2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A New Method and Application for Controlling the Steady-state Probability Distributions of Probabilistic Boolean Networks

Meng Yang
China Ship Development and Design Center,
Wuhan 430064, China
Email: ym_jeff@hotmail.com

Rui Li and Tianguang Chu
State Key Laboratory for Turbulence and Complex Systems,
College of Engineering, Peking University,
Beijing 100871, China
Email: rui.li@pku.edu.cn and chutg@pku.edu.cn

***Abstract*—Probabilistic Boolean networks (PBNs) have been proved to be a useful tool for modeling genetic regulatory interactions. The study of the steady-state probability distribution may help to understand the essential long-run behavior of a PBN. In this paper we focus on a type of PBNs derived from gene expression data collected in a study of metastatic melanoma. The metastatic melanoma model is usually described by a PBN containing seven genes among which WNT5A plays a significant role in the development of melanoma and is known to induce the metastasis of melanoma when highly active. This paper investigates the issue of how to drive the corresponding PBN towards desired steady-state probability distributions so as to reduce the WNT5A's ability to induce a metastatic phenotype.**

I. Introduction

Genetic regulatory networks are widely used in the research of systems biology. Understanding the dynamical behavior of them is essential to advance knowledge of disease, develop modern therapeutic methods, and identify targets in the cell needed to reach a desired goal. The dynamical modeling of gene regulation via network models constitutes a basic problem for genomics. Boolean Networks (BNs) and their extensions have received much attention as they are able to capture the switching behavior of biological processes [1]. Since gene regulation processes exhibit uncertainty and microarray data sets used to infer the model have errors due to experimental noise in the complex measurement processes, it is more realistic to consider a stochastic extension, probabilistic Boolean networks (PBNs) [2], [3]. The control of PBN plays a crucial role for analyzing the behavior of the network. Control inputs may represent therapeutic intervention strategies, such as whether a certain medicine is administered or not. Given a PBN, a natural and important problem is to study its long-run behavior [4]–[7]. It may guide the design of effective intervention strategies for the modeled systems [3]. Therapeutic gene intervention or gene control policy can therefore be developed [8], [9].

The network chosen as an example of gene control in this paper is one developed from data collected in a study of metastatic melanoma [10]. Briefly speaking, the PBN concerned consists of seven genes (including the WNT5A gene) that were chosen from a set of 587 genes that have been subjected to an analysis of their ability to cross predict each other's state in a multivariate setting [11]. Furthermore, it was found that a control strategy that reduces the WNT5A gene's action in affecting biological regulation may reduce the chance of a melanoma metastasizing, a desirable outcome. Therefore, the purpose is to control the corresponding PBN towards desired long-run behavior so that the action of the WNT5A gene in affecting biological regulation is reduced.

Once a PBN is initialized, it will evolve towards its steady-state probability distribution which may represent the essential long-run behavior. Due to the shortage of systematic tools, the issue concerning optimal control for driving a PBN to achieve desired steady-state probability distribution is far from being solved. In [12], a new systematic tool to analyze and control the BNs, called the semi-tensor product (STP) technique, has been proposed recently. By using it, the logical dynamics of BNs is converted into an algebraic iteration in terms of a set of standard difference equations. In particular, the resulting algebraic system allows for application of control theory and methods, giving rigorous analytical results, see, e.g., [13]–[22].

In this paper, we consider how to design control inputs for driving a metastatic melanoma network to desired steady-state probability distribution. We first establish an objective function which explicitly depends on the unknown control inputs, and thus transform the issue concerned into an optimization problem subject to explicit constraints. Since the solution set of the problem is quite huge, a useful algorithm with less complexity is needed. By using the computationally feasible genetic algorithm that we proposed in our former work [23], we can find an optimal solution. To our knowledge, it is the first time that the STP technique is used in the issue concerning optimal control of metastatic melanoma networks, and although very widely used, genetic algorithms are rarely used in the study of the long-run behavior of BNs.

The rest of this paper is organized as follows. Section 2 provides the necessary preliminaries. Section 3 gives the problem formation. The main results are presented in Section 4. Section 5 gives conclusion.

II. Preliminary

A. STP technique

Definition 1 ([12]). Given an $m \times n$ matrix A and a $p \times q$ matrix B, the STP of A and B is defined as

$$A \ltimes B = (A \otimes I_{l/n})(B \otimes I_{l/p}),$$

where "$\otimes$" is the Kronecker product of matrices and l is the least common multiple of n and p.

Observe that if $n = p$, then $A \ltimes B = AB$. Hence, the STP is a generalization of the conventional matrix product.

