IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 18, 2018, accepted October 30, 2018, date of publication November 15, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2881655

Model Checking Optimal Infinite-Horizon Control for Probabilistic Gene Regulatory Networks

LISONG WANG[1], TAO FENG[1], JUNHUA SONG[1], ZONGHAO GUO[1], AND JUN HU[1,2]
[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
[2]College of Engineering, Boise State University, Boise, ID 83725, USA

Corresponding author: Lisong Wang (wangls@nuaa.edu.cn)

This work was supported by the Aeronautical Science Foundation of China under Grant 20165515001.

ABSTRACT Genetic regulatory networks (GRNs) are significant fundamental biological networks through which biological system functions can be regulated. A significant challenge in the field of system biology is the construction of a control theory of GRNs through the application of external intervention controls; currently, context-sensitive probabilistic Boolean networks with perturbation (CS-PBNp) are used as an important network model in research on the optimal GRN control problem. This paper proposes an approximate optimal control strategy approach to the infinite-horizon optimal control problem based on probabilistic model checking and genetic algorithms (GAs). The proposed method first reduces the expected cost defined under the infinite-horizon control to a steady-state reward within a discrete-time Markov chain. A CS-PBNp model with a stationary control policy is then constructed to represent the cost of the fixed control strategy based on a temporal logic with a reward property, and calculations are carried out automatically by a PRISM model checker. The stationary control policy is then encoded as an element of the solution space of a GA. Based on the fitness of each control policy element as calculated by PRISM, an optimal solution can be obtained by using a GA to execute genetic operations iteratively. The experimental results generated by applying the proposed approach to the WNT5A network validate the accuracy and effectiveness of the approach.

INDEX TERMS Genetic regulatory networks, optimal control, probabilistic model checking, genetic algorithm.

I. INTRODUCTION

Research on gene regulatory networks [1] is primarily conducted through the analysis of gene expression data via the application of systems biology methods and techniques to build gene regulatory network (GRN) models that can mimic the regulation of biological systems and enable a better understanding of biological phenomena within their frameworks. A variety of mathematical and computational methods are used in the construction of GRN models that can analyze gene expression using data structures such as Boolean networks (BNs) [2], probabilistic Boolean networks (PBNs) [3], and dynamic Bayesian networks [4].

PBNs, which are extensions of BNs, can capture the rules-based dependencies between genes to elicit translational behavior in genetic processes and are therefore widely used to study the optimal control of GRNs. In a PBN, each node has multiple Boolean functions, each of which can be selected at a given time based on a fixed probability distribution. In this manner, PBNs, which are also known as instantaneous random PBNs, can essentially be viewed as collections of multiple BNs, one of which is randomly chosen at each time step as the evolution rule.

In this paper, we consider the optimal control of a context-sensitive probabilistic BN with perturbations (CS-PBNp) [5]. In a pure PBN, random interference can be introduced to capture the effects of the external environment on the genome. However, real biological systems have an inherent stability reflected in their context-sensitive attributes and, in contrast to the instantaneous random probabilities found in PBNs, the transition between BNs in a context-dependent probability BN is limited within certain probability bounds. Thus, CS-PBNps are more suitable for modeling small biological subnets.

In this paper, we propose a novel approach that combines the application of the PRISM probabilistic model checker with a genetic algorithm (GA) to solve the CS-PBNp infinite-horizon optimal control problem. Recently, Pal *et al.* [6] used discounted- and average-cost formulas to solve the infinite optimization control problem of context-dependent probabilistic BNs, while Abul *et al*. [7] reduced the infinite-horizon control problem to solving one Markov decision-making process. However, these solution methods require the calculation of large numbers of state transitions. Wei *et al*. [19] used PRISM and a GA to solve the CS-PBNp optimal control problem, but did so based on a solution of the finite-horizon optimization control problem.

The goal of the present study was to solve the problem of infinite-horizon optimization control. Probabilistic model checking [8] is a model-based automated verification technique in which model checking algorithms automatically validate properties with specifications defined by probabilistic temporal logic on a probabilistic model to enable broad qualitative or quantitative assessment of the results of system behavior. Under the proposed method, the expected total cost as defined in the infinite-horizon control is first reduced to a steady-state return on a discrete-time Markov chain. A model of context-sensitive probabilistic BNs with perturbations that contains a fixed control strategy is then constructed and a temporal logic formula with reward-based temporal properties is used to represent the PRISM-calculated cost of each fixed control strategy probabilistic model checker. Finally, a GA [9] is used to encode each fixed control strategy as an individual within the genetic algorithm solution space with a fitness value defined based on its control cost, and PRISM is used to obtain an approximate optimal solution by iteratively performing genetic operations on this solution space.

This remainder of this paper is organized as follows. In Section 2, the structure of the CS-PBNp, definition of the infinite-horizon optimization control problem, and genetic algorithm operation mechanism are introduced. Section 3 presents a detailed analysis and process for solving the infinite-horizon optimization control problem. Section 4 describes the application of the proposed method to the WNT5A network and presents the experimental results validating the proposed method. In Section 5, a discussion of related work and, finally, a summary of this paper are presented, the main contribution of this paper is to propose a problem solving method which combines genetic algorithm and probabilistic model checker PRISM, which can effectively solve the CS-PBNp infinite range optimal control problem.

