Interval Algorithm for Random Number Generation

Mamoru Hoshi
Graduate School of Information Systems,
The University of Electro-Communications,
Tokyo, Japan

June 12, 1995

[summary by Vincent Dumas]

1. Introduction

This talk is based on a joint paper with Te Sun Han [1]. It presents an "interval algorithm" that solves the problem of generating a random number X with distribution $\mathbf{q} = (q_1, q_2, \ldots, q_N)$ (i.e. $\Pr[X = k] = q_k$, $1 \le k \le N$) from independent identically distributed tosses with an M-coin of distribution $\mathbf{p} = (p_1, p_2, \ldots, p_M)$. This problem was set by Roche [2] (variants of this problem were studied by von Neumann, Elias, Knuth and Yao). The efficiency of the algorithm is measured by L^*, which is the expected number of tosses required to generate X. Roche proved that the optimal algorithm should satisfy:

$$\frac{H(\mathbf{q})}{H(\mathbf{p})} \le L^* \le \frac{H(\mathbf{q}) + f(\mathbf{p})}{H(\mathbf{p})},$$

where H is the entropy function (see Appendix) and

$$f(\mathbf{p}) = \ln(e/p_{\min}), \qquad \text{where} \qquad p_{\min} = \min_{1 \le j \le M} p_j.$$

The upper bound is satisfied by a probabilistic algorithm.

Han and Hoshi propose an "interval algorithm" that satisfies the upper bound with

$$f(\mathbf{p}) = \ln[2(M-1)] + \frac{h(p_{\max})}{1 - p_{\max}}, \qquad \text{where} \qquad p_{\max} = \max_{1 \le j \le M} p_j,$$

with $h(p) = -p \ln p - (1-p)\ln(1-p)$. No choice of function f seems to be essentially better than any other one. The assumed superiority of the interval algorithm is that it is *deterministic* and *easy to implement.*

2. Interval algorithm

Let $\mathbf{p}$ be the original distribution. Let us fix a partition of $[0,1)$ according to $\mathbf{p}$, that is a sequence

$$\alpha_0 = 0 < \alpha_1 < \cdots < \alpha_M = 1,$$

such that $\alpha_j - \alpha_{j-1} = p_j$ for all j. Now any interval $[a,b)$ may be partitioned into the subintervals $I_j([a,b))$, $1 \le j \le M$, with

$$I_j([a,b)) = [a + (b-a)\alpha_{j-1}, a + (b-a)\alpha_j).$$

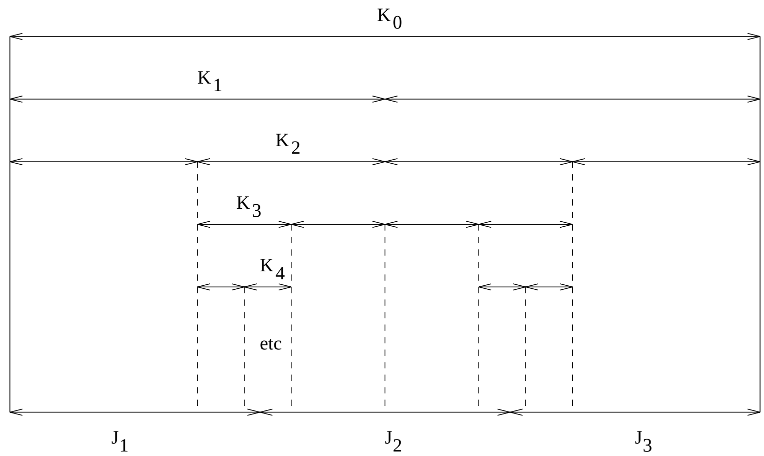

FIGURE 1. Example of sequence (K_n) ($\mathbf{p} = (1/2, 1/2)$, $\mathbf{q} = (1/3, 1/3, 1/3)$).

Let $\mathbf{q}$ be the distribution we want to generate. Fix a partition

$$\beta_0 = 0 < \beta_1 < \cdots < \beta_N = 1$$

of $[0, 1)$ according to $\mathbf{q}$ ($\beta_k - \beta_{k-1} = q_k$), and set $J_k = [\beta_{k-1}, \beta_k)$.

The interval algorithm is defined as follows:

(1) set $n = 0$ and $K_0 = [0, 1)$;
(2) if $K_n \subset J_k$ for some k, then stop the algorithm and set $X = k$;
(3) else flip the M-coin (with probability distribution $\mathbf{p}$). The result is a number $M_n \in \{1, \ldots, M\}$. Set $K_{n+1} = I_{M_n}(K_n)$ and go to (2).

This procedure is illustrated in Figure 1.

With probability one this algorithm terminates in finite time, and generates a random number X, which is a deterministic function of $Y = K_\infty$. Let $\mathcal{Y}$ be the set of all possible values of Y. By construction, $\mathcal{Y}$ is a partition of $[0, 1)$, and any $y \in \mathcal{Y}$ may be obtained with probability $|y|$ (where $|y|$ denotes the length of interval y). In consequence, we fall in J_k with probability $|J_k| = q_k$, which means that X has distribution $\mathbf{q}$ as expected.

