The Annals of Probability
1984, Vol. 12, No. 1, 212–222

TREE ALGORITHMS FOR UNBIASED COIN TOSSING WITH A BIASED COIN

BY QUENTIN F. STOUT AND BETTE WARREN

State University of New York, Binghamton

We give new algorithms for simulating a flip of an unbiased coin by flipping a coin of unknown bias. We are interested in efficient algorithms, where the expected number of flips is our measure of efficiency. Other authors have represented algorithms as lattices, but by representing them instead as trees we are able to produce an algorithm more efficient than any previously appearing. We also prove a conjecture of Hoeffding and Simons that there is no optimal algorithm. Further, we consider generalizations where the input is a sequence of iid discrete random variables and the output is a uniform random variable with N possible outcomes. In this setting we provide an algorithm significantly superior to those previously published.

1. Introduction. We are concerned with efficiently using flips of a coin of unknown bias to simulate a flip of an unbiased coin. This problem is quite natural in that when given an arbitrary coin one should assume that it has some unknown bias. Von Neumann ([7]) seems to have been the first to propose such an algorithm and Hoeffding and Simons ([4]) the first to analyze the problem in depth. Shortly after [4] appeared, both Dwass ([2]) and Bernard and Letac ([1]) gave equivalent generalizations of one of its algorithms to the case where the sequence of iid Bernouilli random variables associated with the original coin is replaced by a sequence of iid random variables from a nondegenerate discrete distribution, and where the goal is to generate one of N equally likely outcomes.

One common characteristic of all the previous algorithms is the practice of reducing a sequence of flips to a record of the numbers of heads and tails occurring. With this reduction, algorithms can be represented as random walks on the non-negative integer lattice points in the plane, where whether an outcome is determined or a walk continues depends solely on the current lattice position. In contrast, our algorithms use all of the sequence information in determining whether to halt or continue, so we are led to represent algorithms as trees. This change, which enlarges the class of algorithms under consideration, seems to be desirable because it allows us to construct an algorithm which is dramatically more efficient than that of Dwass and Bernard and Letac. We are also able to transform a fairly efficient lattice algorithm of Hoeffding and Simons into a more efficient tree algorithm, and for the class of tree algorithms we prove their conjecture that there is no algorithm which has, for all biases of the coin, the minimum expected number of flips. Further, lattice algorithms are more complicated than they first appear for, in any implementation, at least some sequence-

Received September 1982; revised April 1983.
AMS 1980 *subject classifications.* Primary, 60C05; secondary, 60G40.
Key words and phrases. Random numbers, unbiased coin, biased coin, discrete distribution, binary tree.

order information must be retained. Although they can decide from the current lattice position whether or not a sequence terminates, they must distinguish sequences in order to know which value is determined.

Other authors have considered related problems. While we will assume that flips of our coin are independent, Samuelson ([6]) described a method which can be used when successive flips have a simple Markov dependence. He first simulates independent, but still biased, input values and then applies the von Neumann procedure. Of course, once the Markov dependence has been eliminated a more efficient tree algorithm could be used in place of von Neumann's. Elias ([3]) discussed the generation of sequences of unbiased output values, using the expected number of output values per input value as his measure of efficiency. Finally, Knuth and Yao ([5]) used trees to analyze the reverse of our problem—generating biased output from unbiased input.

2. Preliminaries. We assume that there is a mechanism which produces sequences of iid Bernoulli random variables, each taking the value L with probability $p \in (0, 1)$ and the value R with probability $q = 1 - p$. (We use $\{R, L\}$ rather than $\{H, T\}$ in order to facilitate our description and use of binary trees.) We are interested in algorithms that convert sequences of R's and L's into 0's or 1's (the outcomes) in such a way that $\Pr(0) = \Pr(1) = \frac{1}{2}$ for all values of $p \in (0, 1)$. Our strategy is always to algebraically balance the probability of the sequences assigned to 0 with the probability of those assigned to 1, and to verify separately that the procedure stops with probability 1.

A sequence of L's and R's is *reachable* for an algorithm provided no proper initial segment of the sequence causes the algorithm to determine an outcome (and therefore stop). Those reachable input sequences which determine outcomes are called *termination sequences*, and those that do not are called *continuation sequences.*

Algorithms are associated with binary trees in a natural way. Reachable input sequences are identified with paths from the root of the tree; an L is interpreted as a branch to the left and an R a branch to the right. The final node in the path associated with a termination sequence is a *leaf* (ie, it has no branches emanating from it) and is labeled with the outcome assigned to the sequence. It is called a *termination node.* All other nodes are called *continuation nodes* and have branches to both the left and the right.