II. BACKGROUND KNOWLEDGE

A. THE CONTEXT-SENSITIVE PROBABILISTIC BOOLEAN NETWORKS WITH PERTURBATION–CS-PBNp

A BN with n nodes can be defined as $B = (V, F)$, where $V = \{x_1, \ldots, x_n\}$ is a set of nodes in which each node $x_i \in \{0, 1\}(i \in [1 \ldots n])$ indicates the expression state of gene i, $F = \{f_1, \ldots, f_n\}$ is a list of Boolean functions used to represent the regulation interaction rules between genes, and each $f_i : \{0.1\}^n \rightarrow \{0, 1\}$ is a predictive function of gene i. The status of the network at time t can be represented as an n-bit binary vector, $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]$, where $x_i(t)$ is the value of $\mathrm{x_i}$ at time t.

The PBN provides a set of candidate Boolean functions for each gene i denoted by $F_i = \{f_1^{(i)}, \ldots, f_{l(i)}^{(i)}\}$. The probability value $c_j^{(i)}, (j \in [1 \ldots l(i)])$, such that $\sum_{j=1}^{l(i)} c_j^{(i)} = 1$, represents the probability of choosing $\mathrm{f_j^{(i)}}$ to update the expression state of gene i. If we assume that the candidate Boolean functions for each gene are chosen to be independent of each other, the probabilistic Boolean network can essentially be regarded as a collection comprising $N = \prod_{i=1}^{n} l(i)$ Boolean networks.

Context-sensitive PBNs (CS-PBNs), which limit the mutual switching between BNs, are generally considered to be a more appropriate model for the inherent stability of biological systems. A CS-PBN has a binary switch, s, with a small conversion probability q of forcing transition between BNs. If s is in state 1, there is a probability of q that the CS-PBN will choose a new BN as an evolution rule following a fixed probability distribution q. Otherwise, the CS-PBN keeps the current Boolean network unchanged until a transition occurs.

In this process, random interference can be added to the structure of the PBN to capture the effects of external inputs on gene expression status. The occurrence of an interference event on gene $\boldsymbol{I}$ with probability p is represented by the binary random variable $\boldsymbol{per^i}(\boldsymbol{i} \in [\boldsymbol{1} \ldots \boldsymbol{n}])$. When $\boldsymbol{per^i} = \boldsymbol{1}$, for instance, there is a probability p that the expression state of gene $\boldsymbol{i}$ will flip from 1 to 0, and vice versa. In this case, the Markov chain corresponding to the PBN is ergodic, that is, it is reachable between any two states and has a steady state distribution.

B. INFINITE-HORIZON OPTIMIZATION CONTROL PROBLEM

The goal of an infinite-horizon optimization control [6] is to find an optimized fixed control strategy for a control duration approaching infinity to represent a network that is as long as necessary while minimizing the control cost.

A CS-PBNp will have n nodes and n control inputs and a control duration of M. For $M \rightarrow \infty$, $\mathrm{t} = 0, 1, \ldots$ represent different control phases. At any t, $X(t) = [x_1(t), x_2(t), \ldots, x_n(t)]$ represents the current state of the CS-PBNp and $u(t) = [u_1(t), u_2(t), \ldots, u_m(t)]$ represents the current state of all the control inputs, where $u(t) \in \{0, 1\}^m$. The fixed control strategy $\pi : \{0, 1\}^{\mathrm{n}} \rightarrow \{0, 1\}^{\mathrm{m}}$ is a mapping function that maps the network state space to the control space, meaning that the choice of all control inputs depends only on the current state of the network.

Under a horizon of infinite control there is a control cost function $C(X(t), u(t)) : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \mathbb{R}_{\geq 0}$ that is used to represent the cost of applying a control input u (t) when the network is in state $X(t)$. The cost value must be set to capture the cost of intervention control and the relative preference for different states [10] so that, when (t) = 0, the various return values of $C(X(t), u(t))$ can reflect the expectation of different states. A smaller return value indicates that a state is more likely to arrive. When $u(t) = 1$, the return value of $C(X(t), u(t))$ must add an extra control cost based on the degree of state expectation.

Based on how the control cost function is set, for a CS-PBNp model starting from X (0) the total expected cost of the network obtained over an infinite-horizon using the fixed control strategy X (0) is as follows:

$$J_{\pi}\left(X\left(0\right)\right) = \lim_{M \to \infty} \frac{1}{M} E\left[\sum\nolimits_{t=0}^{M-1} C\left(X\left(t\right), u\left(t\right)\right)\right] \quad (1)$$

Thus, finding a solution to the infinite-horizon optimization control problem involves searching an optimized fixed control strategy π^* to minimize the expected total cost, and the optimal total expected cost is therefore defined as follows:

$$J^*_{\pi^*}\left(X\left(0\right)\right) = min \lim_{M \to \infty} \frac{1}{M} E\left[\sum\nolimits_{t=0}^{M-1} C\left(X\left(t\right), u\left(t\right)\right)\right] \quad (2)$$

III. ANALYSIS AND SOLUTION OF INFINITE-HORIZON OPTIMIZATION CONTROL

As a CS-PBNp with control inputs is essentially a Markov decision process, the optimal expected cost solution of a control problem over an infinite-horizon can essentially be equated to a calculation of the minimum steady-state reward value in a Markov decision process (MDP). However, the model checking algorithm in PRISM cannot measure the reward attributes of the stationary state on an MDP model. To address this, the proposed method first adds a specific fixed control strategy with control inputs to the CS-PBNp and then converts the CS-PBNp into a discrete-time Markov chain (DTMC) model. By model checking the steady state reward attributes on this DTMC, the expected total cost of generation under a specific fixed control strategy is obtained. Finally, iterative genetic manipulation is applied to the genetic algorithm to continually add new sets of fixed control strategies to the CS-PBNp through control inputs. These strategies are evaluated by PRISM to obtain the optimal fixed control strategy.

