Adaptive Intervention in Probabilistic Boolean Networks

Ritwik Layek[a], Aniruddha Datta[a], Ranadip Pal[b], Edward R. Dougherty [a,c]

Abstract— **Probabilistic Boolean Networks (PBNs) have been recently introduced as a paradigm for modeling genetic regulatory networks. One of the objectives of PBN modeling is to use the network for the design and analysis of intervention strategies aimed at moving the network out of undesirable states, such as those associated with disease, and into desirable ones. To date, a number of intervention strategies have been proposed in the context of PBNs. However, most of these techniques assume perfect knowledge of the transition probability matrix of the PBN. Such an assumption cannot be satisfied in practice, and may lead to degraded, if not completely unacceptable, performance. To remedy the situation, one can adopt one of two main approaches:(i) design an intervention strategy that is "robust" or somewhat insensitive to the presence of a class of modeling errors, such as uncertainties in the transition probability matrix; or (ii) introduce on-line adaptation or learning into the intervention strategy to ensure satisfactory performance provided the modeling error belongs to a particular class. The first approach has already been developed in an earlier paper. The main goal of this paper is to demonstrate the feasibility of the second approach. Using simulation studies, it is shown that adaptive intervention works well in two different scenarios: first, when we have a family of PBNs whose individual transition probability matrices are reasonably well modeled and the predominant uncertainty is about which member of that family represents the underlying genetic regulatory network; and second, when we have a context sensitive PBN with a low probability of a context change so that there is sufficient time between context changes for the adaptive algorithm to learn the context and exploit it in the intervention design. These results agree quite well with intuitive expectations.**

I. INTRODUCTION

The sequencing of various genomes over the last decade has given a remarkable boost to genomic studies. The improved understanding of the genomes of various organisms, along with advances in microarray technology, have provided us with enormous opportunities for the mathematical modeling of biological networks. There are two major objectives for modeling of genetic regulatory networks: (i) first, to better understand the intergene interactions and relationships on a holistic level, thereby facilitating the diagnosis of disease; and (ii) second, to design and analyze therapeutic intervention strategies for shifting the state of a diseased network from an undesirable location to a desirable one. Many different approaches have been proposed in the literature for modeling the behaviour of genetic regulatory networks. These include linear models, Bayesian networks, neural networks, differential equations, Boolean networks (BNs), and their stochastic generalizations, the Probabilstic Boolean networks (PBNs). Of these, the model which has received the most attention, at least in the context of therapeutic intervention, is the Probabilistic Boolean network model.

To date, a number of approaches have been proposed in the literature for carrying out interventions in Probabilistic Boolean networks [1]. Most of these approaches are based on stochastic optimal control theory for Markov chains and, therefore, assume perfect knowledge of the underlying PBN. Such an assumption cannot be satisfied in practice, and may lead to degraded, if not completely unacceptable, performance. To remedy the situation, one could design a fixed intervention strategy that is "robust" or somewhat insensitive to the presence of a class of modeling errors. Such a design has been considered in [2] where the effect of uncertainties in the transition probability matrix of a PBN on the final intervention outcome was analyzed, and a robust intervention scheme, optimal for the worst-case uncertainty, was developed. In this approach, the intervention strategy is a fixed one and not "tuned" to the actual PBN and consequently, one can expect to get better performance if learning or "on-line" adaptation can be introduced into the intervention design. The aim of this paper is to demonstrate the feasibility of such an adaptive approach. At the very outset, it is important to point out that such a scheme will be feasible only if the uncertainty belongs to a certain class, and prior knowledge about this class can be incorporated into the design. In other words, improved performance results from exploiting prior structural knowledge about the uncertainty.

The paper is organized as follows. In Section II, we formally define Boolean networks and Probabilistic Boolean Networks and derive transition probability expressions for the latter under two different scenarios. In Section III, we introduce notation and quickly recall some earlier results on infinite horizon optimal control for PBNs for the case when the transition probabilities are exactly known. In Section IV, we present an adaptive intervention algorithm which can be used to improve performance when there is structured uncertainty about the transition probabilities of the underlying PBN. Section V presents simulations to demonstrate the performance of the adaptive scheme when applied to a toy example. Finally, Section VI contains some concluding remarks.