Hoeffding and Simons define stopping points and continuation points which are the lattice analogs of our termination and continuation nodes. Since the lattice point (i, j) is associated with every sequence having i R's and j L's, it may be associated with several reachable sequences of probability $p^i q^j$. Lattice algorithms terminate for all of these or for none of them. Tree algorithms are not so constrained, and can therefore exhibit efficiencies unattainable by the lattice algorithms.

We adopt the following conventions.

1. Terminologies of algorithms and trees are used interchangably. Any concept defined for an algorithm carries over to its tree, and vice versa.

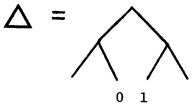


FIG. 1a. *The generator.*

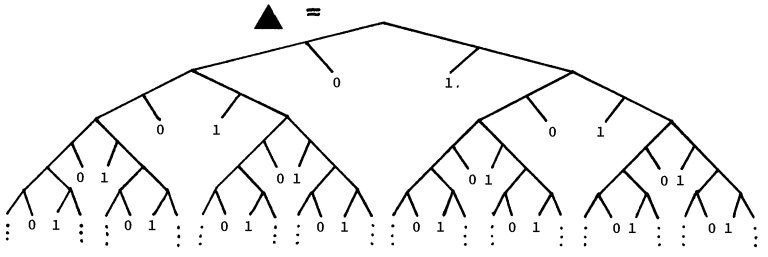
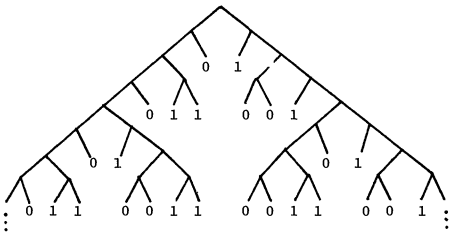


FIG. 1b. *The von Neumann Tree*

2. $L^{(k)}$ denotes a sequence of k L's and $R^{(k)}$ a sequence of k R's.
3. If A is an algorithm and p the probability that an L is produced, then $E(A; p)$ is the expected length of the input sequence required to determine an outcome. When there are $N > 2$ outcomes we denote the expected sequence length by $E(A; p; N)$, and when the input distribution is discrete with probabilities $p_1, p_2 \cdots$ we denote the expectation by $E(A; p_1, p_2, \cdots)$ or $E(A; p_1, p_2, \cdots; N)$
4. If an unshaded figure, eg. Δ, represents a finite tree with some of its leaves unlabeled, then the corresponding shaded figure, eg. ▲, represents the recursive completion of Δ obtained by starting with Δ, replacing each unlabeled leaf with a copy of Δ, replacing each unlabeled leaf of this new tree with a copy of Δ, and so on. (See Figure 1.)
5. The unqualified term *algorithm* will mean a tree algorithm such that Pr(0) = Pr(1) = ½, or, when discussing the case of N equally likely outcomes, the probability of each outcome is $1/N$.

Our notion of when one algorithm is better than another is based on the expected length of an input sequence required to determine an outcome. We say that algorithm A is *E-faster* than algorithm B if for all $p \in (0, 1)$, $E(A; p) \leq E(B; p)$, and for at least one value $p = p_0$, $E(A; p_0) < E(B; p_0)$. This defines a partial order on algorithms, and there are easy examples to show that it is not a total order. We say that A is *E-optimal* if for any algorithm B, either A is E-faster than B or A and B are E-equivalent ($E(A; p) = E(B; p)$ for all p). We say that A is *E-minimal* if no algorithm is E-faster than A.

3. Background. Since input symbols are generated independently, LR and RL are equiprobable. This observation is the basis of the von Neumann algorithm: generate pairs of input values; if the pair is RL then the outcome is 1, if it is LR



Q_2

FIG. 2.

the outcome is 0, and if it is neither the process is repeated. The first few levels of the von Neumann tree, which we represent by ▲, are shown in Figure 1. It is easily verified that $E(\blacktriangle; p) = 1/pq$.