A. THE REDUCTION FROM INFINITE-HORIZON CONTROL TO THE STEADY-STATE REWARD

The expected total cost solution under infinite-horizon control is reduced to the calculation of the steady-state reward [11], the attributes of which can be expressed by the probabilistic computation tree logic (PCTL) state formula $R_{\sim r}[S]$, which is used to solve the system's long-run average expected reward.

Consider a discrete time Markov chain $D = (S, P, L)$, where S is a finite set of states, P is the transition probability matrix, and L is a label function. A reward structure [11] on this chain can be expressed as a two-tuple $r = (\rho, \tau)$, where $\rho : S \times S \to \mathbb{R}_{\geq 0}$ represents a status reward function that sets a reward value for the status of each time step and $\tau : S \times S \to \mathbb{R}_{\geq 0}$ represents a transition reward function that sets a reward value for the state transition on each time step. This reward structure can be used to extend the DTMC model. Using $Exp^D(s, X)$ to represent the expected value starting from state s on the DTMC model based on the reward attributes X, the respective calculation procedures for cumulative and steady-state reward attributes are given as follows:

$$Exp^D\left(s, X_{C\leq k}\right) \begin{cases} 0 & if\ k = 0 \\ \rho(s + (\sum_{s' \in S} P(s, s') \cdot (\tau(s, s') \\ \quad + Exp^D\left(s', X_{C \leq k-1}\right))) & if\ k \neq 0 \end{cases} \quad (3)$$

$$Exp^D\left(s, X_s\right) = \lim_{k \to \infty} \frac{1}{k} Exp^D\left(s, X_{C\leq k}\right) \quad (4)$$

According to the above definition, the model can check the steady-state reward attributes obtained by setting the reward value of the transition reward function $\tau(s, s')$ to zero and using the state reward function $\rho(s)$ to represent the control cost function $C\left(X\left(t\right), u\left(t\right)\right)$ to solve the expected total cost defined under Equation 1 for infinite-horizon control, i.e., the total expected cost generated by adopting a specific fixed control strategy. Note that, in practice the control costs are not related to time.

The S operator from PRISM [12] can also be used to infer the steady-state (long-term operation) behavior of a model. In this case, the steady state probability attribute is expressed by the formula *S bound* [*prop*], which if true indicates that the long-term probability of being in a state satisfying the property *prop* satisfies the requirement of the boundary value *bound*. Using the steady state probability property formula, it is possible to determine the impact of adopting a specific fixed control strategy on the long-term behavior of the CS-PBNp.

B. MODELING OF FIXED CONTROL STRATEGY

Here, we provide a simple example based on the preceding analysis. For a CS-PBNp model with two Boolean nodes (x_1, x_2), control input u, switch probability q = 0.3, and probability of interference p = 0.1, the dynamic equation of the network is defined as follows:

$$x_1\left(t+1\right) = \begin{cases} x_1\left(t\right) \vee u\left(t\right) & c_1^{(1)} = 0.3 \\ x_1\left(t\right) \vee x_2\left(t\right) & c_2^{(1)} = 0.7 \end{cases} \quad (5)$$

$$x_2\left(t+1\right) = \begin{cases} x_1\left(t\right) \vee \neg u\left(t\right) & c_1^{(2)} = 0.2 \\ x_2\left(t\right) & c_2^{(2)} = 0.8 \end{cases} \quad (6)$$

The PRISM modeling code for building a CS-PBNp that contains the specific fixed control strategy π^1 given by $\pi^1(00) = 0, \pi^1(01) = 1, \pi^1(10) = 0, \pi^1(11) = 1$ is shown in Figure 1.

