TIME COMPLEXITY OF SEQUENTIAL AND PARALLEL MONTE CARLO SOLUTION OF LINEAR EQUATIONS

V.C. BHAVSAR

TR87-039, OCTOBER 1987

TIME COMPLEXITY OF SEQUENTIAL AND PARALLEL MONTE CARLO SOLUTION OF LINEAR EQUATIONS

ABSTRACT

The Monte Carlo method proposed by von Neumann and Ulam for solving linear equations is considered in this paper. It is shown that the primary estimate computation processes in this method can be viewed as the realizations of an absorbing Markov chain. Subsequently, the time complexity analysis of the algorithm is carried out using some results from the absorbing Markov chain theory. Two techniques, proper transition probability assignments and the truncation of random walks, are discussed for the time complexity reduction. Finally, the various schemes recently presented for the development of parallel Monte Carlo algorithms are shown to be applicable in implementing the method on parallel computers. It is shown that the time complexity of the method for estimating the solutions of a system of N linear equations on N·K processors, using K primary estimates for each of the unknowns, is independent of N.

INDEX TERMS

Monte Carlo methods, parallel algorithms, linear equations, analysis of algorithms, Markov processes.

Paper: Time Complexity of Sequential and Parallel Monte Carlo Solution of Linear Equations
Author: V.C. Bhavsar
Journal: SIAM SISSC

AMS (MOS) Subject Classifications

15 A 06	Linear Equations
15 A 09	Matrix Inversion
60 J 05	Random Walk
60 K 10	(Special processes) Application
62 E 25	Monte Carlo Studies
62 G 30	Order Statistics
65 C 05	Monte Carlo Methods
65 W 05	Parallel Computation
68 Q 25	Analysis of Algorithms

Abbreviated Title

Monte Carlo Solution of Linear Equations

CONTENTS

1. INTRODUCTION

One of the most fruitful fields of application of the Monte Carlo techniques, since long, has been the solving of linear algebraic equations. Another potential application of the same techniques has been the field of solving large system of integral and/or differential equations (see [13]).

Von Neumann and Ulam were the first to propose a fundamental method (N-U method) for inversion of a matrix (this method was published by Forsythe and Leibler [10]). Later, Wasow also proposed a different method for the inversion and compared its efficiency with the N-U method. Opler [18] and Todd [21] have carried out the inversion of matrices on computing machines. Obviously, these methods can be used to solve linear equations. Akaike [1], Shrieder [19] and Blagoveshchenskii [7] have also proposed different methods for solving linear equations. All these methods of inversion have suffered from slow convergence. In this context, for accelerating the convergence of the N-U method, Halton [11] has proposed a technique, namely, Sequential Monte Carlo, which incorporates sequential improvement in the estimator used. Halton [12] has also proposed least-squares Monte Carlo techniques for solving inconsistent and overdetermined system of approximate equations.

In this paper, we consider the N-U method for discussion since similar methods, since long, have been applied to solving integral equations, eigenvalue problems (see [18,19]), and many other problems, such as thermal multigroup transport equations [20], and moreover the Sequential Monte Carlo method by Halton [11] is also based on the N-U method. The time complexity results presented in this paper can be easily applied to the other Monte Carlo methods for solving linear equations as demonstrated in Bhavsar [4, Chap. 3].

In the next section, we briefly review the N-U method, and in Section 3 a sequential algorithm for the method is presented. It is shown that the primary estimate computation (PEC) processes in this method can be viewed as the realizations of an absorbing Markov chain. Subsequently, the time complexity analysis of the algorithm is carried out using some results from the absorbing Markov chain theory. In Section 4, we propose two techniques for the time complexity reduction of the N-U method. It is shown in Section 5 that the N-U method possesses intrinsic parallelism at all the four levels discussed in [4,6]. Section 6 is devoted to the parallel algorithms for the N-U method. The various schemes for development of parallel Monte Carlo algorithms discussed in [6] are shown to be applicable in implementing the N-U method on parallel computers. Concluding remarks are presented in the last section.

2. THE VON NEUMANN AND ULAM (N-U) METHOD

The principal advantage of the N-U method is that, unlike classical methods for the solution of sets of simultaneous linear equations, it enables to estimate one component of the solution independently of the other components; when N is of the order of thousand or more, this can be very important. This method was first published in [10] and formalized in [13] as follows.

Consider a nonsingular system of equations

$$\sum_{s=1}^{N} A_{rs} \cdot x_s = b_r, \quad r \in S = \{1,2,\ldots, N\}. \tag{1}$$

This equation can be written as a matrix equation $A.x = b$, and will have the unique solution given by

$$x = A^{-1}b. \tag{2}$$

The equation (2) can always be put into the form

$$x = a + Hx, \tag{3}$$

where the spectral radius of H, $\rho(H) < 1$. In this case we have

$$\begin{aligned} x = (I-H)^{-1} a &= a + Ha + H^2a + \ldots + H^ma + \ldots \\ &= \sum_{m=0}^{\infty} H^m a, \end{aligned} \tag{4}$$

and the Neumann series is absolutely convergent.

Now, the Monte Carlo estimates of each of the unknowns in (3) can be obtained <u>independently</u> of the remaining unknowns. The procedure for obtaining the Monte Carlo estimates of an unknown is as follows.

A set of random walks is defined on the augmented index set, $\bar{S} = \{0,1,\ldots, N\}$, the extra index value 0 corresponds to a "stop", i.e. it represents the termination index for the random walk. Let p_{rs} denote the probability of transition from index r to index s, with $p_{rs} \neq 0$ unless $H_{rs} = 0$. Then corresponding to each walk $\Gamma = [r,s_1,s_2,\ldots,s_m,0]$, with $r, s_1, s_2,\ldots, s_m \neq 0$, the primary estimator for an unknown x_r is defined as

$$x_r(\Gamma) = Z_{rs_1} \cdot Z_{s_1s_2} \cdots Z_{s_{m-1}s_m} \cdot Z_{s_m0}, \tag{5}$$

where for $j \neq 0$, $Z_{ij} = H_{ij}/p_{ij}$, and for $j = 0$, $Z_{i0} = a_i/p_{i0}$.

The secondary estimator for x_r is defined by the arithmetic mean of K primary estimates as

$$Y_r = \frac{1}{K} \sum_{i=1}^{K} X_r(\Gamma_i), \tag{6}$$

which has the expectation $E(Y_r) = x_r$, if $\rho(H^+) < 1, (H^+)_{rs} = H^+_{rs} = |H_{rs}|$.

