Computational Statistics & Data Analysis 16 (1993) 205–217
North-Holland

The computer generation of multinomial random variates

Charles S. Davis
University of Iowa, Iowa City, IA 52242, USA

Received October 1991
Revised May 1992

Abstract: The generation of random variates from the multinomial distribution is often required in simulation studies involving methods for the analysis of discrete data. Although several methods have been proposed, comprehensive comparisons of the various algorithms have not been previously carried out. This paper describes seven methods for multinomial variate generation and compares them with respect to the criteria of accuracy, set-up time, simplicity of implementation, storage requirements, and generation time. Based on the results of an extensive empirical study, the Brown and Bromberg (*Amer. Statist.* **38**, 1984) two-stage procedure is generally the fastest algorithm, but requires a complicated set-up phase and extensive storage requirements. The conditional distribution method using the modal binomial variate generator of Kemp (*Commun. Statist.–Theor. Meth.* **15**, 1986) is shown to be a good all-purpose algorithm, in that it has wide applicability, does not require a set-up phase and extensive storage, and is reasonably competitive with the Brown–Bromberg algorithm in terms of generation time.

Keywords: Binomial distribution; Multinomial distribution; Random number generation; Simulation.

1. Introduction

The use of computer simulation to study the properties of new estimators and test procedures is common in many areas of statistical research. Simulation may be used when it is not possible to obtain theoretical results, or as a supplement to large-sample asymptotic theory. Nearly all such studies rely on the computer generation of random variates with specified distributions.

Numerous methods for generating random variates from common univariate distributions have been developed. For example, the extensive literature on Poisson, binomial, normal, gamma, and beta generators is summarized in books by Kennedy and Gentle (1980), Devroye (1986), Ripley (1987), and Dagpunar (1988), among others. In contrast, generation of pseudo-random numbers from

Correspondence to: Dr. C.S. Davis, Dept. of Preventive Medicine and Environmental Health, University of Iowa, Iowa City, IA 52242, USA.

multivariate distributions has received much less attention. The above references discuss methods of generating random variates from continuous multivariate densities such as the multivariate normal and Wishart distributions. In addition, Johnson (1987) describes methods for random vector generation from other continuous distributions. However, the literature concerning generation of discrete random vectors is quite limited.

The multinomial distribution is a common sampling model in problems involving the analysis of discrete data. Although simulation of the finite-sample properties of test statistics used in categorical data analysis often requires multinomial or product-multinomial samples, the development of accurate, general-purpose, and efficient algorithms for generating multinomial variates has not been studied extensively. In addition, comprehensive comparisons of existing algorithms have not been carried out.

In Section 2, four general methods for multinomial variate generation are described. The methodology and results of an extensive empirical comparison of seven specific multinomial generators are presented in Section 3. The algorithms are compared on the basis of accuracy, set-up time, simplicity of implementation, storage requirements, and generation time. Finally, Section 4 gives conclusions and recommendations. Although the two-stage procedure of Brown and Bromberg (1984) is generally the fastest algorithm, it requires a complicated set-up phase and extensive storage requirements. The conditional distribution method using the modal binomial variate generator of Kemp (1986) is shown to be a good all-purpose algorithm, in that it has wide applicability, does not require a set-up phase and extensive storage, and is reasonably competitive with the Brown–Bromberg algorithm in terms of generation time.

2. Methods of generating multinomial random variates

Let $X = (X_1, \ldots, X_k)$ have a multinomial distribution $M_k(N, p)$, where N is a positive integer and $p = (p_1, \ldots, p_k)$, with $p_i > 0$, and $\sum_{i=1}^{k} p_i = 1$. The distribution of X is given by

$$\Pr(X_1 = n_1, \ldots, X_k = n_k) = \frac{N!}{n_1! \cdots n_k!} p_1^{n_1} \cdots p_k^{n_k},$$

where $n_1, \ldots, n_k$ are nonnegative integers satisfying $\sum_{i=1}^{k} n_i = N$. Four basic methods of generating random variates from the $M_k(N, p)$ distribution have been proposed.

2.1. The direct method

The most straightforward approach to generating multinomial random variates is known as the naive (Ho, Gentle and Kennedy, 1979), ball-in-urn (Devroye, 1986, p. 558), or direct (Dagpunar, 1988, p. 168) method. The basic form of the

direct algorithm requires a set-up phase in which the cumulative probability vector p^* for the multinomial distribution is first generated, where $p_i^* = \sum_{j=1}^{i} p_j$, for $i = 1, \ldots, k$. In addition, the components $X_1, \ldots, X_k$ of X are initialized to zero. In the generation phase, N independent random variates U_l are first generated, where U_l is uniformly distributed on the interval (0, 1). For $l = 1, \ldots, N$, the ith component of X is incremented by 1 if $p_{i-1}^* < U_l \leq p_i^*$ (where $p_0^* = 0$).