Let δ_n^i be the i-th column of the identity matrix I_n and let Δ_n be the set consisting of $\delta_n^1, \ldots, \delta_n^n$. Denote by $\mathrm{Col}(A)$ the set of columns of a matrix A and $\mathrm{Col}_i(A)$ the i-th column of matrix A. An $n \times s$ matrix L is called a logical matrix if $\mathrm{Col}(L) \subset \Delta_n$. The set of $n \times s$ logical matrices is denoted by $\mathcal{L}_{n\times s}$. We simply write an $n \times s$ logical matrix $[\delta_n^{i_1}, \delta_n^{i_2}, \ldots, \delta_n^{i_s}]$ as $\delta_n[i_1, i_2, \ldots, i_s]$. Let $\mathcal{D} = \{1, 0\}$. A logical function is a mapping $F : \mathcal{D}^n \to \mathcal{D}$.

Example 1. Consider a simple case with $n = 3$ and $F(x_1, x_2, x_3) = x_1 \vee x_2 \vee x_3$. If $x_1 = x_2 = x_3 = 0$, then $F = 0$; otherwise, $F = 1$.

We identify the elements in $\mathcal{D}$ with 2-dimensional vectors as: $1 \sim \delta_2^1$, $0 \sim \delta_2^2$. Then a logical function $F : \mathcal{D}^n \to \mathcal{D}$ can be regarded as a mapping from Δ_2^n to Δ_2. Let $x = \ltimes_{i=1}^n x_i \in \Delta_{2^n}$. For $n = 3$ case, $x_1 = [q_1, \bar{q_1}]^T$, $x_2 = [q_2, \bar{q_2}]^T$, $x_3 = [q_3, \bar{q_3}]^T$, where $q_i \in \mathcal{D}$, $i = 1, 2, 3$, and $x = x_1 \ltimes x_2 \ltimes x_3 = [q_1q_2q_3,\ q_1q_2\bar{q_3},\ q_1\bar{q_2}q_3,\ q_1\bar{q_2}\bar{q_3},\ \bar{q_1}q_2q_3,\ \bar{q_1}q_2\bar{q_3},\ \bar{q_1}\bar{q_2}q_3,\ \bar{q_1}\bar{q_2}\bar{q_3}]^T$. Thus x includes all the possible minterms of the state variables $\{x_1, x_2, x_3\}$.

Let $F(x_1, \ldots, x_n)$ be a logical function in the vector form. Referring to [12], there exists a unique logical matrix $M_F \in \mathcal{L}_{2\times 2^n}$ such that

$$F(x_1, \ldots, x_n) = M_F \ltimes_{i=1}^n x_i. \tag{1}$$

We call M_F the structure matrix of the logical mapping F, and Eq. (1) is called the algebraic form. It has been proved that the logical form is equivalent to the algebraic form and some easily computable formulae have been provided to convert one from the other. For $x_1,\ \ldots,\ x_n$, if $x_1 \ltimes \cdots \ltimes x_n = \delta_{2^n}^j$ in Eq. (1), then $F(x_1,\ \ldots,\ x_n) = M_F \ltimes \delta_{2^n}^j = \mathrm{Col}_j(M_F)$. That is, $F(x_1,\ \ldots,\ x_n)$, $\forall j \in \{1, \ldots, 2^n\}$ can be directly calculated from M_F.

In order to calculate M_F, we introduce some special logical matrices. Let M_d denote the structure matrices of the logical operator, disjunction $\vee$. Then $M_d = \delta_2[1, 1, 1, 2]$. For each $n \in \mathbb{N}$, let $\varphi(n) = \delta_{2^{2n}}[1, 2^n{+}2, 2{\cdot}2^n{+}3, \ldots, (2^n{-}1)2^n{+}2^n]$. A straightforward computation shows that $x \ltimes x = \varphi(n) \ltimes x, \quad x \in \Delta_{2^n}$. The swap matrix $W_{[m,n]}$ is defined as $W_{[m,n]} = \delta_{mn}[1, m + 1, \ldots, (n - 1)m + 1, 2, m + 2, \ldots, (n - 1)m + 2, \ldots, m, 2m, \ldots, nm]$. It is easy to verify that $x_2 \ltimes x_1 = W_{[m,n]} \ltimes x_1 \ltimes x_2,\ x_1 \in \mathbb{R}^m,\ x_2 \in \mathbb{R}^n$. The dummy logical matrix E_d is defined as $E_d = \delta_2[1, 2, 1, 2]$. Since $E_d\delta_2^1 = E_d\delta_2^2 = I_2$, it follows that $E_dx = I_2$ for $x \in \Delta_2$.

Let us go through an example to understand the process of the calculation of M_F in the STP technique.

Example 2. Consider the logical function in Example 1. We have

$$\begin{aligned} F(x_1, x_2, x_3) &= M_d \ltimes x_1 \ltimes M_d \ltimes x_2 \ltimes x_3 \\ &= M_d \ltimes (I_2 \otimes M_d) \ltimes x_1 \ltimes x_2 \ltimes x_3 \end{aligned}$$

Then the structure matrix M_F can be easily calculated as

$$M_F = \delta_2[1, 1, 1, 1, 1, 1, 1, 2].$$

For more details, readers are referred to [12].

