On the Synthesis of Stochastic Flow Networks

Hongchao Zhou
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125
Email: hzhou@caltech.edu

Ho-Lin Chen
Center for Mathematics of Information
California Institute of Technology
Pasadena, CA 91125
holinc@gmail.com

Jehoshua Bruck
Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125
bruck@caltech.edu

***Abstract*—A stochastic flow network is a directed graph with incoming edges (inputs) and outgoing edges (outputs), tokens enter through the input edges, travel stochastically in the network and can exit the network through the output edges. Each node in the network is a splitter, namely, a token can enter a node through an incoming edge and exit on one of the output edges according to a predefined probability distribution. We address the following synthesis question: Given a finite set of possible splitters and an arbitrary rational probability distribution, design a stochastic flow network, such that every token that enters the input edge will exit the outputs with the prescribed probability distribution.**

The problem of probability synthesis dates back to von Neumann's 1951 work and was followed, among others, by Knuth and Yao in 1976, who demonstrated that arbitrary rational probabilities can be generated with tree networks; where *minimizing the expected path length*, the expected number of coin tosses in their paradigm, is the key consideration. Motivated by the synthesis of stochastic DNA based molecular systems, we focus on designing *optimal-sized* stochastic flow networks (the size of a network is the number of splitters). We assume that each splitter has two outgoing edges and is unbiased (probability $\frac{1}{2}$ per output edge). We show that an arbitrary rational probability $\frac{a}{b}$ with $a \leq b \leq 2^n$ can be realized by a stochastic flow network of size n, we also show that this is optimal. We note that our stochastic flow networks have feedback (cycles in the network), in fact, we demonstrate that feedback improves the expressibility of stochastic flow networks, since without feedback only probabilities of the form $\frac{a}{2^n}$ (a an integer) can be realized.

I. Introduction

The problem of probability synthesis dates back to von Neumann's 1951 work [1], where he considered the problem of simulating an unbiased coin by using a biased coin with unknown probability. He noticed that when a coin is tossed twice, the events HT (Heads and then Tail) and TH (Tail and then Heads) have identical probabilities, hence, in his simulation algorithm HT produces the output 0 and TH produces the output 1. The other two events, namely HH and TT, are ignored. Knuth and Yao [2] gave a procedure to generate an arbitrary probability distribution using an unbiased coin. They use the concept of an edge-labeled tree called generating tree and show that the expected number of coin tosses is upper-bounded by the entropy of the target distribution plus two.

In this paper we generalize the concept of a generating tree and consider general directed graphs. Specifically, we introduce the concept of a stochastic flow network - it is a directed graph with incoming edges (inputs) and outgoing edges (outputs), tokens enter through the input edges, travel stochastically in the network and can exit the network through the output edges. Each node in the network is a splitter, namely, a token can enter a node through an incoming edge and exit on one of the output edges according to a predefined probability distribution.

One application of stochastic flow networks is the synthesis of stochastic DNA based molecular systems [3], which is becoming an alternative way to do computing and control. In such systems, stochasticity plays an important rule. Hence, a natural question is that how to manipulate this stochasticity and synthesize desired probabilities in such systems. Note that people still don't know that how to implement memories using molecular reactions, and usually these systems are used to work as computing or control elements of a biological system, without connecting with electrical devices. So we cannot store some probabilities at first and then post-process them using some mathematical methods (such as Knuth and Yao's scheme). Instead, we can construct stochastic flow networks, where each splitter is implemented with two molecular species such that one incoming token can react with either of the two species with certain probabilities.

Fig. 1 depicts von Neumann's algorithm in the language of stochastic flow networks. Each node is a splitter and the probabilities of the H and T edges are p and $(1-p)$, respectively (the value of p is not known). A notation: A splitter with two outgoing edges, with probabilities p and $(1-p)$ will be called a p-splitter. Assume that a token starts flowing from the root of the tree, at each splitter, it stochastically selects one edge (H with probability p_H or T with probability p_T) to follow. Finally, the token will reach one of the leaves of the tree, called outputs. In general, the outputs of a stochastic flow network have labels denoted by $\{\beta_1, \beta_2, ..., \beta_m\}$. A token will reach an output β_k $(1 \leq k \leq m)$ with probability q_k, and we call $\{q_1, q_2, ..., q_m\}$ the output probability distribution of the network, where $\sum_{k=1}^{m} q_k = 1$.