A simple modification involves sorting the p_i values in descending order prior to calculating the vector of cumulative probabilities. This reduces the generation time of the direct method when the p_i values are unequal.

2.2. The conditional distribution method

The decomposition of the multinomial distribution into marginal and conditional binomial distributions also provides a straightforward approach to the generation of multinomial random variates (Ho, Gentle and Kennedy, 1979; Devroye, 1986, pp. 558–559; Dagpunar, 1988, p. 168). In the conditional method, X_i is generated from the binomial distribution

$$B\left(N - \sum_{j=0}^{i-1} X_j, \frac{p_i}{1 - \sum_{j=0}^{i-1} p_j}\right),$$

for $i = 1, \ldots, k$, where $X_0 = p_0 = 0$. This approach requires only the generation of k binomial random variates and might be expected to be faster than the direct method. In particular, Dagpunar (1988, p. 168) states that the conditional distribution method is likely to be faster than the direct method when N is large relative to k. On the other hand, Devroye (1986, p. 559) cautions that the conditional distribution method will not likely be competitive for small values of N. Neither author suggests a suitable method for generating the required binomial variates or compares this approach to the direct method.

The primary shortcoming of the conditional distribution approach is that an efficient binomial generator is required. Much of the literature on binomial random number generation has focused on the situation in which a large number of identically-distributed variates are desired. A number of fast, specialized methods are available, but these require a set-up phase for each combination of sample size and success probability. However, the difficulty in the case of multinomial generation is that the parameters of the binomial distribution change at every call.

Three binomial generators not requiring a set-up phase have been suggested for use with the conditional distribution method of generating multinomial variates: the Bernoulli method (Kennedy and Gentle, 1980, p. 221); the inverse CDF method (Dagpunar, 1988, p. 144); and the modal method (Kemp, 1986). A serious shortcoming of the inverse CDF method for generating binomial $B(n, p)$

variates is that it can only be used when $n \min(p, 1-p)$ is small. To be more specific, n must be less than $\log(c)/\log(\max(p, 1-p))$, where c is the smallest positive real number.

Kemp's (1986) modal algorithm is a modification of the inverse CDF method in which the search begins at the modal probability and then continues alternately below and above the mode. Since Kemp proposes a simple Stirling-type approximation to quickly and accurately calculate the modal probability, the method is easy to program and well-suited to the variable parameter case. Although Kemp suggested that his algorithm might be useful in conjunction with the conditional distribution method of generating multinomial variates, this possibility was not explored. He did, however, compare the modal method to two other methods of generating binomial variates. Over a wide range of parameter combinations, the modal method was appreciably faster than the exact random median method (Relles, 1972; Ahrens and Dieter, 1974), which was used by Ho, Gentle and Kennedy (1979) in generating multinomial variates. For $np < 500$, the modal method was also faster than the Ahrens and Dieter (1980) acceptance/rejection algorithm. Although the Ahrens and Dieter (1980) method was faster than the modal method when np was greater than 500, it is considerably more complicated in that it requires a high-accuracy algorithm for computing the logarithm of the gamma function.

2.3. The alias method

The alias method (Walker, 1974, 1977) is a fast general method for generating random variates from an arbitrary discrete distribution with a finite number of outcomes. Although originally presented without proof, the theoretical basis was provided by Kronmal and Peterson (1979), who showed that any discrete distribution with a finite number (m) of outcomes can be expressed as an equiprobable mixture of m two-point distributions. The alias method requires a complicated set-up phase in which two arrays of length m are initialized (an array of cut-off values and an array of aliases). The generation step, however, is extremely quick and does not depend on the specific distribution or the number of mass points m. In the implementation described by Kronmal and Peterson (1979), only one uniform random variate on the interval (0, 1) and four operations are required.

The most straightforward implementation of the alias method is to generate observations from the $M_k(1, p)$ distribution. The sum of N such independent random variates has the required $M_k(N, p)$ distribution.

