Assignment of Terminal Penalties in Controlling Genetic Regulatory Networks

Ashish Choudhary, Aniruddha Datta, Michael L. Bittner and Edward R. Dougherty

Abstract— **Probabilistic Boolean Networks have been used to model inter gene relationships and their dynamic behavior. Optimal control of such networks has been proposed by turning on or off individual genes. Though this problem was solved in an earlier work by using a dynamic programming algorithm, issues regarding the assignment of terminal penalties and selection of genes for intervention were not addressed. In this work we provide an algorithm for assigning terminal penalties, taking long term uncontrolled behavior into account. We also discuss the possibility of using gene influence for pre selection of genes to be used for intervention. This is implemented for the popular WNT5A network.**

I. Introduction

Recent advances in microarray technology have made it possible to collect large scale gene expression data. Such data has been critical in verifying previously known inter gene relationships and discovering new ones. Boolean Networks (BN's) were introduced by Kauffman [1] to model such relationships. Even though the BN model was successful in explaining certain biological phenomenon this framework is not adequate for explaining the uncertainties in the relationships.

With the existence of multiple BN models that can be abducted from the data with likelihood very close to the most optimal BN model, a typical approach is to combine a subset of reasonable models in the ratio of their relative importance, than to choose the single best model from an exponential number of possible models[2]. Probabilistic Boolean Networks (PBN's) arise from such a combination of BN models[3].

As briefly reviewed in the next section, the states of a PBN form a homogeneous Markov chain with finite state space having *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.

Intervention in PBN's was proposed in [4] using a subset of genes in the network as *control inputs*. The control inputs were allowed to be ON indicating that intervention was applied to alter the expression level of the gene and OFF indicating no intervention i.e. the system evolves autonomously for the time step. The control objective was to optimally apply one or more treatments over a finite number of steps (which we refer to as a treatment horizon) to guide the system to a more desirable state, by minimizing a cost function. This cost function is dependent on the cost associated with using control and the penalty associated with each of the final destination states.

Once the cost function and the treatment window had been selected, the control problem was 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*(DP), pioneered by Bellman in the 1960's. A detailed discussion on Markov decision processes can be found in [5].

In [4] the terminal penalties for states were chosen based on the expression level of a particular gene. In this work we use a more rational way to assign terminal penalties by looking at the probability with which the system eventually slips into a desirable/undesirable attractor or a cycle. A discussion on the existence of such cycles and attractors can be found in [1]. We proceed to prove a theorem which says that under such a penalty assignment the cost function is non-increasing in the number of control steps, a result not necessarily true for any penalty assignment. We also investigate the use of gene sensitivity for choosing genes for intervention and analyze the performance of the chosen genes.

The paper is organized as follows. In Section II, we provide a brief review of probabilistic Boolean networks. In section III, we review the control problem for PBNs and its solution using the Dynamic Programming technique. Section IV contains a procedure for penalty assignment and the proof of the related theorem. Section V contains a biological example and some instances of gene selection for intervention based on influence. Section VI contains some concluding remarks.

This work was supported in part by the National Science Foundation under Grant ECS-0355227 and in part by the Translational Genomics Research Institute.

A. Choudhary is a PhD student of Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA. cdry@ee.tamu.edu

A. Datta is with Faculty of Department of Electrical Engineering, Texas A&M University, College Station, TX 77843, USA. datta@ee.tamu.edu

M. Bittner is with Translational Genomics Research Institute, 400 North Fifth Street, Suite 1600, Phoenix, AZ 85004, USA. mbittner@tgen.org

E. Dougherty is with Faculty of Department of Electrical Engineering, Texas A&M University, College Station, TX, 77843, USA and of University of Texas M.D. Anderson Cancer Center, Houston, TX 77030, USA. edward@ee.tamu.edu

II. Review of Probabilistic Boolean Networks

In this section, we provide a brief review of probabilistic Boolean networks. We focus on the aspects that are critical to the development in this paper. For a detailed and complete exposition, the reader is referred to [3].

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 of Boolean *predictor functions* $F = (f_1, ..., f_n)$, $f_i : \{0,1\}^n \to \{0,1\}$, $i = 1, ..., n$. 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 kth time step, the ith 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 $x(k) = [x_1(k), x_2(k), \cdots, x_n(k)]$. This vector is sometimes referred to as the *gene activity profile* (GAP) of the network at time k. The function f_i is the predictor function for gene i. Updating the states of all of the genes in B is done synchronously at every time step according to their predictor functions. Such BNs have been studied extensively in [1].

