THE DIRECT SIMULATION METHOD - AN ALTERNATIVE TO THE MONTE CARLO METHOD

Dietrich Fischer
Courant Institute of Mathematical Sciences
New York University

Abstract

A major problem in system simulation is the handling of random phenomena. The Monte Carlo method, which is usually applied for this purpose, is known to converge rather slowly.

A different method is presented here which deals directly with probability distributions instead of random samples. Arithmetic operations on sampled values of random variables are replaced by transformations of their distributions*. The main advantage of this method is that it is arbitrarily precise. Therefore, long runs for gathering statistics are not necessary. However, difficulties can arise from large memory requirements and program complexity.

A comparison with the Monte Carlo method is given on the basis of two examples, the simulation of a signalized traffic network and of a supermarket.

1. Introduction

We take here the following general view of a simulated system (figure 1): It consists of input variables x_i, intermediate variables y_j and output variables z_k. Operators A, B, C, ... convert input and/or intermediate variables into output or other intermediate variables. These variables are functions of time. Some or all of them may be random processes.

One way to deal with such random phenomena is the Monte Carlo method: Input random variables are sampled from their distributions by means of pseudo-random numbers. Each set of input variables gives a specific result. This is done repeatedly and information about the random nature of output variables is obtained by statistical analysis of the results.

The question arises whether it is possible to directly compute the distributions of the output variables by applying some transformations to the distributions of the input variables. As we shall see, this is indeed possible, at least in some cases. We shall call this the direct simulation method or, for short, the direct method, meaning that the detour through generating random numbers and gathering statistics is skipped.

What do such transformations of probability distributions look like? This depends on the type of arithmetic operation that would be performed on the random variables and on the way in which the probability distributions are represented.

Three different ways of representing a probability distribution are

a) by parameters of a theoretical distribution
b) by frequencies for individual values of a discrete random variable or for classes of values of a continuous random variable
c) by moments

The handling of theoretical distribution

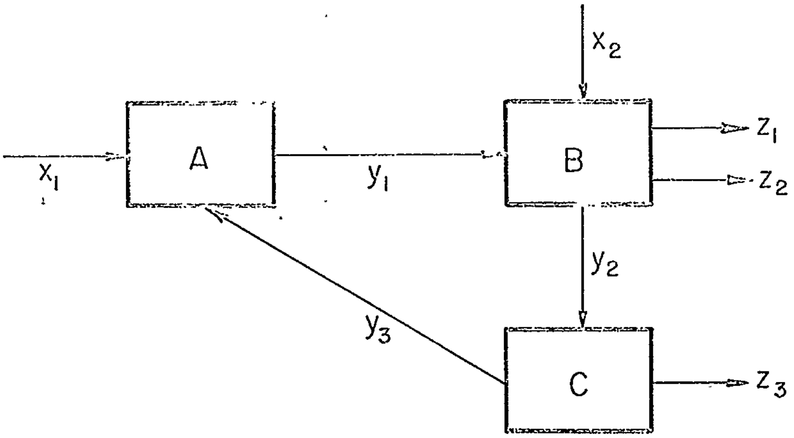

Figure 1: Diagram of a general simulation model

* The basic idea of this approach has been proposed by Professor Walter Nef of the University of Berne, Switzerland.

functions is the field of classical analysis. Here we confine ourselves to distributions that are represented by numerical frequency functions. Some methods for the transformation of distributions that are characterized by their first and second order moments will be described elsewhere[3].

The following examples show a few transformations of probability distributions:

Example 1: The addition of two independent random variables corresponds to the convolution of their distributions. In particular, let X and Y be two independent integer random variables. Then the probability distribution of their sum Z = X + Y is given by

$$P(Z=k) = \sum_{i+j=k} P(X=i)P(Y=j) .$$

As an illustration, consider the following simple numerical example: Let

$$P(X=0) = .8, \quad P(X=1) = .2 \quad \text{and}$$

$$P(Y=3) = .3, \quad P(Y=4) = .4, \quad P(Y=5) = .3 .$$

If X=0, then the sum Z=X+Y can assume the values 3, 4 or 5 with the probabilities

$$\begin{aligned}
P(Z=3|X=0) &= P(X=0)P(Y=3) = (.8)(.3) = .24\\
P(Z=4|X=0) &= P(X=0)P(Y=4) = (.8)(.4) = .32\\
P(Z=5|X=0) &= P(X=0)P(Y=5) = (.8)(.3) = .24
\end{aligned}$$

If X=1, then we have the following possibilities for Z:

$$\begin{aligned}
P(Z=4|X=1) &= P(X=1)P(Y=3) = (.2)(.3) = .06\\
P(Z=5|X=1) &= P(X=1)P(Y=4) = (.2)(.4) = .08\\
P(Z=6|X=1) &= P(X=1)P(Y=5) = (.2)(.3) = .06
\end{aligned}$$

By superposition of these probabilities we obtain the distribution of the sum

$$\begin{aligned}
P(Z=3) &= .24\\
P(Z=4) &= .38\\
P(Z=5) &= .32\\
P(Z=6) &= .06
\end{aligned}$$

If this procedure is applied repeatedly, the range of values of the resulting random variable Z would grow without any limit, if no countermeasure is taken. But extreme values would have only very small probabilities. In order to avoid this, probabilities which are smaller than some given limit EPS are cut off on both sides. EPS = 10^{-6} has been found to be a reasonable value in most applications. The remaining distribution is standardized to 1.

For the distribution of the difference Z=X-Y of two independent integer random variables we find in a similar way

$$P(Z=k) = \sum_{i-j=k} P(X=i)P(Y=j).$$

Among other applications we shall use these operations to add the number of arriving cars to a queue in front of a traffic signal or to subtract the number of cars leaving during a green period.

Example 2: Another operation used in the traffic simulation is the limitation of the range of values of a random variable. Let X be an integer random variable described by frequencies and Y = max(i_o, X). I.e., the integer constant i_o is a lower limit of the random variable Y (figure 3). Then the probability distribution of Y is given by:

$$P(Y=i_o) = \sum_{i\le i_o} P(X=i)$$

$$P(Y=i) = P(X=i) \quad \text{for } i>i_o$$

A similar procedure can be used if i_o itself is a random variable.