2.4. The two-stage method of Brown and Bromberg (1984)

Brown and Bromberg (1984) propose an ingenious two-stage procedure based on the fact that the conditional distribution of independent Poisson random variables is multinomial. In the first stage, k independent Poisson variates Y_i are generated using the alias method, where $Y_i \sim P(\lambda_i)$ and $\lambda_i / \sum_{j=1}^{k} \lambda_j = p_i$. Note

that $\lambda_+ = \sum_{i=1}^{k} \lambda_i$ can be set to any value, subject to the restriction that $\lambda_i = \lambda_+ p_i$. Let N^* denote the sum of these k Poisson variates. If $N^* > N$, the sample is discarded and the first stage is repeated. If $N^* = N$, the first-stage sample has the required $M_k(N, p)$ distribution. If $N^* < N$, then $N - N^*$ observations $Z_j = (Z_{1j}, \ldots, Z_{kj})$ are generated from the $M_k(1, p)$ distribution and $X_i = Y_i + \sum_{j=1}^{N-N^*} Z_{ij}$.

Brown and Bromberg (1984) generate the Z_j values using the alias method for $M_k(1, p)$ random variates. The also provide a simple approximation for λ_+ which minimizes the expected number of uniform random deviates. Although the generation time is less than with the straightforward application of the alias method, the set-up and storage requirements become substantial as k and N increase. In particular, $2k(N+2)$ locations are necessary to store the cutoff values and aliases required to generate the first-stage Poisson sample.

2.5. Comparisons among methods

Limited comparisons among methods for generating multinomial random variates are given in Ho, Gentle and Kennedy (1979) and Brown and Bromberg (1984). Ho, Gentle and Kennedy (1979) compared four multinomial generators (the direct method and three implementations of the conditional method) in a limited simulation study involving only the case $k = 5$. Two of the conditional methods involved approximate methods of binomial variate generation and the resulting multinomial random variates failed goodness-of-fit tests. However, the conditional method using the exact random median method for generating binomial variates (Relles, 1972; Ahrens and Dieter, 1974) produced satisfactory multinomial variates and was considerably faster than the direct method for $N > 20$.

Brown and Bromberg (1984) analytically compared their two-stage procedure with the alias method. Although their algorithm is more efficient than the alias method, the gains are only realized when many tables are generated from the same multinomial distribution, since the efficiency comparisons do not take into account the initial cost of evaluating the Poisson probabilities and setting up the alias tables. In addition, the storage requirements of the two-stage method become excessive as N and k increase.

3. Empirical comparisons

Since the existing literature does not provide sufficient information for comparing methods, the properties of seven algorithms for generating multinomial variates were compared in an extensive empirical study. The specific methods considered were: direct (D) and sorted direct (DS); alias (A); conditional distribution with binomial variates generated using the Bernoulli (CB), inverse CDF (CI), and modal (CM) methods; Brown–Bromberg (1984) two-stage procedure (BB).

Table 1
Probability vectors p used in the comparative study of multinomial variate generators

k	p
3	0.3 0.3 0.4
3	0.2 0.3 0.5
3	0.6 0.3 0.1
3	0.3 0.6 0.1
3	0.1 0.3 0.6
4	0.25 0.25. 0.25 0.25
4	0.4 0.3 0.2 0.1
4	0.1 0.2 0.3 0.4
5	0.2 0.2 0.2 0.2 0.2
5	0.5 0.2 0.1 0.1 0.1
5	0.1 0.1 0.1 0.2 0.5
6	0.167 0.167 0.167 0.167 0.166 0.166
6	0.4 0.2 0.1 0.1 0.1 0.1
6	0.1 0.1 0.1 0.1 0.2 0.4
6	0.101 0.138 0.186 0.138 0.186 0.251
9	0.057 0.077 0.105 0.078 0.105 0.141 0.105 0.141 0.191
12	0.066 0.072 0.078 0.084 0.071 0.078 0.087 0.096 0.076 0.086 0.097 0.109
15	0.024 0.033 0.044 0.059 0.080 0.033 0.044 0.059 0.080 0.108 0.044 0.059 0.080 0.108 0.145
16	0.024 0.031 0.041 0.056 0.031 0.041 0.056 0.075 0.041 0.056 0.075 0.102 0.056 0.075 0.102 0.138
20	0.016 0.020 0.028 0.037 0.050 0.020 0.028 0.037 0.050 0.068 0.028 0.037 0.050 0.068 0.092 0.037 0.050 0.068 0.092 0.124
25	0.010 0.014 0.018 0.025 0.034 0.014 0.018 0.025 0.034 0.045 0.018 0.025 0.034 0.045 0.061 0.025 0.024 0.045 0.061 0.082 0.034 0.045 0.061 0.082 0.111
25	0.5 0.05 0.05 0.05 0.05 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
25	0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.05 0.05 0.05 0.05 0.5 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01
25	0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.05 0.05 0.05 0.05 0.5
30	0.021 0.023 0.025 0.028 0.030 0.033 0.023 0.026 0.028 0.031 0.034 0.037 0.025 0.028 0.031 0.034 0.038 0.042 0.028 0.031 0.034 0.038 0.042 0.047 0.030 0.034 0.038 0.042 0.047 0.052
36	0.021 0.019 0.020 0.021 0.023 0.024 0.019 0.021 0.022 0.024 0.026 0.028 0.020 0.022 0.024 0.027 0.029 0.032 0.021 0.024 0.027 0.030 0.033 0.037 0.023 0.026 0.029 0.033 0.038 0.043 0.024 0.028 0.032 0.037 0.043 0.050
50	0.02 (50 times)
100	0.01 (100 times)

