



# External Control in Markovian Genetic Regulatory Networks

ANIRUDDHA DATTA  
ASHISH CHOUDHARY  
*Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128, USA*

datta@ee.tamu.edu  
cdry@ee.tamu.edu

MICHAEL L. BITTNER  
*National Human Genome Research Institute, National Institutes of Health, Bethesda, MD 20892-4470, USA*

mbittner@nhgri.nih.gov

EDWARD R. DOUGHERTY  
*Department of Electrical Engineering, Texas A&M University, College Station, TX 77843-3128, USA;*  
*Department of Pathology, University of Texas M.D. Anderson Cancer Center, 1515 Holcombe Blvd.,*  
*Houston, TX 77030, USA*

edward@ee.tamu.edu

**Editors:** Paola Sebastiani, Isaac S. Kohane and Marco F. Ramoni

**Abstract.** Probabilistic Boolean Networks (PBN's) have been recently introduced as a rule-based paradigm for modeling gene regulatory networks. Such networks, which form a subclass of Markovian Genetic Regulatory Networks, provide a convenient tool for studying interactions between different genes while allowing for uncertainty in the knowledge of these relationships. This paper deals with the issue of control in probabilistic Boolean networks. More precisely, given a general Markovian Genetic Regulatory Network whose state transition probabilities depend on an external (control) variable, the paper develops a procedure by which one can choose the sequence of control actions that minimize a given performance index over a finite number of steps. The procedure is based on the theory of controlled Markov chains and makes use of the classical technique of *Dynamic Programming*. The choice of the finite horizon performance index is motivated by cancer treatment applications where one would ideally like to intervene only over a finite time horizon, then suspend treatment and observe the effects over some additional time before deciding if further intervention is necessary. The undiscounted finite horizon cost minimization problem considered here is the simplest one to formulate and solve, and is selected mainly for clarity of exposition, although more complicated costs could be used, provided appropriate technical conditions are satisfied.

**Keywords:** gene regulatory network, Markov chain, optimal control, dynamic programming

## 1. Introduction

Probabilistic Boolean Networks (PBN's) have been recently proposed as a paradigm for studying gene regulatory networks (Shmulevich et al., 2002a). These networks which allow the incorporation of uncertainty into the inter-gene relationships, are essentially probabilistic generalizations of the standard Boolean networks introduced by Kauffman (1969, 1993) and Kauffman and Levin (1987). Given a probabilistic Boolean network, the transition from one state to the next takes place in accordance with certain transition probabilities. Indeed, as shown in Shmulevich et al. (2002a), and as will be briefly reviewed in the next section,

the states of a PBN form a homogeneous Markov chain with finite state space. Thus the PBN's form a subclass of the general class of Markovian Genetic Regulatory Networks.

The probabilistic Boolean networks considered thus far in the literature can be described by Markov chains with *fixed* transition probabilities. Consequently, for such a network, given an initial state, the subsequent states evolve according to a priori determined probabilities. This set up provides a model for dynamically tracking the gene activity profile while allowing for uncertainty in the relationship between the different genes. However, it does not provide any effective knobs that could be used to externally guide the time evolution of the PBN, hopefully towards more desirable states.

Intervention has been considered in the context of probabilistic Boolean networks from other perspectives. By exploiting concepts from Markov Chain theory, it has been shown how at a given state, one could toggle the expression status of a particular gene from ON to OFF or vice-versa to facilitate transition to some other desirable state or set of states (Shmulevich, Dougherty, & Zhang, 2002b). Specifically, using the concept of the *mean first passage time*, it has been shown how the particular gene, whose transcription status is to be momentarily altered to initiate the state transition, can be chosen to "minimize" in a probabilistic sense the time required to achieve the desired state transitions. These results come under the category of "transient" intervention which essentially amounts to letting the original network evolve after re-initializing the state to a different value. A second approach has aimed at changing the steady-state (long-run) behavior of the network by minimally altering its rule-based structure (Shmulevich, Dougherty, & Zhang, 2002c). This too constitutes transient intervention, but is more permanent in that it involves structural intervention.

In this paper, we consider probabilistic Boolean networks where the transition probabilities between the various states can be altered by the choice of some auxiliary variables. These variables, which we will refer to as *control inputs*, could then be chosen to increase the likelihood that the network will transition from an undesirable state to a desirable one. Such a situation is likely to arise in the treatment of diseases such as cancer where the auxiliary variables could represent the current status of therapeutic interventions such as radiation, chemo-therapy, etc. To be consistent with the binary nature of the state space associated with PBNs, these auxiliary control inputs will be allowed to be in one of two states: an ON state indicating that a particular intervention is being actively applied at that point in time and an OFF state indicating that the application of that particular intervention has ceased. The control objective here would be to "optimally" apply one or more treatments so that an appropriate cost function is minimized over a finite number of steps, which we will refer to as the *treatment horizon*. The choice of the cost function, as well as the length of the treatment window are two important aspects where the expert knowledge from biologists/clinicians could play a crucial role.

Once the cost function and the treatment window have been selected, the control problem is essentially reduced to that of controlling a Markov Chain over a finite horizon. Control problems of this type have been extensively studied in the controls literature for over four decades. Among the different solution methods available, the most popular one is the technique of *Dynamic Programming*, pioneered by Bellman in the 1960's (Bellman, 1957; Bertsekas, 1976). In this paper, we will formulate the optimal control problem for a probabilistic Boolean network and arrive at a solution based on the dynamic programming

approach. A preliminary version of the ideas in this paper were first presented in the talk “Control Theory and PBNs,” at the *First Workshop on Probabilistic Boolean Networks in Genomic Signal Processing*, National Human Genome Research Institute, Bethesda, February 2002.

The paper is organized as follows. In Section 2, we provide a brief review of probabilistic Boolean networks as introduced in Shmulevich et al. (2002a). In Section 3, we formulate the control problem for PBNs. The solution to this problem using the Dynamic Programming technique is presented in Section 4. Section 5 contains two examples while Section 6 contains some concluding remarks.

## 2. Review of probabilistic Boolean networks

In this section, we provide a brief review of probabilistic Boolean networks. We will only focus on those aspects that are critical to the development in this paper. For a detailed and complete exposition, the reader is referred to Shmulevich et al. (2002a) and Shmulevich, Dougherty, and Zhang (2002b).

A probabilistic Boolean network is a formalism that has been developed for modeling the behaviour of gene regulatory networks. In such a network, each gene can take on one of two binary values, zero or one. A zero value for a gene corresponds to the case when that particular gene is not expressed and a one value indicates that the corresponding gene has been turned ON. The functional dependency of a given gene value on all the genes in the network is given in terms of a single Boolean function or a family of Boolean functions. The case of a single Boolean function for each gene arises when the functional relationships between the different genes in the network are exactly known. Such a situation is not very likely to occur in practice. Nevertheless, networks of this type, referred to as standard Boolean networks (Kauffman, 1993) have been extensively studied in the literature.

To account for uncertainty in our knowledge of the functional dependencies between the different genes, one could postulate that the expression level of a particular gene in the network is described by a family of Boolean functions with finite cardinality. Furthermore, each member of this family is assumed to describe the functional relationship with a certain probability. This leads to a probabilistic Boolean network, as introduced in Shmulevich et al. (2002a).

Our discussion so far has only concentrated on the static relationships between the different genes in the network. To introduce dynamics, we assume that in each time step, the value of each gene is updated using the Boolean functions evaluated at the previous time step. For PBNs, the expression level of each gene will be updated in accordance with the probabilities corresponding to the different Boolean functions associated with that particular gene.

To concretize matters, let us assume that we are attempting to model the relationship between ‘ $n$ ’ genes. Suppose that the activity level of gene ‘ $i$ ’ at time step ‘ $k$ ’ is denoted by  $x_i(k)$ . Thus  $x_i(k) = 0$  would indicate that at the  $k$ th time step, the  $i$ th gene is not expressed while  $x_i(k) = 1$  would indicate that the corresponding gene is expressed. The overall expression levels of all the genes in the network at time step  $k$  is given by the row vector  $x(k) = [x_1(k), x_2(k), \dots, x_n(k)]$ . This vector is sometimes referred to as the *gene activity profile* (GAP) of the network at time  $k$ .

