APPLICATION OF COMPUTER SIMULATION

TECHNIQUES FOR SOME STOCHASTIC MODELS

BY

Gyorgy NEMETH

Dissertation submitted to the Ahmadu Bello University for the degree of

Doctor of Philosophy

Department of Mathematics
Ahmadu Bello University
Zaria, Nigeria.

April,1975

ABSTRACT

Computer simulation is carried out for some Waiting Line Models in this thesis.Before the models are described a short mathematical summary of the most fundamental tool for stochastic simulation on a computer is given, namely the problem of generating pseudo-random numbers and stochastic variates.In this summary the author refers to the "Mixed and Multiplicative Congruential Methods",and lists some basic definitions and theorems by which all statementsand conclusions can be proved.However,the purpose of this dissertation is not to deal with special verifications of these theorems (so proofs are omitted),but to make use of them as important aids in the simulation of the models.

Computer subroutines,based on a strict mathematical background(developed by Neumann [50],Lehmer [37],Greenberger [22],Hutchison [30],Coveyou[12], Good [18 and 19],Juncosa [32],Theichroew [69] etc.) have been employed in the development of both the random number generators and the stochastic variate generators (for frequently used probability distributions such as uniform,exponential and Poisson).These subroutines are included in the introductory part of the paper.

The author proceeds to simulate six stochastic models;the analytical solution of the first five are known,but it is often useful to make an approximation by simulation even when other solutions are available.In these cases the opportunity presents itself to check the validity of simulation and to compare the results obtained by both methods.It turns out that there is a very close agreement between the results got by different ways.

In the formulation of computer models,a flow-chart has been used in each case,which the author considers to be a good,precise and abbreviated language so as to avoid long worded descriptions.Flow-chart language has the additional advantage that it is not orientated to any particular computer.Hence in this paper models appear in flow-chart form.-The computer model is,in fact,the simulation i.e.the set of statements that model stochastic systems.

Annexed to each model a diagram giving a graphic representation of the waiting line in question showing the changing values of some important variables during a short period of the simulation.The construction of the included diagrams is fairly self-evident and makes it possible to follow the flow of events.

In the actual calculations made,it is supposed that arrivals follow Poisson's law and service distributions are governed by exponential disribution.However,the models given in flow-chart form are common models,not restricted to any particular probability distribution and can be used,with little effort,for other distributions as well,e.g. for queueing processes involving the Pearson Type III,Chi-sqare,Erlangian or Gamma distributions.

In the final 3 models (Models 4,5 and 6) sequential queueing systems have been simulated.In Models 4 and 5 Poisson arrivals and exponential serv-

ice time have been assumed.-The analytical study of systems in series is difficult,especially when the interarrival and service time distributions differ from Poisson and exponential.The usual approach is by simulation.-Model 6 is independent of any particular probability distribution.

Finally,let it be noted that calculations were carried out on a computer having 24-bit words,so this had to be taken into consideration in designing the pseudo-random number generator.

Acknowledgements

I am very greateful to Professor I.Abubakar,Head of the Department of Mathematics,Ahmadu Bello University for affording me the opportunity to persue my studies,for helping me to obtain articles and books related to and necessary for my research,and for his assistance througout the composition of this thesis.I also wish to thank to the Ahmadu Bello University for permitting me to undertake this research.

I acknoledge with thanks the moral support of my wife,the encouragement and help of others not listed above without which this work would not have been done.

CONTENTS

	Page
Abstract	I
Acknowledgements	II
Contents	III
P A R T 1.INTRODUCTION	1
1.1 Sampling	1
1.2 Random Number Generation	3
1.3 Mixed Congruential Methods	5
1.3.1 Definitions	5
1.3.2 Theorems	6
1.3.3 Example	7
1.3.4 The Method	8
1.4 multiplicative Congruential Method	10
1.5 Further Considerations	12
1.6 Stochastic Variates	13
1.7 APPENDIX:Computer Subroutines	15
1.7.1 Pseudo-random Number Generator	15
Method	15
Control of Randomness	15
Use	16
Output	16
The Subroutine	16
Example	16
1.7.2 Statistical Tests for Pseudo-random Numbers	17
Method	17
Function	18
Use	18
Output	19
Example	19
1.7.3 Generation of Stochastic Variates (Uniform,Exponential,Poisson)	19
Method	19
(1) Uniform Distribution	20
(2) Exponential Distribution	20
(3) Poisson Distribution	21
Use	22
The Subroutines	22
Example	23

	Page
P A R T 2.SIMULATION MODELS	24
2.1 Introduction	24
2.2 Simulation of a Waiting Line With a Single Station (MODEL 1)	27
2.2.1 Abstract	27
2.2.2 The Model	27
2.2.3 The Subroutine	28
2.2.4 The Flow Chart	30
2.2.5 Check on the Validity	35
2.3 Simulation of a Waiting Line With a Single Station and a Limited Number of Units (MODEL 2)	38
2.3.1 The Model	38
2.3.2 The Flow Chart	39
2.3.3 Check on the Validity	44
2.4 Simulation of a Waiting Line at Several Stations (MODEL 3)	51
2.4.1 The Model	51
2.4.2 The Variables of the Model	51
2.4.3 The Flow Chart	52
2.4.4 The Program Flow Chart	57
2.4.5 Application	59
2.5 Simulation of Waiting Lines in Series	68
2.5-1 A two-station service line in which no queues ahead of station 1 or between the two stations are allowed (MODEL 4)	69
The Model	69
The Flow Chart	71
2.5-2 A two-station system with a waiting line no longer than one unit before the second station but none before the first (MODEL 5)	74
The Model	74
Transitions	75
The Flow Chart	76
Numerical Results	79
2.5-3 Simulation of Waiting Line in Series Under Special Conditions (MODEL 6)	79
The Model	79
The Flow Chart	84

		Page
P A R T	3.POSTSCRIPT	88
	3.1 Computer Simulation	88
	3.2 Definition	88
	3.3 Model	89
	3.4 Flow Chart.Languages	90
	3.5 Advantages	91
REFERENCES		93

P A R T 1

INTRODUCTION

1.1 Sampling

For simulation purposes,a sample is usually utilized to represent the population as part of the input information into a more extensive model. A sample should be representative of the population from which it is drawn; yet in actual practice,it is often quite difficult to assure such a result. If the information contained in the sample is to be of any value to us as representing the salient characteristics of the population,then the seleo-. tion of the elements in the sample must be accomplished in a proper manner. It is obvious that the sample must be selected in a random fashion.In other words,if the sample is to portray the characteristics of a process, then each of the elements produced must have an equal chance of being included in the sample.Sampling is sometimes is the direct selection of a sample from a population,while at other times it involves a simulation.

It is easy to vizualize the random selection of a sample froma population when all the possible items in the population are equally likely to occur. We can randomly select a ball from a hat containing 20 identical balls,numbered 1 through 20.However,if the balls numbered 1,2,...,7 are coloured red and the remaining balls are coloured green,the event of drawing a red ball is not equal to the event of drawing a green one.Nevertheless,a randomly selected ball from the hat is considered a random sample,because the drawing of any one of the 20 balls is a simpler an equally likely event.We can simulate the tossing of a fair die by placing 6 identical balls in a hat,the balls are numbered from 1 to 6,defining the 6 surfaces of the ~~ball~~ die,and then randomly drawing a ball from the hat, successive samples being drawn with replacement.A ball selected by this manner is considered a random sample.If we have an unfair die,and the surfaces to be made appear have the probability of 0.2,0.16,o.16,0.15,0.17,0.16 corresponding to the surfaces denoted by 1,2,3,4,5,6, then we can simulate this experiment by placing 100 balls in a hat numbered 1 through 100.We shall assign the value 1 to the 20 balls numbered 1 through 20,the value 2 to the balls numbered 21 through 36,the value 3 to the balls numbered 37 through 52 etc.By selecting one of the 100 balls,we are randomly selecting a value from the underlying distribution. In this case we have transformed a distribution containing unequally likely events into an equivalent distribution comprised of equally likely,simpler events.

Distributions containing unequally likely events can always be transformed into equivalent distributions comprised of equally likely events by employing some artificial sampling device.This will be true even for continuous distributions.The basic idea is that every "simpler event" in the transformed distribution has an equal and independent chance of being selected.

A sampling procedure that possesses the property,that successive repetitions of the experiment are independent and also the density function of the random variable remains unchanged for all successive repetitions,is said to

be random sampling.This is the case with the continuous and as well as the discrete distributions (sampling with replacement) too.

One can use simulation of artificial methods in selecting samples from a population for which the distribution function of the pertinent random variable is unknown.True simulated sampling usually refers to the sampling of a known distribution without collecting data from the original populatinn of physical objects.We are not trying to learn anything about the random variable (it is known),rather we are trying to select a sample to represent this distribution in a more complex model.The basic problem of simulated sampling is how to select the values of the random variables,whose distributions are known,so that we have simulated their occurences as random processes.If we have a sequence of independent observations of a random variable whose distribution function is known,then we can construct a sequence of independent observations of any other random variable whose distribution is known.The original known sequence is often referred to as the basic sequence and, in practice,it is usually a sequence of random numbers from a uniform distribution.

Whenever we use such a basic sequence to generate a sequence of independent observations of another random variable with a known distribution,it is said to perform Monte Carlo sampling.

Monte Carlo analysis is a simulation technique for problems having a stochastic or probabilistic basis.Two different types of problems give rise to the use of this technique.First,there are those problems that involve some kind of stochastic process.In simulated waiting line models arrival and service times; in inventory models customer demand and delivery times; in research and development events of new product discoveries are examples of economic variables which may be considered to be stochastic in nature. Second,certain completely deterministic mathematical problems cannot be solved easily(if at all) by strict deterministic methods.However,it may be possible to obtain approximate solutions to these problems by simulating a stochastic process whose moments,density function,or oumulative distribution function satisfy the functional relationship or the solution requirements of the deterministic problem.

