社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2008－BIO－14　(8)
2008／9／19

入次数に制約のあるブーリアンネットワークに対する先行状態検出問題および制御問題について

阿久津 達也[1]　林田 守広[1]　張 淑芹[2]　程 王韋王其[3]　呉 國寶[4]

[1] 京都大学　化学研究所　バイオインフォマティクスセンター
[2] 復旦大学　数学科　[3] 香港大学　数学科　[4] 香港浸會大学　数学科

ブーリアンネットワーク（BN）は遺伝子ネットワークの数理モデルの一つである。先行状態検出問題とは、BN の状態が与えられた時、その状態の直前の状態を見つける問題である。また、制御問題は、初期状態、目標状態が与えられた時に、いくつかの頂点の状態を制御することにより目標状態に至らせるような制御系列を見つける問題である。いずれの問題も NP 困難であることが知られているが、本稿では入次数に制約がある場合について、より詳細な結果を示す。また、同様の場合の先行状態検出問題に対する平均的に比較的高速に動作するアルゴリズムを示すとともに、先行状態の平均的な分布について解析を行う。その結果として、同様の場合の制御問題に対する従来より平均的に高速なアルゴリズムを示す。

Analyses and Algorithms for Predecessor and Control Problems for Boolean Networks of Bounded Indegree

Tatsuya Akutsu[1], Morihiro Hayashida[1], Shu-Qin Zhang[2], Wai-Ki Ching[3] and Michael K. Ng[4]

[1] Bioinformatics Center, Institute for Chemical Research, Kyoto University
[2] Department of Mathematics, Fudan University
[3] Department of Mathematics, The University of Hong Kong
[4] Department of Mathematics, Hong Kong Baptist University

We study the predecessor and control problems for Boolean networks (BNs). The predecessor problem is to determine whether there exists a global state that transits to a given global state in a given BN, and the control problem is to find a sequence of 0-1 vectors for control nodes in a given BN which leads the BN to a desired global state. The predecessor problem is useful both for the control problem for BNs and for analysis of landscape of basins of attractions in BNs. In this article, we focus on BNs of bounded indegree and show some harness results on the computational complexity of the predecessor and control problems. We also present simple algorithms for the predecessor problem that are much faster than the naive exhaustive search-based algorithm. Furthermore, we show some results on distribution of predecessors, which leads to an improved algorithm for the control problem for BNs of bounded indegree.

1 Introduction

The *Boolean network* (BN, in short) is one of the well-studied mathematical models of genetic networks [6]. Finding a sequence of control actions for BNs is an important problem on BNs, which is abbreviated as BN-CONTROL in this article. Inspired from works on control of the probabilistic Boolean network (PBN, in short) model [5], we studied BN-CONTROL [1]. We showed that BN-CONTROL is NP-hard even in considerably restricted cases [1]. However, we may be able to develop algorithms that are much faster than exhaustive search based algorithms. Though we have not yet fully succeeded to develop such algorithms, we encountered the problem of finding a global state transiting to a given global state, which is known as the *predecessor problem* for BNs [3, 4] and is abbreviated as BN-PREDECESSOR in this article. In other words, BN-PREDECESSOR is to find an input node to a specified node in a state transition diagram of a BN. It should be noted that the problem is trivial once a state transition diagram is constructed. However, the number of nodes of a state transition diagram is 2^n where n is the number of nodes in a BN. Therefore, faster algorithms should be developed.

For the predecessor problem, some studies have been done. Barrett et al. studied the computational complexity of the predecessor problem for

BNs and other discrete dynamical systems [3]. They showed that BN-PREDECESSOR is NP-hard even for BNs with planar graph structures, whereas they presented a polynomial time algorithm for BNs of bounded tree-width. Coppersmith also studied the computational complexity of BN-PREDECESSOR and distribution of predecessors [4]. She showed that BN-PREDECESSOR for BNs with maximum indegree K can be reduced to K-SAT, where K-SAT denotes the Boolean satisfiability problem for a set of clauses each of which consists of at most K-literals. She also showed that as n grows, the ratio of global states having at least one predecessors converges to $1/e$ and 0 for general BNs and for BNs with maximum indegree K, respectively.