Even some very primitive symmetry is ignored by the von Neumann algorithm. For example, $L^{(2)}R^{(2)}$ and $R^{(2)}L^{(2)}$ are equiprobable and could have been designated termination sequences, one producing a 0 and the other a 1. In fact, any pair $SR^{(k)}L^{(k)}$ and $SL^{(k)}R^{(k)}$, $k > 0$, of reachable sequences with common initial segment S are equiprobable and could terminate. Hoeffding and Simons describe an algorithm Q_2 (see Figure 2) that terminates at precisely such sequences. It maintains the probability balance between 0 and 1 level by level, making assignments, half 0 and half 1, only when there are an even number of sequences of the same probability.

Any algorithm which maintains its probability balance in this manner is called an *even procedure*, and Hoeffding and Simons proved that stopping sets of even procedures can contain only points (i, j) for which the binomial coefficient $\binom{i+j}{j}$ is divisible by 2. Dwass ([12]) showed that Q_2 is the optimal even procedure in the strong sense that no proper initial segment of a termination sequence for Q_2 reaches a stopping point of any other even procedure. This also shows that among even procedures Q_2 is E-optimal.

Q_2 assigns a 0 to a termination sequence of length n if there are an even number of R's in its first $n - 1$ elements, and a 1 if there are an odd number. Hoeffding and Simons noted that the outcome assigned to a termination node at level n is determined by its parent at level $n - 1$, so if both children of a level $n - 1$ continuation node are termination nodes, then they each produce the same outcome. In this situation an E-faster tree can be constructed by replacing the parent at level $n - 1$ with a termination node whose outcome is the outcome previously assigned to the children. The algorithm obtained from Q_2 by making all such substitutions is called Q_3, and is illustrated in Figure 3.

Hoeffding and Simons constructed an algorithm Q_4 for which $E(Q_4; p) < E(Q_3; p)$ for sufficiently small p, showing that Q_3 is not E-optimal. However, since $E(Q_4; \frac{1}{2}) > E(Q_3; \frac{1}{2})$, this does not rule out the possibility that Q_3 is E-minimal. We prove in the next section that it is not.

In addition to our E-based partial order, Hoeffding and Simons considered another partial order on algorithms. Algorithm A is *continuation-better* than

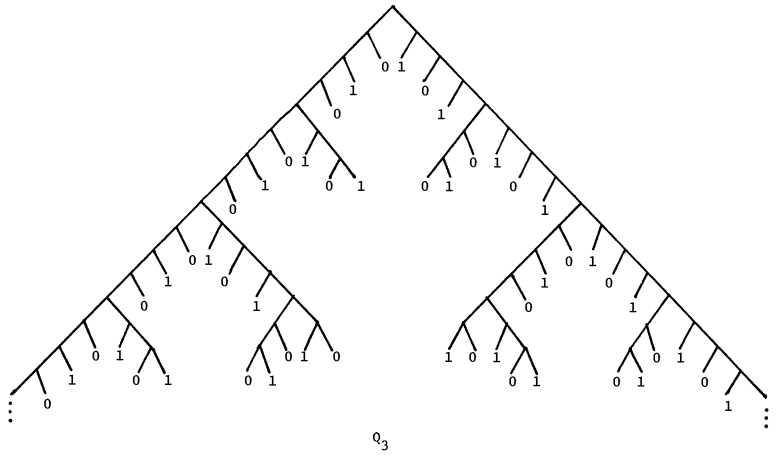


FIG. 3.

FIG. 4.

algorithm B if the continuation sequences of A are a proper subset of the continuation sequences of B. A minimal algorithm under this partial order is called *weakly admissible.* Any E-minimal algorithm clearly must be weakly admissible, but we are uncertain of the converse. Hoeffding and Simons showed that every lattice algorithm is continuation-comparable to a weakly admissible algorithm, and their proof easily extends to the class of tree algorithms. Despite this, neither they nor we have shown that any particular algorithm is weakly admissible.