The work of Knuth and Yao reasons about a generating tree as an algorithm that is maximizing the expected number of desired random bits generated per coin toss. However, motivated by the synthesis of stochastic DNA based molecular systems, we focus on designing *optimal-sized* stochastic flow networks (the size of a network is the number of splitters). This goal is different from the goal in the related literature:

This work was supported in part by the NSF Expeditions in Computing Program under grant CCF-0832824.

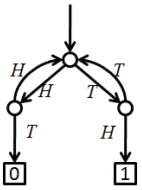


Fig. 1. A network that realizes distribution $\{\frac{1}{2}, \frac{1}{2}\}$, with two p-splitters, where p is unknown.

Elias [4] demonstrated a construction in which the expected number of unbiased random bits generated per coin toss is asymptotically equal to the entropy of the biased coin. Pae and Loui [5] further proved that the mapping function used by Elias is optimal among all n-randomizing functions and is computable in polynomial time. Han and Hoshi [6] and Abrahams [7] considered the case when the tossed coin is a general biased M-sided coin. Blum [8] have studied a general situation that simulating an unbiased coin using sequences produced by an unknown Markov Chain. Gill [9] discussed the problem of generating rational probabilities using a sequential state machine. However, the state machine needs to run for an infinitely long time to get an accurate desired probability. Wilhelm and Bruck [10] proposed a procedure for synthesizing stochastic relay circuits to realize desired binary probabilities. Inspired by PCMOS technology, Qian and Riedel [11] considered the synthesis of of decimal probabilities using combinational logic. However, none of the foregoing approaches considered the problem of generating arbitrary rational probabilities, using a token based approach, while optimizing the network size.

In this paper, we address the following synthesis question: Given a finite set of possible splitters and an arbitrary rational probability distribution, design a stochastic flow network, such that every token that enters the input edge will exit the outputs with the prescribed probability distribution. We assume, without loss of generality, that the probability of each splitter is $\frac{1}{2}$ (since von Neumann's construction in Fig. 1 can use any p-splitter to simulate a $\frac{1}{2}$-splitter). Our goal is to realize the desired probabilities by constructing a network of minimal size. In addition, we study the expected latency, namely the expected number of splitters a token need to pass before reaching the output.

The main contributions of the paper are

1) *General optimal construction:* For any desired rational probability, an *optimal* size construction of stochastic flow network is provided (Section III).
2) *The power of feedback:* With feedback (loops), stochastic flow networks can generate much more probabilities than those without feedback (Section III).
3) *Constructions with well-bounded expected latency:* Two additional constructions with a few more splitters than the optimal construction are proposed such that their expected latencies are well-bounded by a constant (Section IV).

II. Absorbing Markov Chains

Let's consider a flow network with n splitters and m outputs, in which each splitter is associated with a state number in $\{1, 2, ..., n\}$ and each output is associated with a state number in $\{n+1, n+2, ..., n+m\}$. When a token reaches splitter i with $1 \leq i \leq n$, we say that the current state of this network is i. When it reaches output k with $1 \leq k \leq m$, we say that the current state of this network is $n+k$. Note that the current state of the network only depends on the last state, and when the token reach one output it will stay there forever. So we can describe token flow in this network using an absorbing Markov chain. If the current state of the network is i, then the probability of reaching state j in the next instant of time is given by p_{ij}. Here, $p_{ij} = p_H$ ($p_{ij} = p_T$) if and only if state i and state j is connected by an edge H (T).