3. A SEQUENTIAL ALGORITHM FOR THE N-U METHOD

To carry out a step in the random walk from index s1 to index s2, the index s2 can be found out as follows. Define $q_{s1}(n) = \sum_{s2=0}^{n} p_{s1s2}$ and then obtain s2 using a function $\phi(s1,\xi)$, where ξ is a sample from Λ, as

$$s2 = \phi(s1,\xi), \tag{7}$$

such that

$$\phi(s1,\xi) = i, \text{ if } q_{s1}(i-1) \leq \xi < q_{s1}(i) \tag{8}$$

A sequential algorithm based on the N-U method is given below. It obtains the secondary estimates Y_1, Y_2,..., Y_n, of the unknowns x_1, x_2,..., x_N, respectively. In the algorithm the variables are the same as in the earlier discussion. Λ is assumed to be an ideal random number generator.

Algorithm 1: A Sequential Algorithm for the N-U Method

```
begin
    initialize Λ
    for i = 1 step 1 until N do
            Y_i := 0
            for j = 1 step 1 until K do
                X := 1; s1 := i
            while s1 ≠ 0 do pecstep (Λ,X,s1) end do
            Y_i := Y_i + X
    end do
    Y_i := Y_i/K
end
```

```
procedure pecstep (Λ, X, s1);
begin
    obtain a sample ξ from Λ
    s2 := φ(s1,ξ); X := X. Z_{s1s2}
    s1 := s2
end
```

If we obtain successive samples $\xi^{(1)}$, $\xi^{(2)}$,..., $\xi^{(\nu)}$, from Λ to carry out i-th random walk (i.e., a primary estimate computation) for an unknown x_r, and the random walk reaching the 'stop' index at the ν-th sample, then the primary estimate will be given by

$$X_r(\zeta_i) = X_r(\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(\nu)}). \tag{9}$$

The following theorem characterizes the PECs in Algorithm 1.

THEOREM 1. In the N-U method, the PECs for an unknown x_i represent runs through a discrete-time, finite, homogeneous, absorbing Markov chain with state space $\overline{S} = \{0, 1, 2, \ldots, N\}$, the state 0 representing the absorbing state, the transition probability matrix being given as $\underline{P} = [p_{ij}]$, with the initial state i.

Proof. A PEC involves a random walk over finite states (equal to N + 1) with the transition probability to a next state depending only on the present state. The random walk terminates when it reaches state 0. Thus, the PEC process is a finite absorbing Markov chain. It is discrete-time since the state transitions occur at discrete time instants. Finally, the Markov chain is homogeneous because the transition probabilities do not depend on the earlier number of state transitions. Hence, the proof of the theorem.

3.1 Time Complexity Measure

As can be easily seen from Algorithm 1, the execution time of the algorithm is primarily determined by the execution time of the PECs. In the following analysis, we will assume that the secondary estimate computation time is small enough to be negligible compared to PEC times. Further, we initially assume that

A1: The execution time of procedure pecstep is constant.

Subsequently, we will relax this assumption and show its implications to the complexity results. The execution time of a PEC is determined by the number of times the procedure pecstep is executed, which in turn depends on the number of state transitions (ν) required to reach the absorbing state from a given initial state. Thus, the natural time complexity measure for PECs is the number (ν), which also can be considered as the duration of a random walk.

In the ensuing analysis of the algorithms, it is assumed that the set of linear algebraic equations are given in the form of (3); this assumption can be justified in many applications of the N-U method (e.g. the solution of PDEs discussed in [6, 13]).

3.2 Analysis of the Algorithm

Curtiss [9] has carried out a theoretical comparison of the efficiencies of two classical methods (the Gauss elimination method and an iterative method) under some assumptions so that their comparison with the N-U method, which obtains an estimate of the solution (rather than the solution) is possible. Wassow [23] has investigated the mean

number of steps required for random walks in n-dimensional domains in the context of the Monte Carlo method. Wasow [24] has generalized these results by applying similar methods to the study of a moment generating function for a number of steps and its distribution function. Here, we use the results from absorbing Markov chain theory, that become applicable due to Theorem 1, to carry out the time complexity analysis.

The substochastic matrix Q, corresponding to the transition probability matrix $\underline{P}$ is given as

$$Q = [p_{ij}],\ i,j \neq 0, \tag{10}$$

and the fundamental matrix F is given as

$$F = (I - Q)^{-1}, \tag{11}$$

where I is the identity matrix. The element F_{ij} of the fundamental matrix gives the total number of times the chain will be in state j, if the chain has started in state i.

THEOREM 2. In the estimation of an unknown x_i by the N-U method, the PEC times are discrete random variables with mean

$$\mu_i = [FU]_i, \tag{12}$$

and variance

$$\sigma^2_i = [(2FF_{dg})\ U - FU - F_{sq}U]_i, \tag{13}$$

where F is the fundamental matrix, F_{dg} and F_{sq} are derived from F, and U is a unit vector.

Proof: Follows from results of absorbing Markov chain theory [16, p. 49], which become applicable due to Theorem 1.

REMARK 1. μ_i and σ_i^2 of the PEC times for an unknown x_i are independent of N, the size of the system of linear algebraic equations.

THEOREM 3. The μ_i and σ_i^2 of PEC times of an unknown x_i have the upper bounds as follows:

$$\mu_i \leq \frac{1}{(1 - ||Q||_\infty)} \tag{14}$$

$$\sigma_i^2 < \frac{2}{(1 - ||Q||_\infty)^2}, \tag{15}$$

where $||Q||_\infty$ denotes the $\ell - \infty$ norm of the matrix Q.

Proof. By Theorem 2, $\mu_i = [FU]_i = \sum_{j=1}^{N} F_{ij}$, where $F = (I-Q)^{-1}$. For any absorbing Markov chain, (I - Q) has an inverse and we have [16, p. 46]:

$$F = (I - Q)^{-1} = I + Q + Q^2 + \ldots = \sum_{k=0}^{\infty} Q^k, \tag{16}$$

where $Q^k \to 0$ as $k \to \infty$. Hence,

$$\begin{aligned}
\mu_i &= \sum_{j=1}^{N} [I + Q + Q^2 + \ldots]_{ij} \\
&\leq \max_{1 \leq i \leq N} \sum_{j=1}^{N} [I + Q + Q^2 + \ldots]_{ij} \\
&\leq ||I + Q + Q^2 + \ldots||_\infty \\
&\leq ||I||_\infty + ||Q||_\infty + ||Q^2||_\infty + \ldots \\
&= \frac{1}{(1 - ||Q||_\infty)},
\end{aligned} \tag{17}$$

which is the desired upper bound for μ_i.

