A Linear Control Model for Gene Intervention in a Genetic Regulatory Network

Shu-Qin Zhang, Michael K. Ng, Wai-Ki Ching and Tatsuya Akutsu

***Abstract*— In this paper, we propose a linear control model for gene intervention in a genetic regulatory network. At each time step, finite controls are allowed to drive the network states to some target states. The objective is to achieve a target state probability distribution with a minimal control cost. The model can be formulated as a minimization problem with integer variables and continuous variables. Our experimental results show that the control model and the algorithm are efficient for gene intervention problems in genetic networks.**

***Index Terms*— genetic regulatory network, linear control, minimization problem, probabilistic boolean network.**

I. INTRODUCTION

Probabilistic Boolean networks (PBNs) have been proposed to study the dynamic behavior of gene regulatory networks [1]. It is a generalization of the standard Boolean networks. A Boolean network $G(V, F)$ consists of a set of nodes:

$$V = \{v_1, v_2, \ldots, v_s\}$$

and $v_i(t)$ represents the state (0 or 1) of v_i at time t. A list of Boolean functions:

$$F = \{f_1, f_2, \ldots, f_s\}$$

representing rules of regulatory interactions among the nodes (genes):

$$v_i(t+1) = f_i(v(t)), \quad i = 1, 2, \ldots, s,$$

where $v(t) = [v_1(t), v_2(t), \ldots, v_s(t)]^T$. The Boolean network is a deterministic model. However, gene expression is stochastic in nature and there is also experimental noise due to complex measurement process. To overcome the deterministic rigidity of a Boolean network, extension to a probabilistic setting is necessary. In a PBN, for each node, instead of having one Boolean function, there are a number of predictor functions $f_i^{(j)}$ for determining the state of gene v_i if it is chosen. For each gene v_i, $l(i)$ is the number of possible predictor functions and c_i^j is the probability that $f_i^{(j)}$ is being chosen, and it is estimated by using Coefficient of Determination (COD)[6]. By incorporating more possible Boolean functions into each gene, they are able to cope with the uncertainty, which is intrinsic to biological systems. At the same time, they also share the appealing rule-based properties of standard Boolean networks [2], [3], [4]. The dynamics of PBNs also can be understood from the Markov chain point of view. Thus the numerous theories for Markov chains can also be applied to analyze the PBNs.

Although a PBN allows for uncertainty of inter-gene relations during the dynamic process, it will evolve only according to certain fixed transition probabilities. There is no mechanism for controlling this evolution towards more desirable states. To facilitate PBNs to evolve towards some given desired directions, intervention has been studied in some different ways. It has been shown that given a target state, one can facilitate the transition to it by toggling the state of a particular gene from on to off or vice-versa [7]. However, making a perturbation or a forced intervention can only be applied at one time point. The behavior of the system thereafter still depends on the network itself. The network may eventually return to some undesirable state after many steps. Another way is by using structural intervention to change the stationary behavior of the PBNs [8]. This approach also constitutes transient intervention. Since it involves the structural intervention, it is more permanent than the first one.

To increase the likelihood of transitions from an undesirable state to a desirable one in a PBN, more auxiliary variables can be involved in the system. Such variables are called control inputs. They take the binary values: 0 or 1, which indicates that a particular intervention is ceased or actively applied. The control can be applied in finite steps, not only at one time point. In [5], the control problem is formulated as a minimization problem of some costs. Under the supervision of biologists or clinicians, the cost functions are defined as the cost of applying the control inputs in some particular states. For the terminal states, all possible states are assumed to be reachable. Higher terminal costs are assigned to the undesirable states. Then, the control problem is to minimize the total cost under the condition that each step evolution is based on the transition probability which now is a function with respect to the control inputs. Since the system is stochastic in nature, the cost is given by its expectation. The optimal control problem is solved by the technique of stochastic dynamic programming. The simulations for this model indicate that the final state will be the desirable state with higher probability when using controls. For more details, we refer readers to the paper by Datta et al.[5].

In this paper, we formulate the gene intervention problem with a linear control model which is easy to understand and implement. At each time step, finite controls can be put to drive the states to the desirable ones. The objective is to achieve a target state probability distribution with a minimal control cost. The model is formulated as a minimization

Shu-Qin Zhang, Michael K. Ng and Wai-Ki Ching are with the Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong. (email: sqzhang@hkusua.hku.hk, {mng, wkc}@maths.hku.hk).

Tatsuya Akutsu is with the Institute for Chemical Research, Kyoto University, Gokasho Uji, Kyoto 611-0011, Japan.(email: takutsu@kuicr.kyoto-u.ac.jp).