The main objective of any Monte Carlo procedure is the determination of outcomes at random at critical decision points in a stochastic process.The simplest type is a "yes-no" decision point,such as a situation where there are only two possible outcomes.Suppose that H_1 and H_2 are the only possible outcomes of a particular trial and the probability that H_1 occurs is P_1 and the probability that H_2 occurs is $P_2 = 1-P_1$.Thus to determine whether H_1 or H_2 occurs in a given trial,we generate a uniform random number U between the interval 0 and 1,and perform a test.H_1 occurs if $U \leq P_1$;otherwise H_2 occurs.

Another type of decision that must be made in an extension of the previous case.Here,there are several possible outcomes with given probabilities of occurence of each.Suppose that $H_1, H_2, \ldots, H_n$ are the possible outcomes of

a particular trial,and the probability that each occurs is $P_1, P_2, \ldots, P_n$, respectively,and $P_1+P_2+\ldots+P_n = 1$.Thus to determine whether $H_1, H_2, \ldots, H_n$ occurs in a given trial,we generate a uniform random number U,and we perform a test.H_1 occurs if $U \leq P_1$,H_2 occurs if $P_1 < U \leq P_1+P_2$,...,H_k occurs if

$$\sum_{j=1}^{k-1} P_j < U \leq \sum_{j=1}^{k} P_j \qquad (k=2,3,\ldots,n). \qquad (1.1-1)$$

Frequently,such simulations require thousands and sometimes hundreds of thousands of draws from the probability distributions contained in the model.So the basic builing block for simulating complex random phenomena is the generation of random digits.

1.2 Random Number Generation

The main characteristics of random numbers are the underlying distribution of the population of such numbers and the independence associated with the order of occurence of the numbers.The probability of occurence of a specified number should always be the same - not affected by post occurrences.One of the principal requirements of the sequence of random numbers is statistical independence.

In complex sampling experiments we often find it useful to be able to repeat the calculations,including the selection of the random numbers,as a method of checking our results.Further,the accuracy of sampling experiments can be improved if we can repeat the sequence of random numbers.Since any repeatable procedure that has been devised cannot produce truly random numbers otherwise,it would not be repeatable, the need for the generation of a set of such numbers that can be regenerated is selfcontradictory - because of the definition of random numbers.On the other hand,we can generate a set that is almost random.A sequence of numbers can be generated,usually through a computer program by a deterministic formula that permits reproducibility in such a way that any reasonable statistical test will show no significant departure from randomness.From this point of view the method of generating a sequence is totally irrelevant.

Since such a process is not really random,the generated sequence is referred to as being p s e u d o - r a n d o m.Numbers so generated are used as if they were truly random.A sequence of pseudo-random numbers is always cyclic;however,if the cycle is long enough this will present no difficulties.Most pseudo-random number generation schemes attempt to generate a uniformly distributed set of random numbers to become a basic sequence for Monte Carlo simulation purposes.

Four alternative methods have been used by practioners to generate sequence of random numbers:(1) manual methods,(2) library tables,(3) analog computer methods,(4) digital computer methods.The first three suffer from several disadvantages so we apply the last method only.

Three modes of providing random numbers for use on digital computers have been suggested by Tocher[71]:(1) external provision, (2) internal generation by a random physical process and (3) internal generation of sequences of

digits by a recurrence relation.

Without discussing the first two alternatives it is worth mentioning that the (3) alternative overcomes some of the objections to the other methods since there is no problem of input or computer memory capacity and the whole process,depending only on arithmetical steps,is reproducible.The main objection to this solution is that a sequence of digits generated by a purely deterministic rule is the direct antithesis of a random sequence. This objection,as mentioned before,can at least partially be overcome by taking the view that a sequence may be considered random if it satisfies some predetermined set of statistical test of randomness.

We require sequences of numbers which at least appear to be drawn from certain probability distributions and restrict our attention to sequences of numbers which might appear to be drawn from a uniform distribution. The principal value of the uniform distribution for simulation techniques lies in its simplicity and in the fact that it can be used to simulate ran - dom variables from almost any kind of probability distribution.

There are several ways of generating such sequences and nearly all computers use some variation of thecongruential methods developed by Lehmer [37].

It would appear that an acceptable method for generating such numbers must yield sequences of numbers which are

(i) uniformly distributed,

(ii) statistically independent,

(iii) reproducible and nonrepeating for any desired length.

Further such a method must also be capable of(iv)generating pseudo-random numbers at high rates of speed,yet (v) requiring a minimum amount of computer memory capacity.We will concern ourselves only with the methods as mentioned (congruential methods) so much the more because they seem to be best for use on a stored program computer.

Congruential methods for generating random numbers are completely deterministic because the arithmetic processes involved in the calculations uniquely determine each term in a sequence of numbers.Formulas are available for calculating in advance the exact value of the i^{th} number in a sequence of numbers $\{x_1,\ldots,x_i,\ldots\}$ before the sequence is actually generated. Although these processes are not random processes at all,we treat them as though they were,if the sequences that result pass a certain number of statistical tests designed to test various properties of random variables. It turns out rather surprisingly that the statistical behaviour of the sequences is good,with only a few exceptions,as long as they are not repeated too soon.A sequence of numbers generated by one of these methods satisfies requirements from (i) - (v).Our main objective will be to establish the basic number theoretic properties of such sequences,and to summarize the results in two theorems.In order to develop the congruential methods for generating pseudo-random numbers,we use some of the definitions and theorems of number theory and modern algebra. We refer to the definitions and theorems listed in chapter 1.3 many times in the later section of this paper.

Congruential methods are based on a fundamental congruence relationship which may be expressed as the following recursive formula

$$x_i \equiv ax_{i-1} + c \pmod{m} \qquad (1.2\text{-}1)$$

We can define a sequence $\{x_i\}$ of non-negative integers, each less than m, by means of this congruence relation if we have selected a positive integer x_o, called the starting value, an integer a, called the constant multiplier, another integer c, called the additive constant and a fourth integer m, called the modulus, which is positive and greater than the other three in magnitude. From the integers in sequence $\{x_i\}$, rational numbers in the interval $[0,1)$ can be obtained by forming the sequence $\{x_i/m\}$.

It is clear that any sequence must repeat itself eventually because it can contain at most m different numbers, each number in a particular sequence being determined solely by its predecessor. There exists a smallest positive value of i, i=p, such that $x_p = x_o$, where p is the period of the sequence $\{x_i\}$. It folows from Theorem 5 section 1.3.2 that the existence of such a period is equivalent to saying that it is impossible to obtain nonrepeating sequences by congruential methods. However, the period of a sequence can be set satisfactorily high by choosing a sufficient technique. Since such a period does exist, we must prescribe conditions on x_o, a, m and c which will ensure maximumpossible period.

In practice we usually find it desirable to choose m to be a power of 2 on a binary machine, or a power of 10 on a decimal machine. We are then able to avoid the devision which is implicit in the congruence and also the devision to form $\{x_i/m\}$ by shifting the binary or decimal point to the left of the number. Other choices are possible, however m will be taken as given and now the next problem is to find what choice of x_o, a and possibly c will ensure a maximum period. The case in which $c \neq 0$ is the simpler and will be treated first. This class of generators is called Mixed Congruential Method.

1.3 Mixed Congruential Methods

The first part of this chapter contains a set of definitions, examples and theorems of number theory that are pertinent to the understanding of the methods discussed in this and the following chapters.

1.3.1 Definitions

1. For a given a the smallest positive integer x such that $a \equiv x \pmod{m}$ is said to be a residue modulo m.
2. For a given modulus m, the set of m integers congruent in some order to the residues $0,1,\ldots,m-1$ form a complete residue system.
3. A subset of a complete residue system containing all integers that are relatively prime to m is a reduced residue system.
4. The number of positive integers which are less than m and relatively prime to m is defined as Euler's phi-function and is denoted by $\varphi(m)$. A reduced residue system contains $\varphi(m)$ integers. If $m=p$ is a prime, $\varphi(p) = p-1$.
5. Power residues are the residues of the succesive powers of an integer

a modulo m. If the residue of the i^{th} power of a (mod m) is denoted by x_i, then all power residues satisfy the congruence relation

$$x_i \equiv a^i \pmod{m} \quad (i=1,2,\ldots).$$

6. If $(a,m) = 1$, the least positive exponent $i = q$, such that $a^q \equiv 1 \pmod{m}$, is said to be the order of a (mod m). The least positive exponent q is also called the indicator of m when a is said to belong to m. In this case q is denoted $q = \lambda(m)$. The order q of a (mod m) is equal to the total number of distinct numbers, that is, the length of a nonrepeating sequence of power residues of a, called the period of the sequence $\{x_i\}, (i=1,\ldots,q) \pmod{m}$.

7. An integer with order $q = \varphi(m)$ is the primitive root of m. (It is obvious that $\lambda(m) \leq \varphi(m)$.)

1.3.2 Theorems

1. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a \pm c \equiv b \pm d \pmod{m}$ and $ac \equiv bd \pmod{m}$.

2. If $ca \equiv cb \pmod{m}$ and $(c,m) = 1$, then $a \equiv b \pmod{m}$.

3. If $ca \equiv cb \pmod{m}$ and $g = (c,m)$, then $a \equiv b \pmod{m/g}$.

4. Any integer m (other than 0 or ± 1) can be factored uniquely into primes, that is, $m = \prod p_i^{e_i}$ $(i=1,2,\ldots)$.

5. If $(a,m) = 1$, then $a^{\varphi(m)} \equiv 1 \pmod{m}$ (Fermat-Euler theorem), from which it follows that

(i) the largest possible order of a is $q = \varphi(m)$, when a is a primitive root of m;

(ii) for $x < m$ such that $(x,m) = 1$, $xa^q \equiv x \pmod{m}$, where $q = \varphi(m)$;

(iii) the order of an integer a (mod m) is a divisor of $\varphi(m)$.

