algorithms 2 and 1 are employed utilizing sequential computation, to arrive at optimal control solution. The MC sampling is used for the estimation of desirability functions of states. The control sequence obtained is, $u(t) = 0\ \forall t \in \{1,2,3,4\}\ \forall X(t) \in \{1,2,3,4\}$ and $u(5) = 0$ for $X(5) = 1,2,4$ and $u(5) = 1$ for $X(5) = 3$, which matches exactly with the result in [28].

2) ARTIFICIAL 3-GENE NETWORK

The Boolean function and corresponding transition probabilities are given in (32) for a PBCN under consideration. The optimal control problem is formulated such that the expression of gene x_3 is deregulated at end of treatment horizon. This objective can be translated to find the control input that minimizes the cost (33) in finite horizon ($t_f = 2$) case.

$$\mathcal{F}_1 = \begin{cases} f_1^{(1)} = x_3(t) \wedge \neg a(t), c_1^{(1)} = 0.8 \\ f_1^{(2)} = \neg x_3(t) \wedge \neg a(t), c_1^{(2)} = 0.2 \end{cases}$$
$$\mathcal{F}_2 = f_2^{(1)} = x_1(t) \wedge \neg x_3(t), c_2^{(1)} = 1.0$$
$$\mathcal{F}_3 = \begin{cases} f_3^{(1)} = x_1(t) \wedge \neg x_2(t), c_3^{(1)} = 0.7 \\ f_3^{(2)} = x_2(t) \vee a(t), c_3^{(2)} = 0.3 \end{cases} \tag{32}$$
$$v_t(X) = \sum_{\tau=t}^{t_f-1} x_3(\tau) + a(\tau) + x_3(t_f) \tag{33}$$

The optimal solution is achieved by use of MC sampling in Algorithms 1 - 2 and the control actions obtained for all states are $a(0) = 1, a(1) = 1$ which match with results given in [50]. The average cost, starting from all states, over 32000 simulation epochs are as depicted in FIGURE 6. This figure shows that the expression of gene x_3 at the time $t_f = 2$ using improved MC sampling is less in comparison with the MC sampling. Following improved MC sampling from (29) the better costs are achieved as illustrated in FIGURE 6 for

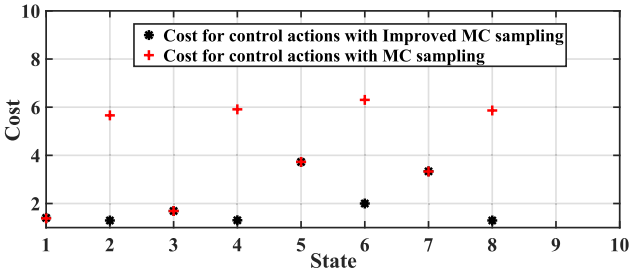

FIGURE 6. Expected costs with MC and entropy-based improved MC sampling.

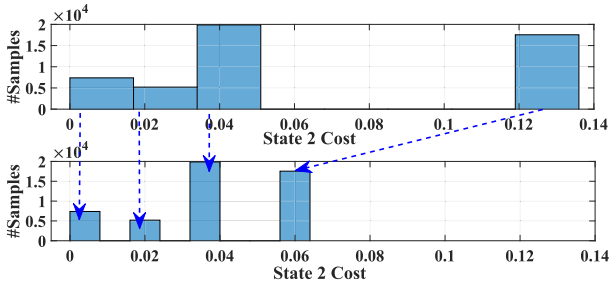

FIGURE 7. Cost distribution for state 2 with entropy-based improved MC sampling.

the following resulting control actions

At $t = 0$,
$a(0) = 2$ for states $\{2, 4, 6, 8\}$, $a(0) = 1$ for remaining states
At $t = 1$,
$a(1) = 1$ for all states. (34)

For the given treatment window, the disease should be treated by using the drug based on the observations and time refereeing to (34).