Further, by Theorem 2:

$$\begin{aligned}
\sigma_i^2 &= [(2F\, F_{dg})\, U - FU - F_{sq}\, U]_i \\
&= 2 \sum_{j=1}^{N} F_{ij}\, F_{jj} - \sum_{j=1}^{N} F_{ij} - \sum_{j=1}^{N} F_{ij}\, F_{ij} \\
&= \sum_{j=1}^{N} F_{ij}\, (2F_{jj} - F_{ij}) - \sum_{j=1}^{N} F_{ij}.
\end{aligned} \tag{18}$$

Hence,

$$\sigma_i^2 < \sum_{j=1}^{N} F_{ij} \, (2F_{jj}) \tag{19}$$

$$< 2 \sum_{j=1}^{N} F_{ij} \cdot \sum_{j=1}^{N} F_{ij} \tag{20}$$

$$< 2 \left(\max_{1 \le i \le N} \sum_{j=1}^{N} F_{ij} \right)^2 \tag{21}$$

Hence, we obtain the desired upper bound

$$\sigma_i^2 < \frac{2}{(1 - ||Q||_\infty^2)} \cdot \tag{22}$$

LEMMA 1. In the N-U method, the PECs carried out to obtain an estimate of an unknown x_i have i.i.d. execution times.

Proof. The algorithm of each of the PECs of an unknown is the same and the computations in each of the PECs are independent of each other. Moreover, the PEC times are random variables by Theorem 2. Hence the lemma follows.

In the ensuring time complexity analyses, the notations used are similar to those used in [15].

THEOREM 4. The sequential Algorithm 1, in carrying out K PECs for an unknown x_i on a sequential computer with one processor, has the expected time complexity

$$E(T_i \ (1, K)) = K \cdot \mu_i, \tag{23}$$

with the variance

$$\text{Var} \ (T_i \ (1, K)) = K \cdot \sigma_i^2. \tag{24}$$

Proof. Follows from Lemma 1 and Theorem 2.

COROLLARY 1. The sequential Algorithm 1 has O(K) expected time complexity in carrying out K PECs for an unknown x_i.

Proof. Since μ_i and σ_i^2 are independent of K as well as N (see Remark 1), the result follows.

REMARK 2. It is evident that the time complexity of the N-U method in the estimation of an unknown in a system of N linear algebraic equations is independent of N, a fact which is well known [11, p. 33].

THEOREM 5. The sequential Algorithm 1, carrying out K PECs for each of the N unknowns in a system of N linear algebraic equations, has the expected time complexity

$$E(T(1, K, N)) = K \cdot \sum_{i=1}^{N} \mu_i, \tag{25}$$

with the variance

$$\text{Var}\ (T(1, K, N)) = K \cdot \sum_{i=1}^{N} \sigma_i^2. \tag{26}$$

Proof. Since PECs for each of the unknowns are carried out independent of each other in Algorithm 1, the Theorem directly follows from Theorem 4.

REMARK 3. The solution of the system of linear equations in (3) involves the inversion of matrix $(I - H)^{-1}$. In order to determine the expected time complexity of the N-U method in solving (3), it is necessary to invert the matrix $(I - Q)$ to find out the μ_i's. The

matrices Q and H are of the same size, i.e. (N x N). Hence, the problem of determining the expected time complexity of the N-U method has the same time complexity as that of the problem being solved with the N-U method. This justifies the need to obtain upper bounds on the mean and the variance of the duration of the random walks to get an appraisal of the time complexity and therefore such bounds are obtained in this section.

COROLLARY 2. The sequential Algorithm 1 has the bounds on the expected time complexity as follows:

$$E(T(1, K, N)) \leq \frac{K \cdot N}{(1 - ||Q||_\infty)}, \tag{27}$$

with the variance

$$\text{Var } (T(1, K, N)) < \frac{2KN}{(1 - ||Q||_\infty)^2}. \tag{28}$$

Proof. Follows from Theorems 3 and 4.

COROLLARY 3. The sequential Algorithm 1 has O(N) expected time complexity in carrying out a fixed number of K PECs for each of the unknowns in a set of N linear algebraic equations.

Proof. Follows from Corollary 2.

REMARK 4. The expected time complexity of the N-U method in solving a system of N linear algebraic equations is only O(N), whereas for direct methods the time complexity has been reported [25] to be $O(N^{2.780})$, and

for iterative methods it is $O(N^2)$, (see [22]). However, the constant implied in the time complexity of the N-U method can turn out to be quite large and hence the N-U method may be efficient compared to the other methods only for large values of N (see [9] or [14, p. 85]).

4. ALGORITHMS FOR $\phi(sl,\xi)$

In the procedure pecstep in Algorithm 1, we did not specify the details of the function $\phi(sl,\xi)$ and we assumed that the execution time of the procedure is constant. However, the execution time of this procedure will usually turn out to be a random variable unless the transition probabilities are assigned properly and an efficient algorithm is used to implement $\phi(sl,\xi)$. Here, we present two algorithms for $\phi(sl,\xi)$ and discuss their implications to the time complexity of Algorithm 1.

Consider the matrix H with all the elements as non-zero (i.e. a dense matrix). Then, the substochastic matrix Q of the transition probability matrix P will have all the elements nonzero, because $p_{rs} \neq 0$ unless $H_{rs} = 0$ (see Section 2). Thus we will have (N + 1) possible next states for any given transient state in the absorbing Markov chain defined in the N-U method.

In the N-U method, it is not specified how the various transition probabilities should be selected except for the requirements that $p_{rs} \neq 0$ unless $H_{rs} = 0$, and $\sum_{s2 \in S} p_{sls2} = 1$ for any state $sl \in \overline{S}$. Hence, the transition probabilities can be selected so that it leads to a smaller execution time of Algorithm 1. Here, we assume that the selection of the transition probabilities has already been carried out. Then, we divide the range of ξ in the interval [0, 1] into (N + 1)

segments corresponding to the (N + 1) possible next states, given a present state s1, as shown in Fig. 1, and make these segment lengths equal to the values of the transition probabilities for the given state s1.