3.1. Methodology

The empirical properties of the above seven algorithms were compared by generating 1000 multinomial vectors from each of 112 distinct $M_k(N, p)$ distributions. These 112 distributions were defined by the combinations of four total sample sizes ($N = 50$, 100, 200, 500) and 28 probability vectors p ranging in dimension from $k = 3$ to $k = 100$. The choices for p are listed in Table 1 and were selected to cover a wide range of practical situations. In particular, the

larger values of k cover situations in which it is desired to generate random contingency tables with specified properties. For selected multinomial dimensions ($k = 3$, 4, 5, 6, and 25), multiple distributions were used in order to investigate the effects of the ordering and relative magnitudes of the components of p.

The seven methods were compared with respect to set-up time, generation time, and goodness-of-fit of the generated variates to the true multinomial distribution. The multinomial generation algorithms were implemented in single-precision using Microsoft FORTRAN version 5.0 on an IBM PS/2 model 80 personal computer. Two minor errors in Kemp (1986) were corrected in the implementation of the modal method. The first term in brackets at the top of page 809 should be raised to the one-half power and the denominator of step 15 on page 810 should be $(m+j)(1-p)$ instead of $(n-m+1-j)(1-p)$. The only machine-specific constant required was the value of the smallest positive real number c; this was needed only for the CI method. In Microsoft FORTRAN (single precision) $c = 1.175 \times 10^{-38}$. Uniform random variates on the interval (0, 1) were generated using the FORTRAN algorithm of Wichmann and Hill (1982).

3.2. Set-Up time

The seven algorithms have widely-varying set-up requirements prior to generating random variates from a specific multinomial distribution. The conditional distribution methods CB, CI, and CM do not require a set-up phase. For the direct and alias methods (D, DS, A), the set-up step relates only to the components of p, and is thus independent of N. Only the BB method has a set-up phase that depends on both N and k.

Table 2 displays set-up times for generators D, DS, A, and BB. For each generator, the set-up time increased with k and was independent of p for fixed k. Therefore, Table 2 displays values for $k = 3$, 6, 12, 25, 50, and 100 only. The direct algorithm D requires the least set-up time, followed by algorithms DS and A, each of which is approximately 10 times slower than D as k increases. The

Table 2
Set-up times (seconds) for multinomial random variate generators

Method	N	k					
		3	6	12	25	50	100
D	–	0.00005	0.00011	0.00017	0.00033	0.00066	0.00132
DS	–	0.00033	0.00082	0.00209	0.00516	0.00654	0.01318
A	–	0.00039	0.00077	0.00143	0.00292	0.00582	0.01153
BB	50	0.035	0.065	0.126	0.254	0.498	0.984
	100	0.063	0.119	0.227	0.461	0.909	1.799
	200	0.115	0.221	0.430	0.868	1.724	3.421
	500	0.267	0.522	1.025	2.089	4.160	8.270

complexity of the set-up phase for algorithm BB is very evident in Table 2. In particular, for $N = 500$ and $k = 100$, a single execution of the BB set-up phase requires more than eight seconds.

Since the set-up phase is executed only when the multinomial parameters change, the execution time required by this phase is less important than factors such as program length, storage requirements, and speed of random variate generation. Thus, although the set-up times required by the various algorithms differ substantially, this factor alone will not likely be the major basis for a choice of generator.

3.3. Generation time

In general, for fixed total sample size N, the time required to generate 1000 random variates increased approximately linearly as k increased. The notable exception was the alias method A, for which the generation time for fixed N was nearly independent of k. In addition, the specific configuration of p for fixed k had little or no effect on the generation times of methods DS, A, CI, CM, and BB. For the remaining two methods (D, CB), configurations where the first component of p was the largest resulted in faster generation times than configurations in which all components were roughly equal or when the first component was the smallest. Even for these two methods, the value of k was still the major factor determining the generation speed for fixed N. Therefore, as an initial summary of the generation times required by the various algorithms, Table 3 gives the minimum ($k = 3$) and maximum ($k = 100$) execution times (seconds) required by each of the seven methods to generate 1000 random variates for $N = 50$, 100, 200, and 500.