0-7803-9017-2/05/$20.00 ©2005 IEEE

problem with integer variables and continuous variables. There are many methods to solve such problems [9]. We use LINGO, a popular software for solving such minimization problem, to get the control input solutions for gene intervention.

The remainder of the paper is organized as follows. In Section 2, we give a brief review on PBNs and in Section 3, we formulate the linear control problem. In Section 4, preliminary numerical results are given to demonstrate the effectiveness of the linear control models and the efficiency of our algorithms. Finally, concluding remarks are given to discuss further research issues in Section 5.

II. Formulation of the Linear Control Model

In this section, we first review the PBNs briefly, we then present our linear control objective. We are interested in modeling the relationship among "n" genes. In such a genetic network, each gene can take one of the two binary values: 0 or 1, or one of the three ternary values: -1, 0 or 1. For the former case, 0 and 1 correspond to the case that a particular gene is not expressed and expressed. For the latter case, -1, 0 and 1 indicate that the gene is down-regulated, unchanged and up-regulated respectively. Here we assume that each gene takes binary values in the discussion.

Suppose that the activity level of gene "i" at time step "k" is denoted by $x_i(k)$ where $x_i(k) = 0$ or 1. The overall expression levels of all the genes in the network at time step k is given by the following column vector

$$x(k) = [x_1(k), x_2(k), \ldots, x_n(k)]^T.$$

This vector is referred to the *Gene Activity Profile* (GAP) of the network at time k. For $x(k)$ ranging from $[0, 0, \ldots, 0]^T$ (all entries are 0) to $[1, 1, \ldots, 1]^T$ (all entries are 1), it takes on all the 2^n possible states of the n genes.

Furthermore, for each gene x_i, there corresponds $l(i)$ possible Boolean functions:

$$f_j^{(i)}, \quad j = 1, \ldots, l(i),$$

and the probability of selecting function $f_j^{(i)}$ is $c_j^{(i)}$, where $f_j^{(i)}$ is a function with respect to $x_1, x_2, \ldots, x_n$, which shows the dependency of x_i on $x_1, x_2, \ldots, x_n$. Since $c_j^{(i)}$ are probabilities, they must satisfy the following condition:

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

For such a PBN with n genes, there are at most $N = \prod_{i=1}^{n} l(i)$ different Boolean networks. This means that there are totally N possible realizations of the network. Let f_k be the kth possible realization,

$$f_k = [f_{k_1}^{(1)}, f_{k_2}^{(2)}, \ldots, f_{k_n}^{(n)}], \quad 1 \le k_i \le l(i), \quad i = 1, 2, \ldots, n.$$

Suppose that P_k is the probability of choosing the kth Boolean network,

$$P_k = \prod_{i=1}^{n} c_{k_i}^{(i)}, \quad 1, 2, \ldots, N. \tag{1}$$

Let a and b be any two column vectors with n entries being either 0 or 1. Then

$$\begin{aligned} &Pr\{x(k+1) = a \mid x(k) = b\} \\ =&\sum_{i=1}^{N} Pr\{x(k+1) = a \mid x(k) = b, \\ &\text{the } i\text{th Network is selected}\} \cdot P_i. \end{aligned} \tag{2}$$

By letting a and b ranging from $00\ldots0$ to $11\ldots1$ independently, we can get the transition probability matrix A. For the ease of presentation, we first transform the $n-$digit binary number vector, as discussed in [1], into a decimal number by

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

As $x(k)$ ranges from $00\ldots0$ to $11\ldots1$, $y(k)$ will cover all the values from 1 to 2^n. Since the mapping from $x(k)$ to $y(k)$ is one-to-one, we can just equivalently work with $y(k)$. Let $w(k)$ be the probability distribution vector at time k, i.e.

$$w_i(k) = Pr\{y(k) = i\}, i = 1, 2, \ldots, 2^n.$$

It is straightforward to check that

$$w(k+1) = Aw(k) \tag{3}$$

where A satisfies

$$\sum_{i=1}^{2^n} A_{ij} = 1$$

and it has at most $N \cdot 2^n$ non-zero entries of the 2^n-by-2^n matrix.

Suppose that the m auxiliary variables, which are called control inputs

$$u_1, u_2, \ldots, u_m$$

are applied to the PBNs at each time step. At each time step k,

$$u(k) = [u_1(k), u_2(k), \ldots, u_m(k)]^T$$

indicates the control status. As in the PBNs, $u(k)$ can take all the possible values from $[0, 0, \ldots, 0]^T$ to $[1, 1, \ldots, 1]^T$. One can still represent the controls with the decimal numbers

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

As $u(k)$ ranges from $[0, 0, \ldots, 0]^T$ to $[1, 1, \ldots, 1]^T$, $v(k)$ can cover all the values from 1 to 2^m.