This research was supported in part by the Translational Genomics Research Institute, in part by the National Science Foundation under Grants ECS0355227, CCF0514644, ECCS0701531 and in part by the National Cancer Institute under Grant CA90301.

[a] Texas A & M University, Electrical and Computer Engineering, College Station, TX, 77843, USA. [b] Department of Electrical and Computer Engineering, Texas Tech University, Lubbock, TX,79410,USA [c] Translational Genomics Research Institute, Phoenix, AZ, 85004, USA

978-1-4244-4524-0/09/$25.00 ©2009 AACC

II. BNs AND PBNs

In this section, we briefly define Boolean Networks and Probabilistic Boolean networks. For a more detailed and motivated development, the reader is referred to [1].

A *Boolean Network* (BN) $B = (V, F)$ on n genes is defined by a set of nodes/genes $V = \{x_1, ..., x_n\}$, $x_i \in \{0, 1\}$, $i = 1, ..., n$, and a list $F = (f_1, ..., f_n)$, of Boolean functions, $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$, $i = 1, ..., n$. Each node x_i represents the state/expression of the gene x_i, where $x_i = 0$ means that gene i is OFF and $x_i = 1$ means that gene i is ON. The function f_i is called the *predictor function* for gene i. Updating the states of all genes in B is done synchronously at every time step according to their predictor functions. A *Probabilistic Boolean Network* (*PBN*) consists of a set of nodes/genes $V = \{x_1, ..., x_n\}$, $x_i \in \{0, 1, ..., Y\}$, $i = 1, ..., n$, $Y \in \mathbb{N}$ and a set of vector valued network functions, $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$, governing the state transitions of the genes, each network function being of the form $\mathbf{f}_j = (f_{j1}, f_{j2}, ..., f_{jn})$, where $f_{ji} : \{0, 1, ..., Y\}^n \rightarrow \{0, 1, ..., Y\}$, $i = 1, ..., n$. In most applications, the discretization is either binary or ternary and in this paper we will use binary, i.e. $Y = 1$. The choice of which network function $\mathbf{f}_j$ to apply is governed by a selection procedure. Specifically, at each time point a random decision is made as to whether to switch the network function for the next transition, with the probability q of a switch being a system parameter. If a decision is made to switch the network function, then a new function is chosen from among $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$, with the probability of choosing $\mathbf{f}_j$ being the selection probability c_j. In other words, each network function $\mathbf{f}_j$ determines a BN and the PBN behaves as a fixed BN until a random decision (with probability q) is made to change the network function according to the probabilities $c_1, c_2, ..., c_k$ from among $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$. The PBN just described is called a *context-sensitive PBN*. In the special case when $q = 1$, the network function is switched at every time point and the PBN is called an *instantaneously random PBN*. We consider context-sensitive PBNs with perturbation, meaning that at each time point there is a probability p of any gene flipping its value uniformly randomly. Since there are n genes, the probability of there being a random perturbation at any time point is $1 - (1 - p)^n$. The state space S of the network together with the set of network functions, in conjunction with transitions between the states and network functions, determines a Markov chain. The random perturbation makes the Markov chain irreducible, meaning that it has the possibility of reaching any state from any other state and that it possesses a steady-state distribution.

The state vector $x(t)$ at any time step t is essentially an n-digit binary number $[x_1 x_2 \cdots x_n]$ whose decimal equivalent is given by

$$z(t) = \sum_{j=1}^{n} 2^{n-j} x_j(t). \qquad (1)$$

As $x(t)$ ranges from $000 \cdots 0$ to $111 \cdots 1$, $z(t)$ takes on all values from 0 to $2^n - 1$. Instead of the vector $x(t)$, one could equivalently consider $z(t)$ to be the state of the network at time t so that the new state space becomes $S = \{0, 1, 2, \cdots, 2^n - 1\}$. We next proceed to derive the transition probability expressions for a context sensitive PBN.