For the specific case $k = 3$, and more generally for small k, the algorithms can be divided into two groups, since D, DS, A, and CB are much slower than CI, CM, and BB. For $N \leq 200$, CI, CM, and BB are generally at least 10 times as fast as the other methods. Although there is more variability among the

Table 3
Minimum ($k = 3$) and maximum ($k = 100$) execution times (seconds) for generating 1000 multinomial variates

Method	$N = 50$		$N = 100$		$N = 200$		$N = 500$	
	Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.
D	11.4	53.7	22.7	107.4	45.5	214.6	113.5	537.3
DS	11.3	53.4	22.6	106.9	45.1	213.6	112.6	534.6
A	12.1	12.1	24.1	24.2	48.1	48.3	120.2	120.8
CB	14.4	519.8	28.7	1036.4	57.3	2068.2	143.0	5151.3
CI	1.9	51.2	2.7	53.7	4.3	70.9	22.3	83.3
CM	2.1	86.8	2.2	89.9	2.5	93.7	2.9	99.8
BB	2.7	31.7	3.4	33.1	4.2	34.3	5.7	71.7

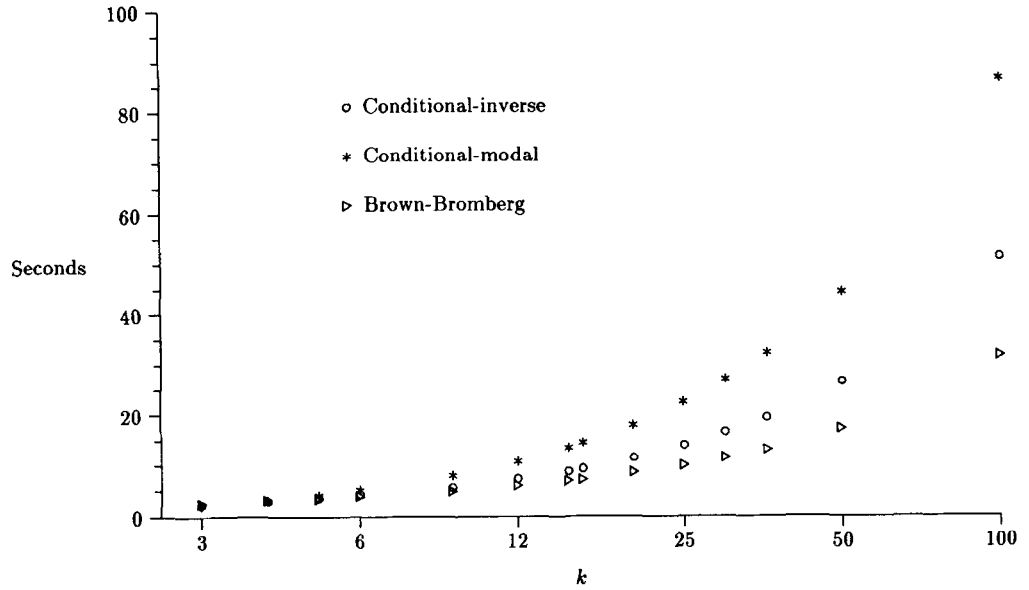

Fig. 1. Execution times for $N = 50$.

execution speeds for CI, CM, and BB when $N = 500$, these algorithms are still much faster than the remaining methods.

For large k, and with particular respect to the specific case $k = 100$, the CB algorithm is much slower than the other methods for all four values of N. The generation times for D and DS are roughly comparable, and although these two methods are competitive with CI, CM, and BB at $N = 50$, they are slower at $N = 100$ and much slower for $N \geq 200$. The execution time for method A is independent of k and, for large k, this method is the fastest for $N \leq 100$. However, this advantage is not maintained at $N = 500$.

The results displayed in Table 3 support the general conclusion that algorithms CI, CM, and BB are preferred over D, DS, A, and CB. For $k = 3$, CI, CM, and BB are much faster than the remaining algorithms for all four values of N. Although A is the fastest algorithm when $k = 100$ and $N \leq 100$, it is considerably slower than CI, CM, and BB for small k. Therefore, the more detailed generation time comparisons that follow consider only CI, CM, and BB.

For $N = 50$, 200, and 500, respectively, Figures 1–3 display the execution times (seconds) of methods CI, CM, and BB for generating 1000 multinomial variates. (Generation times for $N = 100$ are not shown, since they are intermediate to those from $N = 50$ and $N = 200$.) For the cases in which multiple choices of p were considered ($k = 3$, 4, 5, 6, and 25), the specific values of the components of p had little effect on the generation time for fixed k. Therefore, the figures display only the results for the first of the multiple distributions listed in Table 1.

