BIOINFORMATICS

Vol. 00 no. 00
Pages 1–8

Adaptive Intervention in Probabilistic Boolean Networks

Ritwik Layek[1], Aniruddha Datta[1], Ranadip Pal[2], Edward R. Dougherty [1,3*]

[1] Texas A & M University, Electrical and Computer Engineering, College Station, TX, 77843, USA. [2] Department of Electrical and Computer Engineering, Texas Tech University, Lubbock, TX,79410,USA [3] Translational Genomics Research Institute, Phoenix, AZ, 85004, USA

ABSTRACT

Motivation: A basic problem of translational systems biology is to utilize gene regulatory networks as a vehicle to design therapeutic intervention strategies to beneficially alter network and, therefore, cellular dynamics. One strain of research has this problem from the perspective of control theory via the design of optimal Markov chain decision processes, mainly in the framework of probabilistic Boolean networks (PBNs). Full optimization assumes that the network is accurately modeled and, to the extent that model inference is inaccurate, which can be expected for gene regulatory networks owing to the combination of model complexity and a paucity of time-course data, the designed intervention strategy may perform poorly. We desire intervention strategies that do not assume accurate full-model inference.

Results: This paper demonstrates the feasibility of applying on-line adaptive control to improve intervention performance in genetic regulatory networks modeled by PBNs. It shows via simulations that when the network is modeled by a member of a known family of PBNs, an adaptive design can yield improved performance in terms of the average cost. Two algorithms are presented, one better suited for instantaneously random PBNs and the other better suited for context-sensitive PBNs with low switching probability between the constituent BNs.

Contact: edward@ece.tamu.edu

1 INTRODUCTION

There are two major objectives for modeling genetic regulatory networks: (i) to better understand inter-gene interactions and relationships on a holistic level, thereby facilitating the diagnosis of disease; and (ii) 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. Of these, the model which has received the most attention in the context of therapeutic intervention is the probabilistic Boolean network (PBN). To date, a number of approaches have been proposed for carrying out interventions in PBNs based on stochastic optimal control theory for Markov chains (Datta *et al.* (2003); Pal *et al.* (2005b, 2006)). These assume perfect knowledge of the underlying PBN, an assumption, when not satisfied in practice, can lead to degraded or unacceptable performance. To remedy the situation, one could design a fixed intervention strategy that is "robust", or somewhat insensitive, to modeling errors, in particular, to the effect of uncertainties in the transition probability matrix of a PBN (Pal *et al.* (2008)), Another approach is to "tune" the intervention strategy to the actual network via on-line adaptation. The aim of this paper is to demonstrate the feasibility of such an adaptive approach in the framework of PBNs. At the very outset, it is important to point out that such a scheme is feasible only if the uncertainty belongs to a specific class and prior knowledge about this class can be incorporated into the design.

*This research was supported in part by the National Science Foundation (ECCS-0701531).

2 SYSTEMS AND METHODS

2.1 Probabilistic Boolean Networks

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, $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$, $i = 1, ..., n$ (Kauffman (1993)). 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. At time t, the network state is given by $x(t) = (x_1(t), x_2(t), ..., x_n(t))$, called a *gene activity profile* (*GAP*). A *Probabilistic Boolean Network* (*PBN*) consists of a set of nodes/genes $V = \{x_1, ..., x_n\}$, $x_i \in \{0, 1, ..., d\}$, $i = 1, ..., n$, and a set of vector valued network functions, $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$, governing the state transitions of the genes. For $j = 1, 2, ..., k$, $\mathbf{f}_j = (f_{j1}, f_{j2}, ..., f_{jn})$, where $f_{ji} : \{0, 1, ..., d\}^n \rightarrow \{0, 1, ..., d\}$, $i = 1, ..., n$ (Shmulevich *et al.* (2002a,b)) In most applications, the discretization is either binary or ternary. Here we use binary, $d = 1$, which presents no theoretical limitation on the development. 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 the decision is to switch, then a new function is chosen from among $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$, with c_j being the probability of choosing $\mathbf{f}_j$ (network selection is not conditioned by the current network, which can itself be selected). Each network function $\mathbf{f}_j$ determines a BN, the individual BNs being called the *contexts* of the PBN. The PBN behaves as a fixed BN until a decision is made to switch contexts according to the probabilities $c_1, c_2, ..., c_k$ from among $\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_k$. If $q = 1$, the PBN is said to be *instantaneously random*; if $q < 1$ (Brun *et al.* (2005)), the PBN is said to be *context-sensitive.* We consider 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 a random perturbation at any time point is $1-(1-p)^n$. A context-sensitive PBN determines a Markov chain whose states are (context, GAP) pairs. The transition probability from $(s,\mathbf{y})$ to $(r,\mathbf{x})$ is given by

$$\begin{aligned} P_{s,\mathbf{y}}(r,\mathbf{x}) &= \mathbf{1}_{[r=s]}((1-q)+qc_s)\{\mathbf{1}_{[\mathbf{f}_s(\mathbf{y})=\mathbf{x}]}(1-p)^n \\ &+ \mathbf{1}_{[\mathbf{x}\neq\mathbf{y}]}p^{\eta(\mathbf{x},\mathbf{y})}(1-p)^{n-\eta(\mathbf{x},\mathbf{y})}\} \\ &+ \mathbf{1}_{[r\neq s]}qc_r\{\mathbf{1}_{[\mathbf{f}_r(\mathbf{y})=\mathbf{x}]}(1-p)^n \\ &+ \mathbf{1}_{[\mathbf{x}\neq\mathbf{y}]}p^{\eta(\mathbf{x},\mathbf{y})}(1-p)^{n-\eta(\mathbf{x},\mathbf{y})}\}, \end{aligned} \tag{1}$$