4.1 Algorithm 2 for $\phi(s1,\xi)$

Given the present state s1, the next state s2 can be determined as follows. Obtain a sample ξ from Λ and if it falls in the interval

$$q_{s1}(\ell - 1) \leq \xi < q_{s1}(\ell), \quad \ell \in \{1, 2, \ldots, N + 1\},$$

then we have $s2 = (\ell - 1)$, so that the function $\phi(s1,\xi)$, in the line 2 of procedure pecstep in Algorithm 1, can be as given below.

Algorithm 2. An Algorithm for $\phi(s1,\xi)$

```
function φ(s1,ξ)
begin
    if ξ < q_s1(1) then φ := 0
        else if ξ < q_s1(2) then φ := 1
                ...
                ...
            else if ξ < q_s1(N) then φ := (N-1)
                else φ := N
end
```

In Algorithm 2, the probability that the sample ξ falls in interval $q_{s_1}(\ell - 1) \leq \xi < q_{s1}(\ell)$ will obviously equal the value of $p_{s1,(\ell - 1)}$, because the segment lengths are equal to the transition probabilities. Let ν_{s1} denote the number of comparisons required in Algorithm 2 to determine the next state s2, given the present state s1. Then, the

expected value of ν_{s1} will be

$$E(\nu_{s1}) = \sum_{i=1}^{N} i \cdot p_{s1,(i-1)} \tag{29}$$

and the variance of ν_{s1} will be given as

$$\text{Var}(\nu_{s1}) = \sum_{i=1}^{N} i^2 \cdot p_{s1,(i-1)} - E^2(\nu_{s1}). \tag{30}$$

Let ν_i^* denote a random variable, representing the total number of comparisons required in the execution of the Algorithm 1 in obtaining a primary estimate of an unknown x_i, i.e. when the absorbing Markov chain has the initial state equal to i. As already stated in Section 3.2, F_{ij}, the $(ij)^{th}$ element of the fundamental matrix, gives the expected number of times the chain will be in transient state j, if the chain is started in state i. Thus, the expected value of ν_i^* will be given as

$$E(\nu_i^*) = \sum_{j=1}^{N} \{F_{ij} \cdot E(\nu_j)\}. \tag{31}$$

Example 1. Consider the transition probability assignment such that all the transition probabilities are equal, i.e.

$$p_{s1,s2} = \frac{1}{(N+1)}, \quad s1 \in S, \; s2 \in \bar{S} \tag{32}$$

Then for $s1 \in S$ we will have

$$E(\nu_{s1}) = \frac{1}{(N+1)} \sum_{i=1}^{N} i = \frac{N}{2}, \tag{33}$$

and

$$\begin{aligned} \text{Var}(\nu_{s1}) &= \frac{1}{(N+1)} \sum_{i=1}^{N} i^2 - \frac{N^2}{4}, \\ &= \frac{N \cdot (N + \frac{1}{2}) \cdot (N+1)}{(N+1) \cdot 3} - \frac{N^2}{4} \\ &= \frac{N\,(N+2)}{12} \end{aligned} \tag{34}$$

The fundamental matrix F is obtained as follows. The transition probability matrix $\underline{P}$ will be given as

$$\underline{P} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \alpha & \alpha & \alpha & \cdots & \alpha \\ \alpha & \alpha & \alpha & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \alpha & \alpha & \alpha & \cdots & \alpha \end{bmatrix}_{(N+1)\,x\,(N+1)} \tag{35}$$

where $\alpha = \dfrac{1}{(N+1)}$, and the substochastic matrix $Q_{N \times N}$ will be given by the lower partitioned matrix. The fundamental matrix, F, will be

$$F = (I - Q)^{-1} = \begin{bmatrix} (1-\alpha) & -\alpha & \cdots & -\alpha \\ -\alpha & (1-\alpha) & \cdots & -\alpha \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & \cdot & \\ -\alpha & -\alpha & \cdots & (1-\alpha) \end{bmatrix}_{N \times N} \tag{36}$$

$$= \frac{1}{(1 - N\alpha)} \begin{bmatrix} 1 - (N-1)\alpha & \alpha & \cdots & \alpha \\ \alpha & 1 - (N-1)\alpha & \cdots & \alpha \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \alpha & \alpha & \cdots & 1 - (N-1)\alpha \end{bmatrix}$$

$$= I + \frac{\alpha}{(1 - N\alpha)} E_{N \times N}, \tag{37}$$

where I is the identity matrix and $E_{N \times N}$ is a matrix with all the elements equal to 1. Now, if the chain is started in state i, the expected number of state transitions in the chain until absorption will be

$$\sum_{j=1}^{N} F_{ij} = 1 + N \cdot \frac{(\frac{1}{N+1})}{(1 - \frac{N}{N+1})},$$

$$= (N + 1) \tag{38}$$

for any state $i \in S$.

Thus, the expected number of comparisons required in Algorithm 1 to obtain a primary estimate of x_i, if $\phi(s1,\xi)$ is evaluated using Algorithm 2, will be given by

$$E(\nu^*_{s1}) = \frac{N(N + 1)}{2}, \tag{39}$$

where $s1 \in S$.

This implies that if the assumption A1 is not valid (i.e. when the execution time of procedure pecstep is not constant) in the present case, the PEC times will have $O(N^2)$ time complexity and they will no longer be independent of N, as stated in Remark 1. Moreover, the complexity of Algorithm 1 will turn out to be $O(N^3)$, rather than $O(N)$ as stated in Corollary 3. Thus, it is seen that the transition probability assignment can have serious implications on the time complexity of Algorithm 1.

4.2 Algorithm 3 for $\phi(s1,\xi)$

To eliminate the dependence of PEC times on N in Example 1, we propose Algorithm 3 for $\phi(s1,\xi)$, given below. Here we make use of the fact that all the transition probabilities are equal for all the transient states. We determine the next state s2 by multiplying fraction ξ by $(N + 1)$, add one to it and truncate this number so that the index for the next state is obtained directly. Obviously, in this algorithm, the execution time of $\phi(s1,\xi)$ will be constant and

consequently the execution time of procedure pecstep will turn out to be constant, so that the assumption A1 remains valid and all the results derived earlier in Section 3.2 also remain valid.

Algorithm 3. An Algorithm for $\phi(s1,\xi)$

```
begin
    φ := ⌊ ξ (N + 1) + 1 ⌋
end
```

REMARK 5. It is seen from the previous discussion that the transition probability assignment and the algorithm for $\phi(s1,\xi)$ are very crucial to the time complexity of the N-U method and they can change the time complexity drastically, as illustrated in Example 1, from $O(N)$ to even $O(N^3)$. Hence, the transition probability assignment should be carried out properly so that it is convenient computationally and at the same time the expected lengths of the random walks are not excessively large. Moveover, the algorithm for $\phi(s1,\xi)$ should be designed in such a way that it is efficient.

