The Second International Symposium on Optimization and Systems Biology (OSB'08)
Lijiang, China, October 31– November 3, 2008

On Distribution and Enumeration of Attractors in Probabilistic Boolean Networks

Morihiro Hayashida[1] Takeyuki Tamura[1] Tatsuya Akutsu[1]
Wai-Ki Ching[2]

[1]Bioinformatics Center, Institute for Chemical Research, Kyoto University,
Gokasho, Uji-city, Kyoto 611-0011, Japan.
[2]Advanced Modelling and Applied Computing Laboratory,
Department of Mathematics, The University of Hong Kong,
Pokfulam Road, Hong Kong.

Abstract In this paper, we study attractors in Probabilistic Boolean Networks (PBNs). We study the expected number of singleton attractors in a PBN and show that it is 1.5^n if the number of Boolean functions assigned to each node is 2, where n is the number of nodes in a PBN. Then, we present algorithms for identifying singleton and small attractors and perform both theoretical and computational analyses on their average case time complexities. The results of computational experiments suggest that these algorithms are much more efficient than the naive algorithm that examines all possible 2^n states. We also show a simple and interesting relation between the distribution of attractors in a BN and the steady-state distribution in a corresponding variant of a PBN.

Keywords Probabilistic Boolean Network; Attractor; Algorithm

1 Introduction

Understanding of the mechanism of gene regulatory networks is an important topic in computational systems biology. For that purpose, many mathematical models of genetic networks have been proposed, which include Bayesian networks, Boolean networks and probabilistic Boolean networks, ordinary and partial differential equations, and qualitative differential equations [7]. For analyzing these mathematical models, use of algebraic methods, symbolic computational methods and optimization methods is recently attracting researchers in various fields [1]

Among the above mathematical models of genetic networks, a lot of studies have been done on the Boolean network (BN in short). BN is a very simple model [8], which combines genetic networks with Boolean algebra. In this model, each node (e.g., gene) takes either 0 (inactive) or 1 (active) and the states of nodes change synchronously according to regulation rules given as Boolean functions. Though many aspects of Boolean networks have been studied, extensive studies have been done on the distribution of length and number of attractors for randomly generated BNs with average indegree K [2, 5], where an attractor corresponds to a steady-state of a cell. However, exact results have not yet been obtained.

Although BN is a deterministic model, real genetic networks contain some uncertainty. In order to cope with this uncertainty, the probabilistic Boolean network (PBN in short) was proposed as a stochastic extension of BN [10]. Only one Boolean function is assigned to each node in a BN, but multiple Boolean functions can be assigned to each node in a PBN and one Boolean function is randomly selected per each node and per each time step. The dynamics of a PBN can be studied in the context of a standard Markov chain [10]. Therefore, the theory of Markov chains has been applied to the analysis of PBNs, in particular, the analysis of the steady-state probability distribution [3, 4, 10, 12]. Unfortunately, it takes at least $O(2^n)$ computational time because the size of a vector representing the probability distribution is 2^n, where n is the number of nodes in a PBN (i.e., the number of genes). However, in many cases, it might be enough to know approximate probabilities of major states. Furthermore, it may be helpful to know attractors in PBNs because singleton or small attractors may correspond to major states in the steady-state probability distribution. Indeed, Brun et al. studied relations between attractors and steady-state probability distributions [3]. However, they did not provide efficient algorithms for computing attractors in BNs or PBNs.

In this paper, we study attractors in PBNs, where attractors do not correspond to steady-states but correspond to attractors in BNs. First, we study the expected number of singleton attractors in PBNs. In particular, we show that this number is 1.5^n for a PBN in which two Boolean functions are assigned to each node. This is an interesting result because it is known that the expected number of singleton attractors in a BN is 1 [6, 9]. Next, we present algorithms for computing singleton and small attractors by extending the techniques proposed in [13]. We show that the average case time complexity is $o(2^n)$ in many cases, which suggests that computation of singleton attractors in PBNs is easier than computation of steady-state distributions in PBNs. We also perform computational experiments in order to verify the theoretically derived time complexities. Then, we show a simple relation between attractors and steady-state probability distributions for a special class of PBN. Finally, we conclude with future work in the last section.