where r,s denote the rth and sth BNp (Boolean Network with perturbation), which are the BNps at time $t+1$ and t, where $\eta(\mathbf{x},\mathbf{y})$ is the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$, and $\mathbf{1}_{[\mathbf{f}(\mathbf{y})=\mathbf{x}]}$ is the indicator function that takes value 1 if $\mathbf{f}(\mathbf{y})=\mathbf{x}$ according to the rule structure and is equal to 0 otherwise. The random perturbation makes the Markov chain irreducible and ergodic. Thus, it possesses a steady-state distribution. Since there are k contexts and 2^n GAPs in each network, the Markov chain possesses $k2^n$ states and we can relabel the states with $z(t)\in\{0,1,2,\cdots,2^nk-1\}$ being the state that is occupied by the network at time t. For an instanteously random PBN, the Markov chain reduces so that its states are the GAPs of the PBN. The transition probability expression (1) can be used to track the time evolution of the (context, GAP) state. In practice it may be impossible to detect context, only the GAP. We obtain the transition probabilities between the GAPs by taking the expectation of the (context, GAP) transition probabilities over the networks, the transition probability from GAP $\mathbf{y}$ to GAP $\mathbf{x}$ being given by

$$P_{\mathbf{y}}(\mathbf{x}) = (1-p)^n\sum_{i=1}^{k}\mathbf{1}_{[\mathbf{f}_i(\mathbf{y})=\mathbf{x}]}c_i + \mathbf{1}_{[\mathbf{x}\neq\mathbf{y}]}p^{\eta(\mathbf{x},\mathbf{y})}(1-p)^{n-\eta(\mathbf{x},\mathbf{y})} \tag{2}$$

Using the above equations we can compute the $2^n\times 2^n$ transition probability matrix corresponding to the averaged context-sensitive PBN. As shown in Faryabi *et al.* (2009), 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 (2) 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. $c_i=\frac{1}{k},\ i=1,2,\cdots,k$.

2.2 Infinite-horizon control: perfect modeling

In this section, we summarize some results on the infinite-horizon control of PBNs, assuming perfect modeling. A PBN with control can be modeled as a stationary discrete-time dynamic system

$$z_{t+1} = f(z_t,u_t,w_t),\ t=0,1,....., \tag{3}$$

where for all t, the state z_t is an element of a state space S , the control input u_t is an element of a control space C, the disturbance w_t is an element of a space D and $f: S\times C\times D\mapsto S$. [1] In the particular case of a PBN with n genes composed of m Boolean networks with perturbation probability p and network transition probability q, $S=\{0,1,2,.....,m(2^n-1)\}$ and the control input u_t is constrained to take values in the space $C=[0,1,.....,2^k-1]$, where k is the number of binary control inputs. The disturbance w_t is manifested in terms of change of network based on the network transition probability q or change of state due to perturbation probability p. w_t is independent of prior disturbances $w_0,w_1....w_{t-1}$. The objective is to derive a sequence of control inputs, a *control strategy*, such that some cost function is minimized over the entire class of allowable control strategies. We define a cost per stage, $\tilde{g}(i,u,j)$, depending on the origin state i, the destination state j, and the applied control input u. [2] The actual design of a "good" cost function is application dependent and is likely to require considerable expert knowledge. In finite-horizon control one can sum the costs over the number of time points constituting the horizon and take the expectation; however, this cannot safely be done with infinite horizon because the summation of the one-stage costs might diverge to infinity (for all controls), thereby leading to an ill-posed optimization problem. One way to avoid the problem of a possibly infinite total cost is by considering the *average cost per stage* which is defined by

$$J_\pi(z_0) = \lim_{M\to\infty}\frac{1}{M}E\{\sum_{t=0}^{M-1}\tilde{g}(z_t,\mu_t(z_t),z_{t+1})\}. \tag{4}$$

where the expectation is with respect to both origin and destination states. In this formulation, a control policy $\pi=\{\mu_0,\mu_1,\cdots\}$ is chosen to minimize the above cost and the problem is referred to as the *average cost per stage problem*. Minimization of the total cost is feasible if $J_\pi(z_0)$ is finite for at least some admissible policies π and some admissible states z_0. If there is no zero-cost absorbing state (which is the case in context-sensitive PBNs with perturbation), then the total cost will frequently go to ∞. Hence the *average cost per stage* formulation is essential when we are interested in the condition of the patient in the long run and equal importance is given to the patient's condition in all stages. In general, the cost $\tilde{g}(i,u,j)$ of moving from state i to state j under control u may depend on the starting state i; however, in the case of PBNs, we have no obvious basis for assigning different costs based on different initial states. Accordingly, we assume that the penalty $\tilde{g}(i,u,j)$ is independent of the starting state i and its value is based on the control effort and the terminal state j. The penalty is high if the end state is a bad state regardless of the starting state, and vice-versa. Hence $\tilde{g}(i,u,j)=\tilde{g}(u,j)$. Moreover, since in Eq. 4 the cost is obtained by taking the expectation with respect to the origin and destination states, it is possible to replace $\tilde{g}(z_t,u_t,z_{t+1})$ by an equivalent cost

[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.

