An Efficient Method for Generating Discrete Random Variables With General Distributions

ALASTAIR J. WALKER
University of Witwatersrand, South Africa

The fast generation of discrete random variables with arbitrary frequency distributions is discussed. The proposed method is related to rejection techniques but differs from them in that all samples comprising the input data contribute to the samples in the target distribution. The software implementation of the method requires at most two memory references and a comparison. The method features good accuracy and modest storage requirements. It is particularly useful in small computers with limited memory capacity.

Key Words and Phrases· random number generation, probability, arbitrary distributions, statistical tests
CR Categories: 3.24, 5.5

INTRODUCTION

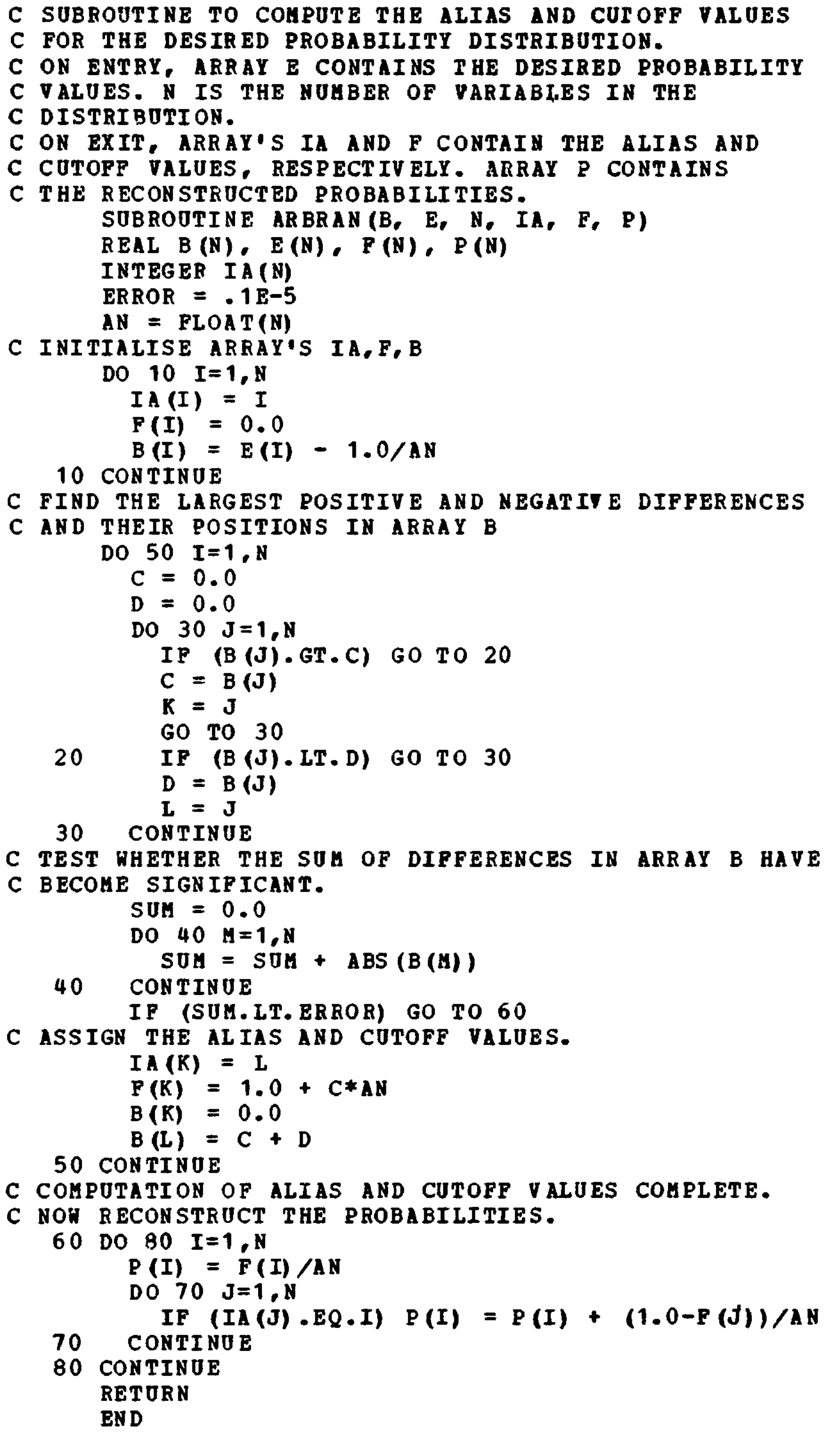
Commonly used criteria [2, 3] for judging the performance of random number generators include speed, precision, memory requirements, generality of the technique, and ease of implementation. The method proposed by Marsaglia [1] has all these attributes but must be used with care in a computer with limited memory capacity since the number of memory locations required for the storage of constants appears to be an order of magnitude greater than the number of variables to be generated. In comparison, the number of memory locations required for the storage of constants by the method proposed in this contribution is only twice the number of variables used. There is no restriction on the other desirable features given above.