6. Primitive roots exist for all power of a prime number $p > 2$, that is a number exists such that $(a,p^e) = 1$ and

$a^{\varphi(p^e)} \equiv 1 \pmod{m}$, where $q = \varphi(p^e)$.

7. If $m = \prod p_i^{e_i}$, then $\varphi(m) = \prod (p_i - 1) p_i^{e_i - 1}$.

8. If $m = p^e$ and p is an odd prime, then $q = \lambda(m) = (p-1)p^{e-1} = \varphi(m)$ for values of a that are primitive roots of m.

Corollary: If $p = 2$, that is $q = \lambda(m) = 2^{e-2}$ for $e > 2$, then

$$\lambda(m) \nmid \varphi(m).$$

(Example: for $m = p^e = 2^4$, $q = \lambda(m) = \lambda(2^4) = 2^2$ and $\lambda(2^4) \nmid \varphi(2^4) = 2^3$.)

9. If $m = \prod p_i^{e_i}$ $(i = 1,2,\ldots,t)$ then:

(i) $\lambda(m)$ = least common multiple $[\lambda(p_1^{e_1}), \lambda(p_2^{e_2}), \ldots, \lambda(p_t^{e_t})]$.

(ii) there exists value of a that have order equal to (e.i. belong simultaneously to) each $\lambda(p_i^{e_i})$.

If $p_1 = 2$, then $\lambda(m) = l.c.m.[\lambda(2^{e_1}), \varphi(p_2^{e_2}), \varphi(p_3^{e_3}), \ldots]$.

Note:

Theorem 9-(ii) follows from the Chinese Remainder Theorem, according to it:

if the positive integers $m_i (i = 1,2,\ldots,t)$ are relatively prime in

pairs and if $a_i (i=1,2,\ldots,t)$ are any given integers, then the t congruences $x \equiv a_i \pmod{m_i}$ $(i=1,2,\ldots,t)$ have a common solution which is unique $(\text{mod } m_1 m_2 \ldots m_t)$.

1.3.3 Example

Suppose that m = 11. It follows that $\varphi(11) = 10$ and there are ten distinct residue classes corresponding to residues of $0,1,2,\ldots,10$. Power residues for $a = 0,1,2,\ldots,10$ are indicated in the next table

	exponent	1	2	3	4	5	6	7	8	9	10	Order
	0	0	repeat									1
	1	1	repeat									1
	2	2	4	8	5	10	9	7	3	6	1	10
	3	3	9	5	4	1	rep.					5
	4	4	5	9	3	1	rep.					5
	5	5	3	4	9	1	rep.					5
a	6	6	3	7	9	10	5	8	4	2	1	10
	7	7	5	2	3	10	4	6	9	8	1	10
	8	8	9	6	4	10	3	2	5	7	1	10
	9	9	4	3	5	1	rep.					5
	10	10	1	rep.								2

$\varphi(11) = 10$, therefore the order of a (mod 11) may only be 1,2,5 or 10.
$\lambda(11) = 5$ when 3 belongs to 11, since $3^5 \equiv 1 \pmod{11}$, but $\lambda(11)=\varphi(11)=10$ when 2 belongs to 11, since $2^{10} \equiv 1 \pmod{11}$ and $2^q \not\equiv 1 \pmod{11}$ if $q < 10$, hence 2 is a primitive root of 11, but 3 is not.

Some important features of power residues can be seen from this example, namely

(i) If m has a primitive root, it will have $\varphi[\varphi(m)]$ of them.
Now $\varphi[\varphi(11)] = 4$, the primitive roots of 11 are 2,6,7 and 8.

(ii) Whenever (a,m) = 1 then in any one sequence a cycle returns to the original first number generated, and no number is repeated within a cycle. The cycle for all sequences begins with the value of a and ends with the number of 1.

(iii) Any value of a greater than or equal to m will be congruent to one of those less than m and will produce the same sequence as those of a's less than m. In other words,

$$a + tm \equiv a \pmod{m} \qquad (1.3\text{-}1)$$

and

$$(a + tm)^i \equiv a^i \pmod{m}. \qquad (1.3\text{-}2)$$

1.3.4 The Method

Theorem A:

The sequence defined by the congruence relation (1.2-1) has a full period m, provided that

(i) c is relatively prime to m;

(ii) $a \equiv 1 \pmod{p}$ for every prime factor p of m;

(iii) $a \equiv 1 \pmod{4}$ if 4 is a factor of m.

If $m = 2^{\alpha}$, as it is used on a binary machine, we need only have c odd and from (ii) and (iii) $a \equiv 1 \pmod 4$. (If $m = 10^{\beta}$, c not dewisable by 2 or 5, and $a \equiv 1 \pmod{20}$, since $a \equiv 1 \pmod 5$, $a \equiv 1 \pmod 4$.)

When $a = 1$ and $(c,m) = 1$, the period is obviously m. ($x_{i+1} \equiv x_i + c \pmod m$ if $(c,m) = 1$ yields m distinct residues (mod m). Example: if $m = 8, c = 3, x_o = 2$, then the series: 2,5,0,3,6,1,4,7.)

We therefore assume that $a \neq 1$.

From $x_{i+1} \equiv ax_i + c$ with $i = 0,1,\ldots,n-1$ we easily obtain

$$x_n \equiv a^n x_o + \frac{(a^n-1)c}{a-1} \pmod m. \qquad (1.3\text{-}3)$$

The period of the x_i sequence is the smallest positive integer n, for which $x_n = x_o$, that is

$$x_o \equiv a^n x_o + \frac{(a^n-1)c}{a-1} \pmod m \qquad (1.3\text{-}4)$$

and after arranging it we have

$$\frac{(a^n-1)}{a-1}\,[x_o(a-1) + c] \equiv 0 \pmod m. \qquad (1.3\text{-}5)$$

Since c and a are both odd integers the bracketed expression is also odd. A condition therefore, which is completely equivalent to this congruence relation is that

$$\frac{a^n-1}{a-1} \equiv 0 \pmod m. \qquad (1.3\text{-}6)$$

It will be shown that if a satisfies the conditions of the Theorem A then this smallest value of n is equal to m.

Lat us consider the first case when $m = p^{\alpha}$, where α is a positive integer, and p is an odd prime. When $\alpha = 1$, then from the (ii) condition $a = 1 + kp^{\beta}$ (β is a positive integer), but $a < m$, therefore $a = 1$, which has been mentioned as the trivial case. We therefore need only consider $\alpha \geq 2$.

It is obvious that a can be written as

$$a = 1 + kp^{\beta} \qquad (1.3\text{-}7)$$

where $(k,p) = 1$ and $k \neq 0$ (and β is a positive integer).

There are two things now to prove:

1. $n = p^{\alpha}$ satisfies (1.3-6),

2. no value of n smaller than p^{α} will satisfies (1.3-6).

To see part 1. we replace a by $1 + kp^{\beta}$, and by means of the binomial theorem we have

$$\frac{a^n-1}{a-1} = n + \binom{n}{2}kp^{\beta} + \binom{n}{3}(kp^{\beta})^2 + \ldots + \binom{n}{i}(kp^{\beta})^{i-1} + \ldots + (kp^{\beta})^{n-1} \qquad (1.3\text{-}8)$$

then replacing n by p^α we obtain

$$\frac{a^n-1}{a-1} = p^\alpha + \frac{p^\alpha(p^\alpha-1)}{2!}kp^\beta + \frac{p^\alpha(p^\alpha-1)(p^\alpha-2)}{3!}(kp^\beta)^2+\ldots + \frac{p^\alpha(p^\alpha-1)\ldots(p^\alpha-i+1)}{i!}(kp^\beta)^{i-1}+\ldots+(kp^\beta)p^{\alpha-1} . \qquad (1.3\text{-}9)$$

We must prove that this expression is congruent to 0 (mod p^α),that is, it is devisable by p^α.

In fact it turns out that each term in this expression is devisable by p^α. To see that it is enough to show that the i^{th} term devisable by p^α.The i^{th} term can be written as

$$\frac{p^\alpha}{i}\left[\frac{(p^\alpha-1)\ldots(p^\alpha-i+1)}{(i-1)!}\right]k^{i-1}p^{\beta(i-1)} . \qquad (1.3\text{-}10)$$

The first part of this term preceeding the factor k^{i-1} is a binomial coefficient $\binom{p^\alpha}{i}$,and is therefore integer.The term in square brackets is also a binomial coefficient $\binom{p^\alpha-1}{i-1}$ and hence integer.So i is a divisor of p^α and since p is prime,i is one of the integral powers of p,that is p or p^2or p^3,...,or p^α.If this exponent of p is denoted by γ,that is $i = p^\gamma$,then γ is less than

$$\frac{i}{p}+\frac{i}{p^2}+\frac{i}{p^3}+\ldots = \frac{i}{p-1} . \qquad (1.3\text{-}11)$$

Since $p > 2$, $\frac{i}{p-1} = i-1$.But the factor p appears this many times in $p^{(i-1)\beta}$ $(\beta\geq 1)$,so we need not devide p^α by i.

It follows that each term on the right side of(1.3-9) devisable by p^α and (1.3-6) is satisfied by $n = p^\alpha$,at least under the stated conditions. To prove (ii) it is possible to show that a value of n will satisfy (1.3-6) if and only if it is a multiple of the samllest such value.We know that $n = p^\alpha$ satisfies (1.3-6),we therefore need to consider only values of n which are powers of p.It is sufficient to show that $n = p^{\alpha-1}$ does not satisfy (1.3-6).

Substituting $n = p^{\alpha-1}$ into (1.3-6),and using that $a = 1 + kp$,we obtain

$$\frac{a^n-1}{a-1} = p^{\alpha-1} + \frac{p^{\alpha-1}(p^{\alpha-1}-1)}{2!}(kp^\beta)+\frac{p^{\alpha-1}(p^{\alpha-1}-1)(p^{\alpha-1}-2)}{3!}(kp^\beta)^2+\ldots$$
$$+(kp)^{p^{\alpha-1}-1} . \qquad (1.3\text{-}12)$$

The right side of this equation is nor devisable by p^α.Since the first term is obviously not devisable by p^α,and so it will be sufficient to show that each of the other terms is devisable by p^α.