PBNs, the probabilistic generalization of BNs were first introduced in [3]. Now suppose that for each gene i, there is a family of $l(i)$ possible Boolean functions

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

that can be used to describe the dependency of x_i on $x_1, x_2, \cdots, x_n$. Furthermore, suppose that $f_j^{(i)}$ is selected with a probability $c_j^{(i)}$ with $\sum_{j=1}^{l(i)} c_j^{(i)} = 1$ Then the expression level of the ith gene transitions according to the equation:

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

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 $P_1, P_2, \cdots, P_N$ be the probabilities associated with the selection of each of these networks. Suppose the kth network is obtained by selecting the functional relationship $f_{i_k}^{(i)}$ for gene i, $i = 1, 2, \cdots, n$, $1 \le i_k \le l(i)$. Then, if the choice of the functional relationship for each gene is assumed to be independent of that for other genes, we have

$$P_k = \prod_{i=1}^{n} c_{i_k}^{(i)}. \tag{2}$$

The binary n-digit state vector $x(k)$ can be mapped to positive integers $z(k)$

$$z(k) = 1 + \sum_{j=1}^{n} 2^{j-1} x_j(k). \tag{3}$$

Then as $x(k)$ ranges from $00\cdots0$ to $11\cdots1$, $z(k)$ 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 $z(k)$.

Now the evolution of the states of the PBN can be described by a finite Markov chain model. By an elementary exercise in probability theory we can deduce that for any two states a and b, $\in \{1, 2^n\}$ the transition probability $Pr\{z(k+1) = a | z(k) = b\}$ is given by

$$\sum_{i=1}^{N} Pr\{z(k+1) = a | z(k) = b, \text{Network } i \text{ is selected}\}.P_i = \sum_{i \in \mathscr{I}} P_i \tag{4}$$

where $\mathscr{I} =$

$$\{i : Pr(z(k+1) = a | z(k) = b, \text{ Network } i \text{ is selected}) = 1\}.$$

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(TPM) 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

$$w(k+1) = w(k)A \tag{5}$$

where the entries of the A matrix have been determined using (4).

A possible way of quantifying the relative importance of different predictor genes on a target was introduced in [3]. The *influence* $I_j(f)$ of the gene x_j on the Boolean function f, with respect to a probability distribution of states $D(x)$ is defined as

$$I_j(f) = E_D\left[\frac{\partial f(x)}{\partial x_j}\right] \tag{6}$$

where E is the expectation operator, $\frac{\partial f(x)}{\partial x_j}$ is defined as $f(x) \oplus f(x^j)$ and x^j is defined as $(x_1, x_2, \ldots, x_{j-1}, x_j \oplus 1, x_{j+1}, \ldots, x_n)$. Essentially influence is the weighted average over states of the change in the value of function f in the event of the flipping of a variable. In the context of PBN's the influence of gene x_k on gene x_i becomes

$$I_k(x_i) = \sum_{j=1}^{l(i)} I_k(f_j^{(i)}).c_j^{(i)} \tag{7}$$

The influence matrix Γ has entries $\Gamma_{ij} = I_i(x_j)$. Also by taking the row sum we can find Γ_i which is the influence of the gene x_i on the network in general under the state distribution D. Under perfect observation D is degenerate, with Γ easy to calculate and interpret.

III. Control in PBNs and DP review

Our intervention strategy would involve the momentary turning ON or OFF of individual genes in the network and then letting the network evolve to the next time step. Such intervention is possible by using gene enhancers or repressors. For a multiple time step procedure our intervention would seek to be globally optimal.

For a PBN with n genes we could have a subset of m possible candidate genes to be used as control variables. Then at any given time step k, the row vector $u(k) \triangleq [u_1(k), u_2(k), \cdots, u_m(k)]$ describes the complete status of all the control inputs. Here the input $u_i(k) = 1$ would represent flipping the status of the ith control gene and $u_i(k) = 0$ would represent leaving the ith gene as it is. As in the case of the state vector, one can equivalently represent the control input status by decimal numbers using equation (3). As $u(k)$ takes on binary values from $[0, 0 \cdots, 0]$ to $[1, 1, \cdots, 1]$, the variable $v(k)$ ranges from 1 to 2^m.