A. Computation of Transition Probabilities for a Context-sensitive PBN

Consider a context-sensitive PBN that is assumed to be completely known. In other words, we know the perturbation probability p, switching probability q, the truth tables of the constituent Boolean Networks and the network selection probabilities c_i, $i = 1, 2, \cdots, k$. We will assume that the following mutually exclusive sequence of events can occur at any time point t: (1)The current network function is applied, the PBN transitions accordingly, and the network function remains the same for the next transition; (2)The current network function is applied, the PBN transitions accordingly, and a new network function (which can be the same as the original one) is selected for the next transition; (3)There is at least one random perturbation and the network function remains the same for the next transition; (4) There is a random perturbation and a new network function (which can be the same as the original one) is selected for the next transition. In (2) and (4) above, we are allowing the possibility that the same network can be selected after a network switch. Thus, if the PBN is currently in Boolean network r ($r \in \{1, 2, \cdots, k\}$), then it will stay in Boolean network r with a probability $1 - q(1 - c_r)$ and transition to some other Boolean network s with probability qc_s. Because of this, the transition probability expressions derived here will be a little different than those obtained in [1] where the underlying assumption was that a network switch would necessarily mean that the BN would have to change. This subtle difference, however, does not affect the qualitative nature of the results in this paper or, for that matter, in the earlier literature.

Clearly, each Boolean Network will have 2^n states $00 \cdots 0$ to $11 \cdots 1$ and the collection of k BNs can be considered to have a set of $2^n k$ states. Let $w(t) \in \{0, 1, 2, \cdots, 2^n k - 1\}$ be the state that is occupied by the network at time t. This means that at time t, the $[w(t)/2^n]^{th}$ BN is used and the current state or gene activity profile in that network is $mod(w(t), 2^n)$. Let $a, b \in \{0, 1, 2, \cdots, 2^n k - 1\}$ be any two states and define $r_1 = [a/2^n]$, $r_2 = [b/2^n]$, $a^{'}$=$mod(a, 2^n)$ and $b^{'}$= $mod(b, 2^n)$. Then it is straight forward to show that the one-step transition probability from a to b is given by:

$$\begin{aligned} & Pr(w(t+1) = b | w(t) = a) = \\ & [g(a,b)\{1 - q + qc_{r_1}\} + \{1 - g(a,b)\} qc_{r_2}] \\ & \times \left[(1-p)^n f_{r_1,a,b} + (1-p)^{n-h} p^h s(h)\right] \qquad (2) \end{aligned}$$

where, h is the Hamming distance between $a^{'}$ and $b^{'}$ represented in binary digits, i.e., the number of genes that differ between the two states,

$$f_{r,a,b} = \begin{cases} 1 & \text{if } a^{'} \text{ transitions to } b^{'} \text{ in 1 step in network } r, \\ 0 & \text{otherwise,} \end{cases}$$

$$g(a,b) = \left\{ \begin{array}{ll} 1 & \text{if } r_1 = r_2 = r \text{ (say)} \\ 0 & \text{otherwise} \end{array} \right.$$

and

$$s(h) = \left\{ \begin{array}{ll} 0 & \text{if } h = 0 \\ 1 & \text{otherwise.} \end{array} \right.$$

The transition probability expression (2) can be used to track the time evolution of the extended state which contains both context as well as gene activity profile information. However, in practice it will be almost impossible to detect the BN from which the current gene activity profile is being emitted. In most cases, we will have knowledge only of the expression status of the genes and not of the context. This motivates the development of an expression for the transition probabilities between the different gene activity profiles by taking the expectation of the extended state transition probabilities over the networks. To do so, let $z(t)$ be the decimal equivalent of the binary gene activity profile at time t. Then for any $s_1, s_2 \in S$, the averaged out transition probability is given by

$$Pr[z(t+1) = s_1 | z(t) = s_2] = \sum_{i=1}^{k}\sum_{j=1}^{k} c_i . Pr[w(t+1) = s_1 + 2^n(j-1) | w(t) = s_2 + 2^n(i-1)]. \quad (3)$$

Using the above equations we can compute the entire transition probability matrix of size $2^n \times 2^n$ corresponding to the averaged context sensitive PBN. As shown in [2], the transition probability matrix for an averaged context sensitive PBN is the same as that of an instantaneously random PBN that makes use of the same constituent Boolean networks.