The proof follows exactly the one given before for the i^{th} term in (1.3-9), but this time we have $p^{\alpha-1}$,instead of p^α,as a factor in the binomial coefficient .Another factor can be obtained from $p^{(i-1)\beta}$provided we now make use of our assumption that p is odd.It is obvious that in this case $\frac{i}{p-1}$ is less than or equal to i-2,instead of i-1 .

So thus Theorem A has been proved when $m = p^\alpha$,and p is odd.The proof when $m = 2^\alpha$ is only slightly different.We must consider the case where

$\alpha \geq 2$ (the case when $\alpha = 1$ is again trivial) which is similar to that of the previous case with the only restriction that $\beta > 1$ in (1.3-7).

In the second case Theorem A can be generalized to the case where $\underline{m}$ is composite.Then we write

$$m = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_t^{\alpha_t} \quad \text{and} \quad a = 1 + kp_1^{\beta_1} p_2^{\beta_2} \cdots p_t^{\beta_t},$$

with p_i prime, α_i a positive integer, $k \neq 0$ and $(k,m) = 1, \beta_i \geq 1$ or, if $p_i = 2$ and $\alpha_i \geq 2, \beta_i \geq 2$.The argument proceeds almost exactly as before,so Theorem A is established in general case.

1.4 Multiplicative Congruential Method

Another class of generators can be obtained from the relation

$$x_i \equiv ax_{i-1} + c \pmod{m}$$

by taking $c = 0$,and is called Multiplicative Congruential Method or sometimes Power Residue Method because

$$x_n \equiv a^n x_o \pmod{m} \qquad (1.4\text{-}1)$$

and if we take x_o to be relatively prime to $\underline{m}$,we are led to

$$a^n \equiv 1 \pmod{m}. \qquad (1.4\text{-}2)$$

(Since $x_n \equiv a^n x_o \pmod{m}$,the period of the x_i sequence is the smallest positive integer $\underline{n}$ such that $x_n = x_o$,that is $x_o \equiv a^n x_o \pmod{m}$.But being $(x_o,m) = 1$,according to theorem 2 (page 6) $a^n \equiv 1 \pmod{m}$.)

After Lehmer [37] had first introduced the congruence theory and power residues into the generation of pseudo-random nmubers,most persons were inclined to choose $\underline{m}$ as a prime number and $\underline{a}$ as a primitive root of this prime $\underline{m}$ (,$(m,a) = 1$ and $(m,x_o) = 1$).The reason for this choice was the assurance that m-1 numbers would be generated through the power residues of $\underline{x}$.(See definitions 5,6 and 7,and theorem 5;p.5-6.)But this technique does not take advantage of characteristic peculiar to the available computer.In computerized version of the multiplicative congruential method a modulus $m = p^e$ can be employed,representing the word size of the computer,where $\underline{p}$ denotes the number of numerals in the number system utilized by the computer and $\underline{e}$ denotes the number of digits in a word.

There are two reasons for choosing $m = p^e$.First,reduction modulo $\underline{m}$ is accomplished by truncating and retaining only the low order $\underline{e}$ digits; and second,conversion to the unit interval (to obtain uniformly distributed variates) involves only moving to binary or decimal point to the left of the number.

We choose $\underline{m}$ to be a power of 2 on a binary machine,or a power of 10 on a decimal machine.If either $m = 2^b$ or $m = 10^d$,then $\lambda(m) < \varphi(m)$ (see theorems 8 and 9;p.6.),so we are not able to choose $\underline{a}$ so that the sequence of numbers has full period $\underline{m}$,but so that the period is quite large.(One of the properties of power residues is that only numbers of the form p^n or $2p^n$, where $p > 2$, have primitive roots,with the exception of the number of 4.)

Our main problem is to describe conditions on $\underline{a}$ which will ensure that the smallest value of $\underline{n}$ satisfying (1.4-2) is a maximum.This value is the

indicator of $\underline{m}$, $\lambda(m)$ when $\underline{a}$ belongs to $\underline{m}$.If $\underline{m}$ is composite as $m=p_1^{\alpha_1}\cdots p_t^{\alpha_t}$, where p_i's are distinct primes then according to theorem 9

$\lambda(m)$ = least common multiple $[\lambda(p_1^{\alpha_1}),\ldots,\lambda(p_t^{\alpha_t})]$,and

there exists $\underline{a}$ belonging simultaneously to each $\lambda(p_i^{\alpha})$ .We must therefore consider the case where $m = p^{\alpha}$($\underline{p}$ is a prime):

1.If $\underline{p}$ is odd,then $\lambda(p^{\alpha}) = (p-1)p^{\alpha-1}$ (theorem 8) and the corresponding values of $\underline{a}$ are primitive roots for p^{α} .

2.If $p = 2$,then we have primitive roots if and only if $\alpha = 1$ or $\alpha = 2$, otherwise $\lambda(2^{\alpha}) = 2^{\alpha-2} < \varphi(2^{\alpha})$ and $2^{\alpha-2}$ is the maximal order (theorem 8).

If we consider a binary computer then $m = 2^{\alpha}$,and according to theorem 8 for $\alpha > 2$,the maximum attainable period is $q = 2^{\alpha-2}$.We have to find constant multipliers having order $2^{\alpha-2}$.From definition 6 and theorem 5,it follows that $\underline{a}$ must be relatively prime to $\underline{m}$.

In order to have $(a,m) = 1$,we must choose $\underline{a}$ as an odd integer.But all odd integers are of the form $8t\pm1$ or $8t\pm3$ (where $\underline{t}$ is an integer), and it can be shown [22] that the latter yields the maximum period $2^{\alpha-2}$, that is $\underline{a}$ must satisfy the congruence relation $a \equiv \pm3 \pmod 8$.

We can summarize now the main properties of the multiplicative congruential methods as follows

Theorem B:

The maximum period of the sequence defined by taking $c = 0$ in $x_i \equiv ax_{i-1}+c \pmod m$ is achieved provided that

(i) x_o is relatively prime to $\underline{m}$;

(ii) $\underline{a}$ is a primitive root for p^{α},if p^{α} is a factor of $\underline{m}$,with $\underline{p}$ odd and α as large as possible,or with $p = 2$ and $\alpha = 1$ or 2;

(iii) $\underline{a}$ belongs to $2^{\alpha-2}$,if 2^{α} is a factor of $\underline{m}$,with $\alpha \geq 2$.

Furthermore,for any $\underline{m}$ there exists values of $\underline{a}$ satisfying these conditions,and the maximum period is the least common multiple of the periods $(p_i-1)p_i^{\alpha_{i-1}}$ or $2^{\alpha-2}$,with respect to the prime power factors.

In practice it is easy to satisfy condition (i) and (iii) (by insisting that $a = 8t\pm3$).Condition (ii) can be satisfied if we find primitive roots of prime powers.

It can be proved [52,p.285-288] that if

(i) $\underline{a}$ is a primitive root for $\underline{p}$ ($\underline{p}$ prime) and

(ii) $a^{p-1} \neq 1+kp$,where $\underline{k}$ is not devisable by $\underline{p}$,then $\underline{a}$ is a primitive root for p^{α}.Moreover,if $\underline{a}$ satisfies the conditions above,then so will any number which is congruent to $\underline{a} \pmod{p^2}$.

For example,if $p = 7$,then 3,5,10,12,17,19,24,26,31,33,38,40,45,47,... are primitive roots for 7.It is easy to verify that 3,5,10,12,17,19,24,26, 38,40,45,47,... satisfy the second condition required for $\underline{a}$,while 31 and 33 do not.In this case the maximum period $6(7^{\alpha-1})$ provided we choose x_o to be not devisable by 7,and $a \equiv (3,5,10,12,17,19,24,26,38,40,45,47) \pmod{49}$.

1.5 Further Considerations

Considering the relative merits of the mixed and multiplicative congruential methods on the basis of the theoretical evidence obtained so far, it appears that the mixed method has some small advantages over the multiplicative.

The mixed methods has larger periods (but the periods are in any case very long, so this is not usually a real advantage). Another advantage of the mixed method lies in its relative simplicity. The basic theorem is easier to establish and the conditions are much easier to remember.

The only disadvantage of the mixed method is in its statistical behaviour In spite of the fact that its behaviour in this respect is generally good, in some cases it is completely unacceptable. There are no such cases among the multiplicative method.

The statistical properties of pseudo-random numbers generated by the methods outlined in the previous section should coincide with the statistical properties of numbers generated by an idealized chance device that selects numbers from the unit interval [0,1) independently and with all numbers equally likely. Clearly, the pseudo-random numbers are completely determined by the starting data, but these pseudo-random numbers can pass the set of statistical tests and can be treated as "truly" random numbers even though they are not. The following statistical tests are among the more important tests for randomness cited in the literature: Frequency test, serial tests, "poker test", "gap test", and so forth.

Before choosing the parameters for generating pseudo-random numbers we must refer to some theoretical results developed by Coveyou [12] and Greenberger [22]. The formulae given by them present theoretical conditions on the optimum values of $\underline{a}$, $\underline{c}$ and $\underline{m}$ in the congruence relation $x_i \equiv ax_{i-1}+c \pmod m$ from the point of view of an a priori determination of the serial correlation between the numbers. The magnitude of the serial correlation coefficient $\rho(x_i, x_{i-1})$ lies between values

$$\rho_{1,2} = \frac{1}{a} - \left(\frac{6c}{am}\right)\left(1 - \frac{c}{m}\right) \pm \frac{a}{m} . \qquad (1.5\text{-}1)$$

Greenberger points out that very small or large values of $\underline{a}$ are to be avoided, and that values of $\underline{a}$ near to $m^{1/2}$ will yield the smallest value for ρ regardless of the value of $\underline{c}$. This result provides a necessary condition for minimizing ρ for a sequence, but it is by no means a sufficient condition.