[2] Note that while finite horizon control problems in the literature allow for costs-per-stage functions that vary from one stage to another, infinite horizon control problems in the literature have typically been derived assuming that the same cost per stage function is used for all stages. For PBNs (both context sensitive and otherwise), this is not of any consequence since all of our earlier finite horizon results also used the same cost per stage function for all stages.

per stage that does not depend on the destination state by taking the expectation with respect to the destination state and leaving only the expectation with respect to the original state. More specifically, we use as cost per stage the expected cost $g(i,u)$ given by Bertsekas (1995):

$$g(i,u) = \sum_{j=0}^{2^n-1} p_{ij}(u)\tilde{g}(i,u,j) = \sum_{j=0}^{2^n-1} p_{ij}(u)\tilde{g}(u,j) \quad (5)$$

where $p_{ij}(u)$ is the transition probability under control u.

To solve the average-cost-per-stage optimal control problem, let Π denote the set of all *admissible* policies π, i.e., the set of all function sequences $\pi = \mu_0, \mu_1, ...$ with $\mu_t(x) : S \rightarrow C, t = 0, 1, ...$ The optimal cost function J^*,which is independent of the initial state (Bertsekas (1995)), is defined by

$$J^* = \min_{\pi \in \Pi} J_\pi(z), z \in S \text{ is arbitrary.} \quad (6)$$

A *stationary policy* is an admissible policy of the form $\pi = \mu, \mu, ...$ Its corresponding cost function is denoted by J_μ. A stationary policy $\pi = \mu, \mu....$ is optimal if $J_\mu(z) = J^*(z)$ for all states z.

To minimize the cost function of Eq. 4, first defiine the mapping

$$J_t(i) = \min_{u \in C} \left[g(i,u) + \sum_{j=0}^{2^n-1} p_{ij}(u) J_{t+1}(j) \right] \quad (7)$$

which, although we will not go into detail, provides the dynamic programming solution for the finite-horizon problem(Bertsekas (1995)). Secondly, for any cost function $J : S \rightarrow \Re$, define the mapping $TJ : S \rightarrow \Re$ by

$$(TJ)(i) = \min_{u \in C} [g(i,u) + \sum_{j=0}^{2^n-1} p_{ij}(u) J(j)], \ i \in S. \quad (8)$$

We note in passing that TJ is the optimal cost function for the one-stage (finite horizon) problem that has stage cost g and terminal cost J. For the average-cost-per-stage problem, the value iteration $J_{t+1}(i) = TJ_t(i)$ cannot be used directly because it may diverge to infinity. Thus, calculating the average cost by taking $\lim_{M \rightarrow \infty}(J_M/M)$ is not feasible. Instead, we consider a *differential cost* h_t obtained by subtracting a fixed component of J_t, say $J_t(n_1)$, from each element of J_t, i.e.,

$$h_t(i) = J_t(i) - J_t(n_1), \forall i \in S. \quad (9)$$

Letting $e = [1, 1, 1, \cdots, 1]^T$, the above relationship can be rewritten in vector form as

$$h_t = J_t - J_t(n_1)e.$$

Some algebraic manipulations (Pal *et al.* (2006)) yield

$$h_{t+1} = Th_t - (Th_t)(n_1)e$$

as the *value iteration algorithm* for the differential cost. Using some additional arguments, we can arrive at the following *policy iteration* algorithm for the average cost case (Bertsekas (1995); Pal *et al.* (2006)):

- 1. (Initialization): An initial policy μ_0 is selected.
- 2. (Policy Evaluation): Given a stationary policy μ_k, we obtain the corresponding average and differential costs λ_k and $h_k(i)$ satisfying

$$\lambda_k + h_k(i) = g(i, \mu_k(i)) + \sum_{j=0}^{2^n-1} p_{ij}(\mu_k(i)) h_k(j), i \in S \quad (10)$$

  This linear system of equations can be solved utilizing the fact that $h_k(n_1) = 0$, where $n_1 \in S$ is any particular reference state.
- 3.(Policy improvement): An improved stationary policy μ_{k+1} satisfying

$$g(i, \mu_{k+1}(i)) + \sum_{j=0}^{2^n-1} p_{ij}(\mu_{k+1}(i)) h_k(j)$$
$$= \min_{u \in C} \left[g(i,u) + \sum_{j=0}^{2^n-1} p_{ij}(u) h_k(j) \right]. \quad (11)$$

  is obtained. The iterations are stopped if $\mu_{k+1} = \mu_k$, else we return to Step 2 and repeat the process.

2.3 Adaptive infinite-horizon control

We now consider an adaptive intervention strategy that can be used in the presence of model uncertainty. We assume that the underlying network is modeled by a member of a known finite family of PBNs and we have no *a priori* knowledge about which member of that family models the actual network. In such a situation, a natural approach is to estimate the model number on-line and then use policy iteration to determine the corresponding controller. This is the principle of adaptive control and considerable theoretical research has been aimed at showing that such *certainty equivalence* schemes can provide the required performance (Ioannou and Sun (1996); Kumar and Varaiya (1986)). Our focus will be to demonstrate via simulations the feasibility of adaptive intervention in the context of gene regulatory networks. We will use a variation of an adaptive control algorithm developed in Kumar and Lin (1982) for unknown Markov chains, to which we refer for technical proofs of convergence. While the scheme in Kumar and Lin (1982) 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.