2 BN, PBN and Attractors

A BN $G(V,F)$ consists of a set $V = \{v_1,\ldots,v_n\}$ of *nodes* and a list $F = (f_1,\ldots,f_n)$ of *Boolean functions*. Each node corresponds to a gene and takes either 0 (gene is not expressed) or 1 (gene is expressed) at each discrete time t. The state of node v_i at time t is denoted by $v_i(t)$, where the states on nodes change synchronously according to given regulation rules. A Boolean function $f_i(v_{i_1},\ldots,v_{i_k})$ with inputs from specified nodes $v_{i_1},\ldots,v_{i_k}$ is assigned to each node, where it represents a regulation rule for node v_i. We use $IN(v_i)$ to denote the set of input nodes $v_{i_1},\ldots,v_{i_k}$ to v_i. Then, the state of node v_i at time $t+1$ is determined by $v_i(t+1) = f_i(v_{i_1}(t),\ldots,v_{i_k}(t))$. Here we let $\mathbf{v}(t) = [v_1(t),\ldots,v_n(t)]$, which is called a *Gene Activity Profile* (GAP) at time t or a (global) state of BN at time t. We also write $v_i(t+1) = f_i(\mathbf{v}(t))$ to denote the regulation rule for v_i. Furthermore, we write $\mathbf{v}(t+1) = \mathbf{f}(\mathbf{v}(t))$ to denote the regulation rule for the whole BN. The number of input nodes to v_i (i.e., $|IN(v_i)|$) is called the *indegree* of v_i. We use K to denote the *maximum indegree* of a BN, which plays an important role in analysis of BNs. An example of BN is shown in Fig. 1(A).

The dynamics of a BN can be well-described by a *state transition diagram* shown in

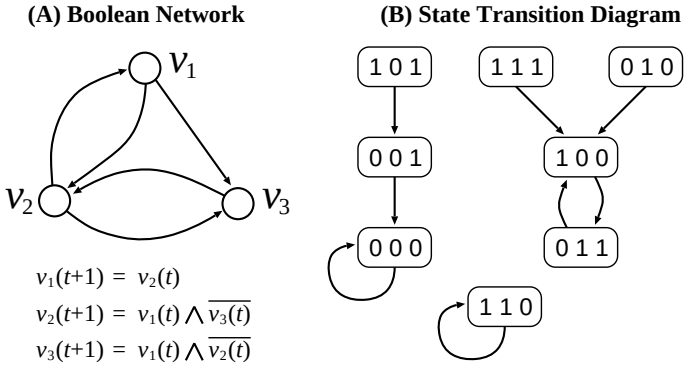

Figure 1: Example of Boolean network (A), and its state transition diagram (B).

Fig. 1(B). For example, an edge from 101 to 001 means that if GAP of BN is $[1,0,1]$ at time t, GAP of BN becomes $[0,0,1]$ at time $t+1$. From this diagram, it can be seen that if $\mathbf{v}(0) = [1,0,1]$, GAP changes as follow:

$$[1,0,1] \Longrightarrow [0,0,1] \Longrightarrow [0,0,0] \Longrightarrow [0,0,0] \Longrightarrow \cdots$$

and the same GAP $[0,0,0]$ is repeated after $t = 1$. It is also seen that if BN begins from $\mathbf{v}(0) = [1,1,1]$, then $[1,0,0]$ and $[0,1,1]$ are repeated alternatively after $t = 0$. These kinds of sets of repeating states are called *attractors*, each of which corresponds to a directed cycle in a state transition diagram. The number of elements in an attractor is called the *period* of the attractor. An attractor with period 1 is called a *singleton* attractor, which corresponds to a fixed point. An attractor with period greater than 1 is called a *cyclic* attractor. In the BN of Fig. 1(A), there are three attractors: $\{[0,0,0]\}$, $\{[1,1,0]\}$, $\{[1,0,0],[0,1,1]\}$, where the first and second ones are singleton attractors and the third one is a cyclic attractor with period 2.

PBN is an extension of BN. The difference between BN and PBN is only that in a PBN, for each vertex v_i, instead of having only one Boolean function, there are a number of Boolean functions (predictor functions) $f_j^{(i)} (j = 1,2,\ldots,l(i))$ to be chosen for determining the state of gene v_i. The probability of choosing $f_j^{(i)}$ is $c_j^{(i)}$, where $c_j^{(i)}$ should satisfy the followings:

$$0 \leq c_j^{(i)} \leq 1 \quad \text{and} \quad \sum_{j=1}^{l(i)} c_j^{(i)} = 1 \quad \text{for} \quad i = 1,2,\ldots,n.$$

An example of PBN is shown in Fig. 2 (A). A part of corresponding state transition diagram is shown in Fig. 2 (B). In Fig. 2 (B), only transitions from [0,0,0],[0,1,0],[1,0,0],[1,1,0] are described since the diagram would be very complex if all transitions were described.