5. TIME COMPLEXITY REDUCTION TECHNIQUES

In this section, we discuss two techniques for the time complexity reduction of the N-U method: (i) Transition Probability Assignment, and (ii) Truncation of Random Walks. The assignment of transition probabilities is only briefly discussed by Curtiss [8, p. 225 and p. 227]. Here, we shall discuss this technique in greater detail and show clarly the interrelationship between the transition probability assignments, the duration of random walks (which primarily determine the time

complexity as seen in the last section) and the variance of the estimators. An illustrative example is also included. We will then introduce the second technique and discuss the effect of truncation of random walks on the mean and the variance of the estimators.

5.1 Transition Probability Assignment

As seen in Section 3.2, the expected duration of the random walks depends on the stochastic matrix $\underline{P}$, i.e. on the transition probabilities. Intuitively, the smaller the absorption probabilities, the larger will be the expected duration of the random walks and vice versa. Although the expected values of the estimators are immune to the changes in the transition probability assignments, the variances of the estimators are affected.

It is known that the variance of the estimator for the unknown x_i is given as [8, p. 223]

$$\mathrm{Var}(X_i) = [(I - B)^{-1} R\, a^2 - ((I - H)^{-1} a)^2]_i, \qquad (40)$$

where

$$R = \begin{bmatrix} 1/p_1 & & & 0 \\ & 1/p_2 & & \\ & & \ddots & \\ 0 & & & 1/p_n \end{bmatrix} \quad (N \times N)$$

$$p_i = 1 - \sum_{j=1}^{N} p_{ij},$$

and B is the matrix with elements given by

$$B_{ij} = H_{ij}^2/p_{ij}.$$

In (40), the first term varies depending on the probability assignment and the second term is constant (equal to the solution).

Since the transition probability assignment determines the duration of the random walks, i.e. time complexity of the N-U method, as well as the variance of the estimator, in order to compare two probability assignments we define the efficiency of Monte Carlo techniques. Since the variance of the estimator is a measure of the magnitude of the errors incurred in the estimation procedure, the efficiency of a Monte Carlo process is defined as being equal to the inverse of the product of the variance of the estimator and the expected time complexity. The interrelationship between the transition probability assignments and the variance of the estimators is illustrated through the following example.

Example 2. Consider the nonsingular system of equations

$$x = Hx + a \tag{41}$$

with $H_{ij} = \beta < 1$; i, j = 1, 2,..., N, and 'a' being the vector with all elements equal to one, so that the solution of (41) is given by

$$x_i = \frac{1}{(1-N\beta)}, \quad i = 1, 2, \ldots, N. \tag{42}$$

Now, let the transition probability assignments be made so that the transition probabilities among the transient states are all equal to $\alpha < 1/N$, i.e. the transition probability matrix is given by

$$\underline{P} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ (1-N\alpha) & \alpha & \dots & \alpha \\ (1-N\alpha) & \alpha & \dots & \alpha \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ (1-N\alpha) & \alpha & \dots & \alpha \end{bmatrix}_{(N+1) \times (N+1)} \tag{43}$$

The expected duration of the walk lengths under the above assignments will be (see Example 1).

$$E(v_i) = \frac{1}{(1 - N\alpha)}. \tag{44}$$

Further, the matrix B will have all its elements equal to $\gamma = \beta^2/\alpha$ and the inverse of $(I - B)$ will be

$$(I - B)^{-1} = \frac{1}{(1-N\gamma)} \begin{bmatrix} 1-(N-1)\gamma & & \gamma \\ & \ddots & \\ \gamma & & 1-(N-1)\gamma \end{bmatrix}_{N \times N} \tag{45}$$

with the diagonal matrix R having all its diagonal elements equal to $1/(1 - N\gamma)$. It can be easily verified that for $i = 1, 2, \dots, N$, the first terms of (40) will be

$$[(I - B)^{-1} \cdot R \cdot a^2]_i = \frac{1}{(1 - N\alpha)(1 - N\gamma)}$$

$$= \frac{\alpha}{(1 - N\alpha)(\alpha - N\beta^2)}, \tag{46}$$

Thus, the variance of the estimator X_i, for the unknown x_i, i = 1,2, ..., N, will be given by

$$\mathrm{Var}(X_i) = \frac{\alpha}{(1 - N\alpha)(\alpha - N\beta^2)} - \frac{1}{(1 - N\beta)^2} . \qquad (47)$$

Let us consider a specific case of equation (41) with N = 10 and β = 0.05. In this case, effect of the changes in the transition probability assignments is depicted in Fig. 2. It is seen that, when α = 0.05, i.e. equal to β, the variance of the estimator reduces to zero. This is not surprising because this case effectively turns out to be the case of unbiased importance sampling with zero variance. Moreover, this is the ideal probability assignment and any deviation from this leads to the decrease in the efficency of Monte Carlo method.

The above example demonstrates that the transition probability assignment is very crucial in determining the expected time complexity, the variance of the estimators and thereby the efficiency of the method.

Thus, there exists an ideal transition probability assignment which leads to zero variance of the estimators and this assignment corresponds to the unbiased importance sampling with zero variance. It is known [8] that such a sampling procedure can be devised only when the solution is known a priori; of course, in such a case we will not resort to the Monte Carlo (or any other) method at all! In conclusion, if the form of the solution is known, it is possible to choose the transition probabilities in such a way that the variance of the estimators is minimized.

5.2 Truncation of Random Walks

As discussed earlier, the time complexity of the N-U method depends on the duration of random walks, and this duration could be arbitrarily

large. In order to reduce the time complexity, we can devise a terminating rule for the random walks, say the walk lengths cannot be greater than n steps, depending on the rapidity of the convergence of the Neumann series.