These principles have already been used to advantage in the design and construction of a high-speed digital hardware-implemented pseudorandom number generator with an arbitrary discrete frequency distribution [5] and as a generator of uniformly distributed random variables with floating point representation [4].

METHOD

The new method is related to rejection techniques but differs from them in that all numbers generated are used. A number is either accepted or replaced with an "alias" number.

Author's address: Department of Electrical Engineering, University of Witwatersrand, 1 Jan Smuts Ave., Johannesburg 2001, South Africa.



```
C SUBROUTINE TO COMPUTE THE ALIAS AND CUIOFF VALUES
C FOR THE DESIRED PROBABILITY DISTRIBUTION.
C ON ENTRY, ARRAY E CONTAINS THE DESIRED PROBABILITY
C VALUES. N IS THE NUMBER OF VARIABLES IN THE
C DISTRIBUTION.
C ON EXIT, ARRAY'S IA AND F CONTAIN THE ALIAS AND
C CUTOFF VALUES, RESPECTIVELY. ARRAY P CONTAINS
C THE RECONSTRUCTED PROBABILITIES.
      SUBROUTINE ARBRAN(B, E, N, IA, F, P)
      REAL B(N), E(N), F(N), P(N)
      INTEGEP IA(N)
      ERROR = .1E-5
      AN = FLOAT(N)
C INITIALISE ARRAY'S IA,F,B
      DO 10 I=1,N
        IA(I) = I
        F(I) = 0.0
        B(I) = E(I) - 1.0/AN
   10 CONTINUE
C FIND THE LARGEST POSITIVE AND NEGATIVE DIFFERENCES
C AND THEIR POSITIONS IN ARRAY B
      DO 50 I=1,N
        C = 0.0
        D = 0.0
        DO 30 J=1,N
          IF (B(J).GT.C) GO TO 20
          C = B(J)
          K = J
          GO TO 30
   20     IF (B(J).LT.D) GO TO 30
          D = B(J)
          L = J
   30   CONTINUE
C TEST WHETHER THE SUM OF DIFFERENCES IN ARRAY B HAVE
C BECOME SIGNIFICANT.
        SUM = 0.0
        DO 40 M=1,N
          SUM = SUM + ABS(B(M))
   40   CONTINUE
        IF (SUM.LT.ERROR) GO TO 60
C ASSIGN THE ALIAS AND CUTOFF VALUES.
        IA(K) = L
        F(K) = 1.0 + C*AN
        B(K) = 0.0
        B(L) = C + D
   50 CONTINUE
C COMPUTATION OF ALIAS AND CUTOFF VALUES COMPLETE.
C NOW RECONSTRUCT THE PROBABILITIES.
   60 DO 80 I=1,N
        P(I) = F(I)/AN
        DO 70 J=1,N
          IF (IA(J).EQ.I) P(I) = P(I) + (1.0-F(J))/AN
   70   CONTINUE
   80 CONTINUE
      RETURN
      END
```