In [5], after applying the controls to the PBNs, the one-step time evolution of the probabilistic distribution vector follows the equation:

$$w(k+1) = A(v(k))w(k) \tag{4}$$

which not only depends on the initial distribution but also on the controls at each time step. By appropriately choosing the control inputs, the states of the network can be led to a more desirable direction. The control problem is then formulated as follows. Given an initial state $y(0)$, find a control law

$$\pi = \{u_0, u_1, \ldots, u_{M-1}\}$$

that minimizes the cost function:

$$J_\pi(y(0)) = E[\sum_{k=0}^{M-1} C_k(y(k), u_k(y(k))) + C_M(y(M))] \quad (5)$$

subject to the constraint

$$Pr\{y(k+1) = j \mid y(k) = i\} = a_{ji}(v(k)). \quad (6)$$

Here $C_k(y(k), v(k))$ are the costs of applying the control input $v(k)$ when the state is $y(k)$. The optimal solution of this problem 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:

$$\begin{cases} J_M(y(M)) = C_M(y(M)) \\ J_k(y(k)) = \min\limits_{v(k)\in\{1,2,\ldots,2^m\}} E\{G(y(k), v(k))\} \\ \qquad\qquad k = 0, 1, 2, \ldots, M-1. \\ G(y(k), v(k)) = C_k(y(k), v(k)) + J_{k+1}(y(k+1)) \end{cases}$$

Furthermore, if $v^*(k) = u_k^*(y(k))$ minimizes the right hand-side of (5) for each $y(k)$, the control law

$$\pi^* = \{u_0^*, u_1^*, \ldots, u_{N-1}^*\}$$

is optimal. For more details, we refer readers to Datta et al.[5].

The above control problem is to put the controls on the transition probability matrix in each time step, such that the system can evolve towards the more desirable states. For the general control problems, the controls can be transferred by a control transition matrix to the whole system such that the probability of the system evolving towards the desired direction will increase.

III. Linear Control Models

We consider a discrete linear control system:

$$w(k+1) = \alpha_k A w(k) + \beta_k B u(k). \quad (7)$$

All the assumptions are the same as the above. Here $w(k)$ is the state probabilistic distribution of all the states in the probabilistic Boolean network, from $[0, 0, \ldots, 0]^T$ to $[1, 1, \ldots, 1]^T$. The matrix A is the transition probability matrix for representing the dynamics from one time step to the next one. The matrix B is the control transition matrix and $u(k)$ is the control vector on the states with $u_i(k), i = 1, 2, \ldots, m$ taking on the binary values 0 or 1. The matrix B can be set in each column to represent the transition from one specific state to another for one particular gene. For example, we can set in the first column such that the first gene makes a transition from 0 to 1, then the first 2^{n-1} entries are 0 and the others are nonzero with the sum being equal to one in this column. Moreover, $u_i(k) = 1$ means the active control is applied at the time step k while $u_i(k) = 0$ indicates that the control is ceased. At this stage, we assume B is given and we need the biologists' or clinicians' guidance or some other methods to compute it. Through the matrix B, the controls are effectively transferred to different states of the PBN. If there are m possible controls at each time step, then the matrix B will be of size: $2^n \times m$.

Starting from the initial state or initial state probability distribution $w(0)$, one may apply the controls

$$u(0), u(1), \ldots, u(k-1)$$

to drive the probability distribution of the system to some desirable state probability distribution at instance k. The evolution of the system now depends on both the initial state probability distribution and the controls in each time step. To make $w(k)$ a probability distribution, it is straightforward to require that

$$\alpha_k + \beta_k = 1.$$

When there is no control at step k, we see that $\alpha_k = 1$. The parameter α_k refers to the intervention ability of the control in a genetic regulatory network.

We remark that the traditional discrete linear control problem does not have such parameters. The main reason is that in the traditional control problem, $w(k)$ is the state of a system. However, $w(k)$ is a probability distribution in this paper. We need to make sure that starting from the initial probability distribution, one apply controls to drive the probability distribution of the system to become some particular target distribution at the time instance k.