If the random walks are limited to n steps, the expected duration of the random walks starting from a transient state i will be

$$E(\nu_i^n) = \sum_{j=1}^{N} [1 + Q + Q^2 + \ldots + Q^n]_{ij}$$

$$= \sum_{j=1}^{N} [(I + Q + Q^2 + \ldots) - (Q^{n+1} + Q^{n+2} + \ldots)]_{ij}$$

$$= \sum_{j=1}^{N} [(I - Q^{n+1}) \quad (I + Q + Q^2 + \ldots)]_{ij}$$

$$= \sum_{j=1}^{N} [(I - Q^{n+1}) \ (I - Q)^{-1}]_{ij}, \tag{48}$$

and the resultant reduction in the expected time complexity will be

$$E(\nu_i) - E(\nu_i^n) = \sum_{j=1}^{N} \left(\sum_{m=n+1}^{\infty} Q^m \right)_{ij} . \tag{49}$$

This truncation of the random walks results in considering only first (n+1) terms of the Neumann series in (4). In this case we will obtain a <u>biased</u> estimate of the unknowns, because the expected value of the estimators will be represented by the vector x', defined by

$$x' = a + Ha + H^2a + \ldots + H^na$$

$$= [(I - H^{n+1}) \ (I - H)^{-1}]a, \tag{50}$$

and the bias introduced for an unknown x_i will be

$$x_i - x_i' = \sum_{m=n+1}^{\infty} [H^m a]_i . \tag{51}$$

Thus, the bias introduced depends on the rapidity of the convergence of the powers of H to zero and the reduction of time complexity is determined by the convergence of the powers of matrix Q to zero. It is well known that the average rate of convergence of the powers of a convergent matrix depends on its spectral radius. As the minimum row-sum approaches unity for any convergent matrix, its eigenvalue of maximum absolute value, i.e. its spectral radius, approaches unity and the rate of convergence to zero of the powers of that matrix becomes smaller and smaller (see [22]).

Further, in the case of truncated random walks the variance of the estimator (X_i^n) for x_i will be

$$\text{Var } (X_i^n) = \{[(I - B^{n+1}) (I - B)^{-1} Ra^2] - [(I - H)^{-1} a]^2\} \qquad (52)$$

Thus, the variance of the estimator decreases as the truncation limit n for the random walks is decreased (of course, at the cost of an increase in the bias). The reduction in variance of the estimator will be

$$\text{Var } (X_i) - \text{Var } (X_i^n) = \sum_{m=n+1}^{\infty} [B^m R a^2]_i. \qquad (53)$$

6. INTRINSIC PARALLELISM IN THE N-U METHOD

The N-U method contains parallelism at all the four levels discussed in [6], as detailed below:

1. Parallelism between the computations of variables. Since the estimation of each of the unknowns can be carried out independently of the other unknowns (see Section 2), their computation can be carried out in parallel. Further, it can easily be seen that these computations are well suited for implementation on SIMD and MIMD computers.

2. Parallelism in the secondary estimate computation. The parallelism in the secondary estimate computation is as discussed in [6].
3. Parallelism between the different PECs. For each of the unknowns, the required K PECs are completely independent of each other and consequently they can be executed in parallel.
4. Parallelism in a PEC. In the N-U method, the computation of $\phi(s1,\xi)$, as given in Algorithm 2, involves a number of comparison operations and these comparisons can be carried out in parallel.

7. PARALLEL ALGORITHMS FOR THE N-U METHOD

The intrinsic parallelism in the N-U method had remained unexploited and we exploited it to develop some parallel algorithms for the method [3, 8].

In the N-U method, the PEC times primarily determine its time complexity because each PEC consists of a number of steps in a random walk and each step in turn consists of a sequence of operations. Hence, the intrinsic parallelism at levels 1 and 3 can be considered to be of primary importance.

The determination of the exact mean and the variance of the PEC times is as time consuming as the solution of the given problem itself, i.e. solution of (2) or (3), and a priori determination of their probability distributions is almost impossible. This implies that in the analysis of algorithms, we should attempt at deriving the distribution-free bounds requiring only the knowledge of the bounds on the mean and the variance of the PEC times.

Thus, we conclude that the assumptions and the approaches given in [6] for the development of parallel algorithms are directly applicable

to the design and analysis of parallel algorithms for the N-U method. Consequently, all the schemes for parallel algorithms presented in [6] can be directly adopted, especially if the estimation of only one unknown is to be carried out. Based on these schemes, parallel algorithms can easily be developed for the N-U method (see [4]). These parallel algorithms can be analysed for their time complexity by substituting the exact values of the mean and variance of the PEC times in (12) and (13), respectively, or alternatively the bounds given in (14) and (15) can be substituted.

In the ensuing subsections, we briefly discuss the developments of parallel algorithms for the N-U method using the static computation assignment (SCA) and the dynamic computation assignment (DCA) schemes discussed in [6]. An SCA SIMD algorithm is given to obtain K primary estimates of each of the N unknowns in parallel.

7.1 SCA Parallel Algorithms

All the four SCA schemes given in [6] are directly applicable for the parallel implementation of the N-U method. If $P \leq K$, it is advisable that the estimation of only one unknown is initiated in parallel, otherwise the time overheads which will be incurred in keeping track of how many estimates have been obtained for each of the unknowns will be large (this is especially recommended for SIMD algorithms). If $P > K$, it is possible to estimate more than one unknowns in parallel.

In SCA Scheme 4 discussed in [6], we initiate more than K PECs for an unknown and choose the earliest K PECs to be completed. This implies that, effectively, we are limiting the number of terms considered from the Neumann series in (4). Consequently, we would obtain a biased

estimate of the solution, as discussed in Section 5.2.

Analysis of an SCA Algorithm

We assume that K primary estimates are required to be obtained for each of the N unknowns and we use N•K processors in the parallel algorithm. In the SCA Scheme 3 (see [6]) algorithm, we initiate all the required K•N PECs in parallel. We assume that the function $\phi(sl,\xi)$ for a sl state can be $\phi := \lfloor \Psi_{sl} \cdot \xi \rfloor + 1$, where Ψ_{sl}, sl = 1, 2, ..., N, are constants.

It is easily seen that the execution time of the algorithm is determined by the largest PEC time of N•K PECs, and it will correspond to the maximum of the extreme order statistic of the N sets of PEC times, each set consisting of K PEC times.

Let the upper bounds on the mean and the standard deviation of the duration of the random walks in (14) and (15) be denoted by μ' and σ', respectively. Then the bounds on the expected time complexity can be easily obtained as follows (see [6]):

$$\mu' \leq E\,(T(NK, NK)) \leq \mu' + \frac{K - 1}{\sqrt{2K - 1}}\,\sigma' \,. \tag{54}$$

Since μ' and σ' are independent of N, we obtain the following.

THEOREM 6. The expected time complexity of the N-U method, implemented on N•K processors to obtain K primary estimates of each of the N unknowns, is independent of N.

7.2 DCA Parallel Algorithms

Since both of the DCA schemes discussed in [6] are directly applicable, the parallel algorithms can easily be developed for the N-U method, and hence these algorithms are not discussed here.