Now suppose that for each gene  $i$ , there are  $l(i)$  possible Boolean functions

$$f_1^{(i)}, f_2^{(i)}, f_3^{(i)}, \dots, f_{l(i)}^{(i)}$$

that can be used to describe the dependency of  $x_i$  on  $x_1, x_2, \dots, x_n$ . Furthermore, suppose that  $f_j^{(i)}$  is selected with a probability  $c_j^{(i)}$  so that

$$\sum_{j=1}^{l(i)} c_j^{(i)} = 1.$$

Then the expression level of the  $i$ th gene transitions according to the equation:

$$x_i(k+1) = f_j^{(i)}(x(k)) \text{ with probability } c_j^{(i)}. \quad (2.1)$$

Let us consider the evolution of the entire state vector  $x(k)$ . Corresponding to a probabilistic Boolean network with  $n$  genes, there are at most  $N = \prod_{i=1}^n l(i)$  distinct Boolean networks, each of which could capture the inter-gene functional relationships with a certain probability. Let  $P_1, P_2, \dots, P_N$  be the probabilities associated with the selection of each of these networks. Suppose the  $k$ th network is obtained by selecting the functional relationship  $f_{i_k}^{(i)}$  for gene  $i$ ,  $i = 1, 2, \dots, n$ ,  $1 \leq i_k \leq l(i)$ . Then, if the choice of the functional relationship for each gene is assumed to be independent of that for other genes, we have

$$P_k = \prod_{i=1}^n c_{i_k}^{(i)}. \quad (2.2)$$

As discussed in Shmulevich et al. (2002a), even when there are dependencies between the choice of the functional relationships for different genes, one can calculate the  $P_i$ 's by using conditional probabilities instead of the unconditional ones  $c_j^{(i)}$ .

The evolution of the states of the PBN can be described by a finite Markov chain model. To do so, we first focus on standard Boolean networks. Then the state vector  $x(k)$  at any time step  $k$  is essentially an  $n$ -digit binary number whose decimal equivalent is given by

$$y(k) = \sum_{j=1}^n 2^{n-j} x_j(k). \quad (2.3)$$

As  $x(k)$  ranges from  $000 \dots 0$  to  $111 \dots 1$ ,  $y(k)$  takes on all values from  $0$  to  $2^n - 1$ . Now to be completely consistent with the development in Shmulevich et al. (2002a), define

$$z(k) = 1 + y(k). \quad (2.4)$$

Then as  $x(k)$  ranges from  $00 \dots 0$  to  $11 \dots 1$ ,  $z(k)$  will take on all values from  $1$  to  $2^n$ . Clearly, the map from  $x(k)$  to  $z(k)$  is one-to-one, onto and hence invertible. Thus instead of the binary representation  $x(k)$  for the state vector, one could equivalently work with

the decimal representation  $z(k)$ . Furthermore, each  $z(k)$  could be uniquely represented by a basis vector  $w(k) \in R^{2^n}$  where  $w(k) = e_{z(k)}$ , e.g. if  $z(k) = 1$ , then  $w(k) = [1, 0, \dots]$ . Then, as discussed in Shmulevich et al. (2002a), the evolution of the vector  $w(k)$  proceeds according to the following difference equation

$$w(k + 1) = w(k)A \quad (2.5)$$

where  $A$  is a  $2^n \times 2^n$  matrix having only one non-zero entry (equal to one) in each row. Equation (2.5) is reminiscent of the state transition equation in Markov Chain theory. The only difference here is that for a given initial state, the transition is completely deterministic. However, Eq. (2.5) can also be easily interpreted within a stochastic framework. For instance, the vector  $w(k)$  does represent the probability distribution over the entire state space at time step  $k$ . Indeed, because of the deterministic nature of the evolution, at each time step  $k$ , the entire probability mass is concentrated on only one out of the  $2^n$  possible states, thereby accounting for the  $2^n$  dimensional vectors  $w(k)$  with only one non-zero entry of one corresponding to the location where the probability mass is concentrated. The matrix  $A$  also qualifies as a bonafide stochastic matrix with the sole non-zero entry in each row being equal to one. Thus, given an initial state, the transition to the next state is deterministic and takes place with probability one.

The stochastic interpretation of (2.5) given above allows us to readily extend (2.5) to accommodate state transitions in probabilistic Boolean networks. Towards this end, let  $a$  and  $b$  be any two basis vectors in  $R^{2^n}$ . Then, using the total probability theorem, it follows that the transition probability  $Pr\{w(k + 1) = a \mid w(k) = b\}$  is given by

$$\begin{aligned} Pr\{w(k + 1) = a \mid w(k) = b\} &= \sum_{i=1}^N Pr\{w(k + 1) = a \mid w(k) = b, \text{ Network } i \text{ is selected}\} \cdot P_i \\ &= \sum_{i \in \mathcal{I}} P_i \end{aligned} \quad (2.6)$$

where

$$\mathcal{I} = \{i : Pr\{w(k + 1) = a \mid w(k) = b, \text{ Network } i \text{ is selected}\} = 1\}.$$

By letting the vectors  $a$  and  $b$  range over all possible basis vectors in  $R^{2^n}$ , we can determine the  $2^n \times 2^n$  entries of the transition probability matrix  $A$ .

Now let  $w(k)$  denote the probability distribution vector at time  $k$ , i.e.  $w_i(k) = Pr\{z(k) = i\}$ . It is straightforward to show that  $w(k)$  evolves according to the equation

$$w(k + 1) = w(k)A \quad (2.7)$$

where the entries of the  $A$  matrix have been determined using (2.6). This completes our discussion of PBNs. For a more rigorous derivation of (2.7), the reader is referred to Shmulevich et al. (2002a).

### 3. Control in probabilistic Boolean networks: Problem formulation

Probabilistic Boolean networks can be used for studying the dynamic behaviour of gene regulatory networks. However, once a probability distribution vector has been specified for the initial state, the subsequent probability distribution vectors evolve according to Eq. (2.7) and there is no mechanism for “controlling” this evolution. Thus the PBNs considered thus far in the literature are “descriptive” in nature in the sense that they can be used to describe the evolution of the probability distribution vector, starting from any initial distribution. For treatment or intervention purposes, we are interested in working with “prescriptive” probabilistic Boolean networks where the transition probabilities of the associated Markov chain depend on certain auxiliary variables, whose values can be chosen to make the probability distribution vector evolve in some desirable fashion.

The use of such auxiliary variables makes sense from a biological perspective. For instance, in the case of diseases like cancer, auxiliary treatment inputs such as radiation, chemo-therapy, etc. may be employed to move the state probability distribution vector away from one which is associated with uncontrolled cell proliferation or markedly reduced apoptosis. The auxiliary variables could also include genes which serve as external master-regulators for all the genes in the network. To be consistent with the binary nature of the expression status of individual genes in the PBN, we will assume that the auxiliary variables (*control inputs*) can take on only the binary values zero or one. The values of the individual control inputs can be changed from one time step to the other in an effort to make the network behave in a desirable fashion.

Suppose that a probabilistic Boolean network with  $n$  genes has  $m$  control inputs  $u_1, u_2, \dots, u_m$ . Then at any given time step  $k$ , the row vector  $u(k) \triangleq [u_1(k), u_2(k), \dots, u_m(k)]$  describes the complete status of all the control inputs. Clearly,  $u(k)$  can take on all binary values from  $[0, 0, \dots, 0]$  to  $[1, 1, \dots, 1]$ . As in the case of the state vector, one can equivalently represent the control input status using the decimal number

$$v(k) = 1 + \sum_{i=1}^m 2^{m-i} u_i(k). \quad (3.8)$$

Clearly, as  $u(k)$  takes on binary values from  $[0, 0, \dots, 0]$  to  $[1, 1, \dots, 1]$ , the variable  $v(k)$  ranges from 1 to  $2^m$ . We can equivalently use  $v(k)$  as an indicator of the complete control input status of the probabilistic Boolean network at time step  $k$ .