Given the objective state or state probability distribution at time k, we aim at finding the optimal controls:

$$u^*(0), u^*(1), \ldots, u^*(k-1),$$

such that the final state or state distribution following formula (7) is just the objective state or state distribution. For simplicity, we set one control at each time step. This means that the total of u_i at each time step should be 1. To make the terminal state to be the desirable state, we define some cost functions. We define $C_k(y_j(k), u(k))$ to be the cost when applying $u(k)$ control input at the k-th step with the state $y_j(k)$. At each step, we hope the state is the desirable state. With this definition, the expected control cost at all the states in step k becomes:

$$E[C_k(y(k), u(k))|y(k-1)]. \quad (8)$$

At each step, we simplify the model and assume that all the states can be reached. We assign a penalty cost of $C_k(y_j(k))$ to the state $y_j(k)$. Whether lower costs or higher costs are assigned depends on whether they are desirable or undesirable. We note that $C_k(y_j(k))$ is still stochastic, therefore we must take its expected value.

The control problem can now be formulated as follows. Given an initial state distribution $w(0)$, find a control law

$$u^*(k) = u_k,$$

such that

$$\min E[(C_k(y(k), u(k)) + C_{k+1}(y(k+1)))|y(k)], \quad \text{for} \quad k = 1, 2, \ldots, M-1$$

subject to

$$\begin{cases} \sum\limits_{l=1}^{m} [u(k)]_l \le 1, & [u(k)]_l \in \{0, 1\}, \\ & k = 0, 1, \ldots, M-1. \\ \alpha_k + \beta_k = 1, & k = 0, 1, \ldots, M-1. \\ \alpha_k + \beta_k \sum\limits_{i=1}^{m} [u(k)]_l = 1, & k = 0, 1, \ldots, M-1. \end{cases} \quad (9)$$

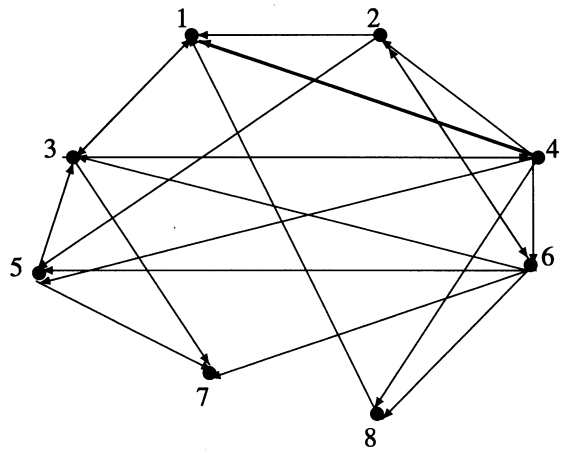

Fig. 1. Probabilistic Boolean Network of the Eight Genes

Since

$$\begin{aligned} & E[C_k(y(k),u(k))|y(k-1)] \\ = & \sum_{j=1}^{2^n} P(y_j(k)) \cdot C_k(y_j(k),u(k)) \\ = & \sum_{j=1}^{2^n} w_j(k) \cdot C_k(y_j(k),u(k)) \end{aligned}$$

the objective function can be written as follows:

$$\min \sum_{j=1}^{2^n} (w_j(k) C_k(y_j(k),u_i(k)) + w_j(k+1) C_{k+1}(y_j(k+1))),$$

where $w(k)$ is obtained from the relation (7). From (9), we see that all α_k and β_k can be represented by $u_i(k)$. Thus this formulation can be seen as only involving $k \times m$ integer variables $u_i(k)$, which constitutes an Integer Programming (IP) model.

IV. Numerical Results

In this section, we present an example to demonstrate the optimal control design by integer linear programming approach. In this example, we consider a PBN of eight genes, $x_1, x_2, \ldots, x_8$. For each gene i, we assume that it can take two values: 0 or 1. Here 1 means the gene is expressed and 0 means it is not expressed.

We further assume that there are two probabilistic Boolean functions: $f_1^{(i)}$ and $f_2^{(i)}$ associated with each gene i. All the probabilistic Boolean functions and their variables are generated randomly. At the same time, the probability of the two Boolean functions being applied to the corresponding particular gene is obtained. Fig.1 shows the network of these eight genes. Suppose that in this example, we expect gene 1 is not expressed. Then controls will be introduced to drive gene 1 from state 1 to state 0. Before solving the optimization problem formulated in the last section, we need to do the following two steps:

(a1) Obtain matrices A and B, where matrices A and B are the corresponding transition matrix and control transition matrix respectively. Since there are two Boolean functions for each gene, there are totally 2^8 networks. From (1), the probability of choosing any one of all the 2^8 can be obtained. By (2), we can get matrix A. To construct matrix B, in practice we need the opinions from the biologists to determine which gene can be easily controlled or have close relation with the target gene. For the purpose of demonstration, we will control gene 1 through all the eight genes and let them move from state 1 to state 0 with equal probability.