In the above code, line 1 ensures that the system's feature model is a discrete-time Markov chain (DTMC). In lines 2–5, the Boolean functions of the network are translated according to the rules of equivalence, while lines 6–9 give the respective control decisions specified by π^1. Module "SWITCH" (lines 10–13) ensures that the context switch s changes to states 1 and 0 with probabilities 0.3 and 0.7, respectively. In module "PER1" (lines 14–17), the random disturbance event per1 changes to states 1 and 0 with probabilities 0.1 and 0.9, respectively. In module "NODE1" (lines 18–27) the migration rules for the state of the Boolean node x_1

```
(01) dtmc
(02) formula  f11=x1+u-x1*u;
(03) formula  f12=x1*x2;
(04) formula  f21=x1*(1-u);
(05) formula  f22=x2;
(06) formula  p1=(x1=0&x2=0&u=0)?1:0;
(07) formula  p2=(x1=0&x2=1&u=1)?1:0;
(08) formula  p3=(x1=1&x2=0&u=0)?1:0;
(09) formula  p4=(x1=1&x2=1&u=1)?1:0;
(10) module  SWITCH
(11) s:[0..1];
(12) [PBN] true -> 0.3:(s′=1) + 0.7:(s′=0);
(13) endmodule
(14) module  PER1
(15) per1:[0..1];
(16) [PBN] true -> 0.1:(per1′=1)+ 0.9:(per1′=0);
(17) endmodule
(18) module NODE1
(19) x1:[0..1];
(20) d1:[0..1];
(21) [PBN]s=1&per1=0&((p1=1)|(p2=1)|(p3=1)|(p4=1))->0.3:(x1'=f11)&(d1'=0)+
0.7:(x1'=f12)&(d1'=1);
(22) [PBN]s=0&per1=0&d1=0&((p1=1)|(p2=1)|(p3=1)|(p4=1))->(x1'=f11);
(23) [PBN]s=0&per1=0&d1=1&((p1=1)|(p2=1)|(p3=1)|(p4=1))->(x1'=f12);
(24) [PBN]per1=0&(p1!=1)&(p2!=1)&(p3!=1)&(p4!=1)->(x1'=x1);
(25) [PBN]per1=1&x1=1->(x1'=0);
(26) [PBN]per1=1&x1=0->(x1'=1)
(27) endmodule
(28) module input
(29) u:[0..1] ;
(30) v:[0..1] ;
(31) [PBN](p1!=1)&(p2!=1)&(p3!=1)&(p4!=1)->(u'=1-u);
(32) [PBN](p1=1)| (p2=1)| (p3=1)| (p4=1)->(u'=u);
(33) endmodule
(34)module  preu
(35) pu:[-1..1];
(36) [csPBN]true->(pu'=u);
(37)endmodule
```

FIGURE 1. **The code of PRISM modeling of CS-PBNp containing stationary control policy.**

are implemented. In the command statements of NODE1 (lines 21–23), the specific fixed control strategy π^1 is implemented as the condition of state transition, while line 24 ensures that the state of the Boolean node x_1 remains unchanged when the state transition conditions do not satisfy the requirement of π^1 (the code for Boolean node x_2 is similar to that for x_1). Lines 28–32 implement the transformation rules that control the value of the control input u in the module "input", while lines 33–36 implement the transformation rules for the value of the variable *pu*, which is used to record the value of *u* at the previous instant, in the module "preu."

The state reward function in the reward structure of PRISM is used to describe how the control cost function $C(X(t), u(t))$ is defined in the infinite-horizon optimization control. The PRISM modeling code shown in Figure 1 sets the network control cost to 1. For states [0,0], [0,1], [1,0], and [1,1], the cost values are set to 0, 2, 4, and 6, respectively, when pu $= 0$ to delineate [0,0] and [1,1] as the most and least desirable states, respectively. These settings are used to define the reward structure shown in Figure 2.

In the above PRISM code, the reward values set in lines 2, 4, 6, and 8 indicate the expectation degrees of

```
( 01 ) rewards "cost"
( 02 )  x1=0&x2=0&pu=0&v=1:0;
( 03 )  x1=0&x2=0&pu=1&v=1:1;
( 04 )  x1=0&x2=1&pu=0&v=1:2;
( 05 )  x1=0&x2=1&pu=1&v=1:3;
( 06 )  x1=1&x2=0&pu=0&v=1:4;
( 07 )  x1=1&x2=0&pu=1&v=1:5;
( 08 )  x1=1&x2=1&pu=0&v=1:6;
( 09 )  x1=1&x2=1&pu=1&v=1:7;
( 10 ) endrewards
```

FIGURE 2. **Code for PRISM modeling of reward structure.**

different states, while those in lines 3, 5, 7, and 9 consider the costs of control based in terms of the respective expectation degrees. Here, the PCTL formula $R = \{``cost''\} =?[S]$ is used to solve for the steady-state reward attribute on the DMTC model reflecting the expected total cost of adopting the fixed control strategy π^1 under infinite-horizon control. This attribute is automatically calculated by the probabilistic model checking algorithm to have a value of 2.14. In addition, the steady state probability attribute formula $S =?[x1 = 0\&x2 = 0]$ is used to obtain the probability (calculated here to be 0.43) that the network remains in state [0,0] for an interval of suitable length under strategy π^1.

C. OPTIMAL FIXED CONTROL STRATEGY OF GENETIC ALGORITHM

In the subsections above, the expected total cost produced by adopting a specific fixed control strategy was obtained through model checking of the steady state reward attributes in PRISM. However, in the infinite-horizon optimization control problem, an optimized fixed control strategy is used to minimize the expected total cost. To obtain the optimal fixed control strategy and optimal total expected cost, we use a process combining a GA with PRISM methodology. In the following, we provide a detailed description of the algorithm used in the implementation process.

1) INITIALIZE THE POPULATION

For a CS-PBNp with n nodes and m control inputs, the fixed control strategy π determines the value of each control input u_i for each state $i \in \{1, 2, \ldots, m\}$. Accordingly, the value of u_i can be encoded into a binary sequence of length 2^n given by $p_i = (i_0, i_1, i_2, \ldots, i_{2^n-1})$, $i_j \in \{0, 1\}$, $0 \le j \le 2^n - 1$. The value of each position in the binary sequence corresponds to the value of the respective control input u_i from among the 2^n states of the network. The network status is sorted from $[0, 0, \ldots 0]$ to $[1, 1, \ldots, 1]$. A fixed control strategy that contains m control inputs is encoded as a two-dimensional matrix $P = [p_1, p_2, p_3, \ldots, p_m]^T$ in which the value of each column indicates the value of all control inputs on its corresponding network state. In the process of initialization, a population $POP = [P_1, P_2, P_3, \ldots, P_N]$ comprising N fixed control strategies P (popsize = N) is randomly generated. Each individual in the population represents a feasible solution to an infinite-horizon optimal control problem.