B. Model description

PBNs can be used to model the metastatic melanoma networks. A PBN with m control input nodes consists of a set of gene nodes $V = \{v_1, \ldots, v_n\}$ and a set of logical functions $\{f^{(1)}, \ldots, f^{(N)}\}$, governing the state transitions of the nodes. Each logical function determines a BN, also called a realization, and the governing BN is randomly chosen at every time step in accordance with a fixed probability distribution. Let $X_i(t) \in \{0,\ 1\}$ be the state of v_i at time t and $U_i^{(j)} \in \{0,\ 1\}$, $j \in \{1, \ldots, N\}$ be the i-th input in the j-th realization. Notice that all the control inputs are designed to vary with the realization in this paper. More formally, given a set of logical functions, $f^{(j)} : \{0,\ 1\}^n \times \{0,\ 1\}^m \to \{0,\ 1\}^n$, $j = 1, \ldots, N$, then the dynamics of a PBN is expressed as

$$X(t + 1) = f\left(U, X(t)\right),\ t = 0, 1, 2, \ldots, \tag{2}$$

where $X(t)$ denotes the n-dimensional state variable at time t, taking value from $\{0,\ 1\}^n$, the function f is selected from among $\{f^{(1)}, \ldots, f^{(N)}\}$ at each time point, with the probability of selecting $f^{(j)}$ being the selection probability p_j, U denotes the m-dimensional control input, taking value from $\{0,\ 1\}^m$, and $U = U^{(j)}$ in the j-th realization.

In the algebraic form, let $x_i(t),\ u_i^{(j)} \in \Delta_2$, and let $x(t) = \ltimes_{i=1}^n x_i(t)$, $u^{(j)} = \ltimes_{i=1}^m u_i^{(j)}$. For the logical function of the i-th node in the j-th realization, denote its structure matrix by $M_{f_i^{j_i}}$. It follows

$$x_i(t + 1) = M_{f_i^{j_i}} \ltimes u^{(j)} \ltimes x(t),\ i = 1, \ldots, n. \tag{3}$$

For the j-th realization, by multiplying all the n state equations in Eqs. (3), we have

$$x(t + 1) = L^{(j)} \ltimes u^{(j)} \ltimes x(t), \tag{4}$$

where $L^{(j)} = M_{f_1^{j_1}} \ltimes_{i=2}^n [(I_{2^{m+n}} \otimes M_{f_i^{j_i}})\varphi(n + m)]$. We call matrix $L^{(j)}$ the network transition matrix of the j-th realization.

Example 3. Consider a PBN consisting of two genes and one control input,

$$\begin{cases} x_1(t + 1) = f_1(u_1, x_1(t), x_2(t)) \\ x_2(t + 1) = f_2(u_1, x_1(t), x_2(t)) \end{cases}$$

where

$$\begin{cases} f_1^1 = u_1 \wedge (x_1(t) \wedge x_2(t)), \\ f_1^2 = u_1 \wedge (x_1(t) \vee x_2(t)), \\ f_2^1 = x_1(t) \leftrightarrow x_2(t), \\ f_2^2 = x_1(t) \wedge x_2(t), \end{cases}$$

and $Pr(f_1 = f_1^1) = 0.2$, $Pr(f_1 = f_1^2) = 0.8$, $Pr(f_2 = f_2^1) = 0.1$, $Pr(f_2 = f_2^2) = 0.9$. The probabilities of selecting each realization are $p_1 = 0.2 \times 0.1 = 0.02$, $p_2 = 0.2 \times 0.9 = 0.18$, $p_3 = 0.8 \times 0.1 = 0.08$, $p_4 = 0.8 \times 0.9 = 0.72$. The network transition matrices of each realization are $L^{(1)} = \delta_4[1, 4, 4, 3, 3, 4,\ 4, 3]$, $L^{(2)} = \delta_4[1, 4, 4, 4, 3, 4, 4, 4]$, $L^{(3)} = \delta_4[1, 2, 2, 3, 3, 4, 4, 3]$, $L^{(4)} = \delta_4[1, 2, 2, 4, 3, 4, 4, 4]$.

Now we consider the control of PBNs in a real study of metastatic melanoma [10], [24]. The corresponding PBN

can be described by a seven-gene network containing the genes WNT5A, pirin, S100P, RET1, MART1, HADHB, and STC2. WNT5A is a gene that plays a significant role in the development of melanoma and is known to induce the metastasis of melanoma when highly active. This suggests a control strategy that reduces the action of WNT5A in affecting biological regulation. There are many ways to do this, one is to let the WNT5A controlled through pirin [25]. Using the algorithms described in [26], four highly probable Boolean networks are presented in [27] as the constituent BNs (realizations) in the PBN, each of which is assumed to be derived from steady-state gene-expression data. In [27], The states are ordered as WNT5A, pirin, S100P, RET1, MART1, HADHB, and STC2. We denote the states of gene WNT5A, S100P, RET1, MART1, HADHB, and STC2 at time t by $x_1(t),\ x_3(t),\ x_4(t),\ x_5(t),\ x_6(t),\ x_7(t)$. The state of the second gene, pirin, is deemed as the control input, and is denoted by $u^{(j)}$ in the j-th realization, $j \in \{1,2,3,4\}$. Then we have

$$\begin{aligned} &x_1(t+1) \ltimes u^{(j)} \ltimes x_3(t+1) \ltimes \cdots \ltimes x_7(t+1) \\ &\qquad = N^{(j)} x_1(t) \ltimes u^{(j)} \ltimes x_3(t) \ltimes \cdots \ltimes x_7(t), \end{aligned}$$

where logical matrices $N^{(j)} \in \mathcal{L}_{2^7 \times 2^7}$ can be easily obtained from the figures in [27].