We now proceed to derive the counterpart of Eq. (2.7) for a probabilistic Boolean network subject to auxiliary controls. Let  $v^*$  be any integer between 1 and  $2^m$  and suppose that  $v(k) = v^*$ . Then, it is clear that the procedure outlined in the last section can be used to compute the corresponding  $A$  matrix which will now depend on  $v^*$  and can be denoted by  $A(v^*)$ . Furthermore, the evolution of the probability distribution vector at time  $k$  will take place according to the following equation:

$$w(k+1) = w(k)A(v^*). \quad (3.9)$$

Since the choice of  $v^*$  is arbitrary, the one-step evolution of the probability distribution vector in the case of a PBN with control inputs takes place according to the equation:

$$w(k+1) = w(k)A(v(k)). \quad (3.10)$$

Note that the transition probability matrix here is a function of all the control inputs  $u_1(k), u_2(k), \dots, u_m(k)$ . Consequently, the evolution of the probability distribution vector of the PBN with control now depends not only on the initial distribution vector but also on the values of the control inputs at different time steps. Furthermore, intuitively it appears that it may be possible to make the states of the network evolve in a desirable fashion by appropriately choosing the control input at each time step. We next proceed to formalize these ideas.

Equation (3.10) is referred to in the control literature as a *Controlled Markov Chain* (Bertsekas, 1976). Markov chains of this type occur in many real life applications, the most notable example being the control of queues. Given such a controlled Markov chain, the objective is to come up with a sequence of control inputs, usually referred to as a control strategy, such that an appropriate cost function is minimized over the entire class of allowable control strategies. To arrive at a meaningful solution, the cost function must capture the costs and the benefits of using any control. The actual design of a “good” cost function is application dependent and is likely to require considerable expert knowledge. We next outline a procedure that we believe would enable us to arrive at a reasonable cost function for determining the course of therapeutic intervention using PBNs.

In the case of diseases like cancer, treatment is typically applied over a finite time horizon. For instance, in the case of radiation treatment, the patient may be treated with radiation over a fixed interval of time following which the treatment is suspended for some time as the effects are evaluated. After that, the treatment may be applied again but the important point to note is that the treatment window at each stage is usually finite. Thus we will be interested in a finite horizon problem where the control is applied only over a finite number of steps.

Suppose that the number of steps over which the control input is to be applied has been a priori determined to be  $M$  and we are interested in controlling the behaviour of the PBN over the interval  $k = 0, 1, 2, \dots, M-1$ . Suppose at time step  $k$ , the state<sup>1</sup> of the probabilistic Boolean network is given by  $z(k)$  and the corresponding control input is  $v(k)$ . Then we can define a cost  $C_k(z(k), v(k))$  as being the cost of applying the control input  $v(k)$  when the state is  $z(k)$ . With this definition, the expected cost of control over the entire treatment horizon becomes

$$E \left[ \sum_{k=0}^{M-1} C_k(z(k), v(k)) \mid z(0) \right]. \quad (3.11)$$

Note that even if the network starts from a given (deterministic) initial state  $z(0)$ , the subsequent states will be random because of the stochastic nature of the evolution in (3.10). Consequently, the cost in (3.11) had to be defined using an expectation. Equation (3.11) does give us one component of the finite horizon cost, namely the cost of control. We now proceed to introduce the second component.

The net result of the control actions  $v(0), v(1), \dots, v(M-1)$  is that the state of the PBN will transition according to (3.10) and will end up in some state  $z(M)$ . Because of the probabilistic nature of the evolution, the terminal state  $z(M)$  is a random variable that could possibly take on any of the values  $1, 2, \dots, 2^n$ . Depending on the particular PBN and

the control inputs used at each step, it is possible that some of these states may never be reached because of non-communicating states in the resulting Markov chains, etc. However, since the control strategy itself has not yet been determined, it would be difficult, if not impossible, to identify and exclude such states from further consideration. Instead, we assume that all the  $2^n$  terminal states are reachable and assign a penalty or terminal cost  $C_M(z(M))$  associated with each one of them. Indeed, in the case of PBNs with perturbation, all states communicate and the Markov chain is ergodic (Shmulevich, Dougherty, & Zhang, 2002b). We next consider penalty assignment.

First, consider the PBN with all controls set to zero i.e.  $v(k) \equiv 1$  for all  $k$ . Then divide the states into different categories depending on how desirable or undesirable they are and assign higher terminal costs to the undesirable states. For instance, a state associated with rapid cell proliferation leading to cancer should be associated with a high terminal penalty while a state associated with normal behaviour should be assigned a low terminal penalty. For the purposes of this paper, we will assume that the assignment of terminal penalties has been carried out and we have at our disposal a terminal penalty  $C_M(z(M))$  which is a function of the terminal state. Thus we have arrived at the second component of our cost function. Once again, note that the quantity  $C_M(z(M))$  is a random variable and so we must take its expectation while defining the cost function to be minimized. In view of (3.11), the finite horizon cost to be minimized is given by

$$E \left[ \sum_{k=0}^{M-1} C_k(z(k), v(k)) + C_M(z(M)) \mid z(0) \right]. \quad (3.12)$$

To proceed further, let us assume that at time  $k$ , the control input  $v(k)$  is a function of the current state  $z(k)$  i.e.

$$v(k) = \mu_k(z(k)) \quad (3.13)$$

where  $\mu_k : \{1, 2, \dots, 2^n\} \rightarrow \{1, 2, \dots, 2^m\}$ . The *optimal control problem* can now be stated as follows: Given an initial state  $z(0)$ , find a control law  $\pi = \{\mu_0, \mu_1, \dots, \mu_{M-1}\}$  that minimizes the cost functional

$$J_\pi(z(0)) = E \left[ \sum_{k=0}^{M-1} C_k(z(k), \mu_k(z(k))) + C_M(z(M)) \right] \quad (3.14)$$

subject to the constraint

$$Pr\{z(k+1) = j \mid z(k) = i\} = a_{ij}(v(k)) \quad (3.15)$$

where  $a_{ij}(v(k))$  is the  $i$ th row,  $j$ th column entry of the matrix  $A(v(k))$ .

#### 4. Solution using dynamic programming

Optimal control problems of the type described by (3.14), (3.15) can be solved by using the technique of *Dynamic Programming*. This technique, pioneered by Bellman in the 1960's

is based on the so-called *Principle of Optimality*. This principle is a simple but powerful concept and can be explained as follows.

Suppose that we have an optimization problem where we are interested in optimizing a performance index over a finite number of steps, say  $M$ . At each step, a decision is made and the objective is to come up with a strategy or *sequence* of  $M$  decisions which is optimal in the sense that the cumulative performance index over all the  $M$  steps is optimized. In general, such an optimal strategy may not exist. However, when such an optimal strategy does exist, the principle of optimality asserts the following: if one searches for an optimal strategy over a subset of the original number of steps, then this new optimal strategy will be given by the overall optimal strategy, restricted to the steps being considered. Although intuitively obvious, the principle of optimality can have far reaching consequences. For instance, it can be used to obtain the following proposition proven in Bertsekas (1976) (Ch. 2, p. 50).

**Proposition 1.** *Let  $J^*(z(0))$  be the optimal value of the cost functional (3.14). Then*

$$J^*(z(0)) = J_0(z(0)),$$

where the function  $J_0$  is given by the last step of the following dynamic programming algorithm which proceeds backward in time from time step  $M - 1$  to time step 0:

$$J_M(z(M)) = C_M(z(M)) \quad (4.16)$$

$$J_k(z(k)) = \min_{v(k) \in \{1, 2, \dots, 2^m\}} E\{C_k(z(k), v(k)) + J_{k+1}[z(k+1)]\} \\ k = 0, 1, 2, \dots, M - 1 \quad (4.17)$$

Furthermore, if  $v^*(k) = \mu_k^*(z(k))$  minimizes the right hand side of (4.17) for each  $z(k)$  and  $k$ , the control law  $\pi^* = \{\mu_0^*, \mu_1^*, \dots, \mu_{N-1}^*\}$  is optimal.

Note that the expectation on the right hand side of (4.17) is conditioned on  $z(k)$  and  $v(k)$ . Hence, in view of (3.15), it follows that

$$E[J_{k+1}(z(k+1)) | z(k), v(k)] = \sum_{j=1}^{2^n} a_{z(k), j}(v(k)) \cdot J_{k+1}(j).$$

Thus the dynamic programming solution to (3.14), (3.15) is given by

$$J_M(z(M)) = C_M(z(M)) \quad (4.18)$$

$$J_k(z(k)) = \min_{v(k) \in \{1, 2, \dots, 2^m\}} \left[ C_k(z(k), v(k)) + \sum_{j=1}^{2^n} a_{z(k), j}(v(k)) \cdot J_{k+1}(j) \right] \\ k = 0, 1, 2, \dots, M - 1 \quad (4.19)$$

## 5. Examples

In this section, we present two examples to show optimal control design using the dynamic programming approach. The first example is a simple contrived one for illustrative purposes only while the second one is a realistic example based on actual gene expression data.