It is possible that some of the transition probabilities computed using (3) may evaluate out to zero. The corresponding transitions are referred to as *forbidden transitions* and the adaptive algorithms to be presented in this paper require that the set F of such forbidden transitions be known.

The transition probability expressions derived in this subsection allow for the possibility of different selection probabilities for the different constituent boolean networks of a PBN. However, in the absence of any prior knowledge, we will henceforth assume a uniform distribution of the selection probabilities, i.e. $c_i = \frac{1}{k},\ i = 1, 2, \cdots, k$.

III. INFINITE HORIZON CONTROL ASSUMING PERFECT MODELING

In this section, we briefly summarize some results on the infinite horizon control of PBNs, assuming that the modeling is perfect. These results serve as a stepping stone for deriving adaptive intervention strategies when perfect modeling is not possible and the modeling error belongs to an appropriate class. We begin by recalling some facts and notation from [1]. As shown in [1], a PBN with control can be modeled as a stationary discrete-time dynamic system[1]

$$z_{t+1} = f(z_t, u_t, w_t),\ t = 0, 1, ..., \quad (4)$$

where for all t, the state z_t is an element of a space S , the control input u_t is an element of a space C, the disturbance w_t, which captures the randomness due to different sources, is an element of a space D and $f : S \times C \times D \mapsto S$. In the particular case of PBNs of n genes composed of k Boolean networks with perturbation probability p and network transition probability q, $S = [0, 1, 2, ..., 2^n - 1]$ and the control input u_t is constrained to take values in the space $C = [0, 1, ..., 2^m - 1]$, where m is the number of control inputs. We next explain the basis for defining C as above. Suppose the PBN has m control inputs $v_1, v_2, \cdots, v_m$, each of which can take on only the binary values 0 or 1. Then at any given time step t, the row vector $v(t) \triangleq [v_1(t), v_2(t), \cdots, v_m(t)]$ describes the complete status of all the control inputs. $v(t)$ can take on all binary-vector values from $[0, 0, \cdots, 0]$ to $[1, 1, \cdots, 1]$. One can equivalently represent the control input status using the decimal number

$$u_t = \sum_{i=1}^{m} 2^{m-i} v_i(t). \quad (5)$$

As $v(t)$ takes on binary values from $[0, 0 \cdots, 0]$ to $[1, 1, \cdots, 1]$, the variable u_t ranges from 0 to $2^m - 1$. We can equivalently use u_t as an indicator of the complete control input status of the probabilistic Boolean network at time step t. Therefore, $C = [0, 1, 2, 3.......2^m - 1]$.

The disturbance w_t is manifested in terms of change of network based on the network switching probability q or change of state due to perturbation probability p. The random disturbances $w_t,\ t = 0, 1, ...$ have identical statistics and are characterized by probabilities $P(.|x_t, u_t)$ defined on D, where $P(.|x_t, u_t)$ is the probability of occurrence of w_t, when the current state and control are x_t and u_t, respectively. w_t is independent of prior disturbances $w_0, w_1....w_{t-1}$.