III. Problem Formulation

Since the j-th realization can be described by Eq. (4), the overall expected value of $x(t+1)$ satisfies $Ex(t+1) = LEx(t)$, where $L = \sum_{i=1}^{N} p_i L^{(i)} \ltimes u^{(i)}$. The matrix L is called the probability transition matrix. Fix $t \geq 0$. If $Ex(t) = [\pi_1, \ldots, \pi_{2^n}]^T \in \mathbb{R}^{2^n}$, then $Pr(x(t) = \delta_{2^n}^i) = \pi_i$, $i = 1, \ldots, 2^n$. The vector $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_{2^n}]^T$ is called the steady-state probability distribution, and it satisfies

$$\boldsymbol{\pi} = L\boldsymbol{\pi}. \tag{5}$$

From Eq. (5), the steady-state probability distribution of a PBN can be easily calculated if the selection probability, the network transition matrix, and the control inputs of each realization are given.

Example 4. Reconsider the PBN in Example 3 and suppose $u_1^{(1)} = u_1^{(2)} = u_1^{(3)} = u_1^{(4)} = [0,1]^T$. Then the probability transition matrix can be easily calculated as

$$L = \sum_{i=1}^{4} p_i L^{(i)} \ltimes u_1^{(i)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0.1 \\ 0 & 1 & 1 & 0.9 \end{bmatrix}.$$

From Eq. (5), the steady-state probability distribution $\boldsymbol{\pi} = [0, 0, 0.0909, 0.9091]^T$.

Now we continue to discuss the PBNs in the study of metastatic melanoma. Without the consideration of the state of pirin in the metastatic melanoma network, let $x(t) = x_1(t) \ltimes x_3(t) \ltimes \cdots \ltimes x_7(t)$. The steady-state probability distribution of the six genes, from state $x = \delta_{64}^1$ to state $x = \delta_{64}^{64}$, is shown in Fig. 1. We observe that the steady-state probability distribution is fixed if no external control is exerted.

Then it is natural to ask how to control a PBN to evolve into a desired steady-state probability distribution. The desired steady-state probability distribution is denoted by vector

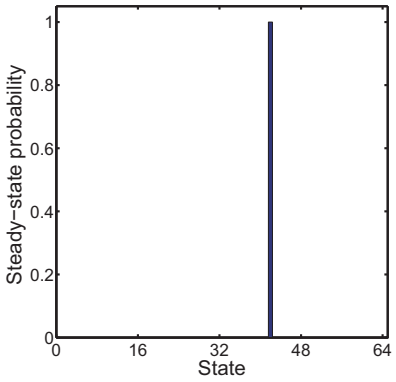

Fig. 1. The steady-state probability distribution of the six genes, WNT5A, S100P, RET1, MART1, HADHB, and STC2 without control. The 42nd element of the steady-state probability distribution vector is 1, and the other elements take the value of 0.

$\boldsymbol{\pi}^* = [\pi_1^*, \ldots, \pi_{2^n}^*]^T$ hereafter. In the potential applications of the control of genetic networks, such as drug discovery and treatment of intractable diseases, some states of the whole network may be of more concern and their probability distributions should be guaranteed firstly. Our work takes into consideration the priority of each state. The priorities of the states $x_1,\ \ldots,\ x_{2^n}$ are described by constant weight values $\omega_1, \ldots, \omega_{2^n}$, respectively, which satisfy $\omega_1 + \cdots + \omega_{2^n} = 1$. Then a measure of distance between $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^*$ is defined as

$$\| \boldsymbol{\pi} - \boldsymbol{\pi}^* \| = \sum_{i=1}^{2^n} \omega_i \mid \pi_i - \pi_i^* \mid . \tag{6}$$

Observe that the set of all possible control inputs of a PBN is finite, of cardinality 2^{mN}. Thus, in most cases, the steady-state probability distribution $\boldsymbol{\pi}$ of a prescribed PBN can not be controlled into $\boldsymbol{\pi}^*$ precisely, and we wish to solve the following problem.

Problem 1. *Consider a PBN (2). The optimal control of steady-state probability distribution is to find control inputs $u^{(1)},\ \ldots,\ u^{(N)}$ such that the distance between the steady-state probability distribution of the PBN and the desired steady-state probability distribution (6) is minimized. For the metastatic melanoma network, the aim is to reduce the steady-state probability of the states at which the WNT5A gene is active.*

IV. Main Results

Since the distance in Eq. (6) is not explicitly a function of the control inputs, we need an adequate objective function to solve Problem 1. To this end, we first used the Theorem in our former work [23].