There are a number of ways in which one can possess a list of PBNs and thereby be presented with the problem of adaptively determining a model number. Several inference procedures produce PBNs by way of first producing Boolean networks satisfying some desired relation to the data. In Pal *et al.* (2005a), Boolean networks are constructed whose attractor structures coincide with data points assumed to be in attractors in the true biological network, along with the networks satisfying certain constraints, such as the number of predictors. Then one or more PBNs are constructed from these Boolean networks by comparing the steady-state distributions of potentially inferred PBNs with the full set of experimental data. In Zhou *et al.* (2004), Boolean networks are inferred by first using a Bayesian approach to generate regulatory graphs (topologies) most compatible with the data and then inferring the predictors

via a nonlinear perceptron model, using a reversible jump Markov chain Monte Carlo (MCMC) method. Then one or more PBNs are constructed from the Boolean networks by using Bayesian scores. In Dougherty and Xiao (2006), a single PBN is constructed such that each constituent Boolean network is consistent with the data, the estimate of the expected distribution of the data generated by the PBN using its steady-steady state distribution agrees with the distribution of the data, and the latter condition cannot be accomplished with less than the number of constituent networks in the inferred PBN. While this leads to a single PBN, in order that the inferred PBN not overfit the data, and in the process be composed of an inordinately large number of Boolean networks, the data are first filtered. Thus, different filtering techniques can lead to different PBNs.

In each of the preceding cases, rather than settle on a single PBN model when applying control, one can take the view that there is a list of potential PBNs and that new data are to be used to adaptively determine the control policy. Moreover, in the cases of Pal *et al.* (2005a) and Zhou *et al.* (2004),one might not even form a PBN and simply treat the problem in the framework of a collection of Boolean networks, in which the adaptation is aimed at selecting a control policy for the governing Boolean network, a view compatible with our proposed algorithms. This latter view, that one has a collection of Boolean networks, absent a PBN structure, was taken in Choudhary *et al.* (2006), where a finite-horizon control policy was determined that performed optimally relative to the family of networks. Here we would proceed adaptively.

In addition to inference, there is another way in which a list of PBNs can naturally occur. In Faryabi *et al.* (2008) and Qian and Dougherty (2009), a PBN is derived from a mammalian cell cycle network proposed in Faure' *et al.* (2006) by assuming a mutation that leads to a cancerous phenotype. Specifically, in the mutation, the gene p27 can never be activated, the result being that the cell can cycle in the absence of any growth factor. A different mutation will lead to a different PBN. Thus, based on a given network, in this case the one proposed in Faure' *et al.* (2006), if one is unsure of the mutation that has led to a cancerous phenotype, then new data utilized in an adaptive fashion can be used to design an intervention strategy.

Suppose the family of controlled PBNs is parametrized by the parameter $\alpha \in A$ where, for any $\alpha \in A$, $\sum_{j\in S} p(i,j,u,\alpha) = 1$ for any $(i,u) \in S \times C$. [3] The only constraint on A is that every element of A results in a set of bonafide transition probabilities. 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 (2). In addition, for a given control gene, the rows of the *controlled* transition probability matrix can be determined as a linear transformation of the rows of the uncontrolled transition probability matrix. As shown in Pal *et al.* (2008), this is a consequence of restricting the class of allowable interventions to the flipping of a chosen control gene. We use the adaptive control algorithm originally derived in Kumar and Lin (1982) by maximizing a modified likelihood criterion. For each $\alpha \in A$, let $J^*(\alpha)$ be the optimal long term average cost obtained for model α using the method of the last sub-section and let $\phi(.,\alpha) : S \to C$ be the corresponding control law attaining it. Let $f : \mathbb{R} \to \mathbb{R}$, $o : \mathbb{Z} \to \mathbb{R}$, and constant m be defined as follows: f is a strictly monotonically increasing continuous function such that $f(\inf_{\alpha\in A} J^*(a)) > 0$; o is any function such that $\lim_{t\to\infty} o(t)t^{-\theta}$ is a positive finite number for some $\theta \in (0,1)$; and m is any integer such that $m > |S|+1$.For our implementation purposes we 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 satisfactorily chosen depending on the cardinality of the state space. The adaptive controller consists of two separate operations, estimation and control:

- *Estimator:* At each time step $0, m, 2m, 3m, ..km, (k+1)m, ...$, estimate α by

$$\hat{\alpha_t} := \text{argmax}_{a\in A} \bar{D}_t(\alpha), \tag{12}$$

where

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

$$K = \frac{1}{f\{J^*(\alpha)\}}^{o(t)}, \tag{14}$$

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

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

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. At time km, knowing the parameter estimate $\alpha_{\hat{km}}$, we can find the optimal cost function $J^*(\alpha_{\hat{km}})$ and the optimal control law $\phi(z_t, \alpha_{\hat{km}})$ which will be used for the next m time steps. The parameter estimate is kept constant at $\alpha_{\hat{km}}$ between time steps km and $(k+1)m-1$.

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

$$u_t := \phi(z_t, \hat{\alpha_t}). \tag{16}$$

The optimal cost function and optimal control law are determined by applying policy iteration to the estimated model. The adaptive algorithm presented here is based on the transition probability expression (2). 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 paper.