Fig. 1. Subroutine for finding the alias and cutoff values for a given probability distribution

```
C A SUBROUTINE TO GENERATE INTEGER RANDOM
C VARIABLES WITH PRESCRIBED PROBABILITY
C DISTRIBUTION.
C ON ENTRY, UA AND UB ARE UNCORRELATED RANDOM
C VARIABLES UNIFORMLY DISTRIBUTED OVER (0,1)
C ARRAY'S IA AND F CONTAIN THE DESIRED ALIAS
C AND CUTOFF VALUES, RESPECTIVELY. N IS THE
C NUMBER OF VARIABLES IN THE DISTRIBUTION.
C ON EXIT, IX IS THE RETURNED RANDOM VARIABLE.
      SUBROUTINE GETONE(UA, UB, IA, F, N, IX)
      REAL F(N)
      INTEGER IA(N)
      AN = FLOAT(N)
C CONVERT UA TO AN INTEGER VARIABLE
      IX = INT(UA*AN) + 1
C COMPARE WITH THE SELECTED CUTOFF.
      IF (UB.GT.F(IX)) IX = IA(IX)
      RETURN
      END
```

Fig. 2. Subroutine for generating a sample IX with the required frequency distribution

It is required to produce a random or pseudorandom integer Y whose probability distribution is Pr $(Y = j) = p_j$, $j = 1$ to n. We have available a random integer X which is uniformly distributed over the range 1 to m, i.e. Pr $(X = j) = q = m^{-1}$. The method requires that $m = n$. It may be convenient to have $m > n$, i.e. to have the range of X larger than that of Y, and in this case we redefine the range of Y to be 1 to m by setting $p_{n+1} = \ldots p_m = 0$.

The method consists of setting

$$Y = \begin{cases} X \text{ with probability } F(X) \\ A(X) \text{ with probability } 1 - F(X) \end{cases}$$

where $A(X)$ is an alias. The functions $A(X)$ and $F(X)$ are chosen according to the algorithm shown in Figure 1 and ensure that Pr $(Y = j) = p_j$, $j = 1$ to n.

After the desired probability values have been entered into array E, the differences in magnitude between the desired distribution and the uniform distribution are found and stored in array B. This array is searched for the largest negative and positive differences, C and D, respectively, and their positions, K and L, respectively. $B(K)$ is then set to zero and $B(L)$ is assigned the sum of C and D. $IA(K)$ is assigned the value of L and $F(K)$ is the normalized value of C added to unity.

A confirmation of the method's working may be required. This consists of the following operation. Find all values $X = x_1, x_2, \ldots, x_r$ such that j is an alias of X_L. Then

$$\Pr(Y = j) = \left[F_j + \sum_{L=1}^{r} 1 - F(x_L)\right]\cdot q.$$

This operation is implemented at the end of the algorithm shown in Figure 1. The reconstructed probabilities are stored in array P at the end of computation.

The practical implementation requires the generation of a pair (X, U) where U is a continuous random variable which has a uniform probability distribution over the range (0, 1) and is independent of X. We then set

$$Y = \begin{cases} X & \text{if } U \leq F(X) \\ A(X) & \text{if } U > F(X). \end{cases}$$

A Fortran implementation of this procedure is given in Figure 2.

DISCUSSION

The method is considered to be particularly well suited for use in the small general purpose computer with limited memory capacity because of the modest storage requirements. Implementation is simple; only two random numbers, one discrete and one continuous, at most two memory references, and a comparison are required to produce a new sample with the required frequency distribution.

Where it is desired to generate discrete random variables from unbounded distributions, the method may be used to handle the bulk of the distribution and a standard computational subroutine used to handle the tail.

Although it has been assumed that X is uniformly distributed, which is the most common case, the method may be extended quite easily to cover any given distribution for X.

ACKNOWLEDGMENTS

The author is most grateful for the helpful comments and suggestions given by Professor D.M. Hawkins, University of Witwatersrand, in the preparation of this paper.

REFERENCES

1. Marsaglia, G. Generating discrete random variables in a computer. *Comm. ACM 6*, 1 (1963), 37-38.
2. Ramberg, J.S., and Schmeiser, B.W An approximate method for generating symmetric random variables *Comm. ACM 15*, 11 (Nov. 1972), 987-990.
3. Sobolewski, J.S , and Payne, W.H. Pseudo-noise with arbitrary amplitude distribution. *IEEE Trans. Computers C-21* (1972), 337-345.
4. Walker, A.J Fast generation of uniformly distrıbuted pseudorandom numbers with floating point representation. *Electron. Lett. 10*, 25/26 (1974), 553-554.
5. Walker, A J. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electron. Lett 10*, 8 (1974), 127-128.

Received Aprıl 1975