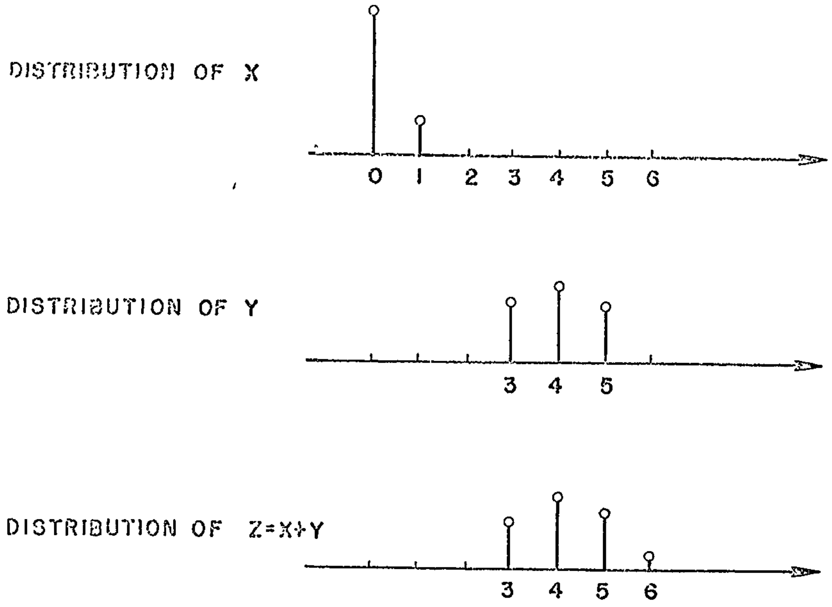

Figure 2: Convolution

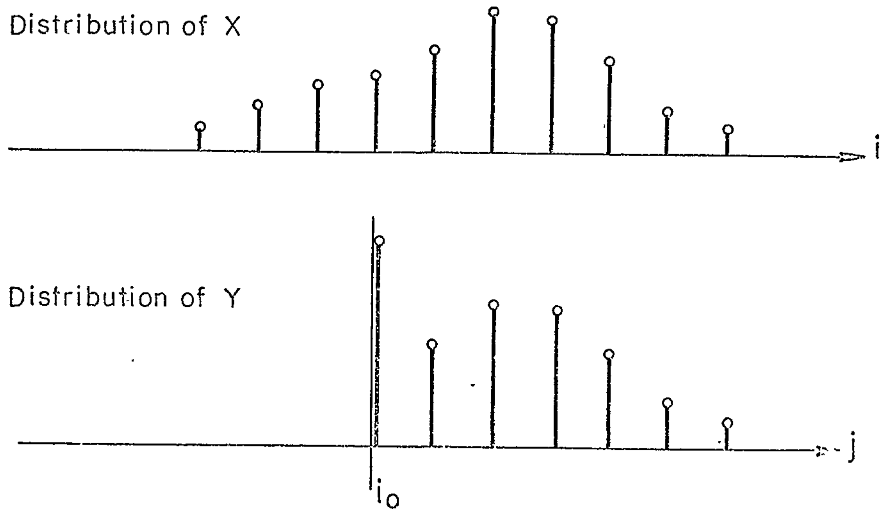

Figure 3: Limitation of the range of values of a random variable ($Y = \max(X, i_o)$)

Example 3: The number of cars arriving at a fork in a road during a given time interval is a random variable X (figure 4). Each car turns left with probability p and right with probability 1-p. What is the distribution of Y, the number of cars turning left? This is a compound distribution given by

$$P(Y=j) = \sum_i P(Y=j/X=i)P(X=i).$$

If X has a fixed value i, then Y has the binomial distribution

$$P(Y=j/X=i) = B_j(i,p) = \binom{i}{j}p^j(1-p)^{i-j} \quad \text{for } j = 0,1,\ldots,i$$

(Model: i independent trials with probability of success equal to p.) Thus the distribution of Y is given by

$$P(Y=j) = \sum_{i\geqslant j} B_j(i,p)P(X=i) .$$

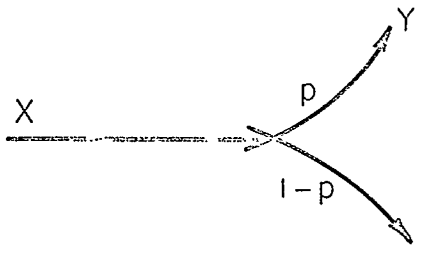

Figure 4: Fork in a road. X = number of car arrivals, Y = number of cars turning left.

So far we have only considered independent random variables. Two dependent random variables can be characterized by their joint distribution (figure 5). To store such a joint distribution by frequencies in a computer, we need an array with two subscripts.

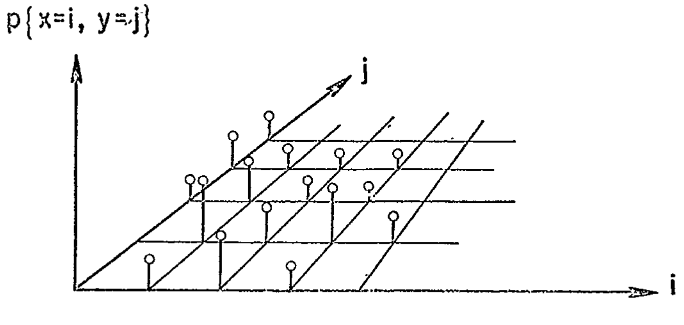

Figure 5: Distribution of a two-dimensional integer random vector (X,Y)

If the two variables X and Y were independent, it would be sufficient to store the marginal distributions

$$P_X(X=i) = \sum_j P(X=i,Y=j) \quad \text{and}$$

$$P_Y(Y=j) = \sum_i P(X=i,Y=j).$$

In that case one would obtain any probability by simple multiplication

$$P(X=i,Y=j) = P_X(X=i)P_Y(Y=j).$$

In the general case, however, this is not correct.

To store the joint distribution of three random variables we need an array with three subscripts, etc. For reasons of clearness we confine ourselves to two dependent random variables. Similar methods can also be applied to three and more dependent random variables. However, not only memory space grows exponentially with the dimension of the arrays used, but also the computer time to handle such amounts of data. This will limit the applicability of this method to relatively few dependent random variables.