In this article, we study BN-PREDECESSOR, its variant, and BN-CONTROL with focusing on cases where the maximum indegee is bounded by some constant K. We show some hardness results on BN-PREDECESSOR, BN-CONTROL and a related problem, all of which strengthen existing hardness results. Next, based on our previous algorithms for identifying singleton attractors [7], we develop algorithms for identifying all predecessors, all of which are much faster than the naive enumeration based algorithm. Then, based on studies by Coppersmith [4], we show some results on distributions of predecessors. Furthermore, by making use of some of these results, we develop an improved algorithm for BN-CONTROL for bounded indegree. Due to space limitation, we do not present details here. Details are given in the journal version [2]

2 Preliminaries

2.1 Boolean Network and BN-PREDECESSOR

A BN consists of a set of n nodes V and n Boolean functions F, where $V = \{v_1, \ldots, v_n\}$ and $F = \{f_1, \ldots, f_n\}$. In general, V and F correspond to a set of genes and a set of gene regulatory rules, respectively. Each node takes either 0 or 1 at each discrete time t, where 1 (resp. 0) means that the corresponding gene expresses (resp. does not express) at time t. The state of v_i at time t is denoted by $v_i(t)$. The *global state* of a BN (or simply the *state* of a BN) at time step t is denoted by the vector $\mathbf{v}(t) = [v_1(t), \ldots, v_n(t)]$. A regulation rule for each node is given in the form of a Boolean function and the states of nodes change synchronously. A node v_i has k_i incoming nodes $v_{i_1}, \ldots, v_{i_{k_i}}$ and the state of v_i at time $t+1$ is determined by $v_i(t+1) = f_i(v_{i_1}(t), \ldots, v_{i_{k_i}}(t))$, where f_i is a Boolean function with k_i input variables. The number k_i is called the *indegree* of node v_i. We also write $v_i(t+1) = f_i(\mathbf{v}(t))$ to denote the regulation rule for v_i and $\mathbf{v}(t+1) = \mathbf{f}(\mathbf{v}(t))$ to denote the regulation rule for the whole BN. A specific global state can be written as an n-dimensional binary vector $[b_1, \ldots, b_n]$. If we consider all 2^n possible states and compute their respective next states, a list of 2^n one-step state transitions can be obtained. These 2^n transitions fully characterize the dynamics of a BN and the table representing these 2^n transitions is called the *state transition table*. We can also associate a directed graph called *state transition diagram* in which a set of nodes is the set of all possible 2^n global states, and there exists a directed edge from $\mathbf{u}$ to $\mathbf{v}$ if and only if $\mathbf{v} = \mathbf{f}(\mathbf{u})$ holds (see also Fig. 1).

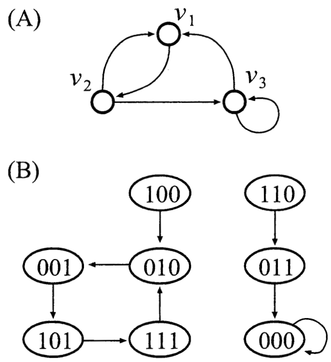


Figure 1: Example of (A) Boolean network and (B) state transition diagram, where $v_1(t+1) = \overline{v_2(t)} \wedge v_3(t)$, $v_2(t+1) = v_1(t)$ and $v_3(t+1) = v_2(t) \oplus v_3(t)$.

For a global state $\mathbf{v}$, a global state $\mathbf{u}$ is called a *predecessor* of $\mathbf{v}$ if $\mathbf{v} = \mathbf{f}(\mathbf{u})$. That is, $\mathbf{u}$ is a predecessor of $\mathbf{v}$ if there is an edge from $\mathbf{u}$ to $\mathbf{v}$ in the state transition diagram of a given BN. Then, BN-PREDECESSOR is defined as follows [3, 4].

Definition 1 (BN-PREDECESSOR) *[3, 4] Suppose that a BN (V, F) and a global state $\mathbf{v}^1$ are given. Then, the problem is to find a global state $\mathbf{v}^0$ such that $\mathbf{v}^1 = \mathbf{f}(\mathbf{v}^0)$. If there does not exist such a global state, "None" should be the output.*

We can generalize the concept of predecessor to k-predecessors. $\mathbf{u}$ is called a *k-predecessor* of $\mathbf{v}$ if k times applications of $\mathbf{f}$ to $\mathbf{u}$ yield $\mathbf{v}$. That is, $\mathbf{u}$ is a k-predecessor of $\mathbf{v}$ if $\mathbf{v} = \overbrace{\mathbf{f}(\mathbf{f}(\cdots}^{k}(\mathbf{u})\cdots))$ holds. Clearly, a usual predecessor is equivalent to a 1-predecessor. We define BN-k-PREDECESSOR to be a problem of finding a k-predecessor of a given global state in a given BN.