Now denote by L the number of tosses necessary to get X. From basic results on entropy in tree algorithms, we get that

$$L^* = \mathrm{E}(L) = \frac{H(Y)}{H(\mathbf{p})}.$$

Moreover, since X is a (deterministic) function of Y, then $H(Y) \geq H(X) = H(\mathbf{q})$, which yields

$$L^* \geq \frac{H(\mathbf{q})}{H(\mathbf{p})}.$$

3. Upper bound

In order to get an upper bound on $H(Y)$ (and then on L^*), the authors introduce a new variable W, such that

(1) W is a function of Y;
(2) W has $2(M - 1)$ possible values;

(3) conditionally on (W, X) being equal to some (w, k), we have

$$Y \succ \mathrm{Geom}(p_{\max}),$$

where $\mathrm{Geom}(p)$ denotes the geometric distribution of parameter p:

$$\Pr[\mathrm{Geom}(p) = i] = (1-p)p^i.$$

Then we will get that: $H(Y) = H(Y, W, X) = H(X) + H(W|X) + H(Y|(W, X))$, with

$$H(X) = H(\mathbf{q}), \quad H(W|X) \leq \ln[2(M-1)], \quad H(Y|(W, X)) \leq H(\mathrm{Geom}(p_{\max})) = \frac{h(p_{\max})}{1 - p_{\max}},$$

which yields the announced bound.

In order to define W, set $X = k$ and consider the possible values of Y, that is all the intervals $y \in \mathcal{Y}$ such that $y \subset J_k$. We may organize them as follows. There is a unique sequence of tosses (M_n) such that, for all n, $K_n = [\gamma, \delta)$ with $\gamma \leq \beta_{k-1}$ and $\delta > \beta_{k-1}$ (resp. with. $\gamma < \beta_k$ and $\delta \geq \beta_k$): this is the *upward* sequence (resp. the *downward* sequence) associated to J_k; it is finite only if $\gamma = \beta_{k-1}$ (resp. if $\delta = \beta_k$) for some K_n. Now any possible value of Y corresponds to a unique, finite sequence of tosses $(M_n(y))_{0 \leq n \leq n(y)}$, and we can check that

$$M_n(y) = M_n, \qquad 0 \leq n < n(y)$$

is valid for (M_n) equal to either the upward sequence or the downward sequence.

For a given y, set $\mathrm{sign}(y) =$ upward (resp. $\mathrm{sign}(y) =$ downward) if y derives from an upward sequence (resp. a downward sequence), and $M(y) = M_{n(y)}(y)$ (the value of the last toss that stops the algorithm at y). One can check that if $\mathrm{sign}(y) =$ upward (resp. if $\mathrm{sign}(y) =$ downward), then $M(y)$ cannot be equal to 1 (resp. $M(y)$ cannot be equal to M); in consequence, there are only $2(M-1)$ possible values for $(\mathrm{sign}(y), M(y))$. We may now define the new random variable $W = (\mathrm{sign}(Y), M(Y))$ which obviously satisfies properties (1) and (2). Moreover, if $X = k$ and $W = (s, m)$, then all the possible values of Y derive from the same upward or downward sequence (M_n), and they may be ordered in a sequence (y_l) such that $n(y_l)$ is strictly increasing. In consequence, the interval algorithm yields y_l with probability

$$p(y_l) = \left(\prod_{n=0}^{n(y_l)-1} p_{M_n} \right) p_m,$$

which implies that $p(y_l) \leq p_{\max} p(y_{l-1})$: property (3) may be deduced from this inequality.

4. Conclusion

The interval algorithm may be adapted to generate the first n terms of a finite state space Markov chain; the average cost L^*/n is then asymptotically optimal. Independent identically distributed tosses with an M-coin may also be replaced by a Markov chain.

Appendix: basic properties of the entropy function

The entropy of a distribution $\mathbf{a} = (a_i)_{i \in I}$ (where I is countable) is defined by:

$$H(\mathbf{a}) = -\sum_{i \in I} a_i \ln a_i.$$

The notation $H(A)$ is also used if A is a random variable with distribution $\mathbf{a}$. If $\mathrm{Card}(I) = P$, then $H(A) = H(\mathbf{a}) \leq \ln P$.

Since a pair of random variables (A, B) is a random variable, one may also consider the entropy $H(A, B)$. If $B = f(A)$ (where f is deterministic), then $H(A) \geq H(B)$ (notice that it implies $H(A, B) = H(A)$).

In the general case, denote by $A/B = b$ the distribution of A conditioned on $B = b$ (it is assumed that $\Pr(B = b) > 0$). Set $f(b) = H(A/B = b)$. Then one may define

$$H(A|B) = \mathrm{E}[f(B)],$$

which satisfies: $H(A|B) = H(A, B) - H(B)$.

Now, consider two distributions $\mathbf{a} = (a_i)_{i\geq 1}$ and $\mathbf{b} = (b_i)_{i\geq 1}$ ordered in decreasing probabilities ($a_i \geq a_{i+1}$ and $b_i \geq b_{i+1}$, for all i). The partial ordering $\mathbf{a} \succ \mathbf{b}$ is defined by:

$$\sum_{i=1}^{j} a_i \geq \sum_{i=1}^{j} b_i, \qquad \forall j \geq 1.$$

If $\mathbf{a} \succ \mathbf{b}$, then $H(\mathbf{a}) \leq H(\mathbf{b})$ (this is indeed valid for all the concave, symmetric functions).

Bibliography

[1] Han (Te Su) and Hoshi (Mamoru). – Interval algorithm for random number generation. – May 1995. Preprint.

[2] Roche (J. R.). – *Efficient generation of random variables from biased coins.* – Bell Technical Report n° 20878, AT&T Laboratories, 1992.