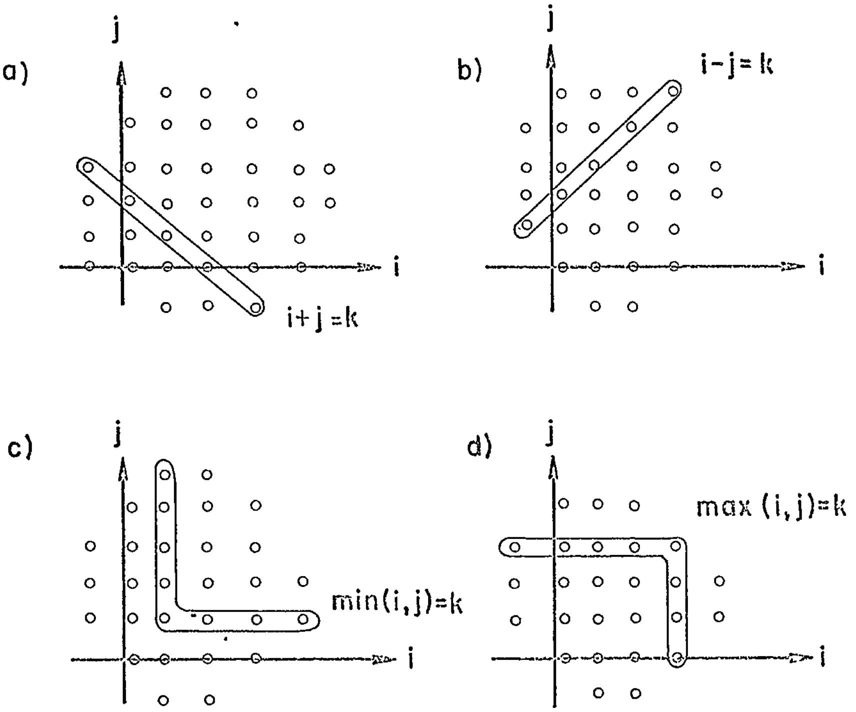

Figure 6: Sum (a), difference (b), minimum (c) and maximum (d) of two dependent random variables. (Probabilities different from zero are marked by dots.)

We now consider some basic operations with the joint distribution of two dependent integer random variables.

Example 4: In analogy to convolution, the sum of two dependent integer random variables is given by the formula

$$P(X+Y=k) = \sum_{\substack{i,j\\ i+j=k}} P(X=i,Y=j) \;.$$

Similarly, for the difference we obtain (figure 6b)

$$P(X-Y=k) = \sum_{\substack{i,j\\ i-j=k}} P(X=i,Y=j) \;.$$

Minimum and maximum of X and Y are given by (figure 6c and 6d)

$$P(\max(X,Y)=k) = \sum_{\substack{i,j\\ \max(i,j)=k}} P(X=i,Y=j)$$

and

$$P(\min(X,Y)=k) = \sum_{\substack{i,j\\ \min(i,j)=k}} P(X=i,Y=j) .$$

Often we are also interested in the marginal distributions

$$P_X(X=i) = \sum_{j} P(X=i,Y=j)$$

and

$$P_Y(Y=j) = \sum_{i} P(X=i,Y=j) \;.$$

By means of the marginal distributions we can immediately obtain the conditional distributions. The conditional distribution of Y given X=i is

$$P(Y=j/X=i) = \frac{P(X=i,Y=j)}{P_X(X=i)} \;.$$

Of course, only that region of X and Y where $P_X(X=i) \neq 0$ is of interest.

In this section we have prepared some of the tools that we are now going to apply to two examples.

2. Simulation of Signalized Road Traffic

Let us first consider the very simplest case of a one-lane one-way street with an isolated traffic signal. We assume that the average car arrival rate and the signal setting are known. The problem posed is to find the distribution of the queue length in front of the signal at the end of a green period.

Time is divided into unit intervals of length $\Delta t \approx 2$ seconds. This interval has been observed to be the approximate minimum time delay between two successive cars. We assume that in each such unit interval a car can arrive with probability p. During a cycle of length $T_c = c \cdot \Delta t$ the number Δl of car arrivals has a binomial distribution in this model:

$$P(\Delta l=k) = B_k(c,p) = \binom{c}{k} p^k (1-p)^{c-k} \;.$$

If we denote the length of the queue at the end of cycle number i by l_i and the number of arrivals during cycle i by Δl_i, we obtain

$$l_i = \max(l_{i-1}+\Delta l_i - g, 0) \;.$$

I.e., during the green time $T_g = g \cdot \Delta t$, a total of g cars leave the queue, provided there are g or more cars. If there are less, the queue becomes completely empty.

Let us compare the Monte Carlo method and the direct simulation method using this example.

At the beginning of the simulation the queue is assumed to be empty. For the Monte Carlo method we generate a sequence of Δl_i with binomial distribution $P(\Delta l=k)$ and compute the corresponding sequence of queue lengths according to the formula

$$l_i = \max(l_{i-1}+\Delta l_i-g,\ 0)\ .$$

The total number of iterations is fixed in advance. At regular intervals, mean and standard deviation of all queue lengths l_i generated so far are printed out.

In the direct method the queue length has initially the distribution

$$P(l_0=j) = \begin{cases} 1 & \text{if } j=0 \\ 0 & \text{otherwise}\ . \end{cases}$$

For each cycle, the following three operations have to be performed:

1. Convolve the distribution $P(l_{i-1}=j)$ with the binomial distribution $P(\Delta l_i=k)$ (see section 1, example 1). The result is the distribution of $l_{i-1}+\Delta l_i$.
2. Shift this distribution by g units to the left. This gives the distribution of $l_{i-1}+\Delta l_i-g$.
3. Limit the distribution obtained in step 2 by zero from below (section 1, example 2). This leads to the distribution of $l_i = \max(l_{i-1}+\Delta l_i-g,\ 0)$.

At regular intervals, mean and standard deviation are computed from the distribution $P(l_i=j)$. The simulation is terminated as soon as the absolute difference between two successive average queue lengths does not decrease any more.

A similar algorithm was used by Bottger[1] (1966) to study the effect of alternate policies at an isolated traffic actuated signal.

Results have been computed by both methods for the parameters c=20 (cycle time in units), g=10 (green time in units) and for three different traffic volumes p (in cars per time unit). Mean value (μ) and standard deviation (σ) of the queue length at the end of green time, as a function of the number of iterations (N) are given in tables 1 and 2.