They showed by a further investigation that the sequence consisting of every k^{th} number of the original sequence is itself a sequence of the form $x_i \equiv x_{i-1}+c \pmod m$, but with multiplier a^k, and additive constant $c_k = (a^k-1)c/(a-1)$ (both parameters should be calculated mod m).

By selecting a constant multiplier which satisfies the Coveyou-Greenberger conditions

$$a = 2^{\alpha/2}+1 \quad (\text{or} \quad a = 10^{\beta/2}+1)$$

on a binary (or on a decimal) machine,is not a sufficient condition,in the case of mixed congruential methods,for minimizing first order serial correlation for these sequences.Only by empirical testing we can have confidence in the statistical properties of sequences.

1.6 Stochastic Variates

Stochastic simulation methods can be used for solving strictly deterministic problems mainly in cases when solutions are not easily obtainable by standard numerical or analytical methods.For a great number of mathematical problems it is possible to find a stochastic process with suitable probability distributions and parameters that satisfy the required mathematical properties of the problem.It may be more efficient to construct such a process and generate statistics on a computer rather than attempt to use standard methods.Among the deterministic problems for which stochastic simulation has been found to be useful are included complex queueing,inventory and job-shop scheduling problems,solutions of higher order differential equations and so forth.

Simulation is merely a technique of numerical analysis and is to be preferred only if its efficiency is superior to that of other techniques.

Stochastic simulation can also be used for problems in which some method of statistical sampling is indicated,but in which the actual taking of a sample is either impossible or not feasible.Such might be the case with data on machine breakdowns in a factory where no accurate records are available,or with demand data for a firm's product at some future data or time interval between accidents,or the arrival of orders at a firm,the arrivals of patients in a hospital,the arrival of customers of a supermarket etc.Statistical data are not easily to obtain,but some knowledge may be available about the statistical population from which the data would originate if they were possible to obtain.In such cases the basic problem for simulation consists in sampling from statistical distribution.

Thus we give a process of selection from the distribution which by repetition will give rise to a frequency distribution of sampled values which matches the frequency distribution required in the simulation.

We use a common source for all types of distribution needed.The generation of simulated statistics (random variates) is carried out by suplying pseudo-random numbers into the process or system and obtaining from it as answers.

In a stochastic simulation we replace an actual statistical population by its theoretical counterpart,describing it by some assumed probability distribution and then sampling fromthis theoretical population by means of some random number generator.When a standard theoretical distribution can not be found then the particular stochastic process can be simu-

lated by sampling from empirical distributions.

Specific techniques for generating - on a computer - random variates from standard probability distributions, as well as from any empirical distributions that may arise in attempting to solve stochastic problems, are known. Three basic methods are, the I n v e r s e T r a n s f o r m a t i o n Method, the R e j e c t i o n Method and the C o m p o s i t i o n Method. Sometimes some variation of these can also be applied. (Subroutines for generating stochastic variates are attached in the Appendix.)

1.7 APPENDIX: Computer Subroutines

1.7.1 Pseudo-random Number Generator

Name and dummy arguments of subroutine: RAND(IA,IX,URN)

Language: FORTRAN IV.

Method

The RAND subroutine can be used to generate uniformly distributed pseudo-random numbers by the Multiplicative Congruential Method. The congruence relation

$$x_{i+1} \equiv ax_i \pmod{m} \qquad (1.7\text{-}1)$$

has been aplied to generate random numbers, where x_i, a, m and x_{i+1} are positive integers and m is greater than the other three in magnitude. To obtain numbers in the interval [0,1) the sequence $\{x_i/m\}$ is formed. The sequence defined by taking (1.7-1) has maximum period provided that

(i) $m = 2^{\alpha}$;

(ii) x_o (starting value) relatively prime to m;

(iii) a (multiplier) is also relatively prime to m;

(iv) $a \equiv \pm 3 \pmod 8$, that is $a = 8t \pm 3$, where t is a positive integer;

(v) a must be close to $2^{\alpha/2}$. Such a value of a will satisfy the Coveyou [12] - Greenberger [22] condition.

The maximum attainable period with these conditions is $p = 2^{\alpha-2}$ (Hull and Dobell [27]).

In consequence the above reasons the following values of the parameters are selected for the subroutine:

(a) $m = 2^{23}$

(b) $x_o = 2^{23}-1$ (it might be any odd number)

(c) $a = 2^{11}+3+8t$ (t positive integer and $0 \leq t \leq 64$).

With these values of the parameters the period $p = 2^{21}$, that is 2 097 152 different numbers can be generated in a random sequence in the interval [0,2 097 152) and can be transformed to obtain a random sequence of numbers in the interval [0,1).

When choosing a value for a, different sequences are obtained using different values of t, but each of them has the same period 2^{21}.

Control of Randomness

The randomness of the numbers generated by RAND can be controlled by another subroutine, named TESTS (see chapter 1.7.2) Two statistical tests - one frequency and one serial test - are applied for control in this subroutine.

RAND is controlled for the value of $a = 2^{11}+3$ and the numbers obtained supply very good statistical behaviour.

Use

Using the notations of the subroutine the arguments corresponding to the parameters applied in the formula (1.7-1) are as follows:

Parameters	Arguments
x_i	IX
a	IA
x_i/m	URN

Before using RAND values must be assigned to the (input) parameters IA and IX.

The RAND subroutine generates one whole number (IX) devides it by the modulus (2^{23}) to have a number (URN) in the interval [0,1).That is the whole number generated by RAND is also denoted by IX and therefore can be used as the next "starting value".

Output

The RAND subroutine does not print any value,but stores the number generated in the IX variable and can deliver up IX to the main program, where it can be used (possibly printed) for further processing.

The Subroutine

```
      SUBROUTINE RAND(IA,IX,URN)
      IX = IX * IA
      IF(IX) 200,300,300
200   IX = 8388607 + IX
      IX = IX + 1
300   XI = IX
      URN = XI/8388608.
      CALL OVERFL()
      RETURN
      END
```

Example

In the example shown below 1024 random numbers are generated and printed. The values of IXS (starting value) and IAS (multiplier) have been given and two vectors (INT and SI) of 8 elements have been defined for collecting 8 numbers generated and printed in one line,8 integers in one line and the corresponding 8 fractions falling into the interval [0,1) in the next line.

IAS (the value of the multiplier) is not changed during one period, but the whole numbers generated is used as a starting value for IXS in the next call of RAND.

Program:

```
      MASTER PROB
      DIMENSION INT(8),SI(8)
      IXS = 8 388 607
      IAS = 2**11 + 3
      N = 0
      DO 5 I = 1,1024
      CALL RAND(IAS,IXS,URNA)
      N = N + 1
```

```
      INT(N) = IXS
      SI(N) = URNA
      IF(N-8) 5,4,4
    4 WRITE(2,6) (INT(I),I = 1,8),(SI(I),I = 1,8)
      N = 0
    5 CONTINUE
      STOP
    6 FORMAT (8I14/8F14.10)
      END
      SUBROUTINE RAND (IA,IX,URN)
      .
      .
      END
      FINISH
```

1.7.2 Statistical Tests for Pseudo-random Numbers

Name and dummy arguments of subroutine:TESTS(NA,NB,IA,IX)

Language:FORTRAN IV.

Location:Source Cards,Computer Centre,Ahmadu Bello University

(written and documented by Gy.Nemeth in February,1973)

Method

The TESTS subroutine investigates the numbers generated by the RAND generator from a statistical point of view;whether the sequence of numbers can be considered to be drawn at random from a uniform distribution.(It is known that these numbers are not drawn at random from a uniform distribution,but for practical purposes it is sufficient that they have the appearance of being so drawn.The Multiplicative Congruential Method which is applied for RAND is completely deterministic,but we can use it to simulate a random process).

The fundamental requirement for the sequence of numbers is to be uniformly distributed over the interval [0,1).To test them for this feature the unit interval is devided into $\underline{k}$ equal subintervals and the number of numbers in the i^{th} subinterval,or in other words,the frequency f_i can be obtained for this sequence.The statistic

$$\chi_1^2 = \frac{k}{n} \sum_{i=1}^{k} (f_i - \frac{n}{k})^2 \qquad (1.7\text{-}2)$$

is used where $\underline{n}$ is the length of the sequence (the number of numbers generated) and $\underline{k}$ is the number of subintervals.This statistic has,for large $\underline{n}$,a χ^2 - distribution with k-1 degrees of freedom.

The result of the above "frequency" test,however,does not depend on the order in which the numbers are generated.Hence,other so-called serial tests are applied attempting to determine whether or not there is any correlation between numbers in the order of their generation.We form a $\underline{k}$ by $\underline{k}$ matrix by determining the frequency of numbers $f_{i,j}$,that are the number of numbers in the i^{th} subinterval followed by a number in the j^{th} subinterval($\underline{i}$ an $\underline{j}$ referring to row number and column number of the matrix, respectively).Then we compute

$$\chi_2^2 = \frac{k^2}{n} \sum_{i,j=1}^{k} (f_{i,j} - \frac{n}{k^2})^2 \qquad (1.7\text{-}3)$$

It has been determined by Good [18-19] and Juncosa [32] that, for large n, the $\chi_2^2 - \chi_1^2$ statistic has asymptotically a χ^2 distribution with k^2-k degrees of freedom, and also that $\chi_2^2 - 2\chi_1^2$ is asymptotically χ^2 with $(k-1)^2$ degrees of freedom.

Function

The TESTS subroutine

(i) generates 1024 numbers by means of the RAND generator;

(ii) computes both χ_1^2 and χ_2^2 for blocks of n = 1024 numbers, using k=8 equal subdivisions of the interval [0,1);

(iii) forms the $\chi_3^2 = \chi_2^2 - \chi_1^2$ and $\chi_4^2 = \chi_2^2 - 2\chi_1^2$ quantities; then

(iv) checks up the significance for χ_1^2, χ_3^2 and χ_4^2 with $k - 1 = 7$, $k^2-k = 56$ and $(k-1)^2 = 49$ degrees of freedom, respectively, on both the 0.05 and 0.01 level.