Later on in section V, we will impose the restriction that only one preselected gene will be used for intervention purposes. In such a case, $v(k)$ will take the following form.

$$v(k) = \begin{cases} 1 & \text{if the intervention gene is flipped} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

We now proceed to derive the counterpart of equation (5) 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:

$$w(k+1) = w(k)A(v^*). \tag{9}$$

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:

$$w(k+1) = w(k)A(v(k)). \tag{10}$$

A. Finite horizon policies

Since the transition probability matrix in (10) is a function of all the control inputs $u_1(k), u_2(k), \cdots, u_m(k)$, 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. These ideas were formalized in [4] to arrive at the following finite horizon optimization problem:

Given an initial state $z(0)$,

$$\min_{\mu_0, \mu_1, \cdots, \mu_{M-1}} E\left[\sum_{k=0}^{M-1} C_k(z(k), \mu_k(z(k)) + C_M(z(M))\right] \tag{11}$$

subject to

$$Pr\{z(k+1) = j | z(k) = i, v(k)\} = a_{ij}(v(k)) \tag{12}$$

where

- $a_{ij}(v(k))$ is the ith row, jth column entry of the matrix $A(v(k))$;
- M represents the treatment/intervention window;
- $\mu_k : [1, 2, 3, \cdots, 2^n] \rightarrow [1, 2, 3, \cdots, 2^m]$, $k = 0, 1, 2, \cdots, M-1$ are functions mapping the state space into the control space;
- $C_k(z(k), v(k))$ is the one step cost of applying the control $v(k)$ at state $z(k)$;
- and $C_M(z(M))$ is the terminal cost associated with the state $z(M)$.

B. Solution Using Dynamic Programming

A technique to solve optimal control problems of the type described by (11) is *Dynamic Programming*. The solution is as follows.[1]

Since the control occurs only at time steps $0, 1, \ldots, M-1$

$$J_M(z(M)) = C_M(z(M)) \tag{13}$$

$$J_k(z(k)) = \min_{v(k) \in \{1, \cdots, 2^m\}} \{C_k(z(k), v(k)) + \sum_{j=1}^{2^n} a_{z(k),j}(v(k)) . J_{k+1}(j)\} \text{ for } k = 0, 1, \cdots, M-1. \tag{14}$$

IV. Assignment of Terminal Penalties

In this section, we develop a method for terminal penalty assignment. In [4] penalties were assigned to states based on the expression level of certain key genes which we call *penalty genes*. In particular we used WNT5A a gene known to be over expressed in metastatic melanoma.

However for a terminal state it would be more logical to look at the long term prospective behavior of the system in the absence of control. A more sophisticated way would be to use the following procedure.

- Partition the states of the Markov Chain into transient and persistent states. In biological networks persistent states usually occur as singleton attractors or cycles with very few states. With an appropriate procedure for PBN construction most attractors would be samples from the data set.
- For singleton attractors the penalty J is set according to the status of the penalty gene or genes, e.g. for the Markov Chain in Figure 1 the penalty gene is gene No.3 and if the gene is upregulated, the corresponding state penalty is $+3$.
- For a cycle the penalty is based on the fraction of time spent in states having penalty gene or genes in undesirable profile.
- For a transient state j, the penalty $J(j) = \sum_i P(S_\infty = i | S_t = j) . J(i)$, where i is a cycle or a singleton attractor.

We illustrate this procedure using the following example

[1] We could also obtain a policy tree starting from a particular initial state descending recursively M steps, may be storing intermediates for use in other sub problems. However, since the state space is finite, we used a bottom up procedure that is easy to implement and provides the complete solution i.e a table of optimal action from any state at any given time.

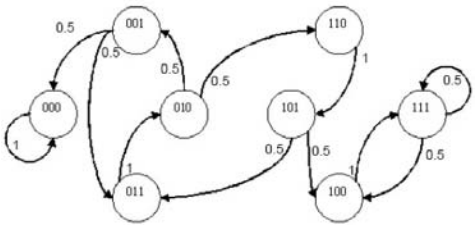


Fig. 1. Markov chain for a 3 gene PBN, $x_3 = 1$ is penalized with $+3$

Consider the markov chain in Figure 1, with upregulated penalty gene No.3 with a penalty 3. There are two persistent equivalence classes. Attractor $\{000\}$ with penalty 0 and cycle $\{100,111\}$ with penalty $\frac{1}{3}\times 0+\frac{2}{3}\times 3=2$ corresponding to the stationary distribution $\pi=[\frac{1}{3},\frac{2}{3}]$ of states $\{100,111\}$.The penalties are listed in the table I. The quantities P($\{000\}$)and P($\{100,111\}$) are the probabilities of falling in the attractor $\{000\}$ and the cycle $\{100,111\}$, respectively.