For the Monte Carlo method (table 1) each of the three examples took 109 seconds of computer time (on a Bull - General Electric Gamma 30S computer at the University of Berne, which has a multiplication time of about 0.4 milliseconds). How much time would be required to reach a relative accuracy of 1% at a confidence level of 95% ? If we make the favorable assumption that the queue lengths in successive cycles are independent, this would mean

$$1.960\sigma/\sqrt{N'} = .01\mu \quad \text{or} \quad N = (196.0\sigma/\mu)^2.$$

Substituting the results after 5000 iterations as estimates for μ and σ we find the required number of iterations and computer time shown in table 3. Actually, the time needed is even greater, because successive queue lengths are positively correlated, and this increases the variance of a sample average (Fishman[4], 1968).

Table 1: Results of Monte Carlo method. Mean (μ) and standard deviation (σ) of the queue length as a function of the number of iterations (N) for three different traffic volumes p (in cars per time unit)

a) p=.3			b) p=.4			c) p=.45		
N	μ	σ	N	μ	σ	N	μ	σ
500	.0380	.2378	500	.3960	.9355	500	1.5020	2.2632
1000	.0280	.1980	1000	.4730	1.1581	1000	1.6240	2.4675
1500	.0280	.1946	1500	.4467	1.1209	1500	1.4713	2.3056
2000	.0255	.1840	2000	.4015	1.0336	2000	1.4440	2.2068
2500	.0232	.1774	2500	.4012	1.0250	2500	1.4960	2.3655
3000	.0233	.1774	3000	.3753	.9800	3000	1.4027	2.2629
3500	.0240	.1805	3500	.3826	.9817	3500	1.4089	2.2860
4000	.0225	.1746	4000	.3730	.9583	4000	1.4390	2.2963
4500	.0220	.1718	4500	.3691	.9463	4500	1.4256	2.2691
5000	.0222	.1735	5000	.3642	.9383	4500	1.3632	2.2160

Table 2: Results of direct simulation method (same notation as in table 1)

a) p=.3			b) p=.4			c) p=.45		
N	μ	σ	N	μ	σ	N	μ	σ
1	.0239	.2017	2	.2865	.7923	10	1.2739	2.0738
2	.0257	.2115	4	.3333	.8915	20	1.3752	2.2496
3	.0259	.2126	6	.3454	.9200	30	1.3949	2.2895
4	.0259	.2128	8	.3490	.9293	40	1.3996	2.2999
			10	.3501	.9325	50	1.4009	2.3028
			12	.3505	.9337	60	1.4012	2.3037
			14	.3507	.9341	70	1.4013	2.3040
			16	.3507	.9343	80	1.4013	2.3040

Table 3: Number of iterations and computer time required to obtain a relative accuracy of 1% at a 95% confidence level by the Monte Carlo method

	p=.3	p=.4	p=.45
number of iterations	2 340 000	254 000	101 000
computer time	14 hours 10 min	1 hour 32 min	37 min

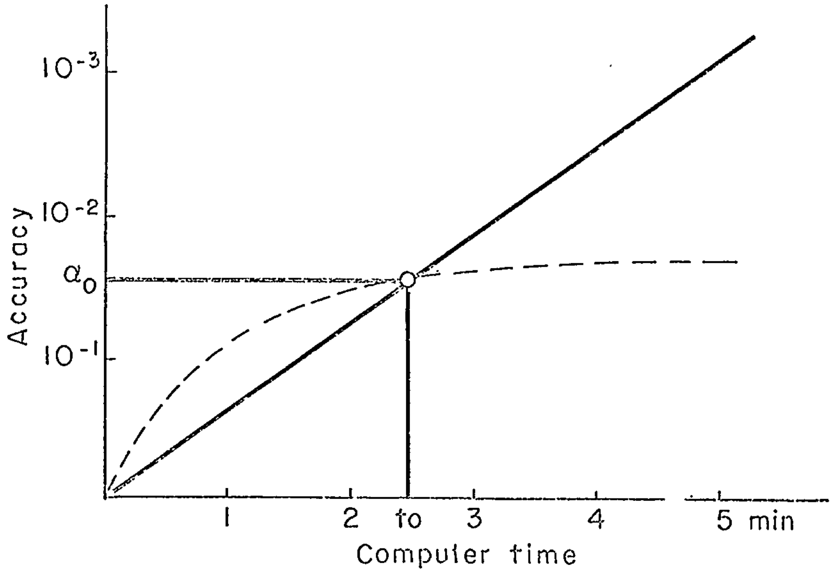

Figure 7: Accuracy versus computer time for the Monte Carlo method (---) and the direct simulation method ( ).

For the direct method (table 2) computer time was 2 sec in case a), 19 sec in case b) and 161 sec in case c). Here no statistical fluctuations occur. In principle, the error can be kept arbitrarily small in the first step, if a sufficient number of digits is taken into account. However, we can observe another effect which could be practically neglected in the Monte Carlo method: The assumption that initially the queue is empty has an effect over several cycles. Only gradually a steady state is reached. This is a disadvantage if one is only interested in the equilibrium state. But it permits a precise study of the system's dynamic response to initial conditions. The Monte Carlo method is not very efficient for this type of investigations.

As in most physical systems, the deviation from the equilibrium state decreases about exponentially with time. If we denote simulated time by T, then the error decreases as $e^{-\alpha T}$ for the direct method, for some α. For the Monte Carlo method the error decreases according to the well-known formula $1/\sqrt{T}$'. In the direct method a single iteration takes longer, but the convergence behavior is better than that of the Monte Carlo method. Which of the two methods is preferable depends on the accuracy desired and on the computer time available (figure 7). If the accuracy desired is less than a_0 or the computer time available less than t_0, then the Monte Carlo method is preferable. Otherwise, the direct method proves more efficient. The precise shape of the two curves in figure 7 depends on the particular problem under investigation. The point (a_0, t_0) must be estimated for each simulated system individually.

Let us now consider a more general traffic model. Instead of assuming a binomial distribution for the car arrivals at a signal, the arrival distribution could be generated by the output of one or more other signals. This permits to simulate traffic flow in a network of arbitrary size and shape. Such a program has been implemented2. As input the user has to specify the relevant geometry of the network, the signal plan, vehicle speeds and traffic volumes. The output consists of distributions and their graphical representations for the waiting time at each signal and for the queue lengths as functions of time. The program is written in FORTRAN and consists of about 1500 instructions. On a CDC 1604 with 32K memory it can handle networks with up to 200 signals, 400 links and a maximum queue length of 50 vehicles.

It is beyond the scope of this paper to describe this program in more detail. Rather we would like to discuss some of the results obtained.