2.2 Control of Boolean Network

In BN-CONTROL [1], there are two types of nodes: *internal nodes* and *external nodes*, where internal nodes correspond to usual nodes (i.e., genes) in BN and external nodes correspond to control nodes. Let a set V of $n+m$ nodes be $V = \{v_1, \ldots, v_n, v_{n+1}, \ldots, v_{n+m}\}$, where $v_1, \ldots, v_n$ are internal nodes and $v_{n+1}, \ldots, v_{n+m}$ are external nodes. For convenience, we use x_i to denote an external node v_{n+i}. Then, $v_i(t+1)$ for $i = 1, \ldots, n$ are determined by $v_i(t+1) = f_i(v_{i_1}(t), \ldots, v_{i_{k_i}}(t))$, where each v_{i_k} is either an internal node or an external node. Here, we let $\mathbf{v}(t) = [v_1(t), \ldots, v_n(t)]$ and $\mathbf{x}(t) = [x_1(t), \ldots, x_m(t)]$. We can describe the dynamics of a BN by $\mathbf{v}(t+1) = \mathbf{f}(\mathbf{v}(t), \mathbf{x}(t))$, where $\mathbf{x}(t)$'s are determined externally. Then, BN-CONTROL is defined as follows (see also Fig. 2).

Definition 2 (BN-CONTROL) *[1]*
Suppose that for a BN, we are given an initial state of the network (for internal nodes) $\mathbf{v}^0$ *and the desired state of the network* $\mathbf{v}^M$ *at the* M*-th time step. Then, the problem is to find a sequence of 0-1 vectors* $\langle \mathbf{x}(0), \ldots, \mathbf{x}(M) \rangle$ *such that* $\mathbf{v}(0) = \mathbf{v}^0$ *and* $\mathbf{v}(M) = \mathbf{v}^M$. *If there does not exist such a sequence, "None" should be the output.*

Datta et al. proposed a dynamic programming (DP) algorithm for control of PBN [5]. Here, we briefly review their method in the context of BN. We use a table $D[b_1, \ldots, b_n, t]$, where each entry takes either 0 or 1. $D[b_1, \ldots, b_n, t]$ takes 1 if there exists a desired control sequence beginning from a state $[b_1, \ldots, b_n]$ at time t. This table is computed from $t = M$ to $t = 0$ by using the following recurrence:

$$D[b_1, \ldots, b_n, M] = \begin{cases} 1, & \text{if } [b_1, \ldots, b_n] = \mathbf{v}^M, \\ 0, & \text{otherwise,} \end{cases}$$

$$D[b_1, \ldots, b_n, t-1] = \begin{cases} 1, & \text{if there exists } (\mathbf{a}, \mathbf{x}) \text{ such that } D[a_1, \ldots, a_n, t] = 1 \text{ and } \mathbf{a} = \mathbf{f}(\mathbf{b}, \mathbf{x}), \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathbf{b} = [b_1, \ldots, b_n]$ and $\mathbf{a} = [a_1, \ldots, a_n]$.

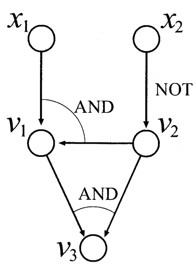
3 Hardness Results

We obtained the following hardness results, which strengthen existing hardness results.

Proposition 3.1 *BN-PREDECESSOR is NP-hard for* $K = 3$.

Theorem 3.2 *BN-CONTROL is NP-hard for* $K = 2$ *and* $M \geq 2$.

Theorem 3.3 *BN-2-PREDECESSOR is NP-hard for* $K = 2$.



$\mathbf{v}^1$ (initial)	0 0 0
$\mathbf{v}^M$ (target)	1 0 1

⇩

t	v_1	v_2	v_3	x_1	x_2
0	0	0	0	1	0
1	0	1	0	1	0
2	1	1	0	1	1
3	1	0	1	0	0

Figure 2: BN-CONTROL is, given initial and desired states of internal nodes (v_1, v_2, v_3), to compute a sequence of states of external nodes (x_1, x_2) that leads to the desired state.

4 Recursive Algorithms for BN-PREDECESSOR

In our previous work [7], we developed a very simple algorithm (called the *basic recursive algorithm*) for identifying all *singleton attractors* along with its variants. Furthermore, we analyzed the average case time complexities. In these algorithms, a partial solution (i.e., a partial global state) is extended one by one according to a given node ordering that leads to a complete solution (i.e., a singleton attractor). If it is found that a partial solution cannot be extended to a complete solution, the next partial solution is examined.

By modifying these algorithm slightly, we can obtain algorithms for BN-PREDECESSOR. For that purpose, we only need to modify the part of

$$\text{it is found that } f_j(\mathbf{v}(t)) \neq v_j(t)$$

in [7] to

$$\text{it is found that } f_j(\mathbf{v}(t)) \neq \mathbf{v}_j^1,$$

where $\mathbf{v}_j^1$ denotes the jth element of a vector $\mathbf{v}^1$. Since both algorithms are almost identical, the same theoretical results on the average case time complexity as in [7] should hold for the modified algorithms (see Table 1). We also performed computational experiments on the modified algorithms and obtained results similar to those in Table 1.