Clearly, we have

$$\begin{array}{ll} \sum_{j=1}^{n+m} p_{ij} = 1 & i = 1, 2, ..., n+m \\ p_{ij} = 0 & \forall i > n \text{ and } i \neq j \\ p_{ii} = 1 & \forall i > n \end{array}$$

Then the network with n splitters and m outputs with different labels can be described by an absorbing Markov chain, where the first n states are transient states and the last m states are absorbing states. The transition matrix of this Markov chain is given by

$$P = \begin{array}{c} \\ n \\ m \end{array} \begin{array}{c} \begin{array}{cc} n & m \end{array} \\ \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix} \end{array}$$

where Q is an $n \times n$ matrix, R is an $n \times m$ matrix, 0 is an $m \times n$ zeros matrix and I is an $m \times m$ identity matrix.

Let B_{ij} be the probability that an absorbing chain will be absorbed in the absorbing state $j+n$ if it starts in the transient state i. Then B is an $n \times m$ matrix, and

$$B = (I - Q)^{-1} R$$

Assume this markov chain starts from state 1 and let S_j be the probability that it will be absorbed in the absorbing state $j + n$. Then S is the distribution of the network

$$S = [1, 0, ..., 0]B = e_1(I - Q)^{-1}R$$

III. Optimal Construction with Feedback

In this section, we consider the scenario that the splitter probability is $\frac{1}{2}$ and we want to demonstrate the importance of feedback (loops) in networks to generate desired probabilities.

A. Loop-free networks

Here, we want to study the expressive power of loop-free networks. We say that there are no loops in a network, that means no token will appear at the same position for more than one time in the given network. For loop-free networks, we have the following theorem:

Theorem 1. *For a loop-free network with n $\frac{1}{2}$-splitters, all probability $\frac{x}{2^n}$ with integer $x (0 \leq x \leq 2^n)$ can be realized,*

and only probability $\frac{x}{2^n}$ with integer $x(0 \le x \le 2^n)$ can be realized.

Proof: (for short) a) For any probability $\frac{x}{2^n}$ with integer $x(0 \le x \le 2^n)$, we can construct a stochastic flow networks with n splitters using Knuth and Yao's scheme.

b) For a network without loops, the probability for a token to reach a given output is $P = \sum_k P_k$, where P_k is the path gain of a forward path from the root to the output. Given n splitters, the length of each forward path should be at most n. So for each k, P_k can be written as $\frac{x_k}{2^n}$ for some x_k. ■

B. Networks with loops

Now, we introduce feedback into networks. We will show that feedback (loops) can play an important rule to enhance the expressibility of flow networks. For any desired rational probability $\frac{a}{b}$ with integers $0 \le a \le b \le 2^n$, we have the following theorem:

Theorem 2. *For a network with n $\frac{1}{2}$-splitters, all rational probability $\frac{a}{b}$ with integers $0 \le a \le b \le 2^n$ can be realized , and only rational probability $\frac{a}{b}$ with integers $0 \le a \le b \le 2^n$ can be realized.*

Proof: a) We prove that all rational probability $\frac{a}{b}$ with integers $0 \le a \le b \le 2^n$ can be realized. When $b = 2^n$, the problem becomes trivial due to the result of Theorem 1. In the following proof, we only consider the case that $2^{n-1} < b < 2^n$ for some n.

We first prove that all probability distributions $\{\frac{x}{2^n}, \frac{y}{2^n}, \frac{z}{2^n}\}$ with integers x, y, z s.t. $(x+y+z = 2^n)$ can be realized with n splitters. Now we construct this network iteratively.

When $n = 1$, by enumerating all the possible connections, the following probability distributions can be realized:

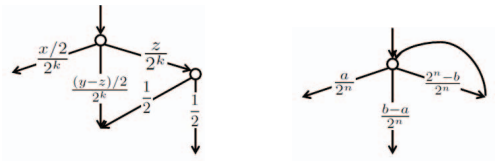
$$\{0,0,1\}, \{0,1,0\}, \{1,0,0\}, \{0, \frac{1}{2}, \frac{1}{2}\}, \{\frac{1}{2}, 0, \frac{1}{2}\}, \{\frac{1}{2}, \frac{1}{2}, 0\}$$

So all probability distributions $\{\frac{x}{2}, \frac{y}{2}, \frac{z}{2}\}$ with integers x, y, z s.t. $(x + y + z = 2)$ can be realized.