It is possible to test any block of 1024 successive numbers within one period (2^{21}), but when the first number of the block is different from that of the full sequence, one must generate the numbers of the sequence preceding the first number of the block in the main program using the RAND generator.

Any number of successive blocks of 1024 numbers can be tested by repetitive application of TESTS, but a suitable value (for the number of successive blocks) must be assigned to the second argument (NB) of the TESTS subroutine. In each repetition of the process RAND generates a block of the next 1024 numbers.

Use

TESTS may only be used together with RAND, therefore the RAND generator must appear in the list of the subroutines in the program. In addition, before calling TESTS adequate values must also be assigned to the (input) parameters IA and IX. (These values might be IX = 8388607 or any odd number and, IA = 2051 + 8t where t can be substituted by an integer from 0 to 64).

TESTS contains two other integer arguments NA and NB for which values must be assigned. When NA = 1, then all the 1024 numbers of the block will not only be calculated, but also printed. Otherwise (when NA any integer different from 1) the first number is merely printed. The argument NB means the number of successive blocks and it depends on the choice of the user.

Output

TESTS prints "new block" when starting a new block followed by the first integer generated and the corresponding fraction being transformed into the [0,1) interval. As indicated above, when NA = 1, the rest of the 1024 numbers in the block will also be printed.

Further on the values of the quantities χ_1^2, χ_2^2, χ_3^2 and χ_4^2 are printed. Then, after testing the quantities χ_1^2, χ_3^2 and χ_4^2 the subroutine prints either the level of significance or that there is no significance. (The significance on the level of 0.05 is still acceptable.)

Example

The program below generates the first $2^{14} = 16384$ numbers of the total period by the RAND generator without printing and testing them. TESTS is then used for generating and testing 4 successive blocks of 1024 numbers (NBB = 4). The program prints the first number of each block only (NAA = 1).

```
*MASTER STAT
      IXS = 2**23 - 1
      IAS = 2**11 + 3 + 8
      DO 1  I = 1,16384
    1 CALL RAND(IAS,IXS,URNA)
      NAA = 2
      NBB = 4
      CALL TESTS(NAA,NBB,IAS,IXS)
      STOP
      END
      SUBROUTINE RAND(IA,IX,URN)
      .
      .
      .
      END
      SUBROUTINE TESTS(NA,NB,IA,IX)
      .
      .
      .
      END
      FINISH
```

1.7.3 Generation of Stochastic Variates (Uniform, Exponential, Poisson)

Name and dummy arguments of subroutines:

1. UNIRA(A,B,X,IA,IX) (for uniform distribution),
2. EXPRA(ALAMB,X,IA,IX) (for exponential distr.),
3. POSRA(ALAM,X,IA,IX) (for Poisson distr.)

Language: FORTRAN IV.

Method

Inverse transformation is applied for generating uniform and exponential random variates.

Let us consider a statistical population having probability density function $f(x)$ and cumulative distribution function $F(x)$. Since $0 \leq F(x) \leq 1$ and $\lim_{x \to \infty} F(x) = 1$, we can find a set of uniformly distributed random numbers (u_i, where $0 \leq u_i \leq 1$) such that

$$u_i = F(x_i) \qquad (1.7\text{-}4)$$

and x_i is uniquely determined by (1.7-4), where x_i an element of the population ($0 \leq x_i < \infty$). It follows, that for any u_i

$$x_i = F^{-1}(u_i) \qquad (1.7\text{-}5)$$

and $F^{-1}(u_i)$ is an inverse transformation of u_i on the unit interval into the domain of $\underline{x}$. That is

$$u = F(x) = \int_{-\infty}^{x} f(v)dv, \qquad (1.7\text{-}6)$$

then

$$P(X \leq x) = P[u \leq F(x)] = P[F^{-1}(u) \leq x] = F(x). \qquad (1.7\text{-}7)$$

Thus the cumulative distribution function of $\underline{x}$ is $F(x)$ and consequently $F^{-1}(u)$ is a variable that has $f(x)$ as its density function.

So, if we wish to generate random variates x_i's from some particular statistical population whose density function is given by $f(x)$, we must first obtain the cumulative distribution function $F(x)$ and its inverse $F^{-1}(u)$, then use repeatedly the relation $x_i = F^{-1}(u_i)$, where $u_i (i=1,2,\ldots)$ uniformly distributed random numbers in the (0,1) interval.

(1) Uniform Distribution

The density and cumulative distribution functions over (a,b) are defined as

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & \text{otherwise} \end{cases} \qquad (1.7\text{-}8)$$

and $$F(x) = \int_{a}^{x} \frac{dv}{b-a} = \frac{x-a}{b-a} \qquad (1.7\text{-}9)$$

If $\underline{u}$ is a uniform random number over the range 0 to 1, then

$$u = F(x) = \frac{x-a}{b-a}, \text{ hence}$$

$$x = a + (b-a)u. \qquad (1.7\text{-}10)$$

The relation (1.7-10) transforms u from the (0,1) interval to the (a,b). So, by generating values of $\underline{u}$ we can obtain a set of uniform random variates (x) falling into the interval (a,b).

(2) Exponential Distribution

The corresponding density and cumulative distribution functions having exponential distribution are defined as

$$f(x) = \lambda e^{-\lambda x} \quad \text{for } x \geq 0 \text{ and } \lambda > 0, \qquad (1.7\text{-}11)$$

and

$$F(x) = \int_{0}^{x} \lambda e^{-\lambda v} dv = 1 - e^{-\lambda x}, \qquad (1.7\text{-}12)$$

where the only parameter λ describing this distribution can be expressed as the reciprocal of the expected value $E(x)$ of $\underline{x}$

$$\lambda = \frac{1}{E(x)}. \qquad (1.7\text{-}13)$$

Since $F(x)$ exists in explicit form we can accomplish the generation of

exponential random variates by inverse transformation technique.

Let v_i denote a uniform random number and set

$$v_i = 1 - e^{-\lambda x_i} \qquad (1.7\text{-}14)$$

so that $x_i = -(\frac{1}{\lambda})\log_e(1 - v_i)$. (1.7-15)

Because of the symmetry of the uniform distribution $F(x_i)$ and $1 - F(x_i)$ are interchangable, $u_i = 1 - v_i$ is itself a uniform random number, therefore

$$x_i = -(\frac{1}{\lambda})\log_e u_i \ . \qquad (1.7\text{-}16)$$

Thus for each value of u_i a unique value of x_i is determined taking only non-negative values ($\log_e u_i < 0$ for $0 \leq u_i < 1$) and having the density function $\lambda e^{-\lambda x}$.

(3) Poisson Distribution

The Poisson distribution can be given by the density function

$$f(x) = \frac{\lambda^x}{x!} e^{-\lambda} \qquad x = 0,1,2,... \qquad (1.7\text{-}17)$$

To simulate a Poisson distribution with parameter λ, we take advantage of the relationship between the exponential and Poisson distribution. If the time intervals between occurence of events, obeying some rather strict rules, are exponentially distributed, then the probability of x events occuring during a certain time is

$$\frac{\lambda^x}{x!} e^{-\lambda},$$

that is the number of events in a fixed period of time has the Poisson distribution.

For generating Poisson variates x, consider the following product of uniform random variables: $\prod_{i=0}^{n} u_i$.

The probability density function of $\prod_{i=0}^{n} u_i$ is

$$f_n(x) = \begin{cases} (-\log x)^n/n! & 0 \leq x \leq 1 \\ 0 & \text{otherwise,} \end{cases} \qquad (1.7\text{-}18)$$

and the cumulative distribution function [73, p.37]

$$F_n(x) = x \sum_{i=0}^{n} \frac{(-\log x)^i}{i!} \qquad (1.7\text{-}19)$$

Now let us test that $\prod_{i=0}^{n} u_i > e^{-\alpha}$ (1.7-20)

The probability of this is $1 - F_n(e^{-\alpha}) = 1 - e^{-\alpha} \sum_{k=0}^{n} \frac{\alpha^k}{k!}$ and, the distribution of the product of the first n uniform random variables for which $\prod_{i=0}^{n} u_i > e^{-\alpha}$ is (1.7-21)

$$F_n(e^{-\alpha}) - F_{n-1}(e^{-\alpha}) = \frac{\alpha^n}{n!} e^{-\alpha} \ . \qquad (1.7\text{-}22)$$

So n has a Poisson distribution of parameter α. With the notation used in equation (1.7-17) we must calculate the products

$$\prod_{i=0}^{x} u_i \geq e^{-\lambda} > \prod_{i=0}^{x+1} u_i \qquad (1.7\text{-}23)$$

to obtain Poisson variates x.

Use

Each of the three subroutine generates one random variate by means of one of the relations(1.7-10),(1.7-16) and (1.7-23) and we can use it in the main program (argument X)A set of stochastic variates can be obtained by repeated applications of any of these subroutines.

All of the subroutines use "RAND" for generating uniformly distributed pseudorandom numbers over the interval (0,1),therefore RAND must appear in the list of the subroutines in the program.

UNIRA and RAND,EXPRA and RAND,POSRA and RAND contain common arguments, denoted by IA and IX (the two last ones in each subroutine).Before calling any of these subroutines proper values must be assigned to IA and IX. IA = 2051 + 8t $(0 \leq t \leq 64)$.The starting value of IXmight be 8 388 607 or any odd number.IXvaries after each calling of RAND,oonsequently the starting value must be assigned to in the beginning of the main program, but this instruction must not occur in any loop.

The other arguments correspond to the parameters of the particular distribution and one of them (X) is the random variate generated in each subroutine.

UNIRA contains two arguments A and B indicating the values of the limits of the interval between which uniformly distributed variates are generated.

EXPRA also has arguments ALAMB and X as well as IA and IX.ALAMB corresponds to λ in the exponential distribution.