2) CALCULATE INDIVIDUAL FITNESS VALUES

The fitness function of a GA, which is also called the evaluation function, is used to assess the excellence degree of each individual within a population. The fitness value of a given individual usually takes on a value of the objective function defined under the optimization problem. In the infinite-horizon optimization control problem, the objective function of optimization is the optimal expected total cost as defined in Equation 2. To calculate the fitness value of each fixed control strategy P_i within the population POP, each $\mathrm{P_i}$ is first replaced with a PRISM model of the CS-PBNp with a fixed control strategy. That is, the formula $\mathrm{P_i}$ is expressed as a fixed control strategy under the PRISM model and the PCTL formula $R = \{``cost''\} =?[S]$ is then model checked to obtain the value of steady state reward as the fitness value of P_i. This fitness value also represents the expected total cost generated by taking the fixed control strategy P_i.

3) SELECTING OPERATION

The selecting operation in a GA selects, with a certain probability, several individuals from a current population based on their respective fitness values. From the *popsize* individuals within a population POP, the individual that can minimize the objective function is found through the use of the roulette selection method, which retains some individuals with low fitness values for as long as possible. The roulette selection method involves first summing the fitness values of all individuals within a population to obtain a total fitness value and then dividing each individual's fitness value by the total fitness value to produce their fitness rate, which is regarded as that individual's probability of being chosen. This is specifically calculated as follows:

$$expected\ (P) = total\_fitness\ (POP)\ /fitness(P) \tag{7}$$

$$selectp(P) = excepted(P)/total\_excepted(POP) \tag{8}$$

where expected is the expected value, fitness is the fitness value, selectp is the selection probability, total_fitness is the total fitness value, and total_excepted is the total expected value. Following the selecting operation, the retained fixed control strategy with the smallest fitness value is used to generate a new population POP′ and is transferred to the next genetic operation, which is applied to a population of the original size.

4) CROSSOVER OPERATION

The crossover operation in a GA is the key to maintaining population diversity. In this operation, two parent individuals are randomly selected with specific probabilities from the overall population and then, in accordance with specified exchange rules, subjected to cross-gene at a certain position to generate two new offspring individuals. In the present application, two fixed control strategies from the population POP′ are selected with a predetermined crossover probability pc to execute popsize × pc uniform crossover operations. Following each execution of the crossover operation, two new fixed control strategies are generated and used to replace the original fixed control strategy. At the end of this process, a new POP″ is generated and passed on to the next genetic operation.

5) MUTATION OPERATION

The mutation operation in a GA is used to induce a gene mutation in each parent within a population by changing the gene value, with a probability based on a specified mutation rule, at a certain position in all parent individuals. New offspring individuals are then generated. In the proposed method, the Exchange Mutation operation is applied with a predetermined mutation probability pm to each fixed control strategy within the population POP″. newly generated fixed control strategy. In this process, each original fixed control strategy is replaced by its mutated version, and a new POP‴ is generated and passed on to the next genetic manipulation.

The selection, crossover, and mutation operations described above are repeated until the following two conditions hold: (1) the maximum iteration number, Max gens, exceeds the set threshold; (2) the optimal solution obtained from the current population remains the same, that is, the optimal solution converges.

IV. CASE STUDY

In this section, we describe the results of applying our proposed infinite-horizon optimization control method to a real gene regulatory network, the WNT5A network. Through analysis of the experimental results, the proposed method is validated.

A. WNT5A NETWORK

The WNT5A network [13] is a gene regulatory network associated with melanoma. Biological experiments have revealed that increasing the concentration of the WNT5A gene can directly enhance the metastasis ability of cancer cells in cell lines and induce cancer cell metastasis. These experimental results suggest that the metastasis of cancer cells can be substantially reduced by down-regulating the concentration of the WNT5A gene. To solve the infinite-horizon optimization control problem in this context, we constructed the following model of context-sensitive probabilistic BNs with perturbation for the WNT5A network:

$$\begin{aligned}
x_1\,(t+1) &= \neg x_6\,(t)\\
x_2\,(t+1) &= (\neg x_2\,(t) \wedge x_4\,(t) \wedge x_6\,(t)) \vee (x_2\,(t)\\
&\qquad \wedge x_4\,(t) \vee x_6\,(t)\,))\\
x_3\,(t+1) &= \neg x_7\,(t)\\
x_4\,(t+1) &= x_4\,(t)\\
x_5\,(t+1) &= \neg x_7\,(t) \vee x_2\,(t)\\
x_6\,(t+1) &= x_3\,(t) \vee x_4\,(t)\\
x_7\,(t+1) &= \neg x_2\,(t) \vee x_7\,(t)
\end{aligned} \tag{9}$$

Where x_1 represents the concentration of WNT5A and x_2, x_3, x_4, x_5, x_6, and x_7 indicate the concentrations of the pirin, S100P, RET1, MART1, HADHB, and STC2 genes, respectively. We chose x_2 as the control gene in the WANT5A network and obtained a CS-PBNp model by using a combination of synchronous and asynchronous updates [14] to expand its PBN dynamically to four BNs with respective probabilities of construction of [0.35,0.15,0.35,0.15]. We also set the context switching probability to $q = 0.1$ and the probability of random interference events to $p = 0.2$.