The min-skew sampling is employed to solve this problem and the cost distribution for state 2 is provided in FIGURE 7. The time and path dependent weighting function used in this case is as follows.

$$w_{s,t}(\tilde{X}) = \begin{cases} 1 & \forall\ \tilde{q}_{s,t\to t_f} = \min_s \tilde{q}_{s,t\to t_f} \\ (1-R_e)^{I_d} & \text{Otherwise.} \end{cases}$$

The entropy and the index factor obtained are

$$H = \frac{1}{3}\sum_i h(x_i)$$
$$= \frac{-1}{3}(0.8\log_2 0.8) + (0.2\log_2 0.2) + (0.7\log_2 0.7) + (0.3\log_2 0.3)$$
$$= 0.5344,$$
$$I_d = \frac{1-H}{H} = 0.8712.$$

The distribution plot of cost in FIGURE 7 clearly depicts the shift in distribution towards minimum and FIGURE 6 shows the improvement over the average cost for all states after incorporation of improved MC sampling.

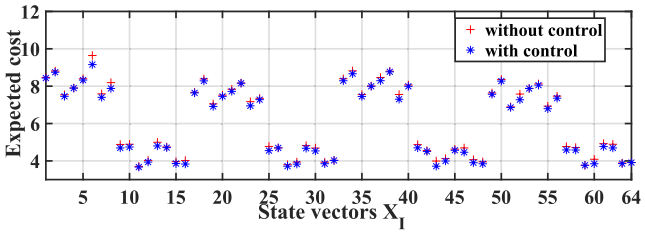

FIGURE 8. Expected cost with and without control for WNT5A network.

3) BIOLOGICAL WNT5A NETWORK

There are seven genes as WNT5A, pirin, S100P, RET1, MART1, HADHB, and STC2 in a biological network which relates to melanoma. The concentration level are given by 0 corresponding to low concentration and 1 corresponding to high concentration. Because the gene pirin is a control input u, we may generate a BCN with dynamics $f_d^{(i)}$ for all genes x_i as follows:

$$\begin{aligned} \text{pirin} : x_1' &= \neg x_5, \ \text{MART1} : x_4' = \neg x_6 \vee a, \\ \text{S100P} : x_2 &= \neg x_6, \text{HADHB} : x_5' = x_2 \vee x_3, \\ \text{RET1} : x_3' &= x_3, \ \text{STC2} : x_6' = x_6 \vee \neg a. \end{aligned}$$

The following update rules transform this BCN to a PBCN by introducing random perturbations with a probability p

$$x_i' = \begin{cases} f_d^{(i)}(x,u), & \text{with a probability of p} \\ 0, & \text{with a probability of (1-p)/2} \\ 1. & \text{with a probability of (1-p)/2} \end{cases} \quad (35)$$

WNT5A expression is directly correlated to the creation of melanoma. As a result, the control goal is to stop WNT5A from expressing at the end of a specified time horizon t_f of dynamic system evolution. Assigning cost $q(x_1) = 1$ for WNT5A and $q(x_i) = 0$ for all other genes, as well as assigning terminal cost $q_{t_f}(x_1) = 10$ for WNT5A and $q_{t_f}(x_i) = 0$ for all other genes, the control objective in mathematical form is

$$J(X) = \min_u \mathbb{E}_{X' \sim P(\cdot|X,u)} \Big[\sum_{k=0}^{t_f-1} \big(x_1(k) + u(k)\big) + 10\, x_1(t_f) \Big].$$

The optimal solution is obtained for two sets of function selection probabilities and random perturbation for the starting state $[1\ 1\ 0\ 1\ 0\ 0\ 0]^T$ for 5 time steps, i.e., $t_f = 5$. The control input sequence for $p = 0.5$ is

$$\{u(0) = 1, u(1) = 1, u(2) = 0, u(3) = 0, u(4) = 0\},$$

while for $p = 0.95$ is

$$\{u(0) = 0, u(1) = 0, u(2) = 0, u(3) = 0, u(4) = 0\}.$$

From this result we can clearly see that for high degree of uncertainty, the optimal choice is to avoid the pirin expression. For all initial states and selection probability of $p = 0.8$, we show the estimated cost with application of control and without control in FIGURE 8.