Let f_j be the jth possible realization,

$$f_j = (f_{j_1}^{(1)}, f_{j_2}^{(2)}, \ldots, f_{j_n}^{(n)}),\ 1 \leq j_i \leq l(i),\ i = 1,2,\ldots,n.$$

The probability of choosing such a realization in an independent PBN (the selection of

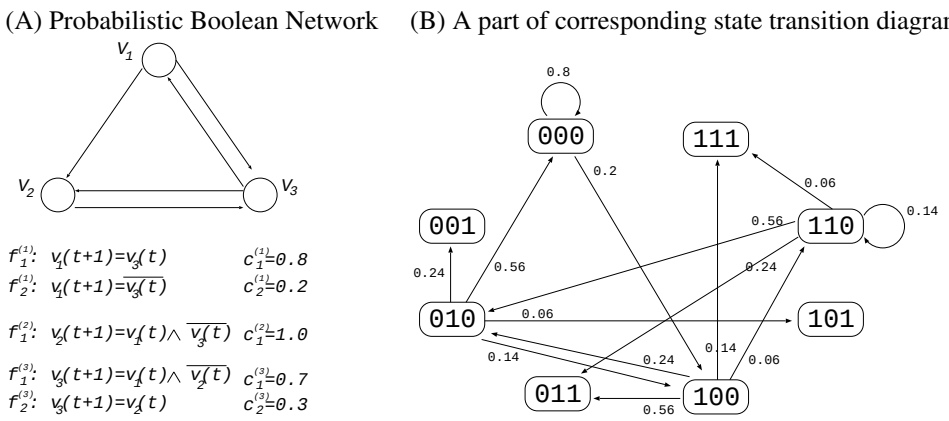

Figure 2: Example of probabilistic Boolean network (A), and a part of its state transition diagram (B).

the Boolean function for each gene is independent) is given by

$$p_j = \prod_{i=1}^{n} c_{j_i}^{(i)}, j = 1,2,\ldots,N$$

where $N = \prod_{i=1}^{n} l(i)$ is the maximum possible number of different realizations of BNs. The dynamics of a PBN can be well described by the state transition probabilities. Let $\mathbf{u}$ be a GAP of a PBN at time t. Then, we can calculate the probability that the PBN takes a GAP $\mathbf{w}$ at time $t+1$. That is, we can calculate the state transition probability $Prob(\mathbf{v}(t+1) = \mathbf{w} \mid \mathbf{v}(t) = \mathbf{u})$ of a given PBN. Since there are 2^n GAPs, these probabilities can be represented by a $2^n \times 2^n$ matrix called the transition probability matrix.

In a PBN, we also call a set of GAPS $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p\}$ an *attractor of period p* if $Prob(\mathbf{v}(t+1) = \mathbf{v}_{i+1} | \mathbf{v}(t) = \mathbf{v}_i) \neq 0$ holds for all $i = 1,2,\ldots,p$, where we identify $p+1$ and 1. As in BN, we also call an attractor with period 1 a *singleton attractor*. In Fig. 2, $\{[0,1,0],[1,0,0]\}$ is an attractor with period 2, and each of $[1,1,0]$ and $[0,0,0]$ is a singleton attractor.

3 Expected Number of Singleton Attractors in PBN

In this section, we show how many singleton attractors there are in a PBN in average. Note that definitions of a singleton attractor and a steady-state of a PBN are different from each other. Recall that $\mathbf{u}$ is called a singleton attractor of a PBN if $f_j(\mathbf{u}) = \mathbf{u}$ holds for some j ($1 \leq j \leq N$). The following theorem shows that the expected number of singleton attractors in a PBN is quite large when compared to that in a BN, which is known to be 1 [6, 9], although the proof of the theorem is omitted due to the page limitation.

Theorem 1.
Suppose that f_i and $|IN(v_i)|$ are randomly assigned for each v_i. When $l(i) = L$ holds for any i, the expected number of singleton attractors in a PBN is $\left\{2 - \left(\frac{1}{2}\right)^{L-1}\right\}^n$.