(a2) Determine the cost of controls and penalty for the states. We assign a cost of 1 to each forcible control. For the states penalty, since we expect gene 1 to be in state 0, we assign a penalty of 0 to the states for which gene 1 equals 0 and a penalty of 3 to all the states for which gene 1 equals 1. We choose the penalty and cost arbitrarily. In practice, we still need some criteria to determine them.

Now we can solve our optimization problem which is an integer programming problem. We choose the control such that it is only applied in three steps: $0, 1, 2$. With the popular software LINGO, we can get the solution in about one minute. The following are some results for initial state being both desirable and undesirable. It is clear to see the effect of this control strategy.

(b1) The initial state is $[0,0,0,0,0,0,0,0]$, which is the desirable state. If we do not apply any control, the probability of this state evolving to the state for which gene 1 equals 0 after three steps is 0.2492.

Under the control strategy, the probability is 0.6517, which is much more than that without any control. The control strategy is as follows: in the first step we control gene 1; in the second step, we control gene 4 and in the third step, we control gene 1 again. As we have assumed, all the corresponding genes will be made to change from 1 to 0.

(b2) The initial state is $[1,1,1,0,1,0,1,1]$, which is not the desirable state. This time if we do not apply any control, the final state will be the desirable state with the probability 0.1913, which is a very small likelihood.

However, with the optimal control, the state will evolve to the desirable state with probability 0.6379. In the first step, we need not apply any control at all; but in the second and the third step, we need to control gene 4 and and gene 1 correspondingly.

A. Computational Cost

To demonstrate the effectiveness of our model, here we made a comparison between our model and the DP model. Assume each gene can take on s states: s can be $0, 1$ or $-1, 0, 1$.

1) Computation aspect: If we get the solution of the Integer Programming model by computing all the possible values of $u_i(k)$ under the constraints and then take the one which can minimize the objective function, the cost is $O((m+1)s^{2n})$. When we apply one control at each step. We know that this cost is the most among all the methods to solve this problem. For the Dynamic Programming (DP) model, the

cost is $O(2^m s^{2n})$. The cost of IP model is much less than that of the DP model.

2) Parameter aspect: In the DP model, for each $v(k)$, there will correspond a $A(v(k))$, thus there will be 2^m matrices which are assumed to be known. In our model, all the control information is included in one matrix B. No matter how many controls we will apply, a matrix B is enough although it is still assumed to be known.

In our numerical tests, we find that our approach takes less than 2 minutes to compute the control solution under the PC with Platinum 4 and 512 kB RAM. However, for the DP approach, the PC cannot get the optimal solution in one day time.

V. Concluding Remarks

In this paper, we introduced a linear control model with the general control model form based on the PBN model of gene regulatory networks. At each step, one or more controls can be put to drive the genes to more desirable states. The control strategy can be used in the real life for therapeutic intervention. The optimal control results presented in this paper assume that the control transition matrix is known. To get a reasonable control transition matrix is our further research topic.

Acknowledgment

The authors would like to thank the support in part by RGC Grant Nos. HKU 7126/02P, 7130/02P, 7046/03P, 7035/04P, and HKU CRCG Grant Nos. 10203919, 10203907, 10204437, 10203501.

References

[1] I. Shmulevich, E. R. Dougherty, S. Kim and W. Zhang, "Probabilistic Boolean Networks: A rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, pp. 261-274, Feb. 2002.

[2] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Theoretical Biology*,vol. 22, no.3, pp. 437-467, Mar. 1969.

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

[4] S. A. Kauffman and S. Levin, "Towards a general theory of adaptive walks on rugged landscapes", *Theoretical Biology*, vol. 128, no. 1, pp. 11-45, Sep. 1987.

[5] A. Datta, M. L.Bittner and E. R. Dougherty, "External control in Markovian genetic regulatory networks", *Machine Learning*, 52, pp. 169-191, 2003.

[6] E. R. Dougherty, S. Kim and Y. Chen, "Coefficient of determination in nonlinear signal processing", *Signal Processing*, vol. 80, no. 10, pp. 2219-2235, Oct. 2000.

[7] I.Shmulevich, E. R. Dougherty and W. Zhang, "Gene perturbation and intervention in probabilistic boolean networks", *Bioinformatics*, vol. 18, pp. 1319-1331, Oct. 2002.

[8] I. Shmulevich, E. R. Dougherty and W. Zhang, "Control of stationary behavior in probabilistic boolean networks by means of structural intervention", *Biological Systems*, vol. 10, no.4, pp. 431-446, 2002.

[9] G. Sierksma, *Linear and Interger Programming: Theory and Practice*, 2nd ed. 2002.