Theorem 1 ([23])**.** *For a PBN with m control inputs (4), let the network transition matrices $L^{(j)}$, the selection probability $p_j,\ j = 1, \ldots, N$, and the desired steady-state probability distribution $\boldsymbol{\pi}^*$ be given, and let matrix $A^{(j)} = p_j L^{(j)} \ltimes W_{[2^n, 2^m]} \ltimes \boldsymbol{\pi}$, $i = 1, \ldots, N$. Suppose $u^{(j)} = \delta_{2^m}^{z_i}$. Then Problem 1 can be equivalently rewritten as the following*

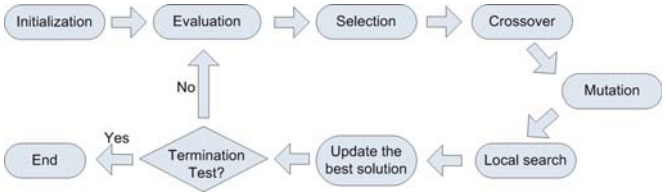

Fig. 2. An overview of the genetic algorithm.

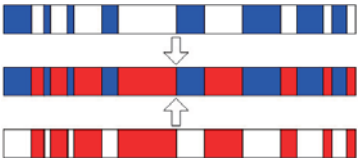

Fig. 3. Two solutions crossover.

problem:

$$\begin{array}{ll} \text{find} & \mathbf{z} = [z_1, \ldots, z_N], \\ \max & H(\mathbf{z}) = \Sigma_{i=1}^{2^n} \omega_i h_i(\mathbf{z}), \\ \text{subject to} & z_j \in \{1, \ldots, 2^m\},\ j = 1, \ldots, N, \end{array} \tag{7}$$

where function $h_i(\mathbf{z}) = -\mid \Sigma_{j=1}^{N} a_{i,z_j}^{(j)} - \pi_i^* \mid$, $a_{i,z_j}^{(j)}$ *is the* i*-th element of the column vector* $\mathrm{Col}_{z_j}(A^{(j)})$.

Observe that the steady-state probability distribution $\boldsymbol{\pi}$ and the matrix $A^{(j)}$ vary with the control inputs. When the distance between $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^*$ is minimum, matrix $A^{(j)}$ can be approximated by $p_j L^{(j)} \ltimes W_{[2^n,2^m]} \ltimes \boldsymbol{\pi}^*,\ j = 1, \ldots, N$. Using the swap matrix $W_{[2^n,2^m]}$ based on the STP technique, matrix $A^{(j)}$ can be easily obtained. Numerical simulation results proved that matrices $A^{(j)}$ provide an approximate data fit. It is worth noting the resulting objective function (7) involves no logical operators. A possible solution is a vector $\mathbf{z} = [z_1, \ldots, z_N] \in \{1, \ldots, 2^m\}^N$.

Exhaustive search is not applicable as it is not easy to check every possible vector of control inputs in each realization. So an optimization algorithm is desperately needed. The genetic algorithms are useful because they are known to cope well with a large solution space. They have been widely applied in different tasks of genetic analysis, as reviewed in [28]. Therefore, we propose a genetic algorithm that is especially suitable to solve the optimal control of the metastatic melanoma network.

The goal of the genetic algorithm is to find a proper vector $\mathbf{z} = [z_1, \ldots, z_N]$ such that the objective function (7) is maximized. Before we introduce the algorithm, we need the following definition at first.

Definition 2. For an integer vector $\mathbf{z} = [z_1, \ldots, z_N]$, integer vector $\mathbf{z}' = [z_1', \ldots, z_N']$ is said to be a neighbor of $\mathbf{z}$ if $\mid \{i \mid z_i' \neq z_i\} \mid \leq 2$.

An overview of the proposed genetic algorithm is depicted in Fig. 2. It consists of the following steps.

a) Initialization: Set parameters for the algorithm, including N_{pop}, k, ε_1 and ε_2 that will be introduced later. Randomly generate an initial set of N_{pop} solutions, which is defined as the set of concerned solutions currently and denoted by $\Psi = \{\mathbf{z}_1, \ldots, \mathbf{z}_{N_{pop}}\}$.

b) Evaluation: Each solution in Ψ is evaluated according to the objective function $H(\mathbf{z})$ in Eq. (7).

c) Selection: For each solution in Ψ, select a solution that will be used in Step "*Crossover*", which is called its parent solution. Solution $\mathbf{z}_i$, $i = 1, \ldots, N_{pop}$ in Ψ is selected as the parent solution with probability q_i, where $q_i = (H(\mathbf{z}_i) - H_{\min}(\Psi))/\sum_{\mathbf{z}\in\Psi}\{H(\mathbf{z}) - H_{\min}(\Psi)\}$, $H_{\min}(\Psi) = \min\{H(\mathbf{z}) \mid \mathbf{z} \in \Psi\}$. Obviously, the solution with higher value of the objective function is more likely to be selected.