Table 1: (A) Theoretically estimated average case time complexity for $p = 1$, $K = \{2,3,4\}$, $L = \{2,3\}$. (B) Theoretically estimated average case time complexity for $p = 1$, $K = \{2,3,4\}$, $L = \{2,3\}$ when $IN(f_{j_1}^{(i)}) = IN(f_{j_2}^{(i)})$ holds for any j_1 and j_2 $(1 \leq j_1, j_2 \leq l(i))$.

(A)

	$L=2$	$L=3$
K=2	$O(1.601^n)$	$O(1.763^n)$
K=3	$O(1.658^n)$	$O(1.790^n)$
K=4	$O(1.698^n)$	$O(1.813^n)$

(B)

	$L=2$	$L=3$
K=2	$O(1.523^n)$	$O(1.750^n)$
K=3	$O(1.565^n)$	$O(1.750^n)$
K=4	$O(1.601^n)$	$O(1.751^n)$

Corollary 1.
When $l(i) = 2$ holds for any i, the expected number of singleton attractors in a PBN is $(1.5)^n$.

4 Algorithms for Computing Attractors in a PBN

In this section, we present an algorithm for finding all singleton attractors of a PBN by extending branch-and-bound type algorithms proposed in [13], where the branch-and-bound method is one of the widely used techniques in combinatorial optimization.

This algorithm can be extended to find cyclic attractors of a PBN. The pseudo code of the algorithm is given below:

Pseudo code

Input: a PBN, **Output:** all singleton attractors

Begin

initialize $m = 1$;

Procedure $PBNAttractor(\mathbf{v}, m)$

if $m = n+1$ **then** output $v_1(t), v_2(t), \ldots, v_n(t)$ **return;**

for $b = 0$ **to** 1 **do** $v_m(t) = b$;

if it is found that $f_j^{(i)}(v_i(t)) \neq v_i(t+1)$ holds for each j for some $i \leq m$ **then continue;**

else $PBNAttractor(\mathbf{v}, m+1)$;

return;

End

Although the detailed analysis is omitted due to the page limitation, theoretically estimated average computational complexity of the above algorithm is shown in Tables 1 (A) and (B) for various parameters.

A problem for finding attractors with period $x \leq p$ of a PBN, which we call *type-1 problem*, can be solved by extending the above algorithm. The overall computational time for type-1 problem can be represented by $time(x = 1) + \cdots + time(x = p)$, where each term will be an exponential of n as shown below. Therefore, when p is small, the overall average time complexity will only be affected by the largest term in the above formula. Thus, the order of the time complexity for a problem of finding attractors with period p, which we call *type-2 problem*, is theoretically the same as that of type-1 problem.

Table 2: (A) Theoretically estimated average case time complexity for $p = 2$, $K = \{2,3,4\}$, $L = \{2,3\}$. (B) Theoretically estimated average case time complexity for $p = 2$, $K = \{2,3,4\}$, $L = \{2,3\}$ when $IN(f_{j_1}^{(i)}) = IN(f_{j_2}^{(i)})$ holds for any j_1 and j_2 $(1 \leq j_1, j_2 \leq l(i))$.

(A)

	$L=2$	$L=3$
K=2	$O(1.753^n)$	$O(1.846^n)$
K=3	$O(1.835^n)$	$O(1.896^n)$
K=4	$O(1.882^n)$	$O(1.926^n)$

(B)

	$L=2$	$L=3$
K=2	$O(1.658^n)$	$O(1.763^n)$
K=3	$O(1.753^n)$	$O(1.813^n)$
K=4	$O(1.816^n)$	$O(1.854^n)$

Therefore, we consider type-2 problem instead of type-1 problem in the following. The pseudo code of the proposed algorithm is given below:

Pseudo code

Input: a PBN, **Output:** all singleton attractors with period p

Begin
define *x-ancestor*(v_i): nodes initializing paths to v_i whose lengths are less than or equal to x.
initialize $m = 1$;
Procedure $PBNcycAttractor(\mathbf{v}, m)$
if $m = n+1$ **then** output $v_1(t), v_2(t), \ldots, v_n(t)$ **return;**
for $b = 0$ **to** 1 **do** $v_m(t) = b$;
$flag = 0$; $i = 1$;
while $flag = 0$ and $i \leq m$
if every *p-ancestor*(v_i) is assigned
$r = 0$;
while $flag = 0$ and $r \leq p-1$
if it is found that $f_j^{(i)}(v_i(t+r)) \neq v_i(t+r+p)$ holds for each j **then** $flag = 1$;
$r = r+1$;
$i = i+1$;
if $flag = 1$ **then continue;**
else $PBNcycAttractor(\mathbf{v}, m+1)$;
return;
End

Although the detailed analysis is omitted due to the page limitation, theoretically estimated average case time complexity for various parameters is as shown in Tables 2 (A) and (B).