### 5.1. A simple illustrative example

In this subsection, we present an example of a PBN with control and work through the details to show how (4.18), (4.19) can be used in arriving at an optimal control strategy. The example we consider is adapted from Example 1 in Shmulevich et al. (2002a). That example involves a PBN with three genes,  $x_1, x_2, x_3$ . There are two functions  $f_1^{(1)}, f_2^{(1)}$  associated with  $x_1$ , one function  $f_1^{(2)}$  associated with  $x_2$ , and two functions  $f_1^{(3)}, f_2^{(3)}$  associated with  $x_3$ . These functions are given by the truth table shown in Table 1. The above truth table corresponds to an uncontrolled PBN. To introduce control, let us assume that  $x_1$  is now going to be a control input whose value can be externally switched between 0 and 1 and the states of the new PBN are  $x_2$  and  $x_3$ . To be consistent with the notation introduced in Section 3, the variables  $x_1, x_2$  and  $x_3$  will be renamed; the variable  $x_1$  now becomes  $u_1$  while the variables  $x_2$  and  $x_3$  become  $x_1$  and  $x_2$  respectively. With this change, we have the truth table shown in Table 2 which also contains the values of the variables  $v$  and  $z$  corresponding to  $u_1$  and  $[x_1, x_2]$  respectively. The values of  $c_j^{(i)}$  in the table dictate that there are two possible networks, the first corresponding to the choice of functions  $(f_1^{(1)}, f_1^{(2)})$  and the second corresponding to the choice of functions  $(f_1^{(1)}, f_2^{(2)})$ . The probabilities  $P_1$  and  $P_2$  associated with each of these networks is given by  $P_1 = P_2 = 0.5$ . We next proceed to compute the matrices  $A(1)$  and  $A(2)$  corresponding to the two possible values for  $v$ .

From Table 2, it is clear that when  $v = 1$ , the following transitions are associated with the network  $N_1$  and occur with probability  $P_1$ :

$$z = 1 \rightarrow z = 1, \quad z = 2 \rightarrow z = 3, \quad z = 3 \rightarrow z = 3, \quad z = 4 \rightarrow z = 2. \quad (5.20)$$

Table 1. Truth table for Example 1 in Shmulevich et al. (2002a).

$x_1 x_2 x_3$	$f_1^{(1)}$	$f_2^{(1)}$	$f_1^{(2)}$	$f_1^{(3)}$	$f_2^{(3)}$
000	0	0	0	0	0
001	1	1	1	0	0
010	1	1	1	0	0
011	1	0	0	1	0
100	0	0	1	0	0
101	1	1	1	1	0
110	1	1	0	1	0
111	1	1	1	1	1
$c_j^{(i)}$	0.6	0.4	1	0.5	0.5

Table 2. Truth table for the example of this section.

$u_1$	$v$	$x_1$	$x_2$	$z$	$f_1^{(1)}$	$f_1^{(2)}$	$f_2^{(2)}$
0	1	0	0	1	0	0	0
0	1	0	1	2	1	0	0
0	1	1	0	3	1	0	0
0	1	1	1	4	0	1	0
1	2	0	0	1	1	0	0
1	2	0	1	2	1	1	0
1	2	1	0	3	0	1	0
1	2	1	1	4	1	1	1
$c_j^{(i)}$					1	0.5	0.5

The corresponding transitions associated with network  $N_2$  that occur with probability  $P_2$  are given by:

$$z = 1 \rightarrow z = 1, \quad z = 2 \rightarrow z = 3, \quad z = 3 \rightarrow z = 3, \quad z = 4 \rightarrow z = 1. \quad (5.21)$$

In view of (5.20) and (5.21), the matrix  $A(1)$  is given by

$$A(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ P_2 & P_1 & 0 & 0 \end{bmatrix}. \quad (5.22)$$

Similarly, we can arrive at the following  $A(2)$  matrix:

$$A(2) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & P_2 & P_1 \\ P_2 & P_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.23)$$

In this example,  $n = 2$  so that the variable  $z$  can take on any one of the four values 1, 2, 3, 4. Also since  $m = 1$ , the control variable  $v$  can take on any one of the two values 1, 2. Suppose that the control action is to be carried out over 5 steps so that  $M = 5$ . Moreover, assume that the terminal penalties are given by

$$C_5(1) = 0, \quad C_5(2) = 1, \quad C_5(3) = 2, \quad C_5(4) = 3. \quad (5.24)$$

Note that the above choices of  $M$  and the values of the terminal penalties are completely arbitrary; in a real-world example, this information would be obtained from biologists. The

current choice of terminal penalties indicates that the most desirable terminal state is 1 while the least desirable terminal state is 4. To set up the optimization problem (3.14), (3.15), we need to define the function  $C_k(z(k), v(k))$ . For the sake of simplicity, let us define

$$C_k(z(k), v(k)) = \sum_{i=1}^m u_i(k) = u_1(k) \quad (5.25)$$

where  $v(k)$  and  $u_i(k)$ ,  $i = 1, 2, \dots, m$  are related by (3.8). Clearly, the cost  $C_k(z(k), v(k))$  captures the cost of applying the input  $u_1(k)$  at the  $k$ th step. The optimization problem (3.14), (3.15) can now be posed using the quantities defined in (5.22), (5.23), (5.24), (5.25). The dynamic programming algorithm resulting from (4.18), (4.19) becomes:

$$J_5(z(5)) = C_5(z(5)) \quad (5.26)$$

$$J_k(z(k)) = \min_{v(k) \in \{1,2\}} \left[ u_1(k) + \sum_{j=1}^4 a_{z(k),j}(v(k)) \cdot J_{k+1}(j) \right], \quad k = 0, 1, 2, 3, 4. \quad (5.27)$$

We proceed backwards step by step from  $k = 4$  to obtain a solution to (5.26), (5.27). The details are given in the Appendix.

The optimal control strategy for this finite horizon problem is:

$$\begin{aligned} \mu_0^*(z(0)) = \mu_1^*(z(1)) = \mu_2^*(z(2)) = \mu_3^*(z(3)) = 1 \text{ for all } z(0), z(1), z(2), z(3) \\ \mu_4^*(z(4)) = \begin{cases} 2 & \text{if } z(4) = 3 \\ 1 & \text{otherwise.} \end{cases} \end{aligned} \quad (5.28)$$

Thus the control input is applied only in the last time step provided the state  $z$  of the system at that time step is equal to 3; otherwise, the optimal control strategy is to not apply any control at all. Let us now consider a few different initial states  $z(0)$  and see whether the optimal control strategy determined above makes sense.

*Case 1.*  $z(0) = 1$ : According to (5.28), (5.29), (5.22), the optimal control strategy in this case is no control. Note from (5.24) that the evolution of the probabilistic Boolean network is starting from the most desirable terminal state. Furthermore, from (5.22), it is clear that in the absence of any control, the state of the network remains at this position. Hence, the control strategy arrived at is, indeed, optimal and the value of the optimal cost is 0.

*Case 2.*  $z(0) = 4$ : In this case, from (5.24), it is clear that the evolution of the probabilistic Boolean network is starting from the most undesirable terminal state. Moreover, from (5.23) note that if the control input was kept turned ON over the entire control horizon, then the state would continue to remain in this most undesirable position during the entire control duration. Such a control strategy cannot be optimal since not only does the network end up in the most undesirable terminal state but also the maximum possible control cost is incurred over the entire time horizon.

To get a more concrete feel for the optimal control strategy, let us focus on the cases where the probabilistic Boolean network degenerates into a standard (deterministic) Boolean network. There are two cases to consider:

(i)  $P_2 = 1, P_1 = 0$ : In this case, from (5.22) we have

$$A(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (5.30)$$

Clearly, if no control is employed then, starting from  $z(0) = 4$ , the network will reach the state  $z(1) = 1$  in one step and stay there forever, after. Thus this no-control strategy is, indeed, optimal and the optimal cost is 0 which does agree with the value determined from (7.10) with  $P_1 = 0$ .