State	P($\{000\}$)	P($\{100,111\}$)	J_{eq}	J_s
000	1	0	0	0
100	0	1	2	0
010	0.5	0.5	1	0
110	0.25	0.75	1.5	0
001	0.75	0.25	.5	3
101	0.25	0.75	1.5	3
011	0.5	0.5	1	3
111	0	1	2	3

TABLE I

TERMINAL PENALTY J_{eq} IS BASED ON THE PROCEDURE IN SECTIONIV. J_s IS BASED ON THE INSTANTANEOUS STATE PROFILE. UPREGULATED GENE NO.3 IS USED AS THE PENALTY GENE WITH WEIGHT $+3$

.

A particular advantage of using the above procedure is that starting from any initial state, we can say that using more control steps is always advantageous. This is illustrated in section V. No such claim can be made for the scheme in [4].

A. Cost function and number of control steps

Let S be the set of states. We now present a proof by induction of the fact that by doing the penalty assignment using the procedure in section IV the cost function J is a non-increasing function of the number of control steps used. To do so, we first make the following observations/assumptions:

- From the definition of the terminal penalties, the following relationship holds

$$J_M(i)=\sum_{j\in S}a_{ij}(1)J_M(j) \tag{15}$$

- In equation (15) the control input $v=1$ corresponds to $u=[0,0,\ldots,0]$, the case with no control input i.e. autonomous evolution. Furthermore $C_M(i,1)=0$, since it is the cost of applying no control input.
- The cost of applying control is stationary and non-negative i.e. $C_M(i,v)=C(i,v)$ and $C(i,v)\geq 0$ for all $v\in A$, where A is the set of all possible control actions.

We now prove that the cost function for a 1 step procedure is less than that of a 0 step procedure. For any $i\in S$ consider $J_{M-1}(i)$, the one step value function. Then from (14),

$$J_{M-1}(i)=\min_{v\in A}(C(i,v)+\sum_{j\in S}a_{ij}(v).J_M(j))$$

$$=\min(\min_{v\in A-\{1\}}(C(i,v)+\sum_{j\in S}a_{ij}(v).J_M(j)),\ C(i,1)+\sum_{j\in S}a_{ij}(1).J_M(j))$$

In view of (15) and $C(i,1)=0$, we have

$$J_{M-1}(i)=\min(\min_{v\in A-\{1\}}(C(i,v)+\sum_{j\in S}a_{ij}(v).J_M(j)),J_M(i))$$

$$\text{so that } J_{M-1}(i)\leq J_M(i) \tag{16}$$

Without loss of generality assume this to hold true for an $M-k-1$ step procedure, i.e.

$$J_{K+1}(i)\leq J_{K+2}(i) \tag{17}$$

Now from (14), we have

$$J_{K+1}(i)=\min_{v\in A}(C(i,v)+\sum_{j\in S}a_{ij}(v).J_{K+2}(j))$$

Let v^* be an input that attains this minimum i.e.

$$J_{K+1}(i)=(C(i,v^*)+\sum_{j\in S}a_{ij}(v^*).J_{K+2}(j)) \tag{18}$$

Now consider the step K:

$$J_K(i)=\min_{v\in A}(C(i,v)+\sum_{j\in S}a_{ij}(v).J_{K+1}(j)) \tag{19}$$

$$=\min(\min_{v\in A-v^*}(C(i,v)+\sum_{j\in S}a_{ij}(v).J_{K+1}(j)),\ C(i,v^*)+\sum_{j\in S}a_{ij}(v^*).J_{K+1}(j))$$

$$\Rightarrow J_K(i)\leq C(i,v^*)+\sum_{j\in S}a_{ij}(v^*).J_{K+1}(j) \tag{20}$$

Now using (18) and (20) we get,

$$J_K(i)-J_{K+1}(i)\leq\sum_{j\in S}a_{ij}(v^*)\{J_{K+1}(j)-J_{K+2}(j)\}$$

Now using (17) we have $J_{K+1}(j)\leq J_{K+2}(j)\ \forall j\in S$,

$$\Rightarrow J_K(i)\leq J_{K+1}(i) \tag{21}$$

Hence for any initial state $i\in S$, the value function $J_K(i)$ is a non increasing function of the number of control time steps used.