Several actual networks have been analyzed under various traffic volumes and signal settings. Two more systematic investigations were the following:

Optimum Cycle Time as a Function of Traffic Volume

If traffic is light, the average waiting time of a car is approximately proportional to the cycle time. Yet the cycle time should not be chosen too short because the constant amber period which is lost in each cycle becomes more and more important. A study has been based on the following model: Consider two intersecting traffic streams with equal volume of p cars per time unit. An amber period of two time units is lost with each switching of the signals. The variable cycle time is $T_c = c \cdot \Delta t$. Each stream is given c/2 - 2 units of green time per cycle.

In figure 8, the average waiting time is displayed as a function of the cycle time c for various traffic volumes p. As expected, the larger the traffic volume p, the larger is the optimum cycle time c which minimizes the average waiting time. If the traffic volume is subject to heavy fluctuations, then it is better to choose a larger cycle time than the one corresponding to the average volume; for the increase in waiting

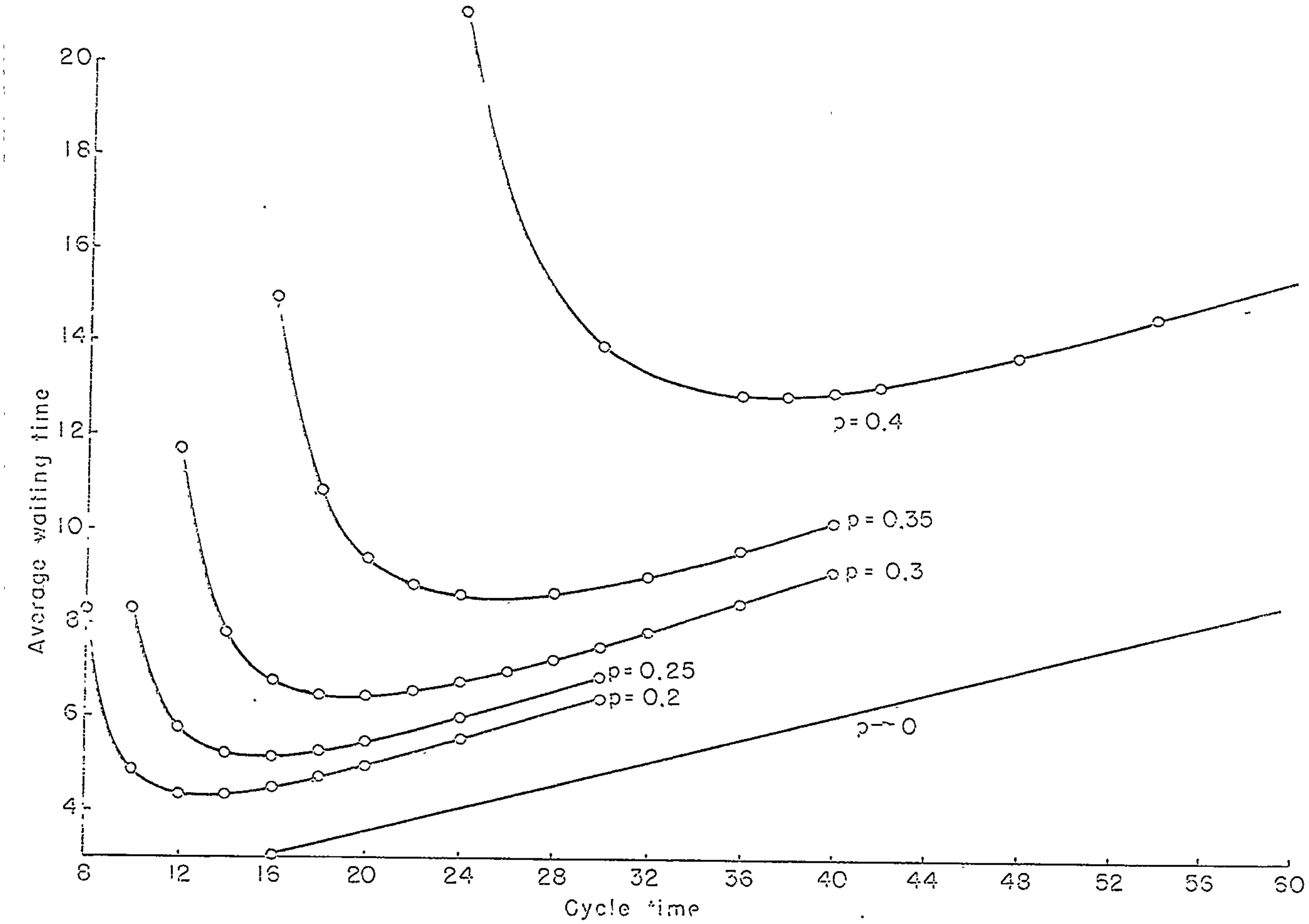

Figure 8: Average waiting time at an isolated traffic signal as a function of cycle time c (in time units) for various traffic volumes p (in cars per time unit). Green time is c/2 – 2 units per cycle.

time is much greater if the cycle time is too short than if it is too long.

Notice that, unlike results of a Monte Carlo simulation, the observed points lie exactly on a smooth curve connecting them.

Traffic behavior through three signals

W. D. Wätjen[6] (1965) has given some interesting results on how the offsets at successive signals affect waiting time. He considered the following model: A one-way street has three successive signals at equal distances. Travel time from one signal to the next is 20 seconds. Arriving traffic is Poisson distributed with an average volume of 686 cars per hour. When the signal is green, cars on the average leave the queue every 2 seconds. This value of 2 sec is not a constant but is normally distributed with a standard deviation of .5 sec. All three signals have a cycle time of 60 sec and 30 sec of green time. Offsets between the first two signals (o_{12}) and between the last two (o_{23}) vary from 0 to 50 sec in steps of 10 sec. This corresponds to 36 different combinations of offsets.

The simulation language used by Wätjen was GPSS. We duplicated this investigation in order to test our program and compare it with a Monte Carlo simulation. Although the model assumptions were slightly different, the results showed a good agreement.

For the waiting time at the first signal, Wätjen obtained the mean value 13.02 sec (as an average of 4825 simulated cars) and a standard deviation of 10.35 sec. Our program gave the mean value 13.54 sec and a standard deviation of 10.48 sec.

Table 4 shows waiting time at signal 2 as a function of the offset o_{12} between signals 1 and 2. In Wätjens results the waiting time does not vanish for an offset of 20 sec (which corresponds to an ideal green wave) because the time interval between successive cars is not a constant but a random variable.