Assume that all probability distribution $\{\frac{x}{2^k}, \frac{y}{2^k}, \frac{z}{2^k}\}$ with integers x, y, z s.t. $(x + y + z = 2^k)$ can be realized by a network with k splitters. Then we show that any desired probability distribution $\{\frac{x}{2^{k+1}}, \frac{y}{2^{k+1}}, \frac{z}{2^{k+1}}\}$ s.t. $x + y + z = 2^{k+1}$ can be realized with one more splitter. Since $x+y+z = 2^{k+1}$, we know that at least one of x, y, z is even. W.l.o.g, we let x be even. Then either both y and z are even, or both y and z are odd.

When both y and z are even, the problem is trivial since the desired probability distribution can be written as $\{\frac{x/2}{2^k}, \frac{y/2}{2^k}, \frac{z/2}{2^k}\}$, which can be realized by a network with k splitters based on our assumption.

When both y and z are odd, W.l.o.g, we assume that $z \le y$. In this case, we construct a network to realize probability distribution $\{\frac{x/2}{2^k}, \frac{(y-z)/2}{2^k}, \frac{z}{2^k}\}$ with k splitters. By connecting the last output with probability $\frac{z}{2^k}$ to an additional splitter, we can get a new network in Fig. 2(a), whose probability distribution is $\{\frac{x}{2^{k+1}}, \frac{y}{2^{k+1}}, \frac{z}{2^{k+1}}\}$.



(a) (b)

Fig. 2. (a) The network to realize $\{\frac{x}{2^{k+1}}, \frac{y}{2^{k+1}}, \frac{z}{2^{k+1}}\}$ iteratively. (b) The network to realize $\{\frac{a}{b}, 1 - \frac{a}{b}\}$.

Iteratively, for any probability distribution $\{\frac{x}{2^n}, \frac{y}{2^n}, \frac{z}{2^n}\}$ with $x + y + z = 2^n$, we can always construct a network with n splitters to realize it.

In order to realize probability $\frac{a}{b}$ with $2^{n-1} < b < 2^n$, we can construct a network with probability distribution $\{\frac{a}{2^n}, \frac{b-a}{2^n}, \frac{2^n-b}{2^n}\}$ with n splitters, then connect the last output (output 2) to the starting point of the network, as shown in Fig. 2(b). Using the method in Section II, we can get that in this new network the probability for a token to reach output 0 is $\frac{a}{b}$.

b) Now we prove that with n splitters, only rational probability $\frac{a}{b}$ with integers $0 \le a \le b \le 2^n$ can be realized. For any flow network with n splitters to generate a probability, it can be described by an absorbing Markov chain with n transient states and 2 absorbing states, whose transition matrix P can be written as

$$P = \begin{pmatrix} p_{11} & \dots & p_{1n} & p_{1(n+1)} & p_{1(n+2)} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ p_{n1} & \dots & p_{nn} & p_{n(n+1)} & p_{n(n+2)} \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$$

where each row consists of two $\frac{1}{2}$ entries and n zeros entries. Then the probability distribution of the network can be written as $e_1(I - Q)^{-1}R$.

In order to prove the result in the theorem, we only need to prove that $(I - Q)^{-1}R$ can be written as $\frac{1}{b}A$ with $b \le 2^n$, where A is an integer matrix.

Let $K = I - Q$, we know that K is invertible if and only $det(K) \neq 0$. In this case, we have

$$(K^{-1})_{ij} = \frac{K_{ji}}{det(K)}$$

where K_{ji} is defined as the determinant of the square matrix of order $(n-1)$ obtained from K by removing the i^{th} row and the j^{th} column multiplied by $(-1)^{i+j}$.

Since each entry of K is chosen from $\{0, \frac{1}{2}, 1\}$, K_{ji} can be written as $\frac{k_{ji}}{2^{n-1}}$ for some integer k_{ji} and $det(K)$ can be written as $\frac{b}{2^n}$ for some integer b. According to the appendix in [13], we have $0 \le det(K) \le 1$, which leads us to $0 < b \le 2^n$ (note that $det(K) \neq 0$).