V. Example Based on Gene Expression Data

In this section, we apply the methodology of this paper to derive an optimal intervention strategy for a particular gene regulatory network. The network chosen is one developed from data collected in a study of metastatic melanoma [6]. In this expression profiling study, the abundance of messenger RNA for the gene WNT5A was found to be highly discriminating 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 [7]. 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 by 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 [8]. 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.

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 10 most significant ones. This was done using the Coefficient of Determination(COD) technique [9] applied to the gene expression patterns across 31 different stress conditions and prior biological knowledge. Subsequent reduction from 3 to 2 levels is done in a way to minimize the loss of entropy for every individual gene. For many genes this process is lossless since they are observed in only 2 of the 3 expression levels.

The companion website [10] shows this 10 gene network and provides insights to the determination of the $2^{10} \times 2^{10}$ matrix of transition probabilities for the Markov Chain corresponding to the dynamic evolution of the gene-activity profile of the 10 gene network. The predictors and functions were determined from the data using COD analysis.

Here it would be appropriate to point out that to apply the 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.

The control objective for this 10-gene network is to externally down-regulate the WNT5A gene. As explained earlier, the reason is that it is biologically known that WNT5A ceasing to be down-regulated is strongly predictive of the onset of metastasis.

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.

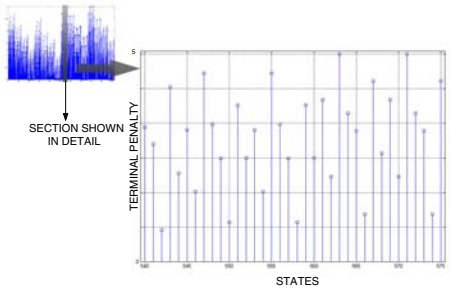


Fig. 2. Terminal penalty with WNT5A as the penalty gene

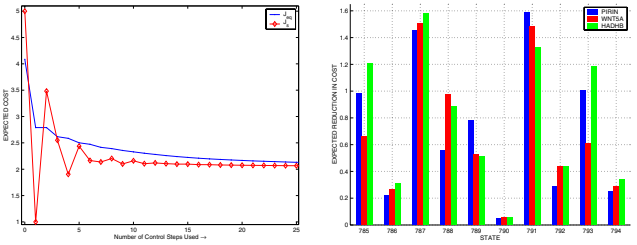
We next consider two different aspects of the control scheme:

A. Comparison of the two Schemes

We used the procedures in section IV and [4] to assign the terminal penalties J_{eq} and J_s respectively, using WNT5A as a penalty gene with a penalty of $+5$, as shown in Figure 2. In the optimal control problem, we used gene 1 (PIRIN) for intervention purposes. In particular consider the state 791 a data point corresponding to $[0,1,1,0,1,0,0,0,1,1]$ as the initial state. For the scheme based on states we observe that the value function J_s is not monotonic. Nevertheless we observe that after a certain number of steps the expected cost function decreases monotonically (12 steps in this case). This lack of monotonicity complicates the problem of selection of an appropriate control horizon particularly if the control horizon cannot be too large. We believe that the number of steps upto which the oscillations occur is related to the distance of the states in the network from the attractors. This is a topic still under investigation. Using the terminal penalty based on equivalence classes mitigates this problem. It is guaranteed that starting from any initial state, using additional control steps, we cannot do any worse even in the short term (Figure 3.1).

B. Selection of genes for intervention

For the purposes of intervention, in theory we could flip a number of genes. However from a biological perspective we would want the intervention to be minimal. Thus it makes sense to choose a particular gene, that is likely to be the most effective in bringing about the desired intervention.



(3.1) (3.2)

Fig. 3. (3.1)With initial state 791 the expected cost is plotted for control with two different types of terminal penalty assignments, J_s based on the individual states and J_{eq} based on equivalence classes. (3.2)Expected reduction in cost by using control for 5 time steps. Observe that different genes dominate when 787,788 and 789 are the initial states.

In principle the optimal control problem could be solved for each gene and then the best gene chosen. However this would be a computationally demanding procedure. Here we suggest two different heuristic approaches for gene selection and compare their performance for the WNT5A example. These two approaches are based on (1) gene influence and (2) a one step control(with 0 control cost).