(ii)  $P_2 = 0, P_1 = 1$ : In this case, from (5.22), (5.23) we have

$$A(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad A(2) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.31)$$

Note from (5.28) that the optimal control strategy is no control over the first four time steps. From (5.31) it follows that with  $z(0) = 4$ , we will have  $z(1) = 2, z(2) = 3, z(3) = 3$  and  $z(4) = 3$ . Then at the last time step, the control input is turned ON and from (5.31), the resulting state is  $z(5) = 2$ . The optimal cost is given by 2 (the sum of the terminal cost and the cost of control) and this value agrees with that determined from (7.10) with  $P_1 = 1$ .

## 5.2. A real world example based on gene expression data

In this subsection, we apply the methodology of this paper to derive an optimal intervention strategy for a particular gene regulatory network. The network chosen as an example of how control might be applied is one developed from data collected in a study of metastatic melanoma (Bittner et al., 2000). In this expression profiling study, the abundance of messenger RNA for the gene WNT5A was found to be a highly discriminating difference between cells with properties typically associated with high metastatic competence versus those with low metastatic competence. These findings were validated and expanded in a second study (Weeraratna et al., 2002). In this study, experimentally increasing the levels of the Wnt5a protein secreted by a melanoma cell line via genetic engineering methods directly altered the metastatic competence of that cell as measured by the standard in vitro assays for metastasis. A further finding of interest in the current study was that an intervention that blocked the Wnt5a protein from activating its receptor, the use of an antibody that binds Wnt5a protein, could substantially reduce Wnt5a's ability to induce a metastatic phenotype. This of course suggests a study of control based on interventions that alter the contribution of the WNT5A gene's action to biological regulation, since the available data suggests

that disruption of this influence could reduce the chance of a melanoma metastasizing, a desirable outcome.

The methods for choosing the genes involved in a small local network that includes the activity of the WNT5A gene and the rules of interaction have been described in Kim et al. (2002). As discussed in that paper, the WNT5A network was obtained by studying the predictive relationship between 587 genes. The expression status of each gene was quantized to one of three possible levels:  $-1$  (down-regulated),  $0$  (unchanged) and  $1$  (up-regulated). Thus in this case, the gene activity profile at any time step is not a binary number but a *ternary one*. However, the PBN formulation and the associated control strategy can be developed exactly as described in Sections 2, 3 and 4 with the only difference that now for an  $n$ -gene network, we will have  $3^n$  states instead of the  $2^n$  states encountered earlier. In this context, it is appropriate to point out that to apply the control algorithm of this paper, it is not necessary to actually construct a PBN; all that is required are the transition probabilities between the different states under the different controls.

A network with 587 genes will have  $3^{587}$  states which is an intractably large number to use either for modeling or for control. Consequently, the number of genes was narrowed down to the ten most significant ones and the resulting multivariate relationship, using the best three-gene predictor for each gene, is shown in figure 1. These relationships were developed using the COD (Coefficient of Determination) technique (Dougherty, Kim, & Chen, 2000; Kim et al., 2000a, 2000b) applied to the gene expression patterns across 31 different conditions and prior biological knowledge. A detailed description of this is available in Kim et al. (2002).

The control objective for this 10-gene network is to externally down-regulate the WNT5A gene. The reason is that it is biologically known that WNT5A ceasing to be down-regulated is strongly predictive of the onset of metastasis. Controlling the 10-gene network using dynamic programming would require us to design a control algorithm for a system with  $3^{10}$  ( $=59,049$ ) states. Although there is nothing conceptually difficult about doing this, it is beyond the computational limits of our current software, which we are in the process of improving.

Accordingly, we further narrowed down the number of genes in the network to 7 by using COD analysis on the 31 samples. The resulting genes along with their multivariate relationship are shown in figure 2. For each gene in this network, we determined their two best two-gene predictors and their corresponding COD's. Using the procedure discussed in Shmulevich et al. (2002a), the COD information for each of the predictors was then used to determine the  $3^7 \times 3^7$  matrix of transition probabilities for the Markov Chain corresponding to the dynamic evolution of the gene-activity profile of the seven gene network.

The optimal control problem can now be completely specified by choosing (i) the treatment/intervention window, (ii) the terminal penalty and (iii) the types of controls and the costs associated with them. For the treatment window, we arbitrarily chose a window of length 5, i.e. control inputs would be applied only at time steps 0, 1, 2, 3 and 4. The terminal penalty at time step 5 was chosen as follows. Since our objective is to ensure that WNT5A is down regulated, we assigned a penalty of zero to all states for which WNT5A equals  $-1$ , a penalty of 3 to all states for which WNT5A equals 0 and a penalty of 6 to all states for which WNT5A equals 1. Here the choice of the numbers 3 and 6 is arbitrary but they do

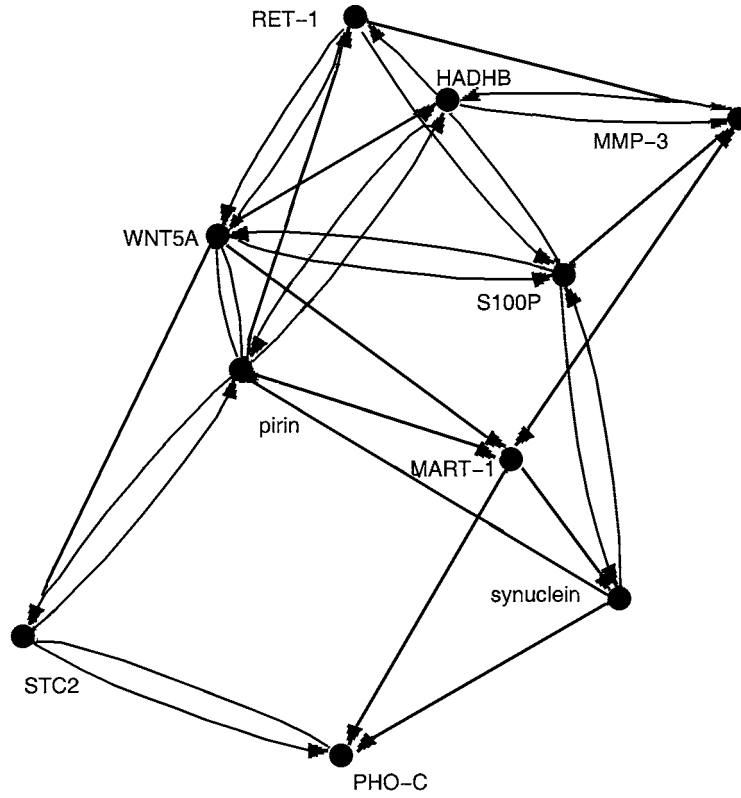


Figure 1. Multivariate relationship between the genes of the 10-gene WNT5A network (Kim et al., 2002).

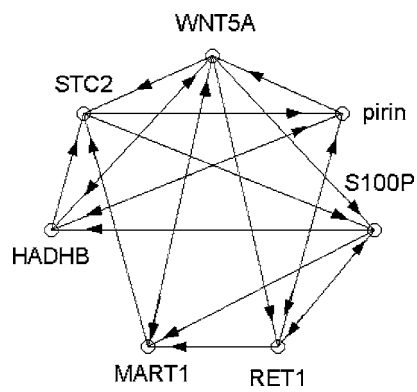


Figure 2. Multivariate relationship between the genes of the 7-gene WNT5A network.

reflect our attempt to capture the intuitive notion that states where WNT5A equals 1 are less desirable than those where WNT5A equals 0. Two types of possible controls were used and next we discuss the two cases separately.

*Case 1. WNT5A Controlled Directly:* In this case, the control action at any given time step is to force WNT5A equal to  $-1$ , if necessary, and let the network evolve from there. Biologically such a control could be implemented by using a WNT5A inhibitory protein. In this case, the control variable is binary with 0 indicating that the expression status of WNT5A has not been *forcibly* altered while 1 indicates that such a forcible alteration has taken place. Of course, whether at a given time step, such intervention takes place or not is decided by the solution to the resulting dynamic programming algorithm and the actual state of the network immediately prior to the intervention. With this kind of intervention strategy, it seems reasonable to incur a control cost at a given time step if and only if the expression status of WNT5A has to be forcibly changed at that time step. Once again, we arbitrarily assigned a cost of 1 to each such forcible change and solved for the optimal control using dynamic programming.