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

3 ALGORITHMS

The schematic diagram of the adaptive control algorithm is shown in Fig. 1. The controller and estimator modules are shown separately with the model set A for the estimator module explicitly indicated. Two different choices for the model set A will lead to the two different algorithms presented in this paper. The family of PBNs is shown schematically in Fig. 2.Each member of the family consists of a number of constituent BNps. The underlying PBN is assumed to come from the family. Any switching from one underlying PBN to another is assumed to be deterministic and very infrequent so that, for all practical purposes, the estimator does not need to track a model changing with time.

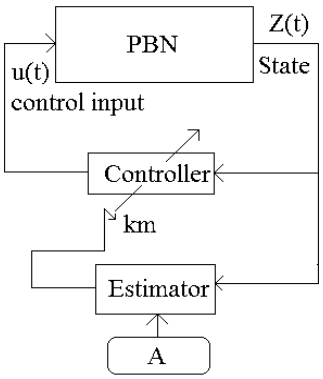

Fig. 1. Adaptive Control Algorithm

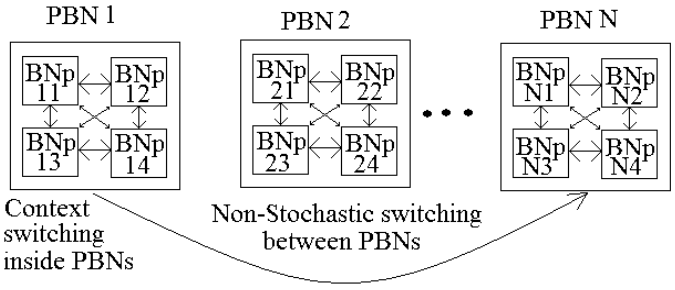

Fig. 2. Family of PBNs

3.1 Algorithm 1

In Algorithm 1, we assume that the family of PBNs constitutes the model set A. Note that this formulation encompasses context-sensitive PBNs, instantaneously random PBNs, and BNs with perturbation (BNps) as they are all special cases of PBNs. For each model (PBN), we can compute the transition probability matrix for the extended state space using equation 1, but it is very difficult to determine the context number from the output state data of the actual PBN. So, constructing the transition counter matrix for the extended state space is practically impossible. For example, suppose each PBN consists of 4 contexts (4 BNps) and the actual underlying PBN is the 2nd PBN in the model set. In addition, suppose at time t there is a transition from state 5 of BNp2 (i.e, context 2) to state 8 of BNp3 (i.e, context 3). In that case, we will observe the $5 \rightarrow 8$ transition; however, in the transition counter matrix there would be 16 elements for that particular $5 \rightarrow 8$ transition (corresponding to the different combinations of 4 source contexts and 4 destination contexts) and there is no way of figuring out which precise context switching occurred. Faced with this hurdle, we compress the transition probability matrix in such a way so that we don't need to find the extended transition counter matrix. This can be done by using equation 2, where the individual transition probability matrices for the different contexts have been averaged out. This averaging out causes no loss of context information when the PBN is instantaneously random since in that case there is no context information to start with; however, even when the PBN is not instantaneously random, and context information is lost, we can still use the averaged transition probability matrix to estimate the model (PBN) number of the underlying PBN. Such an algorithm using the averaged transition probability matrix will henceforth be referred to as Algorithm 1. Clearly, one would expect such an algorithm to perform well for $q = 1$ (i.e, instantaneously random PBN) with performance degradation occurring as the value of q is reduced (i.e, we are moving further and further away from an instantaneously random PBN.)

3.2 Algorithm 2

The main problem with Algorithm 1 is that for small values of the switching probability q (which are typically the more realistic ones), it doesn't perform well. The attractor basin structures of the different constituent BNps of a particular PBN vary significantly and so, averaging of the transition probability matrices of the different constituent BNps is not an appropriate strategy for context-sensitive PBNs with low switching probabilities. For that situation, we can consider the other extreme scenario, where $q = 0$. Then the context-sensitive PBN reduces to a single BNp. A natural question that arises in trying to estimate the underlying PBN from the state transition data is which form of the transition probability matrix should be used. A reasonable answer for $q = 0$ would be to use the individual transition probability matrix for each BNp. This significantly increases the cardinality of the model space A and leads to Algorithm 2. For instance, if we have 4 constituent BNps for each PBN as in Fig. 2, then the cardinality of the model space A will be increased by a factor of 4. Algorithm 2 assumes no context switching and uses the set of constituent BNps as the model set A. This set is used to estimate the model number and the stationary control policy is determined using the policy iteration algorithm. Using simulations it will be shown that Algorithm 2 works better than Algorithm 1 for small values of q. This is quite intuitive because we estimate the model number only after a time interval of m, and if the switching probability q is low, then the number of context switchings inside one estimation time window is expected to be quite low. So, our assumption about the BNp not changing within an estimation window is reasonable. In the next section we will discuss the simulation results for two different sets of data and compare the performance of the two algorithms for three different values of q.

4 DISCUSSION

In this section, we present simulations to demonstrate the efficacy of the proposed adaptive intervention strategies. Such simulation studies are especially important since the theoretical results in Kumar and Lin (1982) guarantee only almost sure convergence and, that in a Cesaro sense[4]. We will consider two different examples of genetic regulatory networks. The first will be an artificial example and the second will be a network derived from gene expression data collected in a metastatic melanoma study. In each case, we will carry out simulation studies using the previously discussed algorithms.