For small N (≤ 100) and small k (≤ 6), the generation times of BB, CM, and CI are similar. For large N (> 100) and small k, CM is faster than BB, which in

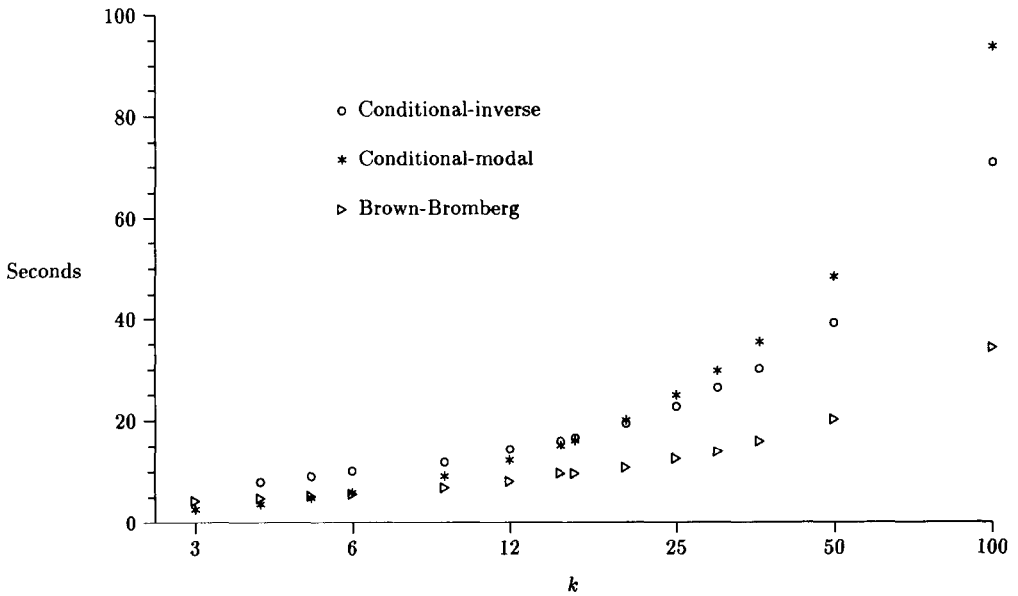

Fig. 2. Execution times for $N = 200$.

turn is faster than CI. For large N and k (> 6), BB is faster than CM and CI. As k increases, CM is the slowest of the three algorithms, i.e., for $k > 16$ when $N = 200$ and for $k > 36$ when $N = 500$.

Of the fastest algorithms, only CM and BB could be used for all of the distributions considered in this study. While the generation speed of BB is often

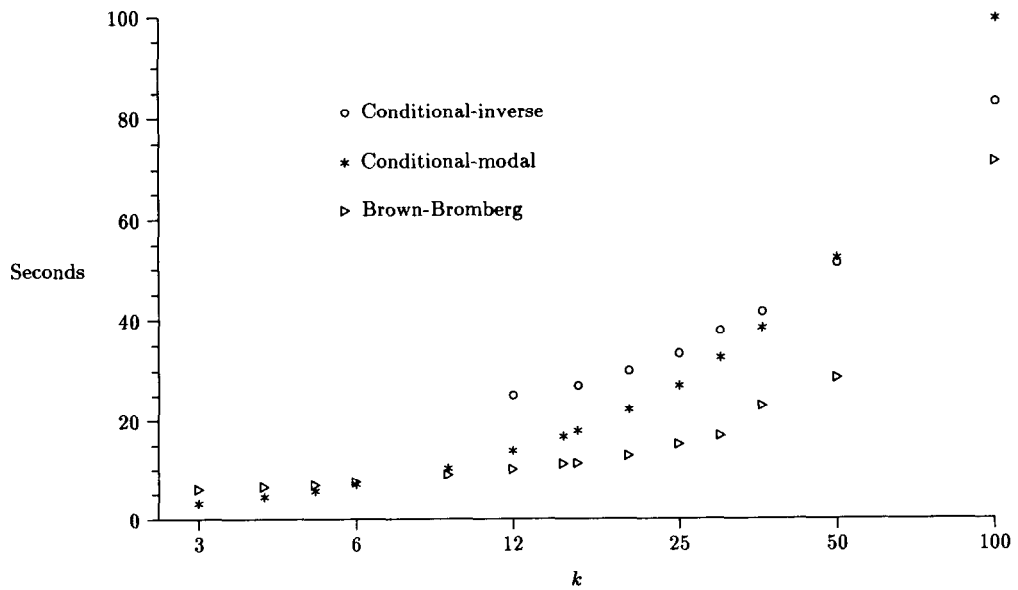

Fig. 3. Execution times for $N = 500$.