4. Improving Q_3.

THEOREM 1. *There is a tree T such that $E(T; p) \leq E(Q_3; p)$ for all $0 < p < 1$, with equality holding only at $p = \frac{1}{2}$.*

PROOF. Consider both Figures 3 and 4. Q_3 has a tree of type $T_1(0, 1)$ attached to the node $L^{(8)}R^{(4)}L$, and one of type $T_2(1, 0)$ attached at $R^{(8)}L^{(4)}R$. Define a new tree T by replacing this $T_1(0, 1)$ with a $T_2(0, 1)$ and the $T_2(0, 1)$ with a $T_1(0, 1)$. Neither substitution destroys the $0 - 1$ probability balance (although the "all or none" lattice property no longer holds, so T does not correspond to a lattice algorithm). The change in expectation is

$$p^9q^4(q - p) + p^4q^9(p - q) = p^4q^4(q - p)(p^5 - q^5).$$

This expression is negative except at $p = \frac{1}{2}$ where it is zero. □

COROLLARY. *Q_3 is not E-minimal.* □

The expectation can be reduced further by substituting $T_2(A, B)$ for $T_1(A, B)$ and $T_1(B, A)$ for $T_2(B, A)$ everywhere a tree of the form $T_1(A, B)$ is attached to a node of probability $p^\alpha q^\beta$ and $T_2(B, A)$ is attached to a node of probability $p^\beta q^\alpha$ with $\alpha > \beta$. Such situations occur repeatedly in Q_3. For all $p \neq \frac{1}{2}$, one of these substitutions increases the expectation, but the other decreases it by a greater amount. If we knew the direction of the bias (the sign of $p - q$) we could make only the advantageous substitutions. In fact, we could substitute at every occurrence of the inefficient T_i, regardless of the probability of the node to which it is attached. (In the portion of Q_3 illustrated in Figure 3, substitutions of $T_1(1, 0)$ for $T_2(1, 0)$ would be made at $L^{(4)}R$, $L^{(12)}R$, and $R^{(8)}L^{(4)}R$ if $p < q$, and if $p > q$, $T_2(0, 1)$ would be substituted for $T_1(0, 1)$ at $R^{(4)}L$, $R^{(12)}L$ and $L^{(8)}R^{(4)}L$.)

5. *E*-optimality. E-minimality is of interest only if there are no E-optimal algorithms. Hoeffding and Simons conjectured that this was the case among lattice algorithm and we shall prove it among tree algorithms. First we need to establish some lower bounds on the E function. No sequence of the form $L^{(n)}$ can ever be a termination sequence, for when $p = (0.6)^{1/n}$ then $L^{(n)}$ has probability .6. Similarly all sequences of the form $R^{(n)}$ are continuation sequences. Hoeffding and Simons showed that any algorithm must have additional continuation sequences. We use $\mathbb{Z}^+$ to denote $\{1, 2, \cdots\}$.

THEOREM 2. (Hoeffding and Simons [4, Theorem 2]). *There is no algorithm which has only $S = \{R^{(n)}, L^{(n)}: n \in \mathbb{Z}^+\}$ as its set of continuation sequences.* □

COROLLARY. (Hoeffding and Simons). $E(A; p) > 1/pq - 1$ for any algorithm A and probability p. □

LEMMA 1. *For any algorithm A, either $E(A; p) \geq 1/pq - 1 + q + o(q)$ as $q \to 0$, or $E(A; p) \geq 1/pq - 1 + p + o(p)$ as $p \to 0$.*

PROOF. Let $S = \{R^{(n)}, L^{(n)}: n \in \mathbb{Z}^+\}$. By Theorem 2 there is an n such that A's continuation sequences contain either $S \cup \{L^{(n)}R\}$ or $S \cup \{R^{(n)}L\}$. In the first case

$$E(A; p) \geq \left(\frac{1}{pq} - 1\right) + p^n q = \left(\frac{1}{pq} - 1\right) + q + o(q) \quad \text{as } q \to 0,$$

while in the second case

$$E(A; p) \geq \left(\frac{1}{pq} - 1\right) + q^n p = \left(\frac{1}{pq} - 1\right) + p + o(p) \quad \text{as } p \to 0. \quad \square$$

THEOREM 3. *There is no E-optimal algorithm.*

PROOF. Suppose there were an algorithm V such that $E(V; p) \leq 1/pq - 1 + O(q^2)$ as $q \to 0$. If we interpret V as a tree and take its mirror image we will have

a tree V^* corresponding to an algorithm for which $E(V^*; p) \leq 1/pq - 1 + O(p^2)$ as $p \to 0$. Let U be any algorithm. By Lemma 1 we know that either $E(U; p) \geq 1/pq - 1 + q + o(q)$ or $E(U; p) \geq 1/pq - 1 + p + o(p)$. In the former case $E(U; p) > E(V; p)$ for q sufficiently small, and in the latter case $E(U; p) > E(V^*; p)$ for p sufficiently small. In either case, U is not E-optimal. Therefore our theorem is proved once we have shown the existence of V, which is accomplished by the following lemma. □

The algorithm V mentioned in the theorem is presented as a tree in Figure 6. It is constructed from the pieces in Figure 5.