B. EXPERIMENTAL RESULTS AND ANALYSIS

For this CS-PBNp model of the WNT5A network, we set the states satisfying $x_1 = 0$ and $x_1 = 1$ as the most expected and unexpected network states, respectively. The control cost was set to 1. The PRISM code of the resulting reward structure is given in Figure 3.

```
(01) rewards "cost"
(02) x1=0&pu=0&v=1:0;
(03) x1=0&pu=1&v=1:1;
(04) x1=1&pu=0&v=1:5;
(05) x1=1&pu=1&v=1:6;
(06) endrewards
```

FIGURE 3. PRISM code of reward structure of CS-PBNp model of WNT5A network.

Following the modeling rules summarized in Section 3.2, the steady-state reward formula $R = \{``cost''\} =?[S]$ was used to solve the total expected cost of the fixed control strategy as the fitness value for each individual within the GA solution space. Pal *et al.* [6] demonstrated that for a Markov chain with ergodicity the calculation of expected total cost will be independent of the initial state of the network. In a CS-PBNp, the impact of random interference events can impose ergodicity to the network state in this manner, resulting in a network with a steady state distribution following long-term evolution. Therefore, it was not necessary to consider the respective initial states of the network in this experiment. We set the population size to 20, the maximum number of iterations to 600, the crossover probability to 0.8, and the mutation probability to 0.1. Because the GA does not always find the optimal solution, we repeated it 30 times to obtain our experimental results.

Figure 4 shows the optimized fixed control strategy obtained by GA, where the state number on the x-axis is used to indicate the decimal value corresponding to the actual binary state value. The total number of states is $2^7 = 128$, with state values ranging from $[0, 0, 0, \ldots, 0]$ to $[1, 1, 1, \ldots, 1]$. The red and blue circles indicate where the control inputs are $u = 1$ and $u = 0$, respectively. Based on the obtained optimized fixed control strategy, the total infinite optimal expected cost is 1.69. Through model checking of the steady-state probability attribute formula $S =?[x_1 = 0]$, we found that when the network state satisfies $x_1 = 0$ the steady-state probability is 0.67.

Figure 5 shows the steady-state probabilities that the network is in various states under the effect of the optimized fixed control strategy shown in Figure 4. With an infinite horizon of optimal control set, the status numbers from 0 to 63 belong to the expected network status while those from 64 to 127 belong to the unexpected network status. Figure 6 shows the steady-state probabilities that the network is in various states without any control. When no control strategy is adopted, the expected cost of the infinite-horizon is 1.87; when the network state satisfies $x_1 = 0$, the steady state probability is 0.59, which is smaller than the steady state probability obtained by adopting the fixed control strategy.

By comparing the experimental results in Figures 5 and 6, it is seen that adopting the optimized fixed control strategy can reduce the probability that the WNT5A network remains in an unexpected state for a long time and increases the probability that it can be moved to an expected state, i.e., a state

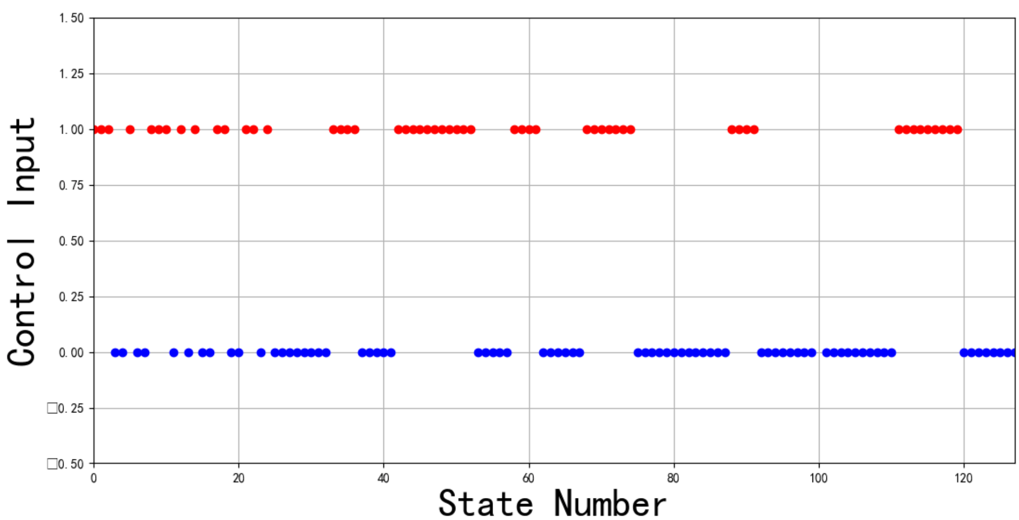


FIGURE 4. The Optimized stationary control policy.