Another equivalent way to represent the dynamical system (4) is as a finite state *Markov Chain* described by the control-dependent one-step transition probability $p_{ij}(u)$ where for any $t = 0, 1, 2, \cdots; i, j \in S$ and $u \in C$,

$$p_{ij}(u) := P(z_{t+1} = j | z_t = i, u_t = u). \quad (6)$$

The infinite horizon optimal control problem for a PBN is formulated as the limit of the following *finite horizon* optimal control problem. Given an initial state z_0:

$$\min_{\mu_0, \mu_1, \cdots, \mu_{M-1}} E\left[\sum_{t=0}^{M-1} g_t(z_t, \mu_t(z_t)) + g_M(z_M)\right] \quad (7)$$

subject to $Pr(z_{t+1} = j | z_t = i, u_t = u) = p_{ij}(u)$, where

- M represents the treatment/intervention window;
- $\mu_t : S \rightarrow C,\ t = 0, 1, 2, \cdots, M-1$ are functions mapping the state space into the control space, i.e. the controls considered are state feedbacks;
- $g_t(z_t, u_t)$ is the one step cost of applying the control u_t at state z_t;
- and $g_M(z_M)$ is the terminal cost associated with the state z_M.

As discussed in [1], the consideration of such an optimization problem can be naturally motivated in the context of cancer

[1] In the rest of this paper, we will be denoting the time dependence of z, u and w by the subscript t. In all other situations, the context will make it clear whether a subscript denotes time dependence or reference to the particular component of a vector.

treatment applications where one must choose between a number of alternative treatments to be applied over a finite horizon of time. Once input from biologists/clinicians has been used to select an appropriate cost function and an appropriate treatment window, the control problem is essentially reduced to that of controlling a Markov Chain over a finite horizon.

By taking the limit of the cost function in (7) as $M \to \infty$, we obtain the *average cost per stage*:

$$J_\pi(z_0) = \lim_{M\to\infty} \frac{1}{M} E\{\sum_{t=0}^{M-1} \tilde{g}(z_t, \mu_t(z_t), w_t)\} \tag{8}$$

where $\tilde{g}(i,u,j)$ denotes the cost of going from state i to state j under control u, and $\pi = \{\mu_0, \mu_1, \cdots\}$ is the control policy. Let Π denote the set of all *admissible* policies. Then the optimal cost function J^*,which is independent of the initial state [3], is defined by

$$J^* = \min_{\pi\in\Pi} J_\pi(z), z \in S \text{ is arbitrary.} \tag{9}$$

As discussed in [3], once the model is perfectly known, the optimal cost function J^* and the *stationary policy* $u_t = \phi(z_t)$ attaining it can be determined by using the *policy iteration* algorithm.

IV. ADAPTIVE INFINITE HORIZON CONTROL

In this section we develop an adaptive intervention strategy that can be used in the absence of perfect information about the PBN. We will assume that the underlying genetic regulatory network is modeled by a member of a known finite family of PBNs, and we have no apriori knowledge about which member of that family models the actual network. In such a situation, the most natural thing to do is to try and estimate the model number on-line, and then use the method discussed in the last section to determine the corresponding controller. This is the underlying philosophy of adaptive control and a considerable amount of theoretical research effort has been dedicated towards showing that such *certainty equivalence* schemes can guarantee the required performance [4]. Our focus in this paper will be to demonstrate via simulations the feasibility of adaptive intervention in the context of genetic regulatory networks. We will use a variation of an adaptive control algorithm developed in [5] for unknown Markov Chains and the interested reader is referred to that paper for a motivated development of the scheme, along with the associated technical proofs of convergence. While the scheme in [5] attempts to estimate all entries of the transition probability matrix, our adaptive algorithm will estimate only the model number since our underlying assumption is that the transition probabilities of the PBN are completely determined, once we know the model number.

Suppose that the family of controlled PBNs is parametrized by the parameter $\alpha \in A$ where[2]

$$A := \left\{ \alpha : \sum_{j\in S} p(i,j,u,\alpha) = 1 \; \forall \; (i,u) \in S \times C \right\}. \tag{10}$$

Notice that the only constraint placed on A is that every element of A results in a set of bonafide transition probabilities. Furthermore, the cardinality $|A|$ of A determines the total number of possible PBNs. For each $\alpha \in A$, we can compute the uncontrolled transition probability matrix by using (3). In addition, for a given control gene, the rows of the *controlled* transition probability matrix can be determined quite easily as a linear transformation of the rows of the uncontrolled transition probability matrix. As shown in [2], this is a consequence of restricting the class of allowable interventions to the flipping of a chosen control gene.