TABLE 1. Comparative Analysis of Sequential Computing and Parallel Computing for Information-theoretic Control of T-cell Signaling Networks for Finite time ($t_f = 5$).

Total samples in thousand	Sequential computing time in sec	Parallel computing time in sec	Improvement factor in time
5.12	36.63	0.23	159.26
10.24	72.37	0.26	278.34
51.2	483.30	0.77	627.66
102.4	960.14	1.37	700.83
512	4693.02	6.20	756.93
1024	9646.89	12.11	796.60

4) 37-GENE BIOLOGICAL T-CELL SIGNALING NETWORK

To illustrate the scalability of the developed method, a biological T-cell signaling model [51] is used. With probability $p = 0.99$, let the T-cell network of 37 genes and 3 control variables follow its Boolean dynamics as shown in TABLE 2. With a probability of $0.99^{37} \approx 0.69$, the system is likely to follow the dynamics, implying a high level of uncertainty (approximately 31%). The optimal control problem is solved to avoid the expression of genes Calcin, DAG, NFkB, Ras, Rlk after 5 time steps. Moreover, the system should avoid the expression of genes DAG, Gads, Calcin, Fos, IKKbeta, JNK, Lck, NFAT, PLCg(act), Rsk, SLP76, Ras for states visited in-between. The genes CD45, CD8, and TCRlig are used as control variables to perform aforementioned task which can be translated to the objective function comprising of the state cost $q(X) = C_q \times [x_1(t) \ldots x_n(t)]^T$, terminal cost $\phi(X) = C_\phi \times [x_1(t_f) \ldots x_n(t_f)]^T$, and control cost $g(U) = [0.01\ 0.05\ 0.02] \times [u_1(t)\ u_2(t)\ u_3(t)]^T$.

$$\begin{aligned} C_q = [\ & 1.93\ 1.33\ 0.0\ \ 1.13\ 2.06\ 0.54\ 0.0\ \ 0.95\ 0.0\ \ 1.02 \\ & 0.0\ \ 0.37\ 0.0\ \ 1.97\ 1.33\ 2.31\ 0.0\ \ 1.54\ 2.48\ 0.0 \\ & 0.96\ 0.0\ \ 2.73\ 0.27\ 0.0\ \ 0.72\ 2.73\ 0.43\ 0.0\ \ 1.37 \\ & 2.78\ 0.0\ \ 1.73\ 0.0\ \ 1.72\ 2.19\ 1.96\ ] \end{aligned}$$

$$\begin{aligned} C_\phi = [\ & 9.84\ 5.55\ 0.0\ \ 6.23\ 0.0\ \ 1.07\ 8.19\ 8.53\ 1.23\ 5.34 \\ & 3.95\ 6.92\ 2.24\ 6.25\ 7.75\ 8.76\ 3.76\ 4.74\ 8.8\ \ 4.94 \\ & 3.31\ 0.4\ \ 0.0\ \ 5.06\ 7.87\ 0.65\ 7.53\ 4.74\ 0.0\ \ 7.54 \\ & 9.57\ 7.91\ 1.24\ 0.0\ \ 7.69\ 3.31\ 4.97\ ]. \end{aligned}$$

Because evaluating the transition probability matrix is impractical in this case, we utilize the matrix-free technique to simulate the dynamics of Boolean network (Algorithm 1) followed by the information-theoretic approach to determine the optimal solution. Using the suggested Algorithm 2, results are compared between sequential implementation and GPU-based parallel implementation. The control inputs and corresponding trajectory starting from an initial state $X_I(0) = 98024258941$ with 4.096×10^8 number of samples are given

TABLE 2. T-cell receptor Dynamics.