In the DCA Scheme 1 (see [6]), when the required K PECs are completed, (P-1) incomplete PECs get aborted and this leads to wasteful utilization of resources. However, for the N-U method this comment needs to be revised. We can consider the incomplete PECs as representing the truncated random walks as discussed in Section 4. Then, if the bias in the estimator due to this truncation is tolerable, the incomplete PECs can be considered. In such a case, if only K primary estimates are desired, we can terminate the computations when the (K - P + 1)-th PEC is completed. Obviously, this will reduce the execution time of the algorithm, as well as utilize the resources efficiently. Such a situation also arises in the SCA Scheme 4. In these cases the modifications in the time complexity results are obvious.

It is evident from the above discussion that DCA Scheme 2, designed specifically for increasing the resource utilization in MIMD computers for the DCA schemes, will not be of any utility.

8. CONCLUSION

In this paper, we have considered the design and analysis of sequential and parallel algorithms for the Monte Carlo solutions of linear algebraic equations, using, in particular, the fundamental method by Von Neumann and Ulam. This method was selected for the discussion because of its applicability to a variety of problems.

It was shown the the PEC processes in the N-U method can be viewed as the realizations of an absorbing Markov chain. This result was subsequently used to derive the expectation and variance of the time complexity of the N-U method. Although the N-U method has O(N) time complexity, it can be efficient compared to the direct and iterative methods for only large values of N due to the large implied constant. The direct comparison of the computational efforts required for the N-U method and the direct and iterative methods is not possible because of the stochastic convergence of the method (see [9]).

It was demonstrated that the time complexity of the N-U method can adversely get affected due to certain transition probability assignments, because the time complexity of the function $\phi(s1,\xi)$ for determining the next state transition depends on the transition probabilities. With two examples it was shown that the time complexity of the N-U method can change from O(N) to $O(N^3)$.

Two techniques for the reduction of time complexity of the N-U method were presented. For the proper transition probability assignments we have illustrated the inter-relationship between the transition probability assignments and the resulting variance of the estimators. We have shown that an ideal transition probability assignment exists and it corresponds to the unbiased importance sampling with zero variance. Unfortunately, such an assignment could be devised only when the solution is known a priori. In the second technique, the truncation of random walks, it was proved that although lowering the truncation limit decreases the expectation and variance of the time complexity, this is achieved at the cost of an increase in bias of the solution.

In the second part of the paper, we have mainly discussed the intrinsic parallelism in the N-U method and its exploitation in the development of parallel algorithms. It was shown that the N-U method possesses intrinsic parallelism at all the four levels discussed in [4,6]. All the schemes for parallel algorithms given in [4,6] were found to be applicable to the development of parallel algorithms for the N-U method. For an SCA scheme algorithm, it was shown that by using (NK) processors to obtain K primary estimates of each of the N unknowns, the expected time complexity is independent of N. This is a significant result when compared to the known parallel algorithms for solving linear equations (see [21]) for which even the lower bound is about $(2 \log_2 N)$. The parallel algorithms based on other schemes can be developed easily along similar lines. Although in this paper we have considered only the N-U method, we believe that the techniques presented in this chapter are general enough to be applicable to the Monte Carlo solutions using other methods.

REFERENCES

[1] H. AKAIKE, Monte Carlo method applied to the solution of simultaneous linear equations, Ann. Inst. Statist. Math., Tokyo, 7 (1956), pp. 107-113.

[2] V.C. BHAVSAR AND V.V. KANETKAR, A multiple microprocessor system (MMPS) for Monte Carlo solution of partial differential equations, in Advances in Computer Methods for Partial Differential Equations, vol. 2, R. Vichnevetsky, ed., IMACS, New Brunswick, Canada, 1977, pp. 205-213.

[3] V.C. BHAVSAR AND J.R. ISAAC, Some parallel algorithms for Monte Carlo solutions of linear equations, in Proc. 15th Annual Convention of Computer Society of India, Bombay, Feb. 8-11, 1980, vol. III, pp. 3.76-3.81.

[4] V.C. BHAVSAR, Parallel algorithms for Monte Carlo solutions of some linear operator problems, Ph.D. thesis, Department of Electrical Engineering, Indian Institute of Technology, Bombay, Nov. 1981.

[5] V.C. BHAVSAR, VLSI algorithms for Monte Carlo solution of linear equations, presented at the APICS Annual Computer Science Seminar, St. Francis Xavier Univ., Antigonish, N.S., Canada, 4-5, Nov. 1983.

[6] V.C. BHAVSAR AND J.R. ISAAC, Design and analysis of parallel Monte Carlo algorithms, SIAM J. Sci. Stat. Comput., vol. 8, no. 1 (1987), pp. s73-s95.

[7] Yu. A. BLAGOVESHCHENSKII, On the effectiveness of the Monte Carlo method for solution of certain problems, Vorprosy Teorii Mathemat. Mashin, Sb. 2, 1962 (In Russian).

[8] J.H. CURTISS, Monte Carlo methods for the iteration of linear operators, J. Math. and Physics, 32 (1953), 209-232.

[9] J.H. CURTISS, A theoretical comparison of the efficiencies of two classical methods and a Monte Carlo method for computing one component of the solution of a set of linear algebraic equations, in Proc. of Symposium on Monte Carlo methods, (Ed.) H.A. Meyer, John Wiley, New York (1956), pp. 191-233.

[10] G.E. FORSYTHE, AND R.A. LEIBLER, Matrix inversion by a Monte Carlo method, Math. Tabs. Aids Comput., 4 (1950), pp. 127-129. Correction to the article in Math. Tabs. Aids Comput., 15 (1951), p. 55.

[11] J.H. HALTON, Sequential Monte Carlo, in Proc. Camb. Phil. Soc., 58 (1962), pp. 57-78. Revised version: MRC Technical Summary Report No. 816, Univ. of Wisconsin, Madison, Wisconsin, 1967.

[12] J.H. HALTON, Least-squares Monte Carlo methods for solving linear systems of equations, Rep. AMD 388/BNL 9678, Brookhaven National Laboratory, Upton, New York, 1965.

[13] J.H. HALTON, A retrospective and prospective survey of the Monte Carlo method, SIAM Rev., 12 (1970), pp. 1-63.

[14] J.M. HAMMERSLEY AND D.C. HANDSCOMB, Monte Carlo Methods, Methuen, London, 1964.