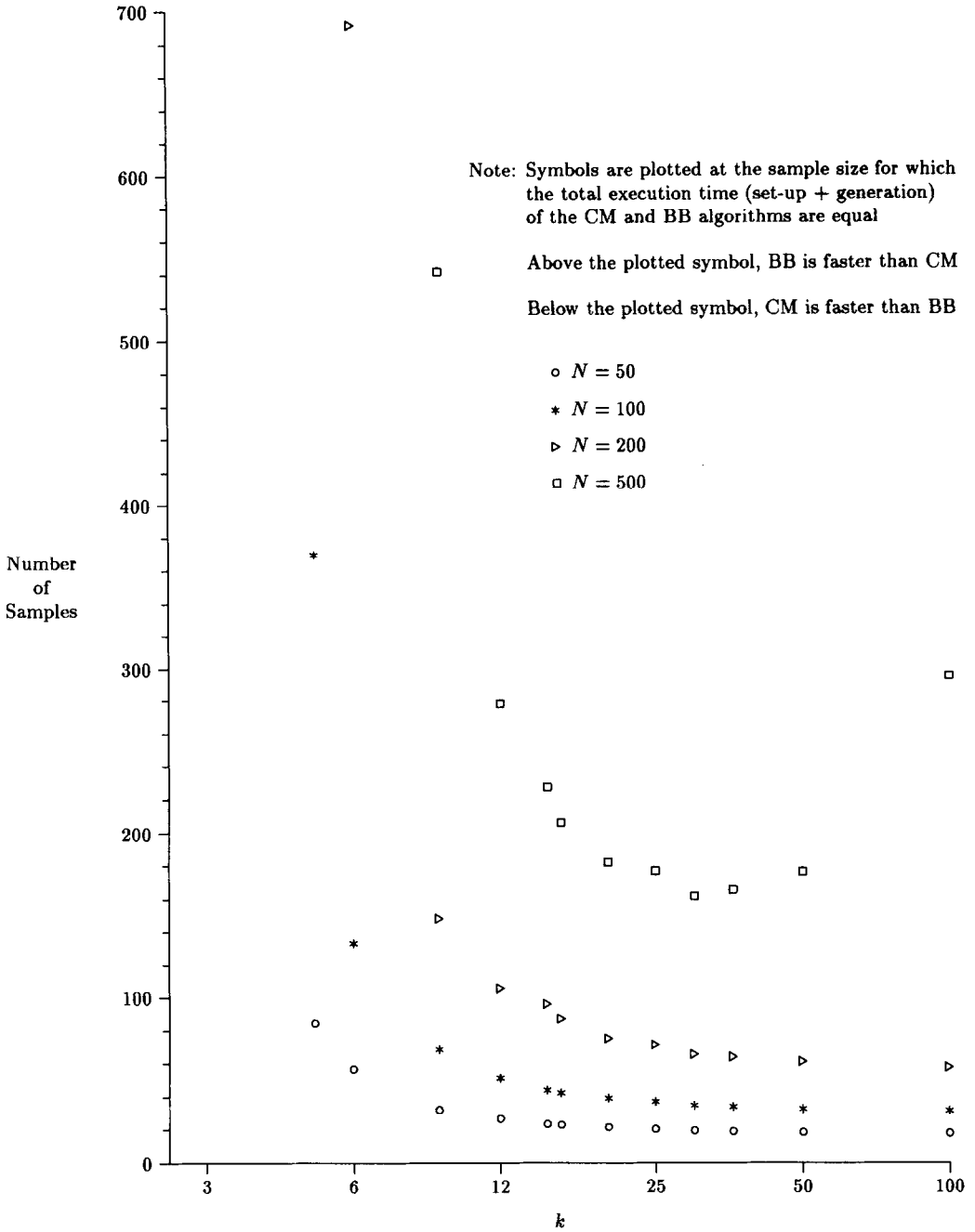

Fig. 4. Comparison of the total generation times (including set-up) of algorithms CM and BB.

much faster than CM, especially when k is large, the comparisons in Figures 1–3 do not take into account the set-up time requirements of BB. Figure 4 displays the number of samples n at which the total times (set-up plus generation) are equal for BB and CM. For small k (≤ 4 at $N = 100$, ≤ 6 at $N = 500$),

CM is faster than BB regardless of n. As N increases, the upper limit of the sample size for which CM is faster than BB also increases. Still, the BB algorithm is nearly always faster if more than 300 samples are to be generated.