LEMMA 2. *In the tree* V, $\Pr(0) = \Pr(1) = \frac{1}{2}$, *and*

$$E(V; p) \leq \frac{1}{pq} - 1 + O(q^2) \text{ as } q \to 0.$$

PROOF. We compute $E(V; p)$ by summing the probabilities of the continuation nodes. Direct calculations from Figures 5 and 6 show that

$$\begin{aligned} E(V; p) &= (\textstyle\sum_{i=0}^{\infty} p^i + \sum_{i=0}^{\infty} q^i - 1) + q^4[E(\blacktriangle; p) - \sum_{i=0}^{\infty} q^i] \\ &\quad + p^4q^3E(\blacktriangle; p) \\ &\quad + pq^3(1 + p + p^2)[1 + qE(\blacktriangledown; p)] \\ &= \left(\frac{1}{pq} - 1\right) + q^4\left(\frac{1}{pq} - \frac{1}{p}\right) + \frac{p^4q^3}{pq} \\ &\quad + pq^3(1 + p + p^2)\left[1 + q\left(\frac{1}{q} + p^4\,\frac{(1 - p^4)}{(1 - p^8)} \cdot \frac{1}{pq}\right)\right] \\ &< \left(\frac{1}{pq} - 1\right) + q^3 + q^2 + 9q^3 \\ &= \left(\frac{1}{pq} - 1\right) + O(q^2) \quad \text{as } q \to 0. \end{aligned}$$

This also proves that the algorithm terminates with probability 1, so we can prove $\Pr(0) = \Pr(1) = \frac{1}{2}$ by showing $\Pr(0) = \Pr(1)$. The values of $\Pr(0) - \Pr(1)$ for the various pieces are given in Figure 5. From these we calculate that for V,

$$\begin{aligned} \Pr(0) - \Pr(1) &= (pq + p^3q + pq^2 - pq - p^2q) \\ &\quad + \frac{p^4}{1 - p^8}\,[-q^2(1 - p^2)(1 - p^4)] \\ &\quad + \frac{pq^4}{1 - p^8}\,(1 + p + p^2)(-(1 - p^4)) \\ &= pq^3\left(1 - \frac{(1 - p^4)}{1 - p^8}\,(1 + p^4)\right) = 0. \quad \square \end{aligned}$$

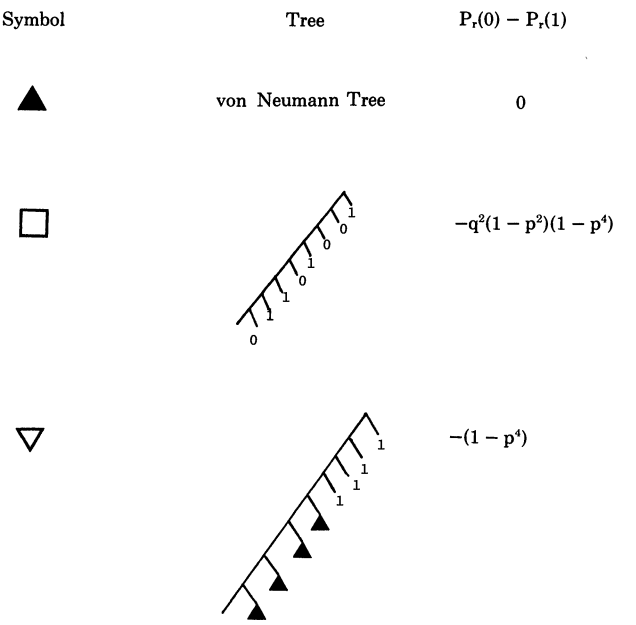


FIG. 5.

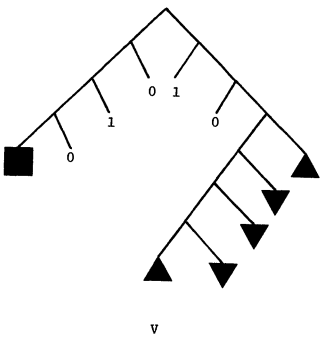


FIG. 6.