Gene	variable	Boolean Function	Gene	variable	Boolean Function
AP1	x_1'	$x_9 \wedge x_{18}$	LAT	x_{19}'	x_{37}
Ca	x_2'	x_{14}	Lck	x_{20}'	$\neg x_{26}) \wedge a_1 \wedge a2$
Calcin	x_3'	x_2	MEK	x_{21}'	x_{28}
cCbl	x_4'	x_{37}	NFAT	x_{22}'	x_3
CRE	x_5'	x_6	NFkB	x_{23}'	$\neg x_{16}$
CREB	x_6'	x_{32}	PKCth	x_{24}'	x_7
DAG	x_7'	x_{25}	PLCg(act)	x_{25}'	$(x_{15} \wedge x_{27} \wedge x_{34} \wedge x_{37}) \vee (x_{27} \wedge x_{31} \wedge x_{34} \wedge y_{37})$
ERK	x_8'	x_{21}	PAGCsk	x_{26}'	$x_{10} \vee (\neg x_{35})$
Fos	x_9'	x_8	PLCg(bind)	x_{27}'	x_{19}
Fyn	x_{10}'	$(x_{20} \wedge a_2) \vee (x_{35} \wedge a_2)$	Raf	x_{28}'	x_{29}
Gads	x_{11}'	x_{19}	Ras	x_{29}'	$x_{12} \vee x_{30}$
Grb2Sos	x_{12}'	x_{19}	RasGRP1	x_{30}'	$x_7 \wedge x_{24}$
IKKbeta	x_{13}'	x_{24}	Rlk	x_{31}'	x_{20}
IP3	x_{14}'	x_{25}	Rsk	x_{32}'	x_8
Itk	x_{15}'	$x_{34} \wedge x_{37}$	SEK	x_{33}'	x_{24}
IkB	x_{16}'	$\neg x_{13}$	SLP76	x_{34}'	x_{11}
JNK	x_{17}'	x_{33}	TCRbind	x_{35}'	$(\neg x_4) \wedge a_3$
Jun	x_{18}'	x_{17}	TCRphos	x_{36}'	$x_{10} \vee (x_{20} \wedge x_{35})$
CD8	a_2	input	ZAP70	x_{37}'	$(\neg x_4) \wedge x_{20} \wedge x_{36}$
CD45	a_1	input	TCRlig	a_3	input

as follows

$$U^*(0 \to t_f - 1) = [5\ 5\ 1\ 1\ 1],$$
$$X_I(0 \to t_f) = [98024258941\ 42441672659\ 56104651553\ 88586447361\ 5119257649\ 5941237809].$$

The state encountered at the terminal time t_f is 5941237809 and its Boolean equivalent is given by $00\bar{0}00\bar{1}0\bar{1}10001000\bar{1}0000000\bar{1}10\bar{0}00\bar{0}110001$. It can be clearly observed from the Boolean representation of state at $t_f = 5$, the gene expression for Calcin(x_3), DAG(x_7), NFkB(x_{23}), Ras(x_{29}), Rlk(x_{31}) is downregulated (shown by bar on the Boolean number.) According to TABLE 1, the time needed to execute the proposed algorithm in sequential manner increases dramatically with the number of samples (Column 2), but the time required to accomplish the same operation using GPU-based parallel processing increases moderately with the number of samples (Column 3).

B. DISCUSSION

Despite the richness and elegance of the dynamic programming (DP) given in [28], solving the Bellman equation for most practical problems is intractable in computational sense. This is due to the fact that the value function must be recorded for each state, and indeed the number of states in PBCNs increases exponentially. As a result, a number of approximation strategies based on solving the Bellman equation approximately have emerged. The policy iteration [34], value iteration [46], [52], and Q-learning [35], [53] are all common approximation approaches in the reinforcement learning field. The policy iteration has been shown to be superior than the value iteration in that it delivers the optimal stationary policy in a finite number of steps, whereas the value iteration may require an infinite amount of steps for convergence [46]. Q-learning [53] is the other technique used, and the learning time grows as the number of genes increases. As a consequence, the authors remark that the Q-learning technique might not always scale to large networks [53]. Furthermore, some of the approaches are matrix-based [34], which operate directly upon matrices and therefore are not suitable for large PBCNs because to memory constraints.