4.1 Artificial example

We consider a 4-gene network modeled by an unknown member of a known family of context-sensitive PBNs. We assume that the cardinality of this family is 7, for each member in this family we have 4 constituent BNps, and $p = 0.01$. The value of q will be chosen differently for various simulations. Since gene values are binary, the cardinality of the state space is 16. Without 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, set to 0. 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. To adaptively intervene in the network, we choose $m = 32$. The cost of control is assumed to be 0.5 and the states are assigned penalties as follows:

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

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. We want to examine how algorithm 1 performs when the true model is deterministically switched. Accordingly, we set up the simulation with the actual model being switched from PBN2 (model number 2) to PBN6 (model number 6) at the 10th estimation window (time = 320). The switching probability (q) is 0.01. This emulates a context-sensitive PBN. Figure 3 shows the convergence results. Each of the following figures shows model and cost comparisons between the non-adaptive regular controller (with complete model information) and the adaptive controller. The top plot shows the estimated and actual models 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 comparison of the cumulative adaptive average cost and the cumulative non-adaptive average cost (assuming perfect knowledge about the true model).From Fig. 3, it is clear that the estimated model converges to the true model and the cumulative adaptive average cost goes towards the cumulative non-adaptive average cost. Figure 4 shows

[4] Roughly speaking, convergence in the Cesaro sense formalizes the notion of convergence of the time average of a signal. This clearly doesn't imply pointwise convergence.

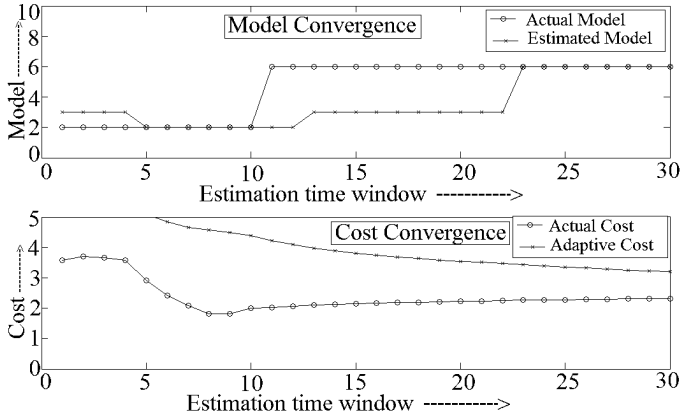

Fig. 3. Artificial Example:Algorithm 1

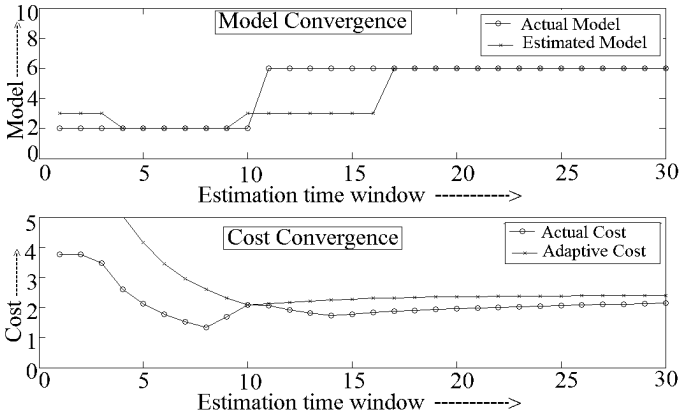

Fig. 4. Artificial Example: Algorithm 2

the simulation results obtained using algorithm 2 on the same simulation set up as above with $q = 0.01$. 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. The estimated model convergence in the case of algorithm 2 is much faster than that obtained using algorithm 1. This is to be expected since, with $q = 0.01$, the underlying assumptions for algorithm 2 are a better fit to the real scenario. We next study the effect of the value of q on the performance of the two algorithms. To compare the two algorithms, we cannot rely on just one simulation. Moreover, we are more interested in achieving controlled cost convergence rather than model convergence as our sole aim in intervention is to minimize the long term average cost. Accordingly, we run the same simulation one hundred times and calculate the difference between the cumulative adaptive and cumulative non-adaptive average costs in each case. We then average the difference sequence over the 100 simulations. Figure 5 shows the results for 30 estimation windows (time = 960) for three different values of q.From Fig. 5, we see that the results match our intuition: algorithm 1 works well for $q = 1$ (instantaneously random PBN) whereas when q is low or 0, algorithm 2 works better.

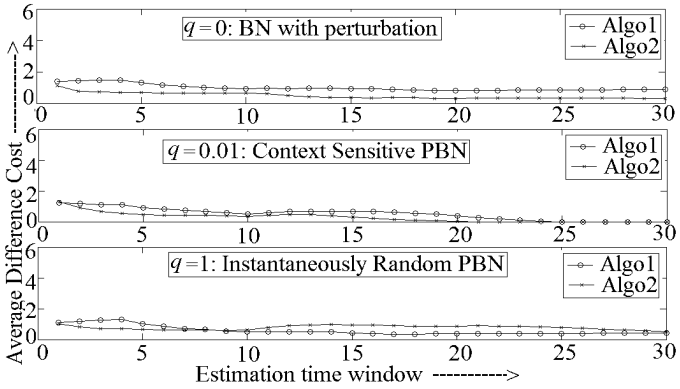

Fig. 5. Artificial Example: Cost difference comparison of the two algorithms for different values of q