We next present the adaptive control algorithm originally derived in [5] by maximizing a modified maximum likelihood criterion. For each $\alpha \in A$, let $J^*(\alpha)$ be the optimal long term average cost obtained for model α using the methods mentioned in the last section, and let $\phi(.,\alpha) : S \to C$ be the corresponding control law attaining it. We next define some functions and constants which will be useful in the description of the adaptive controller. Let the functions $f : R \to R$, $o : Z \to R$ and a constant m be defined as follows [5]:
f is a strictly monotonically increasing continuous function such that

$$f(\inf_{\alpha\in A}(J^*(\alpha)) > 0;$$

o is any function such that

$\lim_{t\to\infty} \frac{o(t)}{t^\theta}$ *is a positive finite number for some* $\theta \in (0,1)$;

and M *is any integer such that*

$$M > |S| + 1. \tag{11}$$

For our implementation purposes we may take f as the logarithmic function and $o(t)$ as the function $o(t) = 2\sqrt(t)$ for which $\theta = 0.5$. The value of m can be chosen to satisfy (11) depending on the cardinality of the state space.

The adaptive controller consists of two operations, estimation and control. The two operations are separate and the estimation is carried out only once every m time steps. We next separately describe the estimator and the controller.

- *Estimator:* At each time step $0, M, 2M, 3M, ..kM, (k+1)M..$ make the estimate of the parameter vector α using the following equations.

$$\hat{\alpha_t} := \underset{\alpha\in A}{argmax} \bar{D_t}(\alpha) \tag{12}$$

where

$$\bar{D_t}(\alpha) := K \prod_{(i,j,u)\in F^c} p(i,j,u,\alpha)^{n_t(i,j,u)} \tag{13}$$

[2] In this section, $p(i,j,u,\alpha)$ denotes $p_{ij}(u)$ when the model α has been selected.

,

$$K = \left[\frac{1}{f\left\{J^*(\alpha)\right\}} \right]^{o(t)} \quad (14)$$

and F^c is the complement of the set of forbidden transitions F which is assumed to be known apriori. These transitions correspond to zero values for $p(i, j, u, \alpha)$. In (13), $n_t(i, j, u)$ is defined as

$$n_t(i, j, u) = 1 + \sum_{s=0}^{t-1} \mathbf{1}(z_s = i, z_{s+1} = j, u_s = u) \quad (15)$$

and can be interpreted as measuring the number of times a transition occurs from i to j under control u. Here $\mathbf{1}(.)$ denotes the indicator function. Now, at time kM, knowing the parameter estimate $\hat{\alpha}_{kM}$, we can find the optimal cost function $J^*(\hat{\alpha}_{kM})$ and the optimal control law $\phi(z_t, \hat{\alpha}_{kM})$ which will be used for the next M time steps. The parameter estimate is kept constant at $\hat{\alpha}_{kM}$ between time steps kM and $(k+1)M - 1$.

- *Controller:* At each time t, the control applied is

$$u_t := \phi(z_t, \hat{\alpha}_t). \quad (16)$$

The optimal cost function and the optimal control law are determined using the techniques of the last section applied to the estimated model.

The adaptive algorithm presented here is based on the transition probability expression (3). Since this expression accurately models an instantaneously random PBN, it is only to be expected that performance degradation will occur as the value of q is reduced from 1 to 0. This will be borne out by our simulations in the next section.

From a practical point of view, the expectation is that the constituent Boolean networks of a PBN switch very infrequently. In other words, the value of q can be reasonably assumed to be very small. In such a scenario, one could consider each constituent Boolean network to be a possible model to be identified by the estimation algorithm. Although this increases the cardinality of the set of possible models, it is expected to result in improved performance especially since a small value of q means that the constituent networks will change very infrequently so that the estimation algorithm will have enough time to identify the current Boolean network. This will also be borne out by the simulation results in the next section.