The net result was a set of optimal control inputs for each of the 2187 ( $=3^7$ ) states at each of the five time points. Using these control inputs, we studied the evolution of the state probability distribution vectors with and without control. For every possible initial state, our simulations indicated that at every time step from 1 to 5, the probability of WNT5A being equal to  $-1$  was higher with control than that without control. Furthermore, with control, WNT5A always reached  $-1$  at the final time point ( $k = 5$ ). Thus, we conclude that the optimal control strategy of Sections 3 and 4 was, indeed, successful in achieving the desired control objective. In this context, it is significant to point out that if the network starts from the initial state  $STC2 = -1$ ,  $HADHB = 0$ ,  $MART-1 = 0$ ,  $RET-1 = 0$ ,  $S100P = -1$ ,  $pirin = 1$ ,  $WNT5A = 1$  and if no control is used, then it quickly transitions to a *bad absorbing* state (absorbing state with  $WNT5A = 1$ ). With optimal control, however, this does not happen.

*Case 2. WNT5A Controlled Through pirin:* In this case, the control objective is the same as in Case 1, namely to keep WNT5A down-regulated. The only difference is that this time, we use another gene, pirin to achieve this control. The treatment window and the terminal penalties are kept exactly the same as before. The control action consists of either *forcing* pirin to  $-1$  (corresponding to a control input of 1) or letting it remain wherever it is (corresponding to a control input of 0). As before, at any step, a control cost of 1 is incurred if and only if pirin has to be *forcibly* reset to  $-1$  at that time step. Having chosen these design parameters, we implemented the dynamic programming algorithm with pirin as the control.

Using the resulting optimal controls, we studied the evolution of the state probability distribution vectors with and without control. For every possible initial state, our simulations indicated that, at the final state, the probability of WNT5A being equal to  $-1$  was higher with control than that without control. In this case, there was, however, no definite ordering of probabilities between the controlled and uncontrolled cases at the intermediate time points. Moreover, the probability of WNT5A being equal to  $-1$  at the final time point was not, in general, equal to 1. This is not surprising given that, in this case, we are trying to control

the expression status of WNT5A using another gene and the control horizon of length 5 simply may not be adequate for achieving the desired objective with such a high probability. Nevertheless, even in this case, if the network starts from the state corresponding to  $STC2 = -1$ ,  $HADHB = 0$ ,  $MART-1 = 0$ ,  $RET-1 = 0$ ,  $S100P = -1$ ,  $pirin = 1$ ,  $WNT5A = 1$  and evolves under optimal control, then the probability of  $WNT5A = -1$  at the final time point equals 0.673521. This is quite good in view of the fact that the same probability would have been equal to zero in the absence of any control action.

## 6. Concluding Remarks

In this paper, we have introduced probabilistic Boolean networks with one or more control inputs. In contrast to the PBNs introduced in Shmulevich et al. (2002a), the evolution of the state of the networks considered here depends on the status of these control inputs. In the case of diseases like cancer, these control inputs can potentially be used to model the effects of treatments such as radiation, chemo-therapy, etc. on the holistic behaviour of the genes. Furthermore, the control inputs can themselves be chosen so that the genes evolve in a more “desirable fashion.” Thus the PBNs with control can be used as a modeling tool to facilitate effective strategies for therapeutic intervention.

In Shmulevich et al. (2002a), it was shown how the state evolution of a PBN can be modeled as a standard Markov Chain. In this paper, we have shown how control can be introduced into a PBN leading to a *controlled* Markov Chain. Furthermore, we also showed how the control inputs can be optimally chosen using the *Dynamic Programming* technique.

The optimal control results presented in this paper assume known transition probabilities and pertain to a finite horizon problem of known length. Their extension to the situation where the transition probabilities and the horizon length are unknown is a topic for further investigation.

## Appendix

The appendix gives a step-by-step solution of (5.26), (5.27).

*Step 4.* We compute  $J_4(z(4))$  for each of the four possible states. Here  $k = 4$  so that  $k+1 = 5$ ; also from (5.24), (5.26), we have

$$J_5(1) = 0, \quad J_5(2) = 1, \quad J_5(3) = 2, \quad J_5(4) = 3. \quad (7.1)$$

From (5.27), setting  $k = 4$ , we obtain

$$J_4(z(4)) = \min_{v(4) \in \{1,2\}} \left[ u_1(4) + \sum_{j=1}^4 a_{z(4),j}(v(4)) \cdot J_5(j) \right].$$

Now  $z(4)$  can take on any of the four values 1, 2, 3, 4. Thus, we will have to compute  $J_4(1)$ ,  $J_4(2)$ ,  $J_4(3)$  and  $J_4(4)$ .

$$\begin{aligned} \text{Now } J_4(1) &= \min_{v(4) \in \{1,2\}} [u_1(4) + a_{12}(v(4)) \cdot 1 + a_{13}(v(4)) \cdot 2 + a_{14}(v(4)) \cdot 3] \\ &\quad \text{(using (7.1))} \end{aligned}$$

$$\begin{aligned}
&= \min\{a_{12}(1) + a_{13}(1).2 + a_{14}(1).3, 1 + a_{12}(2).1 + a_{13}(2).2 + a_{14}(2).3\} \\
&\quad (\text{using (3.8)}) \\
&= \min\{0, 1 + 2\} = 0
\end{aligned}$$

and  $\mu_4^*(1) = 1$ . Similarly,

$$\begin{aligned}
J_4(2) &= \min_{v(4) \in \{1,2\}} [u_1(4) + a_{22}(v(4)).1 + a_{23}(v(4)).2 + a_{24}(v(4)).3] \\
&= \min\{a_{22}(1) + a_{23}(1).2 + a_{24}(1).3, 1 + a_{22}(2).1 + a_{23}(2).2 + a_{24}(2).3\} \\
&= \min\{2, 1 + 2P_2 + 3P_1\} \\
&= 2 \text{ (since } P_1 + P_2 = 1)
\end{aligned}$$

and  $\mu_4^*(2) = 1$ . Again,

$$\begin{aligned}
J_4(3) &= \min_{v(4) \in \{1,2\}} [u_1(4) + a_{32}(v(4)).1 + a_{33}(v(4)).2 + a_{34}(v(4)).3] \\
&= \min\{a_{32}(1) + a_{33}(1).2 + a_{34}(1).3, 1 + a_{32}(2).1 + a_{33}(2).2 + a_{34}(2).3\} \\
&= \min\{2, 1 + P_1\} \\
&= 1 + P_1 \text{ (since } P_1 < 1)
\end{aligned}$$

and  $\mu_4^*(3) = 2$ . Finally,

$$\begin{aligned}
J_4(4) &= \min_{v(4) \in \{1,2\}} [u_1(4) + a_{42}(v(4)).1 + a_{43}(v(4)).2 + a_{44}(v(4)).3] \\
&= \min\{a_{42}(1) + a_{43}(1).2 + a_{44}(1).3, 1 + a_{42}(2).1 + a_{43}(2).2 + a_{44}(2).3\} \\
&= \min\{P_1, 1 + 3\} \\
&= P_1 \text{ (since } P_1 < 1)
\end{aligned}$$

and  $\mu_4^*(4) = 1$ . Thus the net result is

$$J_4(1) = 0, J_4(2) = 2, J_4(3) = 1 + P_1, J_4(4) = P_1 \quad (7.2)$$

and

$$\mu_4^*(1) = 1, \mu_4^*(2) = 1, \mu_4^*(3) = 2, \mu_4^*(4) = 1. \quad (7.3)$$

*Step 3.* Again we compute  $J_3(z(3))$  for each of the four possible states  $z(3) = 1, 2, 3, 4$  using the values  $J_4(1), J_4(2), J_4(3), J_4(4)$  obtained in the previous stage:

$$\begin{aligned}
\text{Now } J_3(1) &= \min_{v(3) \in \{1,2\}} [u_1(3) + a_{12}(v(3)).J_4(2) + a_{13}(v(3)).J_4(3) + a_{14}(v(3)).J_4(4)] \\
&= \min\{a_{12}(1).2 + a_{13}(1).(1 + P_1) + a_{14}(1).P_1, 1 + a_{12}(2).2 \\
&\quad + a_{13}(2).(1 + P_1) + a_{14}(2).P_1\} \\
&= \min\{0, 1 + (1 + P_1)\} \\
&= 0
\end{aligned}$$

and  $\mu_3^*(1) = 1$ .