The argument ALAM in POSRA corresponds to the parameter λ determining the Poisson distribution.

When using any of these subroutines suitable values must be assigned to the input arguments (A and B in UNIRA,ALAMB in EXPRA and ALAM in POSRA).

The Subroutines

1.UNIRA

```
      SUBROUTINE UNIRA(A,B,X,IA,IX)
      CALL RAND(IA,IX,URN)
      X = A + (B-A)*URN
      RETURN
      END
```

2.EXPRA

```
      SUBROUTINE EXPRA(ALAMB,X,IA,IX)
      CALL RAND(IA,IX,URN)
      Y = 1./ALAMB
      X = -Y*ALOG(URN)
      RETURN
      END
```

3.POSRA

```
      SUBROUTINE POSRA(ALAM,X,IA,IX)
      A = EXP(-ALAM)
      B = 1.
      X = 0.
100   CALL RAND(IA,IX,URN)
      B = B*URN
      IF (B-A) 300,200,200
200   X = X+1.
      GOTO 100
300   RETURN
      END
```

Example

The program represents a simple application of EXPRA generating 1536 exponentially distributed random variates by means of a DO-loop and printing these values,8 in a line.

The same program can generate a set of uniformly distributed or Poisson variates if applying UNIRA or POSRA,respectively,instead of EXPRA. Of course,proper values of the input arguments (corresponding to the parameters of the particular distribution) should be assigned.

```
      MASTER VAREX
      DIMENSION SI(8)
      IXS = 8388607
      N = 1
      DO  20 I = 1,1536
      CALL EXPRA (0.1,XE,2067,IXS)
      SI(N) = XE
      N = N+1
      IF (N-9) 20,10,10
10    WRITE (2,30) (SI(K),K=1,8)
      N = 1
20    CONTINUE
      STOP
30    FORMAT (8F14.8)
      END
      SUBROUTINE RAND(IA,IX,URN)
      .
      .
      END
      SUBROUTINE EXPRA(ALAMB,X,IA,IX)
      .
      .
      END
      FINISH
```

P A R T 2

SIMULATION MODELS

2.1 Introduction

Many system simulation experiments are driven by input processes containing elements of random behavior.In such simulations,statistical reliability must be considered if experimental results are to be interpreted properly.Statistical considerations also enter into the evaluation of simulation model designs.Verification,validation and problem analysis are tasks demanding careful statistical consideration.Verification determines whether a model with a particular mathematical structure and data base actually behaves as an experimenter assumes it does.Validation tests whether a simulation model reasonably approximates a real system.Problem analysis seeks to insure the proper execution of the simulation and proper handling of its results;consequently it deals with a host of matters:the concise display of solutions,efficient allocation of computer time,proper design of tests of comparison,and correct estimates of sample sizes for specified levels of accuracy.

In other words,verification and validation insure that a simulation model is properly designed;only after a model has been verified and validated can we justifiably use a model to probe system behavior.Problem analysis mainly deals with the results of experimental probing.

Every simulation model comprises two systems - a data system and a logical system.Both present the model-builder with problems;both contribute equally to the validity of a final simulation model.

When we first look at a simulation model,we see its logical structure, the way in which a system's operations have been analysed and factored into discrete units,and how these units are combined so that the model can be made to reproduce the system's behavior.Looking at the model more deeply,we see that is contains sequences of data comparisons and logical tests. These tests cause a model to take different actions depending on numerical values that are either input from the world outside its boundaries or computed within the model.

The model's behavior is conditional by these data values,and its results are sensitive to data represantations and methods of data generation.

Almost all simulation models have queueing systems imbedded in them. Let us consider a basic queueing model with one service station ,shown in Figure 2.1.Items arrive at the station for processing with average arrival rate λ.If the station is free when a job arrives,it immediately begins service,Which is performed at an average service rate μ .A job that arrives when the service station is engaged, waits in line until it can be processed.When a job is completed,it leaves the service facility (arrow going to the right) freeing the station for another job.If jobs are waiting in the queue,one is selected for service according to a queue discipline

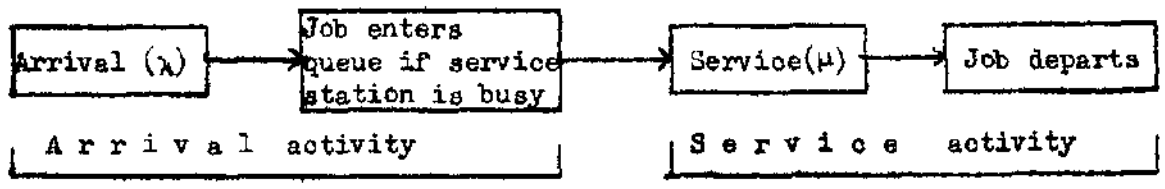


Figure 2.1

and the service station is engaged again.If no jobs are waiting,the service station remains idle until the next job arrives (systems such as this are called queueing systems).

The activities pictured are jobs arriving and jobs being serviced.Jobs arrive at the station at random times.Let the jobs'arrival times be denoted by $TP_1,\ldots,TP_{r+1}$,then the times between job arrivals are:$RAR_1=TP_2-TP_1,\ldots,$ $RAR_r=TP_{r+1}-TP_r$.Inputs to the queueing system are simulated by generating interarrival times at the service facility.When a job arrives,the time when the next job will arrive is computed by random sampling from an interarrival time distribution.Two data problems associated with this simulation are

(1)determining the correct statistical sampling distribution and,

(2)generating random samples from it.

A sequence of job arrival times constitutes a sample from a s i m u l a t i o n i n p u t p r o c e s s.Each arrival generates an interarrival time for the next job and a service time for itself.Figure 2.2 illustrates the sequence of simulation activities when an arrival event occurs:

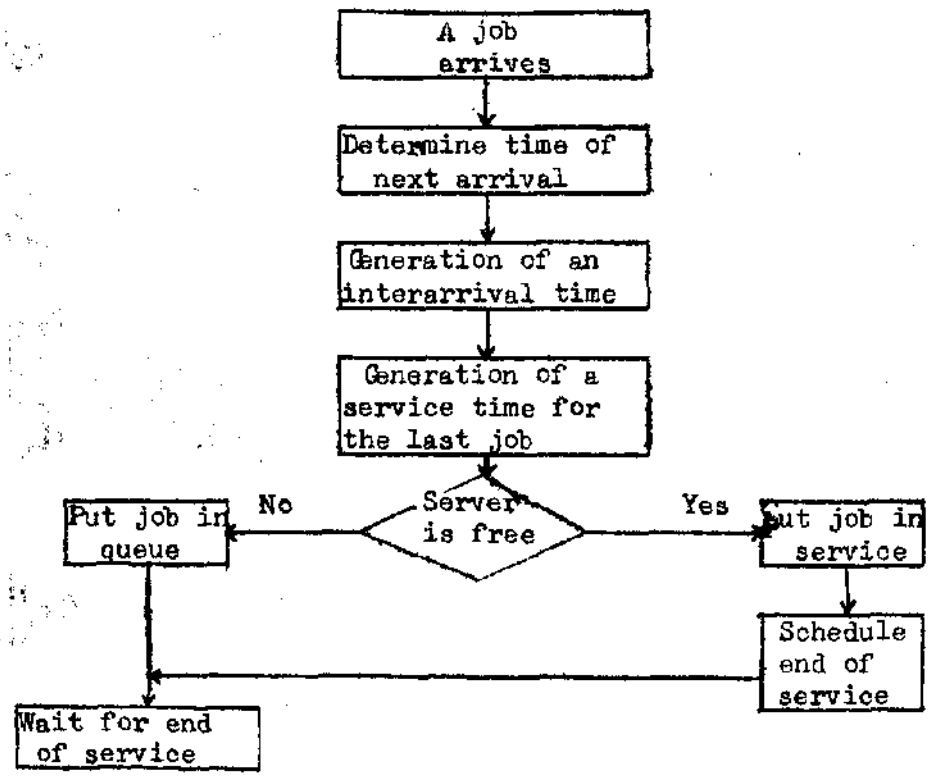


Figure 2.2

the generation of an interarrival time and a service time and placement of a new arrival in process or in queue.When a job arrives,it is placed in service if the server is free ;otherwise,it is placed in queue.

The sequence of service times ($RSR_1,\ldots,RSR_r$) also constitutes a simulation input process,as random samples are drawn from some service time

distribution whenever a job is processed.For each job that passes through the service station,two (random) quantities must be determined - RAR_k and RSR_k.In this model,both of the quantities $RAR_k$and RSR_k are generated when a job enters the station.

All simulation models are driven by some basic force,generally the arrival of a task,job or a request in the simulated system.Each job has some characteristics that contribute to its simulated behavior.A job with n characteristics can be described by a list of these characteristics,which we call an n-tuple.A job in our queueing model is characterized by a 2-tuple (RAR_k,RSR_k).A typical problem encountered when constructing a simulation model is the generation of job characteristics and while checking out a model the examination of a sequence of generated job characterizations.

The sequences of variables whose values are determined by the activities that take place within the simulation model are called s i m u l a t i o n o u t p u t p r o c e s s e s.If TS_k denotes the time that job k leaves the station,then $WT_k = TS_k-(TP_k+RSR_k)$ is the time it spends waiting for service.The sequences of $TS_1,\ldots,TS_r$ and $WT_1,\ldots,WT_r$ are simulation output processes.

A simulation model is designed to generate output processes that can be studied to observe a system's behavior as its data and/or logical structure are changed.Data influence a model through the selection of statistical sampling distributions,random sampling procedures and,activity levels.

The operating rules used to select a job from a waiting line clearly are part of the model structure and influence system behavior.A complex model generally contains many different kinds of operating rules;decision mechanisms,search-and-choice procedures,scheduling heuristics are some that are found most frequently.

A simulation model must therefore be examined in two ways. Its data must be examined,both with respect to the particular representation chosen and the way the model selects samples in its simulation process;its structure must be examined to see that mechanism have been chosen that produce correct system response.