d) Crossover: For each solution $\mathbf{z}_i \in \Psi$, $i \in \{1, \ldots, N_{pop}\}$, assume its parent solution is denoted by $\mathbf{z}_i'$. If $H(\mathbf{z}_i) < H(\mathbf{z}_i')$, then with probability ε_1, a single offspring is generated by substituting $N/2$ randomly selected elements of $\mathbf{z}_i'$ for the corresponding elements of $\mathbf{z}_i$, as shown in Fig. 3.

e) Mutation: The offspring solution mutates randomly to one of its neighbors with probability ε_2. Steps "*Selection*", "*Crossover*" and "*Mutation*" are also called "*Genetic operation*".

f) Local search procedure: Each solution $\mathbf{z}$ in Ψ is specified as an initial solution. (i) Randomly select a neighbor of $\mathbf{z}$ and denote it by $\mathbf{z}'$. Then examine the objective function values of both the current solution $\mathbf{z}$ and the neighbor $\mathbf{z}'$; (ii) If $\mathbf{z}'$ is a better solution than $\mathbf{z}$ (i.e. $H(\mathbf{z}') > H(\mathbf{z})$), replace the current solution with $\mathbf{z}'$ and return to Step (i); (iii): If k neighbors of the current solution $\mathbf{z}$ have been already examined, end this procedure. Otherwise, return to Step (i). Apply this procedure to all solutions in Ψ.

Observe that only a small number of neighbors are examined to prevent the "*Local search procedure*" from spending too much computation time. The high performance of this idea is demonstrated by successfully applying it to flowshop scheduling problems in [29].

g) Update the best solution: The local search procedure provides us a set of candidate solutions ranked by the objective function value. The best solution thus obtained among them is denoted by $\mathbf{z}^*$.

h) Termination test: If a prespecified stopping condition is satisfied (e.g., the best solution was updated 50 times), end the algorithm. Otherwise, return to Step "*Selection*".

The basic idea of the genetic algorithm is shown in Fig. 4.

Remark 1. i) The search direction in the "*Local search procedure*" indicates certain biomedical purposes, e.g. if some states of genes are more important than the others, the corresponding weight values are larger, and it is more likely that the simulation results shown later always favor one specific best solution, like z^* in Fig. 4.

ii) The "*Genetic operation*" is defined according to the optimization problem given specifically. The "*Genetic operation*" we proposed here will be proved to be well suited for the problem defined in Definition 1 by numerical simulations. Parameters in Steps "*Crossover*", "*Mutation*", and "*Local search procedure*" are changeable according to the changing

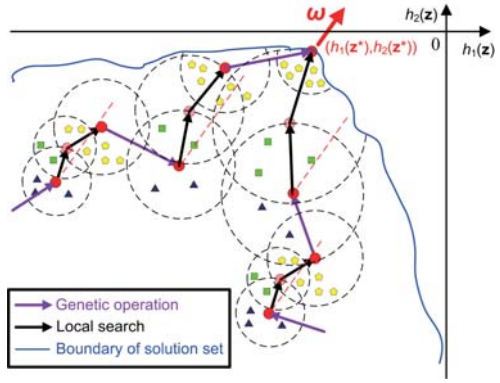

Fig. 4. The genetic algorithm is applied in a 2-dimensional objective space. The purple arrow denotes the result of the "*Genetic operation*". The blue line denotes the boundary of the solution set. Since the search direction has been defined as the red arrow $\boldsymbol{\omega}$, $\mathbf{z}^*$ is the best solution of all. Repeating the "*Genetic operation*" and the "*Local search procedure*", two different initialized solutions both lead to the best solution $\mathbf{z}^*$ in the end.

conditions, e.g. if m takes larger value, it is better to examine more neighbors in the "*Local search procedure*", which means the larger the number of neighbors k is, the better the algorithm performs.

iii) From the "*Local search procedure*", we obtain solutions that are better than their randomly selected neighbors, which can be viewed as locally optimal solutions, and thus obtain the best solution $\mathbf{z}^*$. On the other hand, the "*Genetic operation*" at the next iteration makes $\mathbf{z}^*$ move, very probably eventually moving towards the global optimal solution. In this manner, the simulation result of the genetic algorithm closely approximates to the global optimal solution. It is not easy to theoretically analyze how close the objective function value of $\mathbf{z}^*$ approaches that of the global optimal solution, but the experiments in [23] show that the algorithm works remarkably well.