5 Results of Computational Experiments

In this section, we evaluate expected numbers of singleton attractors in PBN and time complexities of the proposed algorithms by performing some computational experiments on random networks.

Table 3: (A) Empirical numbers of singleton attractors in PBN for $K = \{2,3,4\}$ and $L = \{2,3\}$. (B) Empirical time complexities estimated using 100 randomly generated PBNs with up to 30 nodes for $p = \{1,2\}$, $K = \{2,3,4\}$ and $L = \{2,3\}$ (for the case of $L = 2$ and $p = 1$, 1,000 PBNs were used).

(A)

	$L=2$	$L=3$
$K=2$	1.499^n	1.751^n
$K=3$	1.450^n	1.750^n
$K=4$	1.450^n	1.749^n

(B)

	$L=2$		$L=3$
	$p=1$	$p=2$	$p=1$
$K=2$	$O(1.694^n)$	$O(1.979^n)$	$O(1.855^n)$
$K=3$	$O(1.758^n)$	$O(2.059^n)$	$O(1.904^n)$
$K=4$	$O(1.779^n)$	$O(2.079^n)$	$O(1.920^n)$

Table 4: Empirical time complexities estimated by the results up to $n = \{20,25,30,35\}$ for $p = 1$, $K = \{2,3,4\}$ and $L = 2$.

	$n \leq 20$	$n \leq 25$	$n \leq 30$	$n \leq 35$
$K=2$	$O(1.726^n)$	$O(1.718^n)$	$O(1.694^n)$	$O(1.685^n)$
$K=3$	$O(1.783^n)$	$O(1.772^n)$	$O(1.758^n)$	$O(1.730^n)$
$K=4$	$O(1.830^n)$	$O(1.814^n)$	$O(1.779^n)$	$O(1.761^n)$

For each $K(K = \{2,3,4\})$, we randomly generated 100 or 1,000 PBNs with n ($n = 5,10,15,20,25,30$) nodes and $L(L = \{2,3\})$ Boolean functions for each node, and took the average values. All of these computational experiments were done on a PC with Xeon X5460 3.16GHz CPUs and 10GB memory running under the Linux (version 2.6) operating system, where the icc compiler was used with optimization option -O3. Each experiment was performed on a single thread although the PC has multiple processor cores.

Table 3 (A) shows the empirical numbers of singleton attractors in PBN for each L ($L = \{2,3\}$). We used a tool for GNUPLOT to fit the function a^n to the experimental results. This tool uses the nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm. We can see from the table that the empirical numbers were almost the same numbers as those obtained by theoretical analysis.

Table 3 (B) shows the empirical time complexities of the proposed algorithms estimated by 100 randomly generated PBNs with up to 30 nodes for $p = \{1,2\}$, $K = \{2,3,4\}$ and $L = \{2,3\}$, where for the case of $L = 2$ and $p = 1$, 1,000 PBNs were used. We fit the function $b \cdot a^n$ to the experimental results in order to obtain the empirical time complexities. The empirical time complexities were slightly larger than those derived from the theoretical analysis.

Fig. 3 shows the average elapsed time of the proposed algorithms for $p = 1$, $K = \{2,3,4\}$ and $L = 2$. We can see that the slopes became low-angled as the number of nodes became larger. For each $n(n = 20,25,30,35)$, we fit the function to the results up to n, and obtained the time complexities. Table 4 shows the time complexity estimated using the results up to n for $p = 1$, $K = \{2,3,4\}$ and $L = 2$. The time complexities became smaller as the number of nodes became larger. The empirical time complexities estimated from the results for PBNs with less than 35 nodes were larger than those derived from theoretical analysis. It is highly expected that the empirical time complexities approach