Then, we have that

$$K^{-1} = \frac{2}{b} \begin{pmatrix} k_{11} & k_{21} & \dots & k_{n1} \\ k_{12} & k_{22} & \dots & k_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ k_{1n} & k_{2n} & \dots & k_{nn} \end{pmatrix}$$

Since each entry of R is also in $\{0, \frac{1}{2}, 1\}$, we know that

$$2R = \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \\ \vdots & \vdots \\ r_{n1} & r_{n2} \end{pmatrix}$$

is an integer matrix.

As a result

$$\begin{aligned} K^{-1}R &= \frac{2R}{b} \begin{pmatrix} k_{11} & k_{21} & \dots & k_{n1} \\ k_{12} & k_{22} & \dots & k_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ k_{1n} & k_{2n} & \dots & k_{nn} \end{pmatrix} \\ &= \frac{A}{b} \end{aligned}$$

where each entry of A is an integer. ■

Using the method in the theorem above, we can realize any arbitrary rational probability with optimal size. For example, if we want to realize probability $\frac{14}{29}$, we can first generate a probability distribution $\{\frac{14}{32}, \frac{15}{32}, \frac{3}{32}\}$, which can be realized by adding one splitter to a network with probability distribution $\{\frac{7}{16}, \frac{6}{16}, \frac{3}{16}\}$... Iteratively, we can get a network to generate probability distribution $\{\frac{14}{32}, \frac{15}{32}, \frac{3}{32}\}$, where only 5 splitters are used. After connecting the last output to the starting point, we can get probability $\frac{14}{29}$. Comparing the results in Theorem 2 with those in Theorem 1, we can see that introducing loops into networks can strongly enhance the expressibility of the network.

IV. Constructions with Bounded Expected Latency

In this section, we consider the expected latency as another important issue. Here, the expected latency indicates the expected number of splitters a token need to pass before reaching one of the outputs. Assume the desired probability is $\frac{a}{b}$ with $2^{n-1} < b < 2^n$ for some integer n. First, we analyze the expected latency of the optimal construction (called scheme A). Then we give two other constructions (scheme B and C) and compare their network sizes and expected latencies with those of scheme A. Table I shows the summary of the results in this section, from which we can see that there is a tradeoff between the upper-bound of the network size and the upper-bound of the expected latency. However, it is not easy to say one of the schemes performs absolutely better than the others. Generally, for practical use, we can try all the schemes and choose the best one among them according to our requirements.

A. Scheme A

For the optimal construction described in the section above, we can get the upper bound of its expected latency.

Theorem 3. *Given a network with probability $\frac{a}{b}$ ($2^{n-1} < b < 2^n$) constructed using the optimal scheme (scheme A), its expected latency ET is bounded by* [1]

$$ET \leq (\frac{3n}{4} + \frac{1}{4})\frac{2^n}{b} < \frac{3n}{2} + \frac{1}{2}$$

Proof: For scheme A, we first prove that the expected latency of the network with distribution $\{\frac{a}{2^n}, \frac{b-a}{2^n}, \frac{2^n-b}{2^n}\}$ is bounded by $\frac{3n}{4} + \frac{1}{4}$.

Let's prove this by induction. When $n = 0$ or $n = 1$, this conclusion is true. Assume when $n = k$, this conclusion is true, we want to show that the conclusion is also true for $n = k+2$. Note that in scheme A, a network with size $k+2$ and three outputs can be constructed by adding two more splitters to a network with size k. Let T_k denote the latency of the network with size k, then

$$E[T_{k+2}] = E[T_k] + p_1 + p_2$$

where p_1 is the probability for a token to reach the first additional splitter and p_2 is the probability for a token to reach the second additional splitter. Assume the distribution of the network with size k is $\{q_1, q_2, q_3\}$, then

$$p_1 + p_2 \leq \max_{i \neq j}(q_i + (\frac{q_i}{2} + q_j)) \leq \frac{3}{2}$$

So the conclusion is true for $n = k + 2$. By induction, we know that it holds for all $n \in \{0, 1, 2, ...\}$.