The fundamental advantage of our proposed framework is that, given the cost function, minimization of objective may be executed analytically. We derive the linear Bellman equation, which allows for forward in-time simulation with parallel computing of the cost function evaluation, which is not possible with iterative approaches due to iteration interdependence. Our method could readily be extended to a model-free approach, in which the transition probabilities are learned from the data rather than calculated from known network dynamics.

The effectiveness of optimal control methods proposed in the literature was validated through the use of biological networks such as the 7-gene WNT5A network [29], [33], [46], [52], the 13-gene (9 state, 4 control) ARA OPERON network [34], and the 8-gene artificial network [54], among others. Neither approximation nor analytical PBCN optimal control approaches have been validated using a large biological network of 40 genes (37 states, three control), to the best of the authors' knowledge. The results of applying information-theoretic PBCN optimal control to a T-cell signaling network demonstrate the scalability of the method with large PBCNs.

VII. CONCLUSION

For PBCN, which is set in a traditional MDP form, an information-theoretic optimal control formulation is developed by adopting the stochastic optimal control theory. A nonlinear control problem is transformed into a linear problem using the proposed solution method. The resultant formulation is solved analytically using a matrix-free technique, allowing large systems to be solved. The suggested

framework's scalability is validated through the use of GPU-based parallel processing for speedy estimate of the desirability function. The methodology described is general and can be applied to a variety of PBCN optimal control tasks. The proposed method is not confined to a specific type of distribution and could also be used to regulate context-sensitive PBCN. Cross-fertilization of a concept from stochastic optimal control and MDP can be used to develop the receding horizon approach. The method presented in this research is not limited to PBCN and can be used to other sequential decision-making issues including uncertainty. The methodology proposed is a model-based approach that begins with the PBCN model of gene regulation networks. We intend to investigate the state-of-the-art machine learning techniques to create a PBCN model using time-course gene expression data and solve the optimal control problem.

REFERENCES

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

[2] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2001.

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

[4] E. Fornasini and M. E. Valcher, "Observability and reconstructibility of probabilistic Boolean networks," *IEEE Control Syst. Lett.*, vol. 4, no. 2, pp. 319–324, Apr. 2020.

[5] R. Zhou, Y. Guo, and W. Gui, "Set reachability and observability of probabilistic Boolean networks," *Automatica*, vol. 106, pp. 230–241, Aug. 2019.

[6] Z. Jing and L. Zhenbin, "Observability of probabilistic Boolean networks," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 183–186.

[7] F. Li and J. Sun, "Controllability of probabilistic Boolean control networks," *Automatica*, vol. 47, no. 12, pp. 2765–2771, 2011.

[8] J. Wang, Y. Liu, and H. Li, "Finite-time controllability and set controllability of impulsive probabilistic Boolean control networks," *IEEE Access*, vol. 8, pp. 111995–112002, 2020.

[9] A. Yerudkar, C. Del Vecchio, N. Singh, and L. Glielmo, "Reachability and controllability of delayed switched Boolean control networks," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 1863–1868.

[10] T. Leifeld, Z. Zhang, and P. Zhang, "Fault detection for probabilistic Boolean networks," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2016, pp. 740–745.

[11] Y. Guo, R. Zhou, Y. Wu, W. Gui, and C. Yang, "Stability and set stability in distribution of probabilistic Boolean networks," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 736–742, Feb. 2019.

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

[13] A. Yerudkar, C. D. Vecchio, and L. Glielmo, "Sampled-data set stabilization of switched Boolean control networks," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6139–6144, 2020.

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

[15] L. Tong, Y. Liu, Y. Li, J. Lu, Z. Wang, and F. E. Alsaadi, "Robust control invariance of probabilistic Boolean control networks via event-triggered control," *IEEE Access*, vol. 6, pp. 37767–37774, 2018.

[16] T. Akutsu and A. A. Melkman, "Identification of the structure of a probabilistic Boolean network from samples including frequencies of outcomes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 8, pp. 2383–2396, Aug. 2019.