4.2 Melanoma application

In a study of metastatic melanoma it was found that experimentally increasing the levels of the Wnt5a protein secreted by a melanoma cell line via genetic engineering methods directly altered the metastatic competence online as measured by the standard *in vitro* assays for metastasis (Weeraratna *et al.* (2002)). Furthermore, it was found that an intervention that blocked the Wnt5a protein from activating its receptor, the use of an antibodyh that binds the Wnt5a protein, could substantially reduce Wnt5a's ability to induce a metastatic phenotype. This suggests a control strategy that reduces the WNT5A gene's action in affecting biological regulation, since the data suggest that disruption of this influence could reduce the chance of a melanoma metastasizing, a desirable outcome. PBNs derived from the same expression data have been used in Datta *et al.* (2003); Pal *et al.* (2005b, 2006, 2008) for demonstrating earlier non-adaptive intervention strategies. We consider 7-gene PBNs containing the genes WNT5A, pirin, S100P, RET1, MART1, HADHB and STC2 obtained via the algorithms described in Pal *et al.* (2005a). The states are ordered as above, with WNT5A as the most significant bit (MSB) and STC2 as the least significant bit (LSB).

We have constructed 7 PBNs with four constituent BNs in each. The adaptive intervention strategy has been applied to the family of PBNs with pirin as the control gene ($u = 1$, state of pirin is reversed, and $u = 0$, no intervention), $m = 256$, and $p = 0.01$. The value of q varies between simulations. The cost of control is assumed to be 0.5 and the states are assigned penalties as follows:

$$\tilde{g}(u, j) = \begin{cases} 5 & \text{if } u = 0 \text{ and WNT5A is 1 for state } j \\ 5.5 & \text{if } u = 1 \text{ and WNT5A is 1 for state } j \\ 0.5 & \text{if } u = 1 \text{ and WNT5A is 0 for state } j \\ 0 & \text{if } u = 0 \text{ and WNT5A is 0 for state } j \end{cases}$$

Since our objective is to down-regulate the WNT5A gene, a higher penalty is assigned for destination states having WNT5a up-regulated. Also, for a given WNT5A status for the destination state, a higher penalty is assigned when the control is active versus when it is not. Figures 6 and 7 show the performance of the adaptive intervention schemes using algorithms 1 and 2, respectively. In each case, the genetic regulatory network is initially described by PBN4 (model number 4) and at estimation window number 10 (corresponding to time =2560), the underlying model is deterministically switched to PBN2 (model number 2). The switching probability (q) is assumed to be 0.01. From the model convergence plots in Figs. 6 and 7, it is clear that the estimated models track the actual model quite well. Furthermore, the model tracking using algorithm 2 is better than with algorithm 1. This is consistent with our expectation since for the small q, the underlying assumption for algorithm 2 represents a closer approximation to reality. The cumulative adaptive average costs also appear to converge to the non-adaptive ones.

To see if these results are representative, we ran the same simulation one hundred times and calculated the differences between the cumulative adaptive and non-adaptive costs for each of the two algorithms.We then averaged the difference sequence over the one hundred simulations. Figure 8 shows the plots of the average difference sequence over 30 estimation windows (time = 7680) for three different values of the switching probability q. From the figure, we see that the results match our intuition: algorithm 1 works well for

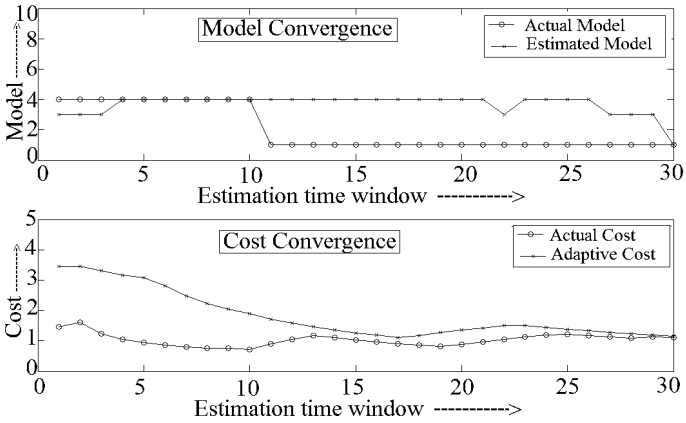

Fig. 6. Melanoma Application: Algorithm 1

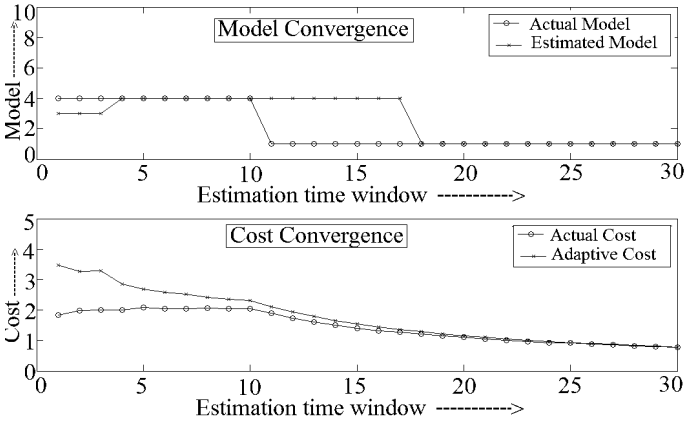

Fig. 7. Melanoma Application: Algorithm 2

$q = 1$ (instantaneously random PBN) whereas when q is low or 0, algorithm 2 works better.