Secondly, we prove that if the expected latency of the network with distribution $\{q_1, q_2, q_3\}$ is ET', then by connecting its last output to its starting point (feedback), we can get a network such that its expected latency is $ET = \frac{ET'}{q_1+q_2}$. This conclusion can be obtained immediately from

$$ET' = ET + q_3(ET)$$

The theorem holds based on the two conclusions above. ■

	Scheme A	Scheme B	Scheme C
Network size	$\leq n$	$\leq n+3$	$\leq 2(n-1)$
Expected latency	$\leq (\frac{3n}{4} + \frac{1}{4})\frac{2^n}{b}$	$\leq 6\frac{2^n}{b}$	$\leq 3.585\frac{2^n}{b}$

TABLE I
The comparison of different schemes. Here $\frac{2^n}{b} < 2$.

B. Scheme B

In the subsection above, we showed that the expected latency of the optimal construction may increase as the network size increases. Here, we propose another construction (scheme B) with a few more splitters than the optimal one, such that its expected latency is well-bounded by a constant.

Assume a and b are relative prime numbers, and let $c = b - a$. Then a and c can be expressed using binary extension.

$$a = \sum_{i=0}^{n} a_i 2^i, c = b - a = \sum_{i=0}^{n} c_i 2^i$$

[1] By making scheme A more sophisticated, we can reduce the upper bound to $(\frac{n}{2} + \frac{3}{4})\frac{2^n}{b}$.

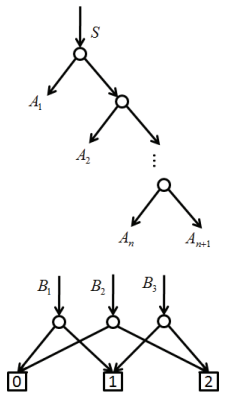


Fig. 3. The network to realize probability $\frac{a}{b}$ when $2^{n-1} < b < 2^n$.

Starting from the structure in Fig. 3, we connect A_i with $1 \leq i \leq n+1$ to one of B_1, B_2, B_3 and output 2, such that the probability distribution of the outputs is $\{\frac{a}{2^{n+1}}, \frac{b-a}{2^{n+1}}, \frac{2^{n+1}-b}{2^{n+1}}\}$. Based on the values of a_i, c_i with $0 \leq i \leq n-1$, we have the following rules for these connections:

1) If $a_i = c_i = 1$, connect A_{n-i} with B_1.
2) If $a_i = 1, c_i = 0$,connect A_{n-i} with B_2.
3) If $a_i = 0, c_i = 1$, connect A_{n-i} with B_3.
4) If $a_i = c_i = 0$, connect A_{n-i} with output 2.
5) Connect A_{n+1} with output 2.

So far, the distribution of the network is $\{\frac{a}{2^{n+1}}, \frac{b-a}{2^{n+1}}, \frac{2^{n+1}-b}{2^{n+1}}\}$. Similar as Theorem 2, by connecting the output 2 to the starting point (feedback), we can get a new network with probability $\frac{a}{b}$. Note that comparing with the optimal scheme, 3 more splitters are used to realize the desired probability. For this network, we can get the upper bound for its expected latency:

Theorem 4. *Given a network with probability $\frac{a}{b}$ ($2^{n-1} < b < 2^n$) constructed using scheme B, its expected latency ET is bounded by*

$$ET \leq 6\frac{2^n}{b} < 12$$

C. Scheme C

In this subsection, we propose another scheme, called scheme C, which is similar to Scheme A. Both Scheme A and Scheme C is try to realize the distribution $\{\frac{a}{2^n}, \frac{b-a}{2^n}, \frac{2^n-b}{2^n}\}$ first. However, the difference is that in Scheme C, this distribution is realized by applying Knuth and Yao's scheme [2]. Generally, Knuth and Yao's scheme can be described as follows [12]. Assume we want to realize the distribution $\{p_1, p_2, ...\}$. Let the binary expansion of the probability p_i be $p_i = \sum_{j \geq 1} p_i^{(j)}$, where $p_i^{(j)} = 2^{-j}$ or 0. Then the atoms of the expansion are $\{p_i^{(j)} : i = 1, 2, ..., m, j \geq 1\}$.