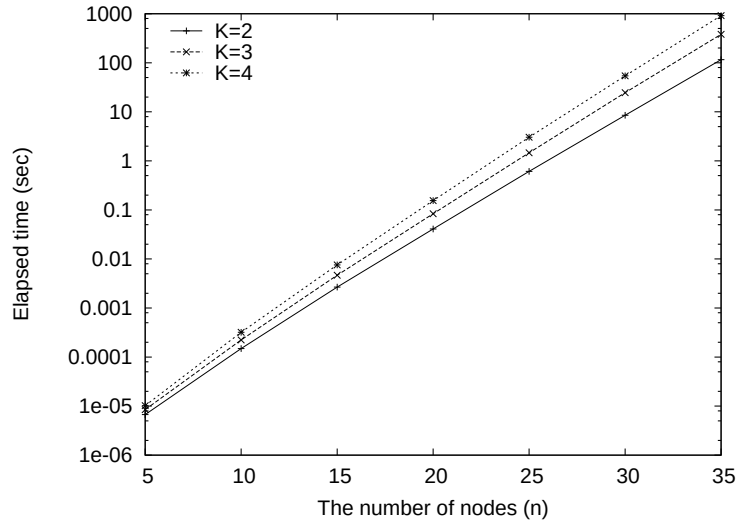

Figure 3: Average elapsed time (seconds) obtained by 1,000 randomly generated PBNs for $p = 1$, $K = \{2,3,4\}$ and $L = 2$.

asymptotically to those derived from theoretical analysis if we can perform experiments for sufficient large n. Though we did not examine enough numbers of repetitions, the algorithm could handle BNs with up to $45 \sim 50$ nodes for the case of $K = L = 2$. In the case of $n = 45$, the algorithm took several or several tens of thousands seconds. Though it is not very fast, it should still be much faster than the naive algorithm which examines all 2^n states.

The empirical time complexities for $p = 1$ and $L = 2$ when $IN(f_{j_1}^{(i)}) = IN(f_{j_2}^{(i)})$ holds for any j_1 and j_2 ($1 \leq j_1, j_2 \leq l(i)$) were $O(1.639^n)$, $O(1.674^n)$, $O(1.715^n)$ for $K = 2,3,4$ respectively. In this case, the time complexities estimated from the results of computational experiments are a little different from those obtained by the theoretical analysis. However, it is reasonable because we assumed that the number of nodes is very large in the theoretical analysis.

6 Relation between Steady-State Probabilities and Attractors

In this section, we discuss a simple relation between the steady-state probability distribution in a PBN and the structure of attractors in a BN. Though Brun et al. have already derived some relations [3], we derive a simpler relation for a special case.

Let $\mathbf{u}, \mathbf{w}$ be GAPs of a BN at time t and $t+1$, respectively. Since GAP at time $t+1$ is given deterministically from GAP at time t in BN, we can write $Prob(\mathbf{v}(t+1) = \mathbf{w} | \mathbf{v}(t) = \mathbf{u}) = 1$. For the other GAPs $\mathbf{w}'$, we can write $Prob(\mathbf{v}(t+1) = \mathbf{w}' | \mathbf{v}(t) = \mathbf{u}) = 0$.

Here, we construct a variant of PBN by introducing slight perturbation to this BN. Recall that dynamics of a PBN can be represented by a $2^n \times 2^n$ matrix. If $\mathbf{w}$ is the next

GAP to $\mathbf{u}$, we let

$$Prob(\mathbf{v}(t+1) = \mathbf{w} | \mathbf{v}(t) = \mathbf{u}) \ = \ 1 - \frac{n-1}{n}\varepsilon.$$

For other $\mathbf{w}'$, we let

$$Prob(\mathbf{v}(t+1) = \mathbf{w}' | \mathbf{v}(t) = \mathbf{u}) \ = \ \varepsilon.$$

Let $P(\varepsilon)$ be the $2^n \times 2^n$ matrix corresponding to these transition probabilities and $\mathbf{x}$ be the corresponding steady-state probability vector (i.e., $\mathbf{x} = P(\varepsilon)\mathbf{x}$). It should be noted that $\mathbf{u}, \mathbf{w}, \mathbf{w}'$ are n-dimensional 0-1 vectors whereas $\mathbf{x}$ is a 2^n-dimensional real vector, and each element of $\mathbf{x}$ corresponds to a probability of some GAP $\mathbf{u}$ in a BN. It is also to be noted that $\mathbf{x}$ is uniquely determined because its norm is 1 and all the transition probabilities are non-zero.