Now we apply the methodology above to find control inputs to solve the optimal control problem of the steady-state distribution probability of the metastatic melanoma network with 7 genes. (So far the STP technique can be used to analyze BNs with no more than 20 genes.) The objective of the control is to let $x_1(t) = \delta_2^2$, and since each constituent BN has several attractors [27], the steady-state probability of the attractors should be higher than other states. To this end, we may assume without loss of generality the desired steady-state probability distribution $\boldsymbol{\pi}^* = [\pi_1^*, \ldots, \pi_{64}^*]^T$, where $\pi_{42}^* = 0.26$, $\pi_i^* = 0$ if $i \in \{1, \ldots, 32\}$, $\pi_i^* = 0.05$ if $i \in \{33, 36, 44\}$, $\pi_i^* = 0.1$ if $i \in \{34, 50\}$, $\pi_i^* = 0.015$ else. We assume the selection probability of each realization $p_i = 0.25$ and the weight values of all states $\omega_1 = \cdots = \omega_{64} = 1/64$. It follows from the STP technique that

$$\begin{aligned} x(t+1) &= E_d \ltimes u^{(j)} \ltimes x(t+1) \\ &= E_d \ltimes W_{[2,2]} \ltimes x_1(t+1) \ltimes u^{(j)} \ltimes \cdots \ltimes x_7(t+1) \\ &= E_d \ltimes W_{[2,2]} \ltimes N^{(j)} \ltimes x_1(t) \ltimes u^{(j)} \ltimes \cdots \ltimes x_7(t) \\ &= E_d \ltimes W_{[2,2]} \ltimes N^{(j)} \ltimes W_{[2,2]} \ltimes u^{(j)} \ltimes x(t). \end{aligned}$$

So the the network transition matrix of the j-th realization is $L^{(j)} = E_d \ltimes W_{[2,2]} \ltimes N^{(j)} \ltimes W_{[2,2]}$.

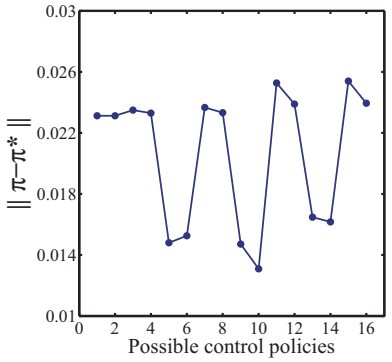

Fig. 5. The distance between $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^*$.

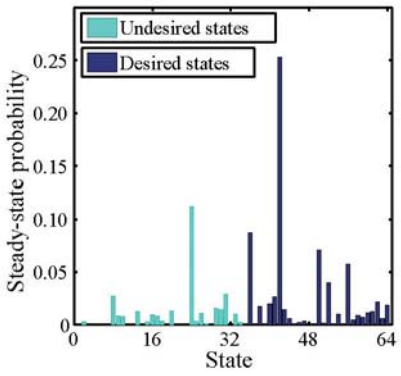

Fig. 6. The steady-state probability distribution of the six genes, WNT5A, S100P, RET1, MART1, HADHB, and STC2 after optimal control.

With the help of Theorem 1 and the genetic algorithm, the best solution is obtained as $\mathbf{z}^* = [2, 1, 1, 2]$. Hence the corresponding control inputs are $u^{(1)} = \delta_2^2$, $u^{(2)} = \delta_2^1$, $u^{(3)} = \delta_2^1$, $u^{(4)} = \delta_2^2$. Let $\delta_{16}^i = u^{(1)} \ltimes u^{(2)} \ltimes u^{(3)} \ltimes u^{(4)}$. Then $i = 10$. Since $u^{(1)} \ltimes u^{(2)} \ltimes u^{(3)} \ltimes u^{(4)} = \delta_{16}^{10}$, the result is the same as what we observed from Fig. 5, which depicts the distance between $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^*$ of all the 16 control strategies.

The steady-state probability distribution of the six genes after optimal control is shown in Fig. 6. For states from $x(t) = \delta_{64}^{33}$ to $x(t) = \delta_{64}^{64}$, the state of gene WNT5A is $x_1(t) = \delta_2^2$, which means that the action of WNT5A in affecting biological regulation is reduced. So states from $x(t) = \delta_{64}^{33}$ to $x(t) = \delta_{64}^{64}$ are the desired states. We observe that the steady-state probability of the desired states is much larger than the undesired ones.

V. Conclusion

This paper considered the issue concerning optimal control of the steady-state probability distributions of a widely studied metastatic melanoma network. Based on the STP technique, we have established an objective function which explicitly depends on the unknown control inputs and thus transformed the formulated problem into the one of maximizing the objective function. Then the genetic algorithm we proposed before can find the optimal solution, for any given weight values which indicate the importance of each element of the steady-

state probability distribution vector. In addition, the proposed genetic algorithm is always applicable no matter how large the number of realization N is. This is mainly because the number of neighbors k is adjustable in "*Local search procedure*". Since we obtain the optimal control solution of the steady-state probability distributions of the PBNs in a real biological problem, the validity of the proposed genetic algorithm is verified.

ACKNOWLEDGMENT

This work was supported by NSFC (No. 61273111) and National 973 Program (No. 2012CB821200).

REFERENCES

[1] S. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[2] I. Shmulevich, E. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[3] I. Shmulevich, R. Edward, and W. Zhang, "Control of stationary behavior in probabilistic Boolean networks by means of structural intervention," *Journal of Biological Systems*, vol. 10, no. 04, pp. 431–445, 2002.