6. A generalization of Q_2. Let $M(i, j)$ be the number of Q_2-reachable sequences of probability p^iq^j. The termination rule for Q_2 can be described in two ways:

(1) assign outcomes to all sequences of probability p^iq^j if $M(i, j)$ is even.

(2) assign outcomes to all sequences of probability p^iq^j if $M(i, j)$ is even, and to all but one if $M(i, j)$ is odd.

The two rules are equivalent because $M(i, j)$ is always 0, 1, or 2 (Q_2 has $\binom{n}{r} \bmod 2$ continuation strings of probability p^rq^{n-r} for all integers $0 \leq r \leq n$).

Suppose we have input values drawn from a discrete (possibly infinite) distribution with non-zero probabilities $p_1, p_2, \cdots$, and a set of outcomes of cardinality $N < \infty$. The algorithm suggested by Dwass ([2]), and Bernard and Letac ([1]), hereafter referred to as DBL, generalizes rule 1, terminating all of the reachable

sequences of probability $\prod p_i^{n_i}$ if and only if there are a multiple of N of them, or equivalently if and only if N divides the multinomial coefficient $\binom{\sum n_i}{n_1, n_2, \ldots}$. We propose an algorithm, hereafter referred to as SW, generalizing rule 2: terminate all but $\binom{\sum n_i}{n_1, n_2, \ldots} \bmod N$ of the sequences of probability $\pi p_i^{n_i}$. The continuation sequences of SW are a proper subset of the continuation sequences of DBL.

SW and DBL are actually families of algorithms. In order to specify an algorithm of either type we need a rule for distributing outcomes among the termination sequences. Further, for SW we need a rule to determine which reachable sequences of a given probability terminate when there are more than N but not a multiple of N of them. There are many simple rules available (for example, a "first come first served" rule based on lexical ordering of sequences). However, the expected input sequence lengths are independent of these rules, depending only on the p_i's, N and the number of continuation sequences of each probability. When we make assertions about SW or DBL we mean that they are true of every algorithm of that type.

We would like to determine the behavior of $E(\text{DBL}; p_1, p_2, \cdots; N)$ for fixed $p_1, p_2, \cdots$ as $N \to \infty$. Unfortunately we are unable to do so. However, we determine enough of its behavior to show that SW is vastly superior to DBL.

PROPOSITION 1. *For fixed $p_1, p_2, \cdots$, there are infinitely many N for which $E(\text{DBL}; p_1, p_2, \cdots, N) > N$.*

PROOF. If N is prime then it cannot divide any multinomial coefficient $\binom{n}{n_1, n_2, \cdots}$ with $n < N$, so all DBL termination sequences have length $\geq N$. Since some sequences of length N do not terminate, the expected input sequence in length exceeds N. □

We will show that for fixed $p_1, p_2, \cdots$, SW has an expected sequence length of the form $O(\log N)$, proving it first in the case of binary inputs.

THEOREM 4. *For fixed p, $E(SW; p; N)$ has order $\log(N)$.*

PROOF. Assume that $p > \frac{1}{2}$, the cases $p < \frac{1}{2}$ and $p = \frac{1}{2}$ being similar. Let L be the least integer which is no less than $-\log_p N$.

$$
\begin{aligned}
E(\text{SW}; p; N) &= \sum_{n=0}^{\infty} \sum_{i+j=n} \left[\binom{n}{i} \bmod N \right] p^i q^j \\
&\leq \sum_{n=0}^{L} \sum_{i+j=n} \binom{n}{i} p^i q^j + \sum_{n=L+1}^{\infty} \sum_{i+j=n} N p^i q^j \\
&< L + 1 + N \sum_{n=L+1}^{\infty} p^n \left(\sum_{i=0}^{\infty} (q/p)^i \right) \\
&= L + 1 + N \frac{p^{L+2}}{q(p-q)} < L + 1 + \frac{p^2}{q(p-q)} \\
&= O(\log N).
\end{aligned}
$$

On the other hand, $\binom{n}{i} \leq N$ whenever $n < \log_2 N$, so

$$E(SW; p; N) > \log_2 N.$$

Therefore $E(SW; p; N)$ is of exact order log (N). □

Notice that $E(A; p; N) > \log_2 N$ for all algorithms A and probabilities p. To see this, suppose $k < \log_2 N$. If p were ½ then each node at level k would have probability $2^{-k} > 1/N$, so none could be a termination node. Therefore $E(A; p; N) \geq \log_2 N$, and since $L^{(k)}$ can never be a termination node we have $E(A; p; N) > \log_2 N$.