Gene influence as described in section II is a property of the underlying PBN and depends only on the state distribution. It is independent of the cost of control, terminal penalties and time steps. Unlike the optimal control problem which would have to be solved every time the cost functions are changed gene influence has to be calculated only once. We could use gene influence to narrow down the pool, that can then be studied using dynamic programming. This in essence is similar to the popular feature selection problem in the pattern recognition area.

One of the ten genes is to be preselected to be used as control. At each time step the control action is chosen according to equation (14) as either flipping that gene or leaving it as is. We found that genes 1(PIRIN), 2(WNT5A) itself and 8(HADHB) dominate other genes in reducing the expected cost after 5 steps of control from any of the 2^{10} initial states. However there is no one particular gene that performs better than other genes for all initial states. This is clear from Figure 3.2.

This motivated us to use the rank expectation to rank the genes. We used a uniform distribution over

- S: All $2^{10} = 1024$ states.
- S_{DATA}: States in the dataset.
- $S_{DATA\_WNT5A=1}$: States in dataset with WNT5A upregulated(9 in number).

In general we observed that the influence heuristic performs better if the number of states over which the ranks are averaged are in particular, the states which need more intervention. The heuristic does not perform well when we use averaging over all states since the majority of states need very little or no intervention.

We also found that gene influence was very effective in ruling out genes that should not be used for intervention. For the WNT5A network we discovered that the set of genes with least influence matched very closely the set of genes which were least effective when used for intervention. In particular the set of genes ranked in the bottom 20% by influence matched the set ranked by expected cost reduction in the 5 step optimal control with an accuracy ranging from $50-100\%$ for all states. We display the detailed results on the companion website [10].

VI. Concluding Remarks

In this paper we have refined our method of assignment of terminal penalties based on the individual state profile[4] by using equivalence classes of states. We also proved that such a terminal penalty assignment ensures that using more control steps produces better results, something that is not necessarily true for individual state based assignment in the short run. We also introduced gene influence as a simple heuristic to narrow down the pool of candidate genes to be used for intervention purposes by selecting genes with high influence or more so by rejecting genes with low influence. This is important since the states in the network grow exponentially with the number of variables, and it may not be possible to check all candidate genes using the dynamic programming approach. The optimal control results presented here assume that Boolean networks are combined in a certain way to produce PBNs. Appropriate methods for combining such biological models is still an open question. The redundancy in biological data and the causal inference procedures in general complicate the abduction of networks. The sensitivity of the performance of control to the accuracy of inference is also not well understood at the present time and a topic for further investigation.

References

[1] S. A. Kauffman, *The Origins of Order: Self-organization and Selection in Evolution*, Oxford University Press, New York, 1993.

[2] W. Burtine, " A guide to the Literature on Learning Probabilistic Networks from Data,"*IEEE Transactions on Knowledge and Data Engineering*,Vol. 8, No .2, 195-210, 1994.

[3] I. Shmulevich, E. R. Dougherty, S. Kim & W. Zhang, "Probabilistic Boolean Networks: A Rule-based Uncertainty Model for Gene Regulatory Networks," *Bioinformatics*, 18, 261-274, 2002.

[4] A. Datta, A. Choudhary, M. L. Bittner & E. R. Dougherty, "External Control in Markovian Genetic Regulatory Networks," *Machine Learning*, Vol. 52, 169-191, 2003.

[5] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley, 1994.

[6] M. L. Bittner *et al*, "Molecular Classification of Cutaneous Malignant Melanoma by Gene Expression Profiling", *Nature*, 406, 536-540, 2000.

[7] A. T. Weeraratna, Y. Jiang, G. Hostetter, K. Rosenblatt, P. Duray, M. L. Bittner & J. M. Trent," Wnt5a Signaling Directly Affects Cell Motility and Invasion of Metastatic Melanoma", *Cancer Cell*, 1, 279-288, 2002.

[8] S. Kim, H. Li, E. R. Dougherty, N. Cao, Y. Chen, M. L. Bittner, & E. Suh," Can Markov Chain Models Mimic Biological Regulation ?" *Journal of Biological Systems*, Vol. 10, No. 4, 437-458, 2002.

[9] S. Kim, E. R. Dougherty, M. L. Bittner, Y. Chen, K. Sivakumar, P. Meltzer & J. M. Trent, "A General Framework for the Analysis of Multivariate Gene Interaction via Expression Arrays," *Biomedical Optics*, Vol. 4, No. 4, 411-424, 2000.

[10] http://ee.tamu.edu/∼cdry/acc05.pdf