5 Results on Distribution of Predecessors

Coppersmith showed that the probability that a randomly chosen global state has a predecessor is bounded by $(1 - 2^{-2^K})^n$ [4], which approaches to 0

Table 1: Theoretically estimated average case time complexities of basic, outdegree-based, and BFS-based algorithms for the singleton attractor detection problem [7]. The same results should hold for the modified algorithms for BN-PREDECESSOR.

K	2	3	4	5
basic	1.35^n	1.43^n	1.49^n	1.53^n
outdegree-based	1.19^n	1.27^n	1.34^n	1.41^n
BFS-based	1.16^n	1.27^n	1.35^n	1.41^n

as n grows. Based on her idea, we estimate a lower bound of the expected number of predecessors for a global state having at least one predecessor.

Proposition 5.1 *Suppose that for each node, K input nodes are randomly selected and then a Boolean function is randomly selected from 2^{2^K} possible Boolean functions (including constant Boolean functions). Then, the expected number of global states having no predecessor is greater than $2^n \cdot \left(\frac{2^L-1}{2^L}\right)$ where $L = \frac{n}{2^{2^K+1}}$.*

Proposition 5.2 *Suppose that for each node, K input nodes are randomly selected and then a Boolean function is randomly selected from 2^{2^K} possible Boolean functions. Then, once a global state has a predecessor, it is expected to have $2^{n/(2^{(2^K+1)}-1)}$ or more predecessors.*

6 An Improved Algorithm for BN-CONTROL

BN-CONTROL is NP-hard but can be solved in $O(nM2^{m+n})$ time for BNs of bounded indegree by using the DP algorithm in Section 2.2. Though some practically faster algorithm was proposed, no improvement has been done on the theoretical computational complexity. Here, we show an improved algorithm for BN-CONTROL whose average case time complexity is $O(nM2^{m+\beta n})$, where β (< 1) depends on K.

The idea of the improved algorithm is quite simple but non-trivial. We can assume without loss of generality that the constant function 0 is assigned to each of the first $n/2H$ nodes with high probability, where $H = 2^{2^K}$ [2]. Then, we can ignore these nodes and thus we can only consider $2^{n-\frac{n}{2H}}$ states for internal nodes, instead of 2^n states.

As in Section 5, we let $L = \frac{n}{2^{2^K+1}} = \frac{n}{2H}$. For a global state $\mathbf{v}$, $\mathbf{v_0}$ denotes the global state defined by

$$[\overbrace{0, 0, \ldots, 0}^{L}, \mathbf{v}_{L+1}, \mathbf{v}_{L+2}, \ldots, \mathbf{v}_n].$$

Then, the following proposition follows from the definition of $\mathbf{v_0}$.

Proposition 6.1 *Suppose that the constant function 0 is assigned to each of the first L nodes in a BN with external nodes. Then, $\mathbf{f}(\mathbf{v}, \mathbf{x}) = \mathbf{f}(\mathbf{v_0}, \mathbf{x})$ holds for all $\mathbf{v}$.*

Based on this proposition, we can replace $D[b_1, \ldots, b_n, t]$ in the original DP algorithm with $D'[c_1, \ldots, c_{n-L}, t]$. Using this improved algorithm, we can obtain the following theorem.

Theorem 6.2 *Suppose that for each node, K input nodes are randomly selected and then a Boolean function is randomly selected from 2^{2^K} possible Boolean functions. Then, BN-CONTROL for bounded indegree K can be solved in $O(nM2^{m+(1-(1/2^{(2^K+1)}))n})$ time on the average.*

References

[1] T. Akutsu, M. Hayashida, W-K. Ching, and M. K. Ng (2006) Control of Boolean networks: Hardness results and algorithms for tree-structured networks, *Journal of Theoretical Biology*, 244, 670–679.

[2] T. Akutsu, M. Hayashida, S-Q. Zhang, W-K. Ching, and M. K. Ng (2008) Analyses and algorithms for predecessor and control problems for Boolean networks of bounded indegree, *IPSJ Transactions on Bioinformatics*, in press.

[3] C. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns, and M. Thakur (2007) Predecessor existence problems for finite discrete dynamical systems, *Theoretical Computer Science*, 386, 3–37.

[4] S. N. Coppersmith (2007) Complexity of the predecessor problem in Kauffman networks, *Physical Review E*, 75, 051108.

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

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

[7] S-Q. Zhang, M. Hayashida, T. Akutsu, W-K. Ching, and M. K. Ng (2007) Algorithms for finding small attractors in Boolean networks, *EURASIP Journal on Bioinformatics and Systems Biology*, 2007, 20180.