Since $\sum_i p_i = 1$, the sum of the probabilities of these atoms is 1. Now, we allot all the atoms to leaves of a tree such that the depth of atom 2^{-j} is j. We can see that all the depth of these atoms satisfy the Kraft inequality, and hence we can always construct such a tree.

Knuth and Yao showed that the expected number of fair bits required by the procedure above to generate a random variable X with distribution $\{p_1, p_2, ...\}$ lies between $H(X)$ and $H(X) + 2$. Based on this result, we have the following theorem about Scheme C.

Theorem 5. *Given a network with probability $\frac{a}{b}$ ($2^{n-1} < b < 2^n$) constructed using scheme C, its network size is bounded by $2(n-1)$ and its expected latency ET is bounded by*

$$ET \leq (log_2 3 + 2)\frac{2^n}{b} < 7.2$$

Proof: Let's first consider the network with distribution $\{\frac{a}{2^n}, \frac{b-a}{2^n}, \frac{2^n-b}{2^n}\}$, which is constructed using Knuth and Yao's scheme.

1) The network size is bounded by $2(n-1)$. That is because for each j with $2 \leq j \leq n$, there are at most two atoms with value 2^{-j}. If $j = 1$, there are at most one atom with value 2^{-j} (except that the target distribution is $\{\frac{1}{2}, \frac{1}{2}\}$).

2) The expected latency ET' of the network with distribution $\{\frac{a}{2^n}, \frac{b-a}{2^n}, \frac{2^n-b}{2^n}\}$ is bounded by $ET' \leq (log_2 3 + 2)$. That is because that this expected latency ET' is equal to the expected number of fair bits required. According to the result of Knuth and Yao's scheme, it is not hard to get this conclusion.

Now we can get a new network by connecting the last output to the starting point (feedback). We can see that the network size keeps unchanged and the expected latency of the new network is $ET = ET'\frac{2^n}{b}$. ■

REFERENCES

[1] J. von Neumann, "Various techniques used in connection with random digits", Appl. Math. Ser., Notes by G.E. Forstyle, Nat. Bur. Stand., vol. 12, pp. 36-38, 1951.

[2] D. Knuth and A. Yao, "The complexity of nonuniform random number generation", in Algorithms and Complexity, New Directions and Results, J.F. Traub, Ed. New York: Academic, pp. 357-428, 1976.

[3] D. Soloveichik, G. Seelig, and E.Winfree, "DNA as a universal substrate for chemical kinetics", In LNCS 5347, pp. 57-69, 2009.

[4] P. Elias, "The efficient construction of an unbiased random sequences", Ann. Math. Statist., vol. 43, pp. 865-870, 1972.

[5] S. Pae, M.C. Loui, "Optimal random number generation from a biased coin", Proceeding of ACM-SIAM symposium on discrete algorithms, pp. 1079-1088, 2005.

[6] T.S. Han, M. Hoshi, "Interval algorithm for random number generation", IEEE Transaction on Information Theory, vol. 43, No. 2, pp. 599-611, 1997.

[7] J. Abrahams. "Generation of discrete distributions from biased coins", IEEE Transactions on Information Theory. Vol. 42, pp. 1541-1546, 1996.

[8] M. Blum, "Independent unbiased coin flips from a correlated biased source - a finite state markov chain", Combinatorica, Vol. 6, No. 2, pp. 97-108, 1986.

[9] R. Gill, "On a weight distribution problem, with application to the design of stochastic generators", Journal of the ACM (JACM), Vol. 10, pp. 110-121, 1963.

[10] D. Wilhelm and J. Bruck, "Stochastic switching circuit synthesis", IEEE International Symposium on Information Theory, pp. 1388-1392, 2008.

[11] W. Qian and M.D. Riedel, "The synthesis of combinational logic to generate probabilities", International Conference on Computer-Aided Design, 2009.

[12] T.M. Cover, J.A. Thomas. "Elements of information theory", Wiley-Interscience, Second Edition, 2006.

[13] H. Zhou, H. Chen, and J. Bruck. "On the synthesis of stochastic flow networks", Tech. Report at http://paradise.caltech.edu/etr.html.