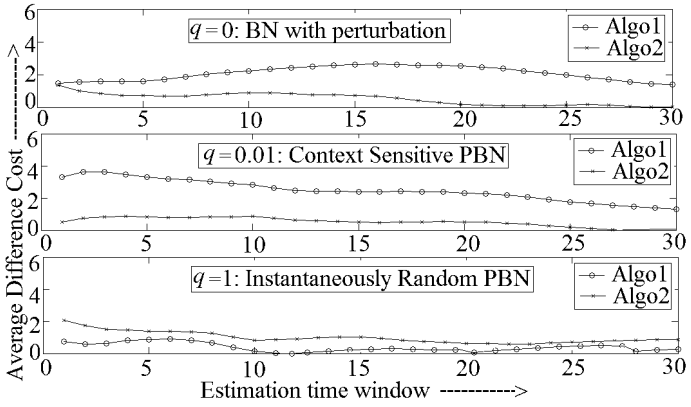

Fig. 8. Melanoma Application: Cost difference comparison of the two algorithms for different values of q

4.3 Concluding remarks

We have demonstrated the feasibility of applying adaptive intervention to improve intervention performance in genetic regulatory networks modeled by PBNs. Specifically, we have shown via simulations that when the 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 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 have 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 with low switching probability between the constituent BNs. Our simulation results confirm these intuitive expectations.

REFERENCES

Bertsekas, D. (1995). *Dynamic Programming and Optimal Control*. Athena Scientific.

Brun, M.and Dougherty, E. R., , and Shmulevich, I. (2005). Steady-state probabilities for attractors in probabilistic boolean networks. *EURASIP Journal on Signal Processing*, pages 1993–2013.

Choudhary, A., Datta, A., Bittner, M. L., and Dougherty, E. R. (2006). Intervention in a family of boolean networks. *Bioinformatics*, **22**, 226–232.

Datta, A., Choudhary, A., Bittner, M., and Dougherty, E. (2003). External Control in Markovian Genetic Regulatory Networks. *Machine Learning, VOL. 52*, pages 169–191.

Dougherty, E. R. and Xiao, Y. (2006). Design of probabilistic boolean networks under the requirement of contextual data consistency. *IEEE Transactions on Signal Processing*, **54**, 3603– 3613.

Faryabi, B., Chamberland, J. F., Vahedi, G., Datta, A., and Dougherty, E. R. (2008). Optimal intervention in asynchronous genetic regulatory networks. *IEEE Journal of Selected Topics in Signal Processing*, **2**, 412–423.

Faryabi, B., Vahedi, G., Chamberland, J.-F., Datta, A., and Dougherty, E. (2009). Intervention in context-sensitive probabilistic boolean networks revisited. *EURASIP Journal on Bioinformatics and Systems Biology*.

Faure', A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics*, **22**, 124–131.

Ioannou, P. A. and Sun, J. (1996). *Robust Adaptive Control A Unified Approach*. Prentice Hall, Englewood Cliffs.

Kauffman, S. (1993). *The Origins of Order Self-Organization and Selection in Evolution*. Oxford Univ. Press,USA.

Kumar, P. and Lin, W. (1982). Optimal adaptive controller for unknown markov chains. *IEEE Transactions on Automatic Control,VOL. AC-27, NO. 4*, **AC-27**, 765–774.

Kumar, P. and Varaiya, P. (1986). *Stochastic Systems Estimation, Identification and Adaptive Control*. Prentice Hall, Englewood Cliffs.

Pal, R., Ivanov, I., Datta, A., Bittner, M. L., and Dougherty, E. R. (2005a). Generating boolean networks with a prescribed attractor structure. *Bioinformatics*, **21**, 4021–4025.

Pal, R., Datta, A., Bittner, M., and Dougherty, E. (2005b). Intervention in context-sensitive probabilistic boolean networks. *Bioinformatics,VOL. 21, NO. 7*, pages 1211–1218.

Pal, R., Datta, A., and Dougherty, E. (2006). Optimal infinite-horizon control for probabilistic boolean network. *IEEE Transaction on Signal Processing,VOL. 54, NO. 6*, pages 2375–2387.

Pal, R., Datta, A., and Dougherty, E. (2008). Robust intervention in probabilistic boolean networks. *IEEE Transaction on Signal Processing,VOL. 56, NO. 3*, pages 1280–1294.

Qian, X. and Dougherty, E. R. (2009). On the long-run sensitivity of probabilistic boolean networks. *Theoretical Biology*, **257**, 560–577.

Shmulevich, I., Dougherty, E., and Zhang, W. (2002a). From boolean to probabilistic boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE,VOL. 90, NO. 11*, pages 1778–1792.

Shmulevich, I., Dougherty, E., Kim, S., and Zhang, W. (2002b). Probabilistic boolean networks a rule based uncertainty model for gene regulatory networks. *Bioinformatics,VOL. 18, NO. 2*, pages 261–274.

Weeraratna, A. T., Jiang, Y., Hostetter, G., Rosenblatt, K., Duray, P., Bittner, M., and M., T. J. (2002). Wnt5a signalling directly affects cell motility and invasion of metastatic melanoma. *Cancer Cell*, **1**, 279–288.

Zhou, X., Wang, X., Pal, R., Ivanov, I., Bittner, M. L., , and Dougherty, E. R. (2004). A bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics*, **20**, 2918–2927.