[4] M. Brun, E. Dougherty, and I. Shmulevich, "Steady-state probabilities for attractors in probabilistic Boolean networks," *Signal Processing*, vol. 85, no. 10, pp. 1993–2013, 2005.

[5] I. Shmulevich, I. Gluhovsky, R. Hashimoto, E. Dougherty, and W. Zhang, "Steady-state analysis of genetic regulatory networks modelled by probabilistic Boolean networks," *Comparative and Functional Genomics*, vol. 4, no. 6, pp. 601–608, 2003.

[6] W. Li, L. Cui, and M. Ng, "On computation of the steady-state probability distribution of probabilistic Boolean networks with gene perturbation," *Journal of Computational and Applied Mathematics*, vol. 236, no. 16, pp. 4067–4081, 2012.

[7] S. Zhang, W. Ching, M. Ng, and T. Akutsu, "Simulation study in probabilistic Boolean network models for genetic regulatory networks," *International Journal of Data Mining and Bioinformatics*, vol. 1, no. 3, pp. 217–240, 2007.

[8] W. Ching, S. Zhang, Y. Jiao, T. Akutsu, N. Tsing, and A. Wong, "Optimal control policy for probabilistic Boolean networks with hard constraints," *IET Systems Biology*, vol. 3, no. 2, pp. 90–99, 2009.

[9] A. Datta, A. Choudhary, M. Bittner, and E. Dougherty, "External control in Markovian genetic regulatory networks," *Machine learning*, vol. 52, no. 1, pp. 169–191, 2003.

[10] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor *et al.*, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.

[11] S. Kim, H. Li, E. R. Dougherty, N. Cao, Y. Chen, M. Bittner, and E. B. Suh, "Can Markov chain models mimic biological regulation?" *Journal of Biological Systems*, vol. 10, no. 04, pp. 337–357, 2002.

[12] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of* B*oolean Networks: A Semi-tensor Product Approach*. Springer Verlag, 2011, vol. 81.

[13] F. Li, J. Sun, and Q.-D. Wu, "Observability of Boolean control networks with state time delays," *IEEE Transactions on Neural Networks*, vol. 22, no. 6, pp. 948–954, 2011.

[14] H. Li and Y. Wang, "Boolean derivative calculation with application to fault detection of combinational circuits via the semi-tensor product method," *Automatica*, vol. 48, no. 4, pp. 688–693, 2012.

[15] X. Xu and Y. Hong, "Matrix expression and reachability analysis of finite automata," *Journal of Control Theory and Applications*, vol. 10, no. 2, pp. 210–215, 2012.

[16] R. Li and T. Chu, "Complete synchronization of Boolean networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 840–846, 2012.

[17] R. Li, M. Yang, and T. Chu, "Synchronization design of Boolean networks via the semi-tensor product method," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 6, pp. 996–1001, 2013.

[18] R. Li, M. Yang, and T. Chu, "State feedback stabilization for Boolean control networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1853–1857, 2013.

[19] R. Li, M. Yang, and T. Chu, "State feedback stabilization for probabilistic Boolean networks," *Automatica*, vol. 50, no. 4, pp. 1272–1278, 2014.

[20] M. Yang and T. Chu, "Evaluation of attractors and basins of asynchronous random Boolean networks," *Physical Review E*, vol. 85, no. 5, p. 056105, 2012.

[21] M. Yang, R. Li, and T. Chu, "Controller design for disturbance decoupling of Boolean control networks," *Automatica*, vol. 49, no. 1, pp. 273–277, 2013.

[22] M. Yang, R. Li, and T. Chu, "Construction of a Boolean model of gene and protein regulatory network with memory," *Neural Networks*, vol. 52, pp. 18–24, 2014.

[23] M. Yang, R. Li, and T. Chu, "Optimal control of steady-state probability distributions of probabilistic Boolean networks," in *32nd Chinese Control Conference*. IEEE, 2013, pp. 2269–2274.

[24] A. T. Weeraratna, Y. Jiang, G. Hostetter, K. Rosenblatt, P. Duray, M. Bittner, and J. M. Trent, "Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma," *Cancer cell*, vol. 1, no. 3, pp. 279–288, 2002.

[25] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks: the imperfect information case," *Bioinformatics*, vol. 20, no. 6, pp. 924–930, 2004.

[26] R. Pal, I. Ivanov, A. Datta, M. L. Bittner, and E. R. Dougherty, "Generating Boolean networks with a prescribed attractor structure," *Bioinformatics*, vol. 21, no. 21, pp. 4021–4025, 2005.

[27] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite-horizon control for probabilistic Boolean networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2375–2387, 2006.

[28] S. Pal, S. Bandyopadhyay, and S. Ray, "Evolutionary computation in bioinformatics: A review," *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*, vol. 36, no. 5, pp. 601–615, 2006.

[29] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392–403, 1998.