3.4. Goodness of fit

Although all of the methods are exact, the goodness of fit of the generated variates to the specified multinomial distribution was assessed in order to detect any difficulties in implementation. The empirical distributions of generated goodness-of-fit statistics were compared with the corresponding chi-square reference distributions by testing for differences between empirical p-values and the nominal p-value of 0.05. Although the observed proportion of replications exceeding the 95th percentile is occasionally outside the 95% normal-theory confidence interval, the departures are generally small and usually occur when expected frequencies are small (and the chi-square approximation might not be adequate).

The single exception to the adequacy of the fit was for algorithm BB when $N = 500$ and $k \geq 36$. This was due to the fact that, as N and k become large, it is difficult to accurately compute the required Poisson probabilities in single precision. This problem did not occur when double-precision calculations were used.

4. Conclusions and recommendations

Of the seven multinomial random variate generators studied in this paper, the direct, sorted direct, alias, and conditional-Bernoulli methods are much slower than the conditional-inverse, conditional-modal, and Brown–Bromberg methods. Although the Brown–Bromberg algorithm is clearly the preferred generator of multinomial random variates in terms of speed, this advantage is gained at the expense of requiring a complex set-up phase and a large number of storage locations. While the storage issue is not likely to be a problem on mainframes or minicomputers, it is a major issue when carrying out simulation studies on personal computers. In addition, the set-up phase computations must be carried out carefully in order to guarantee the accuracy of the generated variates. The conditional-inverse method is the second-fastest generator for $N \leq 100$, does not have a set-up phase, and can be implemented much more simply than the Brown–Bromberg method. Unfortunately, computational difficulties impose severe restrictions on the use of this method, especially as N increases.

The conditional-modal method is faster than the Brown–Bromberg algorithm when k is small and faster than the conditional-inverse method when N is large. It has no set-up or array storage requirements and does not face the computational problems of the conditional-inverse method. Although it is often less than half as fast as the fastest method (Brown—Bromberg), the difference in the generation speed of these two methods is small relative to the generation speeds

of some of the other methods which have been proposed in the literature. In addition, the time required to generate 1000 multinomial variates by the CM algorithm never exceeds that of the BB method by more than one minute over the wide range of distributions considered in this study. This difference in speed is not likely to be significant in most simulation studies. Thus, the conditional-modal method is a good all-purpose algorithm for multinomial random variate generation, since it provides a nice balance between the competing criteria of execution speed and simplicity.

Acknowledgements

This research was partially supported by Grant CA39065 from the National Cancer Institute and by Grant MH46011 from the National Institute of Mental Health. Helpful comments from the associate editor and referees improved the organization of this paper; their assistance is gratefully acknowledged.

References

Ahrens, J.H. and U. Dieter, Computer methods for sampling from gamma, beta, Poisson and binomial distributions, *Computing* **12** (1974) 223–246.

Ahrens, J.H. and U. Dieter, Sampling from binomial and Poisson distributions: a method with bounded computation times, *Computing* **25** (1980) 193–208.

Brown, M.B. and J. Bromberg, An efficient two-stage procedure for generating random variates from the multinomial distribution, *Amer. Statist.* **38** (1984) 216–219.

Dagpunar, J., *Principles of Random Variate Generation* (Clarendon Press, Oxford, 1988).

Devroye, L., *Non-Uniform Random Variate Generation* (Springer-Verlag, New York, 1986).

Ho, F.C.M., J.E. Gentle and W.J. Kennedy, Generation of random variates from the multinomial distribution, *Proc. ASA Statistical Computing Section* (1979) 336–339.

Johnson, M.E., *Multivariate Statistical Simulation* (Wiley, New York, 1987).

Kemp, C.D., A modal method for generating binomial variables, *Commun. Statist.–Theor. Meth.* **15** (1986) 805–813.

Kennedy, W.J. and J.E. Gentle, *Statistical Computing* (Marcel Dekker, New York, 1980).

Kronmal, R.A. and A.V. Peterson, On the alias method for generating random variables from a discrete distribution, *Amer. Statist.* **33** (1979) 214–218.

Relles, D.A., A simple algorithm for generating binomial random variables when N is large, *J. Amer. Statist. Assoc.* **67** (1972) 612–613.

Ripley, B.D., *Stochastic Simulation* (Wiley, New York, 1987).

Walker, A.J., New fast method for generating discrete random numbers with arbitrary frequency distributions, *Electronics Letters* **10** (1974) 127–128.

Walker, A.J., An efficient method for generating discrete random variables with general distributions, *ACM Trans. Math. Software* **3** (1977) 253–256.

Wichmann, B.A., and I.D. Hill, Algorithm AS 183: an efficient and portable pseudo-random number generator, *Appl. Statist.* **31** (1982) 188–190.