$$\begin{aligned} J_3(2) &= \min_{v(3) \in \{1,2\}} [u_1(3) + a_{22}(v(3)).J_4(2) + a_{23}(v(3)).J_4(3) + a_{24}(v(3)).J_4(4)] \\ &= \min\{a_{22}(1).2 + a_{23}(1).(1 + P_1) + a_{24}(1).P_1, 1 + a_{22}(2).2 \\ &\quad + a_{23}(2).(1 + P_1) + a_{24}(2).P_1\} \\ &= \min\{(1 + P_1), 1 + P_2(1 + P_1) + P_1^2\} \\ &= (1 + P_1) \end{aligned}$$

and  $\mu_3^*(2) = 1$ .

$$\begin{aligned} J_3(3) &= \min_{v(3) \in \{1,2\}} [u_1(3) + a_{32}(v(3)).J_4(2) + a_{33}(v(3)).J_4(3) + a_{34}(v(3)).J_4(4)] \\ &= \min\{a_{32}(1).2 + a_{33}(1).(1 + P_1) + a_{34}(1).P_1, 1 + a_{32}(2).2 \\ &\quad + a_{33}(2).(1 + P_1) + a_{34}(2).P_1\} \\ &= \min\{(1 + P_1), (1 + 2P_1)\} \\ &= 1 + P_1 \end{aligned}$$

and  $\mu_3^*(3) = 1$ .

$$\begin{aligned} J_3(4) &= \min_{v(3) \in \{1,2\}} [u_1(3) + a_{42}(v(3)).J_4(2) + a_{43}(v(3)).J_4(3) + a_{44}(v(3)).J_4(4)] \\ &= \min\{a_{42}(1).2 + a_{43}(1).(1 + P_1) + a_{44}(1).P_1, 1 + a_{42}(2).2 \\ &\quad + a_{43}(2).(1 + P_1) + a_{44}(2).P_1\} \\ &= \min\{2P_1, 1 + P_1\} \\ &= 2P_1 \end{aligned}$$

and  $\mu_3^*(4) = 1$ . Thus

$$J_3(1) = 0, \quad J_3(2) = 1 + P_1, \quad J_3(3) = 1 + P_1, \quad J_3(4) = 2P_1 \quad (7.4)$$

and

$$\mu_3^*(1) = 1, \quad \mu_3^*(2) = 1, \quad \mu_3^*(3) = 1, \quad \mu_3^*(4) = 1. \quad (7.5)$$

*Step 2.* Computation of  $J_2(z(2))$  for each of the four possible states  $z(2) = 1, 2, 3, 4$  using the values  $J_3(1), J_3(2), J_3(3), J_3(4)$  obtained in the previous stage.

$$\begin{aligned} J_2(1) &= \min_{v(2) \in \{1,2\}} [u_1(2) + a_{12}(v(2)).J_3(2) + a_{13}(v(2)).J_3(3) + a_{14}(v(2)).J_3(4)] \\ &= \min\{a_{12}(1).(1 + P_1) + a_{13}(1).(1 + P_1) + a_{14}(1).2P_1, 1 \\ &\quad + a_{12}(2).(1 + P_1) + a_{13}(2).(1 + P_1) + a_{14}(2).2P_1\} \\ &= \min\{0, 1 + (1 + P_1)\} \\ &= 0 \end{aligned}$$

and  $\mu_2^*(1) = 1$ .

$$\begin{aligned} J_2(2) &= \min_{v(2) \in \{1,2\}} [u_1(2) + a_{22}(v(2)).J_3(2) + a_{23}(v(2)).J_3(3) + a_{24}(v(2)).J_3(4)] \\ &= \min\{a_{22}(1).(1 + P_1) + a_{23}(1).(1 + P_1) + a_{24}(1).2P_1, 1 + a_{22}(2).(1 + P_1) \\ &\quad + a_{23}(2).(1 + P_1) + a_{24}(2).2P_1\} \\ &= \min\{1 + P_1, 1 + P_2(1 + P_1) + 2P_1^2\} \\ &= 1 + P_1 \end{aligned}$$

and  $\mu_2^*(2) = 1$ .

$$\begin{aligned} J_2(3) &= \min_{v(2) \in \{1,2\}} [u_1(2) + a_{32}(v(2)).J_3(2) + a_{33}(v(2)).J_3(3) + a_{34}(v(2)).J_3(4)] \\ &= \min\{a_{32}(1).(1 + P_1) + a_{33}(1).(1 + P_1) + a_{34}(1).2P_1, 1 + a_{32}(2).(1 + P_1) \\ &\quad + a_{33}(2).(1 + P_1) + a_{34}(2).2P_1\} \\ &= \min\{1 + P_1, 1 + P_1(1 + P_1)\} \\ &= 1 + P_1 \end{aligned}$$

and  $\mu_2^*(3) = 1$ .

$$\begin{aligned} J_2(4) &= \min_{v(2) \in \{1,2\}} [u_1(2) + a_{42}(v(2)).J_3(2) + a_{43}(v(2)).J_3(3) + a_{44}(v(2)).J_3(4)] \\ &= \min\{a_{42}(1).(1 + P_1) + a_{43}(1).(1 + P_1) + a_{44}(1).2P_1, 1 + a_{42}(2).(1 + P_1) \\ &\quad + a_{43}(2).(1 + P_1) + a_{44}(2).2P_1\} \\ &= \min\{P_1(1 + P_1), 1 + 2P_1\} \\ &= P_1(1 + P_1) \end{aligned}$$

and  $\mu_2^*(4) = 1$ . Thus

$$J_2(1) = 0, \quad J_2(2) = 1 + P_1, \quad J_2(3) = 1 + P_1, \quad J_2(4) = P_1(1 + P_1) \quad (7.6)$$

and

$$\mu_2^*(1) = 1, \quad \mu_2^*(2) = 1, \quad \mu_2^*(3) = 1, \quad \mu_2^*(4) = 1. \quad (7.7)$$

*Step 1.* Computation of  $J_1(z(1))$  for each of the four possible states  $z(1) = 1, 2, 3, 4$  using the values  $J_2(1), J_2(2), J_2(3), J_2(4)$  obtained in the previous stage.

$$\begin{aligned} J_1(1) &= \min_{v(1) \in \{1,2\}} [u_1(1) + a_{12}(v(1)).J_2(2) + a_{13}(v(1)).J_2(3) + a_{14}(v(1)).J_2(4)] \\ &= \min\{a_{12}(1).(1 + P_1) + a_{13}(1).(1 + P_1) + a_{14}(1).P_1(1 + P_1), 1 \\ &\quad + a_{12}(2).(1 + P_1) + a_{13}(2).(1 + P_1) + a_{14}(2).P_1(1 + P_1)\} \\ &= \min\{0, 1 + 1 + P_1\} \\ &= 0 \end{aligned}$$

and  $\mu_1^*(1) = 1$ .

$$\begin{aligned} J_1(2) &= \min_{v(1) \in \{1,2\}} [u_1(1) + a_{22}(v(1)).J_2(2) + a_{23}(v(1)).J_2(3) + a_{24}(v(1)).J_2(4)] \\ &= \min\{a_{22}(1).(1 + P_1) + a_{23}(1).(1 + P_1) + a_{24}(1).P_1(1 + P_1), 1 \\ &\quad + a_{22}(2).(1 + P_1) + a_{23}(2).(1 + P_1) + a_{24}(2).P_1(1 + P_1)\} \\ &= \min\{(1 + P_1), 1 + P_2(1 + P_1) + P_1^2(1 + P_1)\} \\ &= 1 + P_1 \end{aligned}$$

and  $\mu_1^*(2) = 1$ .

$$\begin{aligned} J_1(3) &= \min_{v(1) \in \{1,2\}} [u_1(1) + a_{32}(v(1)).J_2(2) + a_{33}(v(1)).J_2(3) + a_{34}(v(1)).J_2(4)] \\ &= \min\{a_{32}(1).(1 + P_1) + a_{33}(1).(1 + P_1) + a_{34}(1).P_1(1 + P_1), 1 \\ &\quad + a_{32}(2).(1 + P_1) + a_{33}(2).(1 + P_1) + a_{34}(2).P_1(1 + P_1)\} \\ &= \min\{1 + P_1, 1 + P_1(1 + P_1)\} \\ &= 1 + P_1 \end{aligned}$$

and  $\mu_1^*(3) = 1$ .