We can use Theorem 4 to give information about SW algorithms with nonbinary input. Let $p_1, p_2, \cdots$ be the (non-zero) probabilities associated with an input process and let $I_2, I_2, \cdots$ be any partition of its index set. Define a restricted input process with probabilities $q_i = \sum_{j \in I_i} p_j$ by identifying values whose indices fall into the same I_i. It is easy to show that if SW_F is an SW algorithm based on the full (original) input process and SW_R is based on the restricted process, then for all $n \in \mathbb{Z}^+$, the probability that SW_F continues beyond level n is no greater than the probability that SW_R continues. In particular, if I is any proper subset of the index set and $p = \sum_{i \in I} p_i$, then $E(\mathrm{SW}; p_1, p_2, \cdots; N) \leq E(\mathrm{SW}; p; N)$. Combining this observation with Theorem 4 gives

THEOREM 5. *For fixed* $p_1, p_2, \cdots$, $E(\mathrm{SW}; p_1, p_2, \cdots; N) = O(\log N)$. □

Dwass showed that DBL is continuation-optimal among all algorithms making counter-balancing assignments to sequences of equal probability and terminating either all or none of the sequences of a given probability. SW has the first property; it is the lattice-inspired "all or nothing" characteristic of DBL that makes it so much less efficient.

7. Remarks and questions.

(1) $E(Q_4; p) = 1/pq - 1 + q + o(q)$ as $q \to 0$, and $E(V; p) = 1/pq - 1 + q^2 + o(q^2)$. What is the largest k for which a $1/pq - 1 + O(q^k)$ algorithm exists? Hoeffding and Simons' proof of Theorem 2 showed that there is an upper bound on k.

(2) Can Q_3 be improved at $p = ½$? That is, is $E(Q_3; ½) = \min\{E(A; ½): A \text{ an algorithm}\}$?

(3) Is every tree algorithm E-comparable to an E-minimal algorithm? Hoeffding and Simons showed that every lattice algorithm is continuation-comparable to a continuation-minimal algorithm, but their Zorn's Lemma based argument does not extend to the more complex E-ordering.

(4) What is the asymptotic behavior of $E(\mathrm{DBL}; p_1, \cdots; N)$ or at least what is it for other special cases? Bernard and Letac showed that for primes the expectation grows linearly. Is it possible that for some infinite sequence (such as products of successive pairs of primes) sublinear asymptotic behavior can

be achieved? Also, no one has yet shown that the expectation is never worse than linear.

REFERENCES

[1] BERNARD, JACQUES and LETAC, GÉRARD (1973). Construction d'evenements equiprobables et coefficients multinomiaux modulo p^n. *Ill. J. Math.* **17** 317–332

[2] DWASS, MEYER (1972). Unbiased coin tossing with discrete random variables. *Ann. Math. Statist.* **43** 860–864.

[3] ELIAS, PETER (1972). The efficient construction of an unbiased random sequence. *Ann. Math. Statist.* **43.** 865–870.

[4] HOEFFDING, WASSILY and SIMONS, GORDON (1970). Unbiased coin tossing with a biased coin. *Ann. Math. Statist.* **41** 341–352.

[5] KNUTH, DONALD E. and YAO, ANDREW C. (1976). The complexity of nonuniform random number generation. In *Algorithms and Complexity, New Directions and Results*, ed. J. F. Traub. Academic, New York. 357–428.

[6] SAMUELSON, PAUL A. (1968). Constructing an unbiased random sequence. *J. Amer. Statist. Assoc.* **63** 1526–1527.

[7] VON NEUMANN, JOHN (1951). Various techniques used in connection with random digits. Notes by G. E. Forsythe. National Bureau of Standards, *Applied Math Series* **12** 36–38. Reprinted in von Neumann's *Collected Works* **5** (Pergamon Press, 1963), 768–770.

MATHEMATICAL SCIENCES
STATE UNIVERSITY OF NEW YORK
BINGHAMTON, NEW YORK 13901