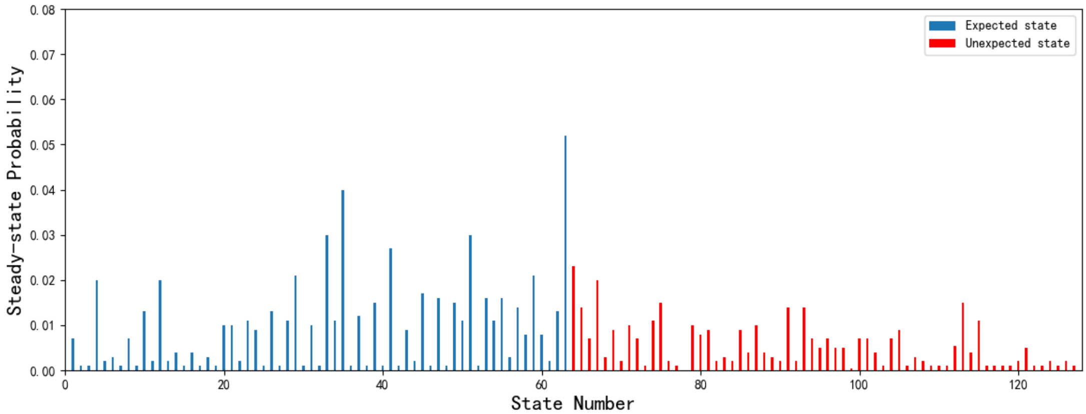


FIGURE 5. Steady-state probability distribution applying a optimized stationary control policy.

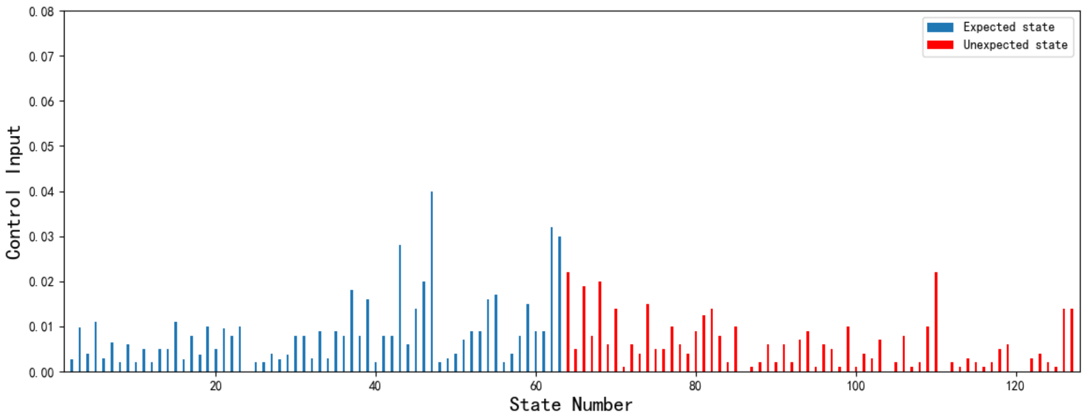


FIGURE 6. Steady-state probability distribution without applying a control policy.

satisfying $x_1 = 0$. This corresponds to reducing the ability of cancer cells to metastasize if the goal of an infinite-horizon optimal control is met. If, instead, the method of exhaustive traversal is used to find the most accurate solution to the problem, the search scale of the solution explodes to 2^{128}, which in practice is not feasible. Thus, the proposed method represents an effective approach to solving large-scale search problems.

V. RELATED WORK

In recent years, much in-depth research has been conducted to explore the probabilistic BN optimization control problem, resulting in many feasible solutions, including the matrix half-tensor and probability dynamic programming methods. However, these methods often require users to compute a large number of state transitions, making it difficult to build and analyze network models. Model checking is an automatic method of formal verification through the use of formal modeling languages that can accurately and flexibly describe network models and apply temporal logic to express their behavior, At present, the probability model checker PRISM [15] is used to verify the reachability and security of probabilistic Boolean control networks. However, the effects of stochastic disturbances in probabilistic BNs have not been considered [16], and there is to date no method for applying probability model checking to automatically solve the optimal control problem

Datta *et al.* [10] defined the objective function of the finite-horizon optimization control problem of probabilistic BNs and proposed using a probabilistic dynamic programming algorithm to accurately solve the optimal control strategy. Pal *et al.* [6] explored context-sensitive probabilistic BNs in the context of the infinite-horizon optimization control problem and used discounted- and average-cost formulas to, respectively, solve the minimum expected total cost and obtain the stationary control policy. Yang *et al.* [17] and Ching *et al.* [18] proposed the application of a GA to approximate the finite-horizon optimization control problem of a probabilistic BN and identified the hard constraint condition in the control strategy, namely, that the number of control iterations is limited. Wei *et al.* [19] used a hybrid model combining PRISM with a GA to solve the CS-PBNp optimal control problem, although their main objective was to solve the problem of finite-horizon optimal control. Mizera *et al.* [20], [21] proposed a tool—ASSA-PBN—for the modeling, simulation, and analysis of probabilistic BNs. ASSA-PNB applies an efficient statistical method to calculating the steady-state probabilities of a large PBN. Meanwhile, Mizera *et al.* [21]

proposed using a GA as a parameter estimation algorithm to calculate probabilities in the context of a BN control problem.