V. SIMULATION RESULTS

In this section, we present simulations to demonstrate the efficacy of the adaptive intervention strategies that we have proposed. Such simulation studies are especially important since the theoretical results in [5] guarantee only almost sure convergence and, that too, in a Cesaro sense. We will consider a toy example and carry out simulation studies using two different adaptive intervention algorithms. In the first scheme, henceforth referred to as Algorithm 1 (or algo 1), the model space for the estimation algorithm is composed of all of the possible PBNs. In the second scheme, henceforth referred to as Algorithm 2 (or algo 2), the model space for the estimation algorithm is made up of all the constituent Boolean networks for all the possible PBNs.

We consider a 4-gene genetic regulatory network that is modeled by an unknown member of a known family of context-sensitive PBNs. We assume that the cardinality of this family is 7, and for each member in this family we have 4 constituent Boolean networks and $p = q = 0.01$. Since the gene expressions are binary, the cardinality of the state space becomes 16. Without any loss of generality, we assume that the first gene, i.e. the gene corresponding to the most significant bit (MSB) in the gene activity profile, is the gene that needs to be down-regulated, i.e. zeroed. In addition, we assume that the second gene is the control gene that can be flipped with $u = 1$ and $u = 0$ denoting the "flipping" and "no flipping" actions respectively. In order to adaptively intervene in such a network, we choose $M = 32$. In addition, the cost of control is assumed to be 0.5 and the states are assigned penalties as follows:

$$\tilde{g}(u, j) = \begin{cases} 5 & \text{if } u = 0 \text{ and MSB is 1 for state } j \\ 5.5 & \text{if } u = 1 \text{ and MSB is 1 for state } j \\ 0.5 & \text{if } u = 1 \text{ and MSB is 0 for state } j \\ 0 & \text{if } u = 0 \text{ and MSB is 0 for state } j \end{cases}$$

The penalty assignment is based on the fact that for infinite-horizon problems, there is no terminal penalty; instead, the cost $\tilde{g}(u, j)$, of transitioning to state j under control u, contains the penalties of each state. Since our objective is to down-regulate the MSB gene, a higher penalty is assigned for destination states having the MSB gene up-regulated. Also for a given MSB gene status for the destination state, a higher penalty is assigned when the control is active versus when it is not.

Figure 1 shows the convergence results obtained using algorithm 1. In each of the figures showing convergence, the top plot shows the estimated model and the actual model as functions of the estimation time steps. The x-axis is calibrated in terms of the number of estimation windows with each window being 32 time steps long. Similarly, the bottom plot in each of the convergence figures shows the cumulative average adaptive and non-adaptive costs (assuming perfect knowledge about the true model). From Figure 1, it is clear that the estimated model converges to the true model and the cumulative adaptive average cost converges to the cumulative non-adaptive average cost.

We next wish to examine how algorithm 1 performs when the true model is deterministically switched. Accordingly, we repeated the simulations with the actual model being switched from PBN2 (model number 2) to PBN6 (model number 6) at the 10th estimation window (time=320). The results are shown in Figure 2. From Figure 2, it is clear that the model estimate tracks the actual model quite well and the cumulative average cost also converges.

Figure 3 shows the simulation results obtained using algorithm 2 on the same network. Here, once again, the actual network switches from PBN1 to PBN4 at the 10th estimation