$$\begin{aligned} J_1(4) &= \min_{v(1) \in \{1,2\}} [u_1(1) + a_{42}(v(1)).J_2(2) + a_{43}(v(1)).J_2(3) + a_{44}(v(1)).J_2(4)] \\ &= \min\{a_{42}(1).(1 + P_1) + a_{43}(1).(1 + P_1) + a_{44}(1).P_1(1 + P_1), 1 \\ &\quad + a_{42}(2).(1 + P_1) + a_{43}(2).(1 + P_1) + a_{44}(2).P_1(1 + P_1)\} \\ &= \min\{P_1(1 + P_1), 1 + P_1(1 + P_1)\} \\ &= P_1(1 + P_1) \end{aligned}$$

and  $\mu_1^*(4) = 1$ . Thus

$$J_1(1) = 0, \quad J_1(2) = 1 + P_1, \quad J_1(3) = 1 + P_1, \quad J_1(4) = P_1(1 + P_1) \quad (7.8)$$

and

$$\mu_1^*(1) = 1, \quad \mu_1^*(2) = 1, \quad \mu_1^*(3) = 1, \quad \mu_1^*(4) = 1. \quad (7.9)$$

*Step 0.* Computation of  $J_0(z(0))$  for each of the four possible states  $z(0) = 1, 2, 3, 4$  using the values  $J_1(1), J_1(2), J_1(3), J_1(4)$  obtained in the previous stage.

$$\begin{aligned} J_0(1) &= \min_{v(0) \in \{1,2\}} [u_1(0) + a_{12}(v(0)).J_1(2) + a_{13}(v(0)).J_1(3) + a_{14}(v(0)).J_1(4)] \\ &= \min\{a_{12}(1).(1 + P_1) + a_{13}(1).(1 + P_1) + a_{14}(1).P_1(1 + P_1), 1 \\ &\quad + a_{12}(2).(1 + P_1) + a_{13}(2).(1 + P_1) + a_{14}(2).P_1(1 + P_1)\} \\ &= \min\{0, 1 + (1 + P_1)\} \\ &= 0 \end{aligned}$$

and  $\mu_0^*(1) = 1$ .

$$\begin{aligned} J_0(2) &= \min_{v(0) \in \{1,2\}} [u_1(0) + a_{22}(v(0)).J_1(2) + a_{23}(v(0)).J_1(3) + a_{24}(v(0)).J_1(4)] \\ &= \min\{a_{22}(1).(1 + P_1) + a_{23}(1).(1 + P_1) + a_{24}(1).P_1(1 + P_1), 1 \\ &\quad + a_{22}(2).(1 + P_1) + a_{23}(2).(1 + P_1) + a_{24}(2).P_1(1 + P_1)\} \\ &= \min\{(1 + P_1), 1 + P_2(1 + P_1) + P_1^2(1 + P_1)\} \\ &= 1 + P_1 \end{aligned}$$

and  $\mu_0^*(2) = 1$ .

$$\begin{aligned} J_0(3) &= \min_{v(0) \in \{1,2\}} [u_1(0) + a_{32}(v(0)).J_1(2) + a_{33}(v(0)).J_1(3) + a_{34}(v(0)).J_1(4)] \\ &= \min\{a_{32}(1).(1 + P_1) + a_{33}(1).(1 + P_1) + a_{34}(1).P_1(1 + P_1), 1 \\ &\quad + a_{32}(2).(1 + P_1) + a_{33}(2).(1 + P_1) + a_{34}(2).P_1(1 + P_1)\} \\ &= \min\{(1 + P_1), 1 + P_1(1 + P_1)\} \\ &= 1 + P_1 \end{aligned}$$

and  $\mu_0^*(3) = 1$ .

$$\begin{aligned} J_0(4) &= \min_{v(0) \in \{1,2\}} [u_1(0) + a_{42}(v(0)).J_1(2) + a_{43}(v(0)).J_1(3) + a_{44}(v(0)).J_1(4)] \\ &= \min\{a_{42}(1).(1 + P_1) + a_{43}(1).(1 + P_1) + a_{44}(1).P_1(1 + P_1), 1 \\ &\quad + a_{42}(2).(1 + P_1) + a_{43}(2).(1 + P_1) + a_{44}(2).P_1(1 + P_1)\} \\ &= \min\{P_1(1 + P_1), 1 + P_1(1 + P_1)\} \\ &= P_1(1 + P_1) \end{aligned}$$

and  $\mu_0^*(4) = 1$ . Thus

$$J_0(1) = 0, \quad J_0(2) = 1 + P_1, \quad J_0(3) = 1 + P_1, \quad J_0(4) = P_1(1 + P_1) \quad (7.10)$$

and

$$\mu_0^*(1) = 1, \quad \mu_0^*(2) = 1, \quad \mu_0^*(3) = 1, \quad \mu_0^*(4) = 1. \quad (7.11)$$

In view of (7.3), (7.5), (7.7), (7.9), (7.11), the optimal control strategy for this finite horizon problem is given by (5.28), (5.29).

### Acknowledgments

This work was supported in part by the National Cancer Institute under Grant CA90301, by the National Human Genome Research Institute, by the National Science Foundation under Grant ECS-9903488 and by the Texas Advanced Technology Program under Grant No. 000512-0099-1999.

## Note

1. In the rest of this paper, we will be referring to  $z(k)$  as the state of the probabilistic Boolean network since, as discussed in Section 2,  $z(k)$  is equivalent to the actual state  $x(k)$ .

## References

- Bellman, R. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press.
- Bertsekas, D. P. (1976). *Dynamic Programming and Stochastic Control*. Academic Press.
- Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., Radmacher, M., Simon, R., Yakhini, Z., Bendor, A., Sampas, N., Dougherty, E., Wang, E., Marincola, F., Gooden, C., Lueders, J., Glatfelter, A., Pollock, P., Carpten, J., Gillanders, E., Leja, D., Dietrich, K., Beaudry, C., Berens, M., Alberts, D., & Sondak, V. (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, *406*:6795, 536–540.
- Dougherty, E. R., Kim, S., & Chen, Y. (2000). Coefficient of determination in nonlinear signal processing. *Signal Processing*, *80*:(10), 2219–2235.
- Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Theoretical Biology*, *22*, 437–467.
- Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. New York: Oxford University Press.
- Kauffman, S. A. & Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Theoretical Biology*, *128*, 11–45.
- Kim, S., Dougherty, E. R., Bittner, M. L., Chen, Y., Sivakumar, K., Meltzer, P., & Trent, J. M. (2000a). A general framework for the analysis of multivariate gene interaction via expression arrays. *Biomedical Optics*, *4*:(4), 411–424.
- Kim, S., Dougherty, E. R., Chen, Y., Sivakumar, K., Meltzer, P., Trent, J. M., & Bittner, M. (2000b). Multivariate measurement of gene-expression relationships. *Genomics*, *67*, 201–209.
- Kim, S., Li, H., Dougherty, E. R., Cao, N., Chen, Y., Bittner, M., & Suh, E. (2002). Can Markov chain models mimic biological regulation? *Journal of Biological Systems*, *10*, 447–458.
- Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002a). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, *18*, 261–274.
- Shmulevich, I., Dougherty, E. R., & Zhang, W. (2002b). Gene perturbation and intervention in probabilistic Boolean networks. *Bioinformatics*, *18*, 1319–1331.
- Shmulevich, I., Dougherty, E. R., & Zhang, W. (2002c). Control of stationary behavior in probabilistic Boolean networks by means of structural intervention. *Biological Systems*, *10*, 431–446.
- Weeraratna, A. T., Jiang, Y., Hostetter, G., Rosenblatt, K., Duray, P., Bittner, M., & Trent, J. M. (2002). Wnt5a signalling directly affects cell motility and invasion of metastatic melanoma. *Cancer Cell*, *1*, 279–288.

Received June 18, 2002

Revised December 23, 2002

Accepted December 23, 2002

Final manuscript January 14, 2003