An important result of Wätjens paper is the fact that waiting time does not only

Table 4: Mean (μ) and standard deviation (σ) of waiting time at signal 2 as a function of the offset o_{12} between signals 1 and 2

offset o_{12} (sec)	waiting time (sec)			
	Monte Carlo method (GPSS)		direct simulation method	
	μ	σ	μ	σ
0	14.26	13.19	16.09	14.06
10	6.72	11.19	6.43	11.91
20	1.15	3.42	0.00	0.00
30	8.74	4.05	8.84	2.11
40	19.44	4.01	18.84	2.13
50	28.62	3.39	28.84	2.13

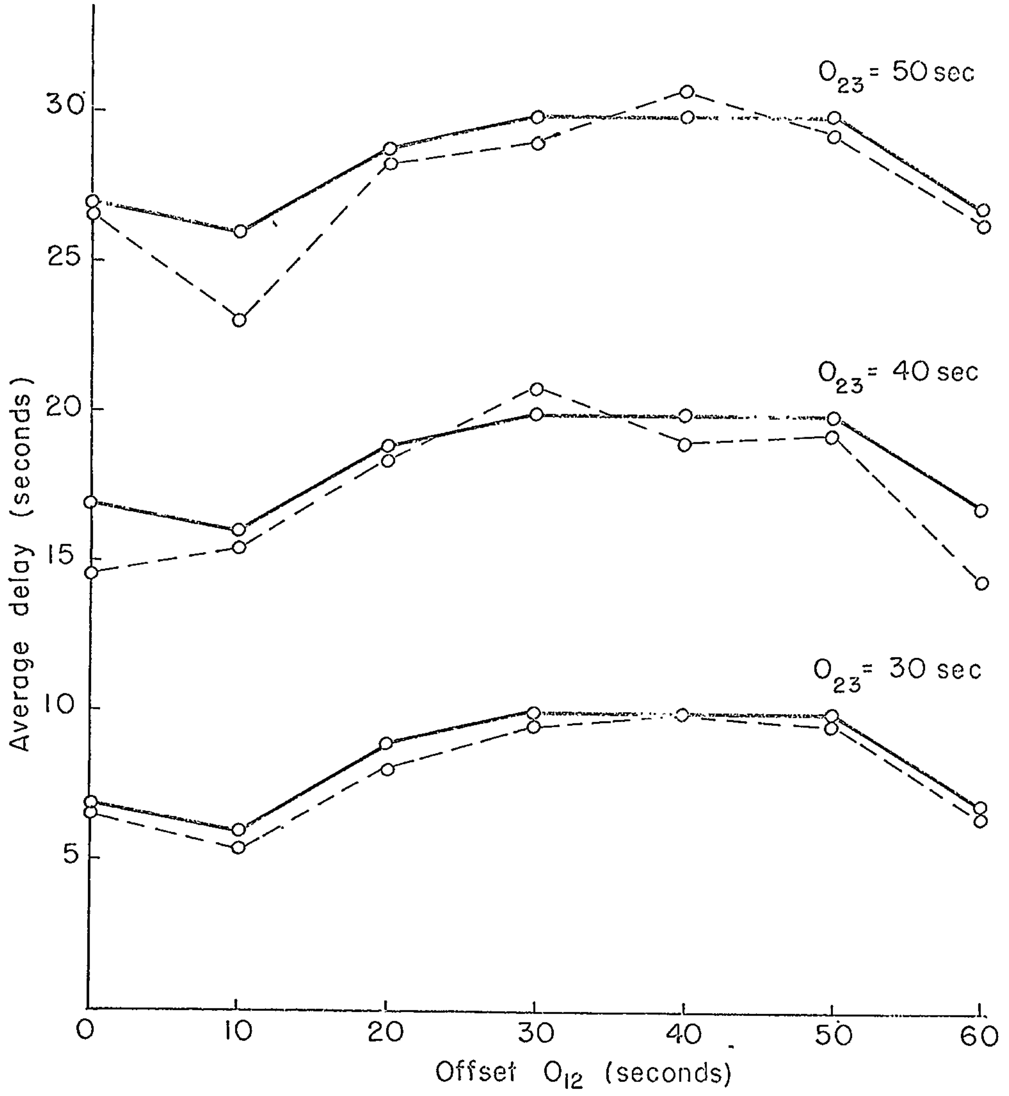

Figure 9: Traffic behavior through three signals. Average delay at signal 3 (in seconds per vehicle) as a function of the offset o_{12}, for 3 different offsets o_{23}. Monte Carlo method (---) and direct simulation method (——).

depend on the offset between a signal and its predecessor, but also on the offset between previous signals. In figure 9 the average waiting time at signal number 3 is shown as a function of the offset o_{12} between signals 1 and 2 (for three different offsets o_{23}).

Some of the oscillations in the curves obtained by GPSS are due to random effects and do not reflect any law. For a more thorough discussion of the results we refer to Wätjen.

The ultimate purpose of this traffic simulation program is to find the optimal signal plan for a whole network for any given traffic situation. As it stands now, the user has to specify the signal plan in advance and then select the best among a few alternatives, in view of the results.

A next step would be to combine this program with an optimizing algorithm. So far the program is too slow for this purpose. The simulation of a system takes several minutes, depending on the number of signals and other parameters (e.g. 18 minutes for a system with 11 signals on a Bull - General Electric Gamma 30S computer with a multiplication time of 0.4 milliseconds). But if the speed of computers continues to increase at the current rate such an optimization may soon become feasible. Convolution, the slowest part in the program, could make extensive use of parallel processing.

3. A Simple Supermarket Model

In the last section we dealt with a model that contains only independent variables. Let us now consider an example that contains a pair of mutually dependent variables. We have chosen a simplified version of the supermarket model that is discussed in Gordon[5] (1969) on pages 221-227.

Customers of the supermarket are obliged to take a basket before they begin to shop. There is a limited number of baskets and, if no basket is available when they arrive, customers leave without shopping. If they get a basket, customers shop and then go to the checkout counter. If the counter is occupied, they join a queue. After checking out, they return the baskets and leave the supermarket.