[17] I. Apostolopoulou and D. Marculescu, "Tractable learning and inference for large-scale probabilistic Boolean networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2720–2734, Sep. 2019.

[18] A. Yerudkar, C. Del Vecchio, and L. Glielmo, "Output tracking control design of switched Boolean control networks," *IEEE Control Syst. Lett.*, vol. 4, no. 2, pp. 355–360, Apr. 2020.

[19] A. Yerudkar, C. D. Vecchio, and L. Glielmo, "Output tracking control of probabilistic Boolean control networks," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 2109–2114.

[20] L. Wang, T. Feng, J. Song, Z. Guo, and J. Hu, "Model checking optimal infinite-horizon control for probabilistic gene regulatory networks," *IEEE Access*, vol. 6, pp. 77299–77307, 2018.

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

[22] M. Toyoda and Y. Wu, "On optimal time-varying feedback controllability for probabilistic Boolean control networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2202–2208, Jun. 2020.

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

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

[25] Q. Liu, Y. He, and J. Wang, "Optimal control for probabilistic Boolean networks using discrete-time Markov decision processes," *Phys. A, Stat. Mech. Appl.*, vol. 503, pp. 1297–1307, Aug. 2018.

[26] R. Dehghannasiri, M. S. Esfahani, and E. R. Dougherty, "An experimental design framework for Markovian gene regulatory networks under stationary control policy," *BMC Syst. Biol.*, vol. 12, no. S8, pp. 5–20, Dec. 2018.

[27] B. Faryabi, G. Vahedi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, "Optimal constrained stationary intervention in gene regulatory networks," *EURASIP J. Bioinf. Syst. Biol.*, vol. 2008, no. 1, 2008, Art. no. 620767.

[28] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Mach. Learn.*, vol. 52, nos. 1–2, pp. 169–191, 2003.

[29] O. Wei, Z. Guo, Y. Niu, and W. Liao, "Model checking optimal finite-horizon control for probabilistic gene regulatory networks," *BMC Syst. Biol.*, vol. 11, no. S6, p. 104, Dec. 2017.

[30] A. Acernese, A. Yerudkar, L. Glielmo, and C. D. Vecchio, "Double deep-Q learning-based output tracking of probabilistic Boolean control networks," *IEEE Access*, vol. 8, pp. 199254–199265, 2020.

[31] B. Faryabi, A. Datta, and E. R. Dougherty, "On reinforcement learning in genetic regulatory networks," in *Proc. IEEE/SP 14th Workshop Stat. Signal Process.*, Aug. 2007, pp. 11–15.

[32] A. Acernese, A. Yerudkar, L. Glielmo, and C. Del Vecchio, "Model-free self-triggered control co-design for probabilistic Boolean control networks," *IEEE Control Syst. Lett.*, vol. 5, no. 5, pp. 1639–1644, Nov. 2021.

[33] G. Papagiannis and S. Moschoyiannis, "Deep reinforcement learning for control of probabilistic Boolean networks," in *Proc. Int. Conf. Complex Netw. Their Appl.* Madrid, Spain: Springer, 2020, pp. 361–371.

[34] Y. Wu, Y. Guo, and M. Toyoda, "Policy iteration approach to the infinite horizon average optimal control of probabilistic Boolean networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 2910–2924, Jul. 2021.

[35] A. Acernese, A. Yerudkar, L. Glielmo, and C. Del Vecchio, "Reinforcement learning approach to feedback stabilization problem of probabilistic Boolean control networks," *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 337–342, Jan. 2021.

[36] P. Bajaria, A. Yerudkar, and C. Del Vecchio, "Aperiodic sampled-data stabilization of probabilistic Boolean control networks: Deep Q-learning approach with relaxed Bellman operator," in *Proc. Eur. Control Conf. (ECC)*, 2021, pp. 836–841.

[37] W. H. Fleming and R. W. Rishel, *Deterministic and Stochastic Optimal Control*, vol. 1. New York, NY, USA: Springer-Verlag, 2012.