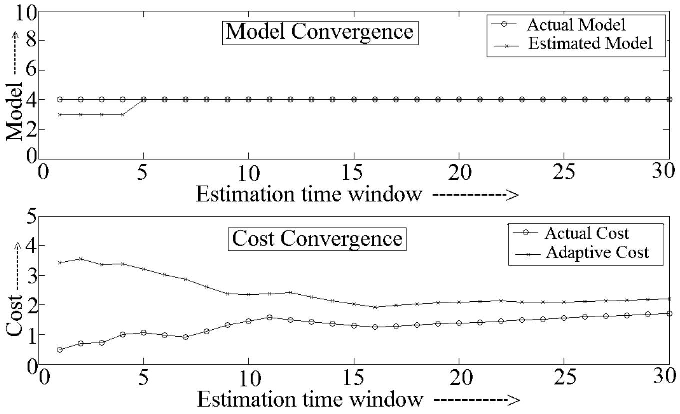

Fig. 1. Toy Example: Algorithm 1

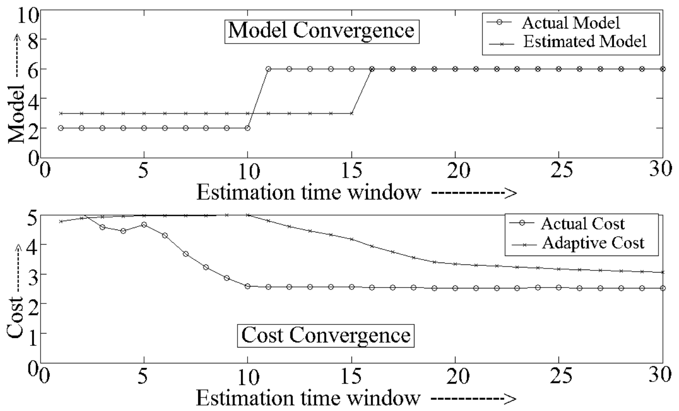

Fig. 2. Toy Example 2: Algorithm 1

window (time=320). Clearly, the estimated model converges to the true model and the cumulative adaptive average cost converges to the cumulative non-adaptive average cost for the true model. We notice that the estimated model convergence in this case is a lot faster than that obtained using algorithm 1. This is only to be expected since with $q = 0.01$, the underlying assumptions for algorithm 2 are a better fit to the real scenario.

VI. CONCLUDING REMARKS

In this paper, we have demonstrated the feasibility of applying adaptive intervention to improve performance in controlled genetic regulatory networks modeled by PBNs. Specifically we showed via simulations that when the genetic regulatory network is modeled by a member of a known family of PBNs, one can use adaptation and carry out a certainty equivalence design that leads to improved performance in terms of the cumulative average cost. These simulation studies are important since the theoretical results in the literature guarantee only almost sure convergence and, that too, in the Cesaro sense. We presented two different algorithms for model estimation, and argued that while one of the algorithms is well suited for instantaneously random PBNs, the other is much better for context-sensitive PBNs,

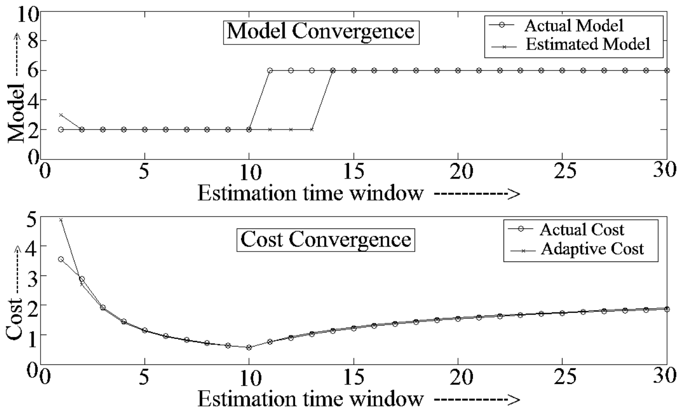

Fig. 3. Toy Example 2: Algorithm 2

with low switching probability between the constituent BNs. Our simulation results seem to confirm these intuitive expectations.

REFERENCES

[1] A. Datta and E.R. Dougherty, *Introduction to Genomic Signal Processing with Control*, CRC Press, 1 edition, 2006.

[2] R.Pal, A.Datta, and E.R.Dougherty, “Robust intervention in probabilistic boolean networks,” *IEEE Transaction on Signal Processing, VOL. 56, NO. 3*, pp. 1280–1294, March 2008.

[3] D.P.Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific,Belmont,Massachusetts, 3 edition, 2007.

[4] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification and Adaptive Control*, Prentice Hall, Englewood Cliffs, 1986.

[5] P.R.Kumar and Woei Lin, “Optimal adaptive controller for unknown markov chain,” *Transactions on Automatic Control,VOL.AC-27, NO.4*, pp. 765–774, August 1982.