The arrival times of customers are Poisson distributed (i.e., the interarrival times are exponentially distributed). For the shopping time and the service time of customers at the checkout counter we also assume an exponential distribution, in deviation from Gordons model. There is only one checkout counter. The number of baskets (5) is kept small in order to save computer time for the matrix operations in the direct method. The average interarrival time of customers is 240 sec, the average shopping time 600 sec and the average checkout time 180 sec. With these parameters the model is completely specified.

We are interested in the distribution of the number of customers shopping (SHP), the number of customers checking out (CHK) and the total number in the store (TOTAL). We are interested in the steady state distributions after initial oscillations have balanced out.

Although the relation TOTAL = SHP+CHK holds, the distribution of TOTAL is not the convolution of the distributions of SHP and CHK. SHP and CHK are strongly dependent of each other. Their sum is bounded from above by the number of baskets (figure 10).

In order to compare the direct method with the Monte Carlo method, we have also simulated this model in GPSS. First 100 customers are simulated without results being printed out. In this way the system goes from its initial empty state into a statistical equilibrium. Then statistics are gathered for running times of 25, 100, 400 and 1600 customers. From an increase of the running time by a factor of 4 we can expect the accuracy to roughly double. Results are given in table 6a. Computer time for this example was 1.96 minutes on an IBM 360/50 at New York University.

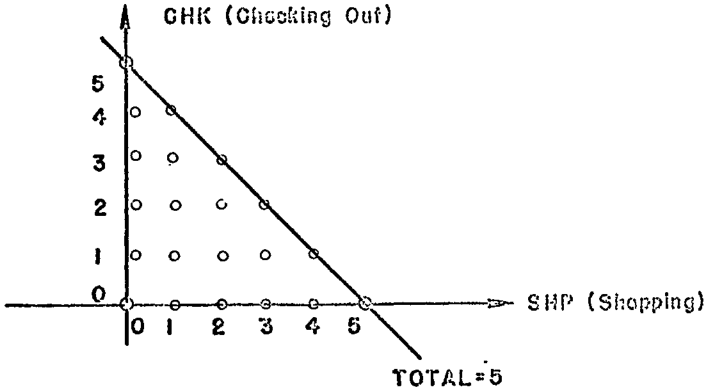

Figure 10: Number of customers shopping (SHP), checking out (CHK) and sum (TOTAL). The point (3,1) e.g. represents the probability that 3 customers are shopping and 1 is checking out.

For the direct method we have one pair of dependent variables, the number of customers shopping (SHP) and the number of customers checking out (CHK). Their distribution is represented in a matrix. At the beginning of the simulation this distribution has the form

P(SHP=0, CHK=0) = 1

P(SHP=i, CHK=j) = 0 for i≠0 or j≠0.

After a certain time, all the points within the feasible region

SHP≥0
CHK≥0
SHP+CHK≤5

are assigned a positive probability (figure 10).

Table 5: Variable names and values of parameters

ARR	number of customers arriving in DT
CHK	number of customers at the checkout counter
DELTA	bound of convergence
EPS	lower limit for probabilities considered in the calculation (10-6)
N	number of baskets (5)
SHP	number of customers shopping
SRV	number of customers that can be served in DT
TARR	average interarrival time (240 sec)
TOTAL	total number of customers (=SHP+CHK)
TSHP	average shopping time (600 sec)
TSRV	average service time at the checkout counter (180 sec)

As mentioned before, time is advanced in unit intervals. During each iteration that corresponds to such an interval, the following three steps have to be carried out. (A summary of variable names used is given in table 5.)

1. Add the number of customers arriving (ARR) to the number of customers shopping (SHP). Take into account that the total number of customers does not exceed the number of baskets (N).

Table 6: Mean values and distributions of the number of customers shopping (SHP), checking out (CHK) and their sum (TOTAL)

Mean values

a) Monte Carlo method (GPSS)

customers simulated	SHP	CHK	TOTAL
25	2.174	.756	2.931
100	1.811	1.141	2.952
400	1.965	1.117	3.082
1600	2.030	1.058	3.089

b) direct simulation method

number of iterations	SHP	CHK	TOTAL
1	.226	.017	.243
2	.431	.045	.476
3	.616	.080	.696
4	.783	.120	.903
5	.934	.162	1.096
10	1.478	.388	1.866
15	1.760	.596	2.355
20	1.886	.759	2.645
25	1.932	.875	2.807
100	1.984	1.076	3.060
178	1.988	1.080	3.068

Distribution of the number of customers shopping (SHP)

a) Monte Carlo method

number of customers simulated	probability that SHP =					
	0	1	2	3	4	5
25	.0734	.2505	.2321	.3290	.1010	.0137
100	.1471	.3373	.2301	.1739	.0648	.0466
400	.1120	.2610	.3270	.1777	.0934	.0285
1600	.1109	.2543	.2894	.2119	.1048	.0284

b) direct method (after 178 iterations)

.1146	.2634	.2927	.2050	.0966	.0276

Distribution of the number of customers checking out (CHK)

a) Monte Carlo method

number of customers simulated	probability that CHK =					
	0	1	2	3	4	5
25	.5652	.2348	.1201	.0523	.0129	.0143
100	.4062	.2680	.1619	.1079	.0540	.0018
400	.3897	.2828	.1913	.0996	.0288	.0075
1600	.4246	.2724	.1684	.0946	.0338	.0060

b) direct method (after 178 iterations)

.4130	.2790	.1733	.0915	.0357	.0075

Distribution of TOTAL = SHP + CHK

a) Monte Carlo method

number of customers simulated	probability that TOTAL =					
	0	1	2	3	4	5
25	.0250	.1739	.1218	.3324	.2174	.1292
100	.0737	.1041	.2109	.2087	.2116	.1907
400	.0309	.1093	.2138	.2328	.2176	.1952
1600	.0342	.1142	.1987	.2211	.2436	.1878

b) direct method (after 178 iterations)

.0368	.1170	.1959	.2332	.2253	.1918

2. Part of the customers shopping go to the checkout counter. The total number of customers does not change.

3. The number of customers at the checkout counter (CHK) is decreased by the number of customers served (SRV) in the interval DT. Observe that CHK never drops below 0.

The following path in figure 11 corresponds to a possible sequence of these three steps:

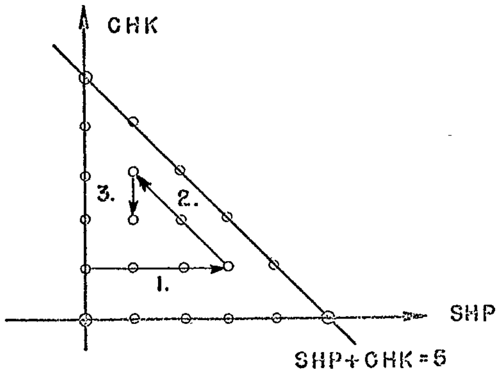

Figure 11: A possible sequence of states in the system

We now describe what operations correspond to each of those three steps.