[38] R. F. Stengel, *Stochastic Optimal Control: Theory and Application*. New York, NY, USA: Wiley, 1986.

[39] H. J. Kappen, "Path integrals and symmetry breaking for optimal control theory," *J. Stat. Mech., Theory Exp.*, vol. 2005, no. 11, Nov. 2005, Art. no. P11011.

[40] H. J. Kappen and H. C. Ruiz, "Adaptive importance sampling for control and inference," *J. Statist. Phys.*, vol. 162, no. 5, pp. 1244–1266, 2016.

[41] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1603–1622, Dec. 2018.

[42] E. A. Theodorou and E. Todorov, "Relative entropy and free energy dualities: Connections to path integral and KL control," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 1466–1473.

[43] E. Todorov, "Linearly-solvable Markov decision problems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1369–1376.
[44] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. New York, NY, USA: Academic, 1976.
[45] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
[46] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite-horizon control for probabilistic Boolean networks," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2375–2387, Jun. 2006.
[47] E. Todorov, "Efficient computation of optimal actions," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 28, pp. 11478–11483, Jul. 2009.
[48] T. D. Han and T. S. Abdelrahman, "HiCUDA: High-level GPGPU programming," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 1, pp. 78–90, Jan. 2011.
[49] B. Daga, A. Bhute, and A. Ghatol, "Implementation of parallel image processing using NVIDIA GPU framework," in *Proc. Int. Conf. Adv. Comput., Commun. Control*. Berlin, Germany: Springer, 2011, pp. 457–464.
[50] K. Kobayashi and K. Hiraishi, "Optimization-based approaches to control of probabilistic Boolean networks," *Algorithms*, vol. 10, no. 1, p. 31, Feb. 2017. [Online]. Available: https://www.mdpi.com/1999-4893/10/1/31
[51] S. Klamt, J. Saez-Rodriguez, J. A. Lindquist, L. Simeoni, and E. D. Gilles, "A methodology for the structural and functional analysis of signaling and regulatory networks," *BMC Bioinf.*, vol. 7, no. 1, pp. 1–26, Dec. 2006.
[52] R. Layek, A. Datta, R. Pal, and E. R. Dougherty, "Adaptive intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 25, no. 16, pp. 2042–2048, Aug. 2009.
[53] B. Faryabi, A. Datta, and E. R. Dougherty, "On approximate stochastic control in genetic regulatory networks," *IET Syst. Biol.*, vol. 1, no. 6, pp. 361–368, Nov. 2007.
[54] Q. Liu, X. Guo, and T. Zhou, "Optimal control for probabilistic Boolean networks," *IET Syst. Biol.*, vol. 4, no. 2, pp. 99–107, Mar. 2010.

K. SONAM (Member, IEEE) received the B.Tech. degree in electronics engineering from Shivaji University, India, in 2013, and the M.Tech. degree in control systems from the Veermata Jijabai Technological Institute, Mumbai, India, in 2017, where she is currently pursuing the Ph.D. degree in electrical engineering. Her current research interests include stochastic control, optimal control, reinforcement learning, control, and optimization of biological systems and power systems under uncertainty.

SARANG SUTAVANI received the B.E. degree in electronics engineering and the M.Tech. degree in control systems from Mumbai University, India, in 2013 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Department of Mechanical Engineering, Clemson University, USA. His current research interests include reinforcement learning, cyber security, Boolean networks, and stochastic control.

S. R. WAGH (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from The University of Western Australia, Perth, WA, Australia, in 2012. From 2015 to 2016, she was a Visiting Scholar with Tufts University, Medford, MA, USA. She is currently an Assistant Professor with the Veermata Jijabai Technological Institute, Mumbai, India. Her current research interests include power system dynamics, stability and control, and smart grid.

N. M. SINGH received the Ph.D. degree in electrical engineering from the IIT Bombay, Mumbai, India, in 1990. He is currently an Adjunct Professor with the Veermata Jijabai Technological Institute, Mumbai. His current research interests include geometric control theory, complex networks, and stochastic control.