Let A be an attractor with period p in a BN (i.e., $p = |A|$). Let B be the *basin* of the attractor A. That is, B is the set of GAPs that eventually fall into A (we assume that A is included in B). Let $\mathbf{u}$ be an arbitrary GAP in A. We consider the case when ε is close to 0.

Let $\pi(\mathbf{u})$ be the steady-state probability of $\mathbf{u}$. Let $\pi(B) = \sum_{\mathbf{u} \in B} \pi(\mathbf{u})$. Then, as shown in [3], we have

$$\pi(\mathbf{w}) \approx \begin{cases} (1/p) \cdot \pi(B), & \text{if } \mathbf{w} \in A, \\ 0, & \text{if } \mathbf{w} \in B - A. \end{cases}$$

For each basin B, we consider its complement $V - B$. Since the probability emitting from B and the probability incoming to B should be balanced, we have

$$|B|\varepsilon(1 - \pi(B)) \approx (|V| - |B|)\varepsilon\pi(B).$$

Therefore, we have $\pi(B) \approx \frac{|B|}{2^n}$ and thus we have

$$\pi(\mathbf{w}) \approx \frac{|B|}{|A| \cdot 2^n}$$

for each state $\mathbf{w}$ in an attractor A with basin B.

This result is interesting because it relates steady-state probabilities with the sizes of attractors and their basins.

7 Concluding Remarks

In this paper, we studied attractors in PBNs. We showed theoretical and experimental results on the number of singleton attractors in PBNs. Extension of the results for cyclic attractors is an important future work though it seems very difficult since no rigorous results are known for the number of cyclic attractors even on BNs.

We presented algorithms for finding singleton and small attractors in PBNs and performed theoretical and empirical analyses of their average case time complexities. Though the proposed algorithms are much faster than the naive algorithm that examines all 2^n states, these cannot yet handle very large PBNs. Therefore, improvement of the algorithms is also an important future work.

In order to relate attractors in BNs and steady-state distributions in PBNs, we derived a simple and interesting relation. However, this relation holds only for a special variant of PBN. Therefore, derivation of more general relations is left as a future work.

Acknowledgements

We would like to thank Susumu Goto, Atsushi Mochizuki and Kei Tokita for helpful discussions.

References

[1] Anai, H., Horimoto, K. and Kutsia, T., *Algebraic Biology, Second International Conference*, Springer, 2007.

[2] Bilke, S. and Sjunnesson, F., Number of attractors in random Boolean networks, *Physical Review E*, **72**, 016110, 2005.

[3] Brun, M., Dougherty, E. R. and Shmulevich, I., Steady-state probabilities for attractors in probabilistic Boolean networks, *Signal Processing*, **85**, 1993–2013, 2005.

[4] Ching, W., Zhang, S., Ng, M. and Akutsu T., An approximation method for solving the steady-state probability distribution of probabilistic Boolean networks, *Bioinformatics*, **23**, 1511–1518, 2007.

[5] Drossel, B., Mihaljev, T. and Greil, F., Number and length of attractors in a critical Kauffman model with connectivity one, *Physical Review Letters*, **94**, 088701, 2005.

[6] Harvey, I. and Bossomaier, T., Time out of joint: Attractors in asynchronous random boolean networks, *Proc. 4th European Conference on Artificial Life*, 67–75, MIT Press, 1997.

[7] Jong, H. D., Modeling and simulation of genetic regulatory systems: a literature review, *Journal of Computational Biology*, **9**, 67–103, 2002.

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

[9] Mochizuki, A., An analytical study of the number of steady states in gene regulatory networks, *Journal of Theoretical Biology*, **236**, 291–310, 2005.

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

[11] Shmulevich, I., Gluhovsky, I., Hashimoto, R. F., Dougherty, E. R. and Zhang, W., Steady-state analysis of genetic regulatory networks modeled by probabilistic Boolean networks, *Comparative and Functional Genomics*, **4**, 601–608, 2003.

[12] Zhang, S., Ching, W., Ng, M. and Akutsu, T.. Simulation study in probabilistic Boolean network models for genetic regulatory networks, *Journal of Data Mining and Bioinformatics*, **1**, 217–240, 2007.

[13] Zhang, S., Hayashida, M., Akutsu, T., Ching, W. and Ng, M., Algorithms for finding small attractors in Boolean networks, *EURASIP Journal on Bioinformatics and Systems Biology*, Article ID 20180, 2007.