VI. CONCLUSIONS

In this paper, we proposed an approach to solving the CS-PBNp infinite-horizon optimal control problem by combining a GA with the PRISM probability model checker. The proposed method first reduces the solution of the total expected cost defined under the infinite-horizon control to the calculation of the steady-state reward on a discrete-time Markov chain. The specific fixed control strategy is then added to the PRISM model of the CS-PBNp with control inputs. A reward structure is used to represent the control cost functions defined in infinite optimization control and the infinite-horizon optimization control is found by encoding each fixed control strategy as a two-dimensional matrix of structural characteristics represented by an individual in the GA solution space. The respective fitness values are then calculated in PRISM, and the GA iteratively performs various genetic operations on the solution space to obtain the optimal total expected cost and the optimized fixed control strategy. The results of a number of experimental evaluations applied to a real biological network model validated the proposed approach.

ACKNOWLEDGMENT

The authors would like to express their special thanks to Dr. Ou Wei for his very helpful discussions in the creative process of this paper.

REFERENCES

[1] E. H. Davidson, *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution*. Amsterdam, The Netherlands: Elsevier, 2006, pp. 508–509.

[2] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proc. IEEE*, vol. 90, no. 11, pp. 1778–1792, Nov. 2002.

[3] 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, no. 2, pp. 261–274, 2001.

[4] P. Li, C. Zhang, E. J. Perkins, P. Gong, and Y. Deng, "Comparison of probabilistic Boolean network and dynamic Bayesian network approaches for inferring gene regulatory networks," *BMC Bioinform.*, vol. 8, no. 7, p. S13, 2007.

[5] B. Faryabi, "Intervention in context-sensitive probabilistic Boolean networks revisited," *EURASIP J. Bioinform. Syst. Biol.*, vol. 2009, no. 1, 2009, Art. no. 360864.

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

[7] O. Abul, R. Alhaj, and F. Polat, "Markov decision processes based optimal control policies for probabilistic Boolean networks," in *Proc. 4th IEEE Symp. Bioinform. Bioeng. (BIBE)*, May 2004, pp. 337–344.

[8] V. Forejt, "Automated verification techniques for probabilistic systems," in *Formal Methods for Eternal Networked Software Systems*. Springer, 2011.

[9] M. Kumar, "Genetic algorithm: Review and application," *Int. J. Inf. Technol. Knowl. Manage.*, vol. 2, no. 2, pp. 451–454, 2010.

[10] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," in *Proc. Amer. Control Conf.*, 2003, pp. 3614–3619.

[11] M. Kwiatkowska, G. Norman, and D. Parker, "Stochastic model checking," in *Proc. Int. Conf. Formal Methods Perform. Eval.*, 2007, pp. 220–270.

[12] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: Probabilistic symbolic model checker," in *Computer Performance Evaluation: Modelling Techniques and Tools*, vol. 2324. 2002, pp. 200–204.

[13] M. Bittner *et al.*, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, 2000.

[14] K. Kobayashi and K. Hiraishi, "Verification and optimal control of context-sensitive probabilistic Boolean networks using model checking and polynomial optimization," *Sci. World J.*, vol. 2014, Jan. 2014, Art. no. 968341.

[15] D. A. Parker, "Implementation of symbolic model checking for probabilistic systems," Ph.D. dissertation, School Comput. Sci., Univ. Birmingham, Birmingham, U.K., 2002.

[16] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, Oct. 2002.

[17] C. Yang, C. Wai-Ki, T. Nam-Kiu, and L. Ho-Yin, "On finite-horizon control of genetic regulatory networks with multiple hard-constraints," *BMC Syst. Biol.*, vol. 4, no. 2, p. S14, 2010.

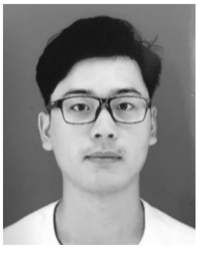
[18] W. K. Ching, H. Y. Leung, S. Zhang, and N. K. Tsing, "A genetic algorithm for optimal control of probabilistic Boolean networks," in *Proc. 2nd Int. Symp. Optim. Syst. Biol. (OSB)*, Lijiang, China, Oct./Nov. 2008, pp. 29–35.

[19] O. Wei, Z. Guo, Y. Niu, and W. Liao, "Model checking optimal finite-horizon control for probabilistic gene regulatory networks," *BMC Syst. Biol.*, vol. 11, no. 6, pp. 75–88, 2017.

[20] A. Mizera, J. Pang, and Q. Yuan, "ASSA-PBN: An approximate steady-state analyser for probabilistic Boolean networks," in *Proc. Int. Symp. Automat. Technol. Verification Anal.* Springer, 2015, pp. 214–220.

[21] A. Mizera, J. Pang, and Q. Yuan, "ASSA-PBN 2.0: A software tool for probabilistic Boolean networks," in *Computational Methods in Systems Biology*. Springer, 2016.

LISONG WANG received the M.S. degree in computer science and the Ph.D. degree in computer applications technology from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1995 and 2010, respectively. He is currently an Associate Professor with the College of Computer Science and Technology, NUAA. His research interests include avionics safety analysis, data management in distributed environments, formal methods and model-based safety analysis, and wireless sensor networks.



TAO FENG is currently pursuing the master's degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His major is formal methods and model-based safety analysis.

JUNHUA SONG is currently pursuing the master's degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. Her major is formal methods and model-based safety analysis.

ZONGHAO GUO is currently pursuing the master's degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His major is formal methods and model-based safety analysis.

JUN HU received the Ph.D. degree from Nanjing University in 2006. He is currently an Associate Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. He is also a Visiting Scholar with Boise State University. His research interests include model-based system engineering, avionics safety analysis, formal methods, and software analysis and verification.