1. Addition of new customers

The distribution of the number of customers arriving in an interval DT, P(ARR=k), is a Poisson distribution with the parameter λ = DT/TARR. For each possible value CHK = 0, 1, ..., N we compute the conditional distribution P(SHP=i/CHK=j) and convolve it with the distribution P(ARR=k). Since the total number of customers does not exceed N, the result of this convolution has to be bounded from above by the maximum value N-j. After having done these two operations for all values of j, we can return to the joint distribution by multiplying the conditional distributions by the marginal distribution of CHK:

P(SHP=i,CHK=j) = P(SHP=i/CHK=j)P(CHK=j) .

This step corresponds to a horizontal shifting of probabilities to the right in figure 10.

2. Transfer of customers from shopping to checkout

The shopping time of customers is exponentially distributed, with the average TSHP. This means that after the time DT has elapsed a customer is still shopping with probability p = exp(-DT/TSHP) and has transferred to the checkout with probability 1-p. The total number of customers does not change in this step. By a coordinate transformation from (SHP,CHK) to (SHP,TOTAL) we obtain

P(SHP=i,TOTAL=j) = P(SHP=i,CHK=j-i).

Then we can get the conditional distributions P(SHP=i/TOTAL=j) for j = 0, 1, ..., N. If a fixed number SHP=i customers are shopping, then the number of those who are still shopping after an interval DT has a binomial distribution with the parameters p and i. If the number of customers shopping is not fixed but has a certain distribution, then we get a compound distribution for the number of customers still shopping after the interval DT. As in step 1, we have to return to the joint distribution at the end.

Step 2 corresponds to a shifting of probabilities from the lower right to the upper left in figure 10.

3. Subtraction of customers served

The distribution of the number of customers that can be served in the interval DT, P(SRV=k), is a Poisson distribution with the parameter μ = DT/TSRV. Customers served are subtracted from the number of customers checking out (CHK). The number of customers shopping does not change. Similarly as in the first step we compute the conditional distribution P(CHK=j/SHP=i). Then we convolve this distribution with P(SRV=k) with negative sign, to subtract customers served. Since CHK does not drop below zero, we have to limit the resulting distribution by zero from below. Finally we return to the joint distribution.

This third step corresponds to shifting probabilities vertically downwards in figure 10.

These three steps are repeated for a given time interval DT until the distributions converge to their asymptotic shape. The following test for convergence has been used: The procedure is terminated as soon as the absolute differences of two successive mean values of SHP, CHK and TOTAL drop below a given bound DELTA or do not decrease any more. As a measure of security, a maximum number of iterations has also been prescribed.

A particular problem arises in selecting the best time interval DT. Although the state of the system (the vector (SHP,CHK)) is a discrete function of time, the underlying probability distribution that we consider here is a continuous function of time. As with the numerical integration of differential equations, we can expect that the smaller the time interval DT is, the more accurate are the results. On the other hand, the smaller DT is, the larger is the number of iterations required to bring the system from its initial state into a stable equilibrium. In order to save computer time without losing accuracy, the following compromise was chosen: First the simulation was started with a large time interval in order to bring the system from the initial state into an approximate equilibrium in as few steps as possible. Then the time interval was gradually decreased to improve upon the accuracy.

Computer time for this example was 4.5 sec on a CDC 6600 at New York University. The program is written in FORTRAN. It consists of a short main program, which describes the system, and 26 subroutines with a total of about 1000 instructions. The subroutines are not related to this specific example but can be used for the solution of other problems as well.

The example considered in this section is very simple and hardly of any practical value. But by extending these basic methods one can also attack problems that are considerably more complex.

4. Conclusions

We have seen that in some instances the direct simulation method can give a precise solution to a problem within reasonable computer time. The time required by the Monte Carlo method to produce results of comparable quality would be considerably longer.

A possible disadvantage of the direct method is its excessive consumption of computer memory. A simulated system can usually assume a very large, if not infinite, number of possible states. In a Monte Carlo simulation, the system will go only through a limited random selection of these states. In the direct method we theoretically consider the set of all states at the same time and assign a probability to each one. Even if we combine individual states into classes, the number of classes may still be too large. Sometimes we may be able to factor a system into independent subsystems and bring it down to a manageable size. But in other examples this may not be possible without the loss of essential information. There will always be large systems with complex interdependence which can be investigated only by Monte Carlo methods.

Another difficulty of the direct method is that it requires rather voluminous programs. It is clear that an algorithm which transforms probability distributions is more complex than a simple operation with random numbers. But this should not be an obstacle to the use of this method. These algorithms need be programmed only once and can then be applied to many different problems.

This is a preliminary report and further work is planned. The methods presented in this paper are far from being exhaustive, but we hope it will encourage similar investigations in this direction.

Acknowledgements

I wish to thank Professor Walter Nef of the University of Berne who has kindly suggested this investigation to me and given me valuable advice. I also acknowledge partial support of this work by the Hasler Foundation, Berne, and free computer time made available by Polytechnic Institute of Brooklyn, New York, and New York University.

References

1. Böttger, R. Verkehrsabhaengige Signalregelung bei instationärem Fahrzeugfluss. Zeitschrift für angewandte Mathematik und Mechanik, Bd. 46 (1966), T92-T94.

2. Fischer, D. Simulation eines Verkehrsnetzes. University of Berne, 1968 (unpublished).

3. Fischer, D. Zur Behandlung abhangiger Variablen in der direkten Methode. In: Nef and Bauknecht (Ed.), Digitale Simulation. Lecture Notes in Operations Research and Mathematical Systems, Springer Verlag (scheduled to appear soon).

4. Fishman, George S. Estimating Reliability in Simulation Experiments. Digest of the Second Conference on Applications of Simulation, New York (Dec. 2-4, 1968), 6-10.

5. Gordon, Geoffrey. System Simulation. Prentice-Hall, Englewood Cliffs, New Jersey, 1969.

6. Wätjen, W. D. Computer simulation of traffic behaviour through three signals. Traffic Engineering and Control (February 1965), 623-626.