[15] D. HELLER, A survey of parallel algorithms in numerical linear algebra, SIAM Rev., 20 (1978), pp. 740-777.

[16] J.G. KEMENY, AND J.L. SNELL, Finite Markov chains, Van Nostrand, N.J., 1960.

[17] A.R. MITCHELL, AND D.F. GRIFFITHS, The finite difference method in partial differential equations, Wiley, Chinchester, New York, 1980.

[18] A. OPLER, Monte Carlo matrix calculations with punch card machines, Math. Tabs. Aids Comput., 5 (1951) pp. 115-120.

[19] Yu. A. SHREIDER, Solution of systems of linear algebraic equations by the Monte Carlo method, Voprosy Teorii Matemat. Mashin, Sb.1 (1958) pp. 167-171, in Russian. English translation by C.A.R. Hoare, The Theory of Mathematical Machines, Pergamon Press, Oxford, 1963.

[20] J. SPANIER AND E.M. GELBARD, Monte Carlo principles and neutron transport problems, Addison-Wesley, Reading, MA, 1969.

[21] J. TODD, Experiemtns on the inversion of a 16x16 matrix, Nat. Bur. Stand. Appl. Mat. Ser., 29 (1953), pp. 113-115.

[22] R.S. VARGA, Matrix iterative analysis, Prentice-Hall, Englewood Cliffs, N.J., 1962.

[23] W.R. WASOW, On the mean duration of random walks, J. Res. Nat. Bur. Standards, 46 (1951), pp. 462-471.

[24] W.R. WASOW, On the duration of random walks, Ann. Math. Statist., 22 (1951), pp. 199-216.

[25] P.V. YA, New fast algorithms for matrix operation, SIAM J. Computing, 9 (1980), pp. 321-342.

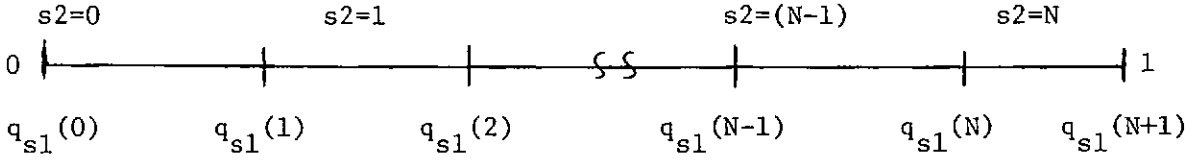

Figure 1. A possible segmentation of the range of ξ, the real unit interval [0, 1].

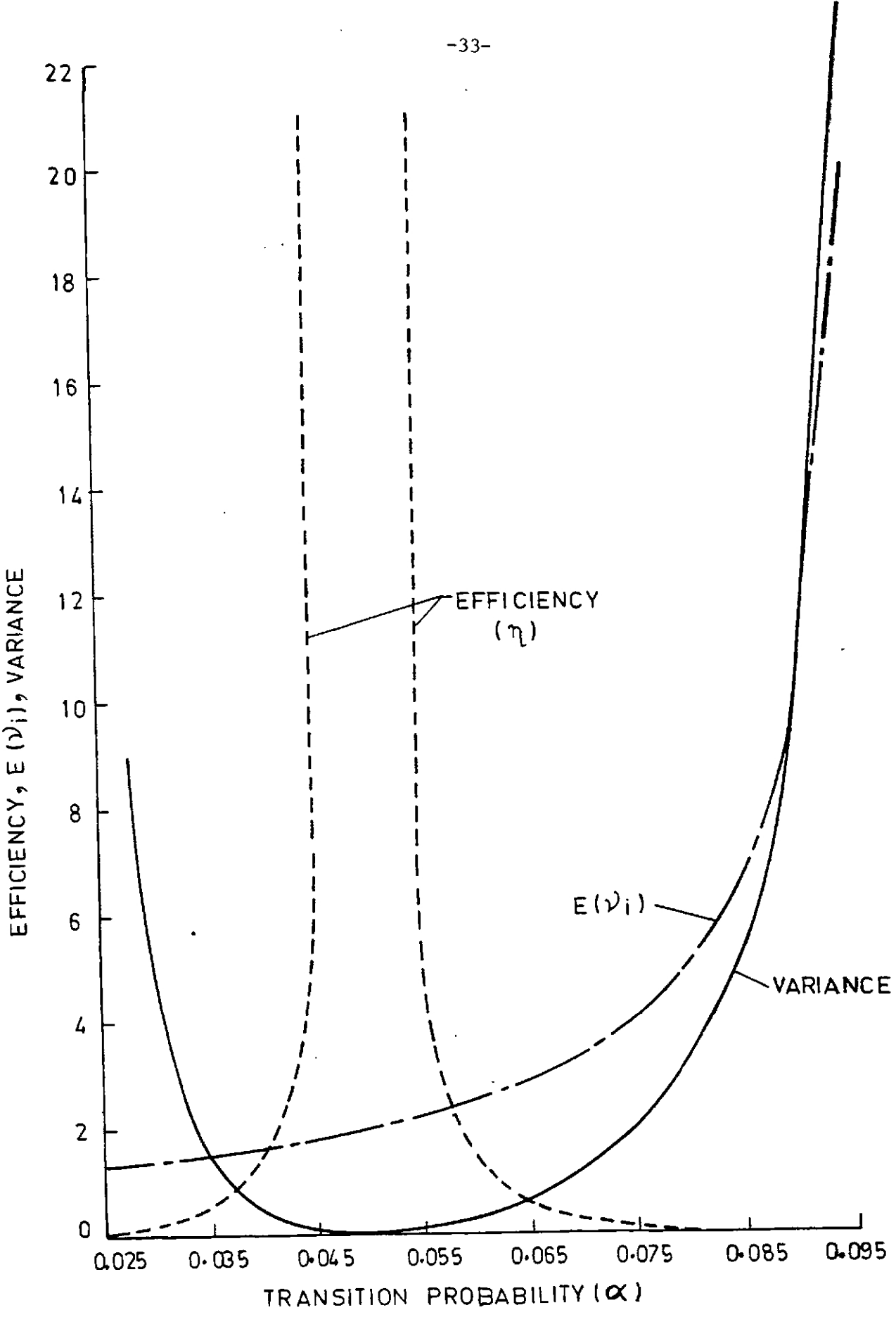

FIG. 2. INFLUENCE OF TRANSITION PROBABILITY ASSIGNMENT ON EXPECTED DURATION OF RANDOM WALKS [E(νi)], SOLUTION VARIANCE, AND EFFICIENCY