One purpose of system simulation experiments is to compare system responses to different operating rules.Another purpose is to determine functional relationships between input factors and system response.Regardless of purpose,there are several statistical questions common to all problem analyses and to structural verification and validation as well.Some of them will be discussed in the following chapters.

2.2 Simulation of a Waiting Line With a Single Station (Model 1)

2.2.1 Abstract

A mathematical model is formulated in this chapter to describe a single-channel single-service-station waiting line situation.Variables and functional relationship among the variables are specified to determine the system.After defining the mathematical model,it is converted into a flow chart form that may appear as a computer subroutine being useful to evaluate a model of a queueing process by means of simulation on a digital computer. A detailed flow chart has been drawn that can easily be programmed on any standard computer.The object of this chapter is to describe some experiments concerning a technique for simulating the behavior of waiting lines on a digital computer.However the characteristics of the model are known and can be determined,under certain conditions,by strict analytical means,its simulated counterpart may be useful

to compare the results obtained by simulation to that of analytical methods;
to apply this simulation when one cannot fit any theoretical distribution well to the experimental data;or
to develope more complicated queueing models.

2.2.2 The Model

The model simulates a waiting line with single service station where arrivals occur singly and the queue discpline is specified according to the rule that "first in,first out".It is assumed that the variability of the time intervals at which successive items join the queue follow a probability distribution (I),and this interarrival time distribution does not change its statistical character in time i.e.is stationary.Furthermore it is supposed that all customers' service time belong to the same probability distribution and the service time distribution(II) is stationary as well.The sets of consecutive service and interarrival times are assumed to be independent.

Sample values for both of the distributions mentioned before are provided by random variate generators made for particular (theoretical or empirical) distributions and written in subroutine forms.In these external subroutines Monte Carlo technique has been applied to generate uniform random numbers,then stochastic variates from required probability distributions (Appendix p.20-23.)and,the generators can be fitted to the model to simulate stochastic processes.These subroutines are defined separately from the model and may be altered depending on the specific distributions required for sampling,in the course of simulation. This particular simulation technique lends itself to relatively easy interpretation of the simulated system.

2.2.3 The Subroutine

It was mentioned before that the model appears in flow chart form which can be handled as a subroutine.This formulated subroutine determines

(1) the total waiting time,
(2) the total idle time,
(3) the number of arrivals who wait,
(4) the number of occurences of idle time,
(5) the total arrival time,
(6) the total service time,and calculates
(7) the frequency and probability distribution of the number of items in the system (so the maximum queue size as well),
(8) the frequency and probability distribution of the number of waiting units in the system,
(9) the frequency and probability distribution of the number of waiting units when assuming non-empty queues.

The symbols used in the subroutine are defined and described below:

RAR:the random time interval between two successive arrivals,
RSR:the random service time for a particular arrival unit,
S_{max}:maximum queue size that has been reached during the simulation,
L_p:limit for the number of performances,
k:the current number of units in the system,
n:the current number of units in the queue,
i:the current number of arrivals,
i_w:the number of arrivals who wait,
i_{dc}:the number of occurences of idle time,
j:a marker to denote whether the server is active or not
($j = 0$ if idle; $j = 1$ if active)
jn:a counter to determine the correct values of the total waiting time,
TC:current time that has been reached in the simulation,
TP:predicted time of the next arrival,
TS:time for the end of a single arrival'service,
WT:waiting time for a single arrival,
ID:particular idle time,
T_k:amount of time that $\underline{k}$ units spend in the system
($k = 0,1,\ldots,S_{max}+1$)
WA_n:amount of time that $\underline{n}$ units spend in the queue
($m = 0,1,\ldots,S_{max}$)
TWT:total waiting time,
TID:total idle time,
TST:total service time,
TAT:total arrival time,
TATS:interval between the times that the first arrival enters and the last arrival finishes the service (total time of simulation).

Some obvious relationshipsare shown by the following identities:

$$TAT = \sum_{i} RAR$$

$$TST = \sum_{i} RSR$$

$$TWT = \sum_{i} WT$$

$$TID = \sum_{i_{dc}} ID$$

TATS = TAT + $(TS^{(last)} - TE)$, where $TS^{(last)}$ is the time for ending the simulation, and TE the time when the last arrival enters the system.

Using the notations introduced the following calculation are executed by the subroutine:

the probability that there are k units in the system

$$p_k = \frac{T_k}{TATS}, \quad (k = 0,1,\ldots,S_{max}+1)$$

the probability that there are n units in the queue

$$q_n = \frac{WA_n}{TATS}, \quad (n = 0,1,\ldots,S_{max})$$

the probability that there are w units in a non-empty queue

$$r_w = \frac{WA_w}{SUM}, \quad (w = 1,2,\ldots,S_{max}) \text{ and, } \quad SUM = \sum_{w=1}^{S_{max}} WA_w .$$

It is obvious that

$$\sum_{k=0}^{S_{max}+1} p_k = 1, \quad \sum_{n=0}^{S_{max}} q_n = 1, \text{ and } \quad \sum_{w=1}^{S_{max}} r_w = 1,$$

(and $T_m = WA_{m-1}, p_m = q_{m-1}$ when $m > 1$).

Other characteristics of the system are calculated as follows:

(1) Expected number of units in the system: $E(k) = \sum_{k=0}^{S_{max}+1} kp_k$

(2) Expected queue length: $E(n) = \sum_{n=0}^{S_{max}} nq_n$

(3) Expected length of non-empty queues: $E(w/w>0) = \sum_{w=1}^{S_{max}} wr_w$

(4) Expected time which an arrival spends in system:

$$E(T_{ws}) = \frac{TWT+TST}{i}$$

(5) Expected waiting time of an arrival:

$$E(WT) = \frac{TWT}{i}$$

(6) Expected waiting time of an arrival that waits:

$$E(WT/WT > 0) = \frac{TWT}{i_w}$$

(7) Expected idle time:

$$E(ID) = \frac{TID}{i}$$

(8) Expected fraction of total time service-station is idle:

$$E(f_{id}) = p_o$$

(9) Expected fraction of total time service station is occupied:

$$E(f_{occ.}) = 1-p_o \ .$$

Before describing the flow chart of the model a simple diagram is given (Figure 2.2-1) to show a part of the process and to indicate the current values of some characteristic variables. In this case neither the interarrival nor the service time distributions follow any theoretical distribution. The variable $TP^{(i)}$ indicates the time when the i^{th} unit enters the system and $TS^{(i)}$ means the time for ending the service of the i^{th} unit.

2.2.4 The Flow Chart

The flow chart consists of several main parts:

(1) generation of stochastic variates for arrival times (block 9),

(2) generation of stochastic variates for service times (block 15),

(3) calculation of waiting times (block 22),

(4) calculation of idle times (block 30),

(5) condition for finishing the simulation process (blocks 8 and 28),

and after ending the simulation

(6) to perform the calculations using the data collected and calculated so far (block 37).

At the start of the simulation, that is when $i = 1$, the queue is empty, the service station is idle and the total arrival time, total waiting time, total idle time, total service time and the length of the queue are all assumed to be equal to zero. In other words the following starting conditions are assumed to be in effect:

$$TAT = 0$$
$$TWT = 0$$
$$TID = 0$$
$$TST = 0$$
$$n = 0$$

and furthermore

$$T_k = 0 \quad (k = S_{max}+1)$$
$$TC = 0$$
$$TP = 0$$
$$i_w = 0$$
$$i_{do} = 0,$$

and so the first arrival enters the system at zero time. The starting conditions appear in block 1 by setting all the listed variables equal to 0.

Due to the start of the process, $j = 0$ (the service station is idle) and a large number is set to TS indicated by $TS = \infty$ in block 2, to make

Current values of some variables during the simulation

Arrival time (TP)	$TP^{(1)}$	$TP^{(2)}$		$TP^{(3)}$	$TP^{(4)}$		$TP^{(5)}$	$TP^{(6)}$		$TP^{(7)}$			$TP^{(8)}$			$TP^{(9)}$		$TP^{(10)}$
Time for the end of service (TS)		$TS^{(1)}$			$TS^{(2)}$			$TS^{(3)}$		$TS^{(4)}$		$TS^{(5)}$	$TS^{(6)}$		$TS^{(7)}$	$TS^{(8)}$		$TS^{(9)}$	$TS^{(10)}$
Number of units in the system (k)	1	2	1	2	3	2	3	4	3	4	3	2	3	2	1	1	0	1	
Number of units in the queue (n)	0	1	0	1	2	1	2	3	2	3	2	1	2	1	0	0	0	0	
Waiting time (WT)		WT_1		WT_1	WT_2	WT_1	WT_2	WT_3	WT_2	WT_3	WT_2	WT_1	WT_2	WT_1					
Idle time (ID)																	ID		
Time that k units spend in the system (T_k)	T_1	T_2	T_1	T_2	T_3	T_2	T_3	T_4	T_3	T_4	T_3	T_2	T_3	T_2	T_1	T_1	T_0	T_1	

Figure 2.2-1

its value larger then TP in order to generate arrival time in the first run and whenever idle time occurs.

Customers are served in the order of their arrivals,and the simulation is finished when the number of generated arrival times exceeds a predetermined limit (L_p).The number of performances of the process is controlled by a counter (i),which is incremeneted by one each time a new arrival enters the system.When $i > L_p$ the process stops as soon as the items that have entered the system have been serviced.

As a unit arrives,the time for arrival of the next unit can be predicted by drawing a sample from I (see p.27)and as a unit commences service the time at which it will be completed can be predicted by drawing a sample from II.At any stage the next event to occur is found from the minimum of these two sample values (block 5).In other words,the subroutine is constructed so that the next arrival time is generated only if the current value of the total service and idle time is greater than or equal to the present value of TP.So the condition for generating the next value of TP is

$$TST_{current}+TID_{current} \geq TP.$$

On the contrary,a new service time is generated only,if

$$TST_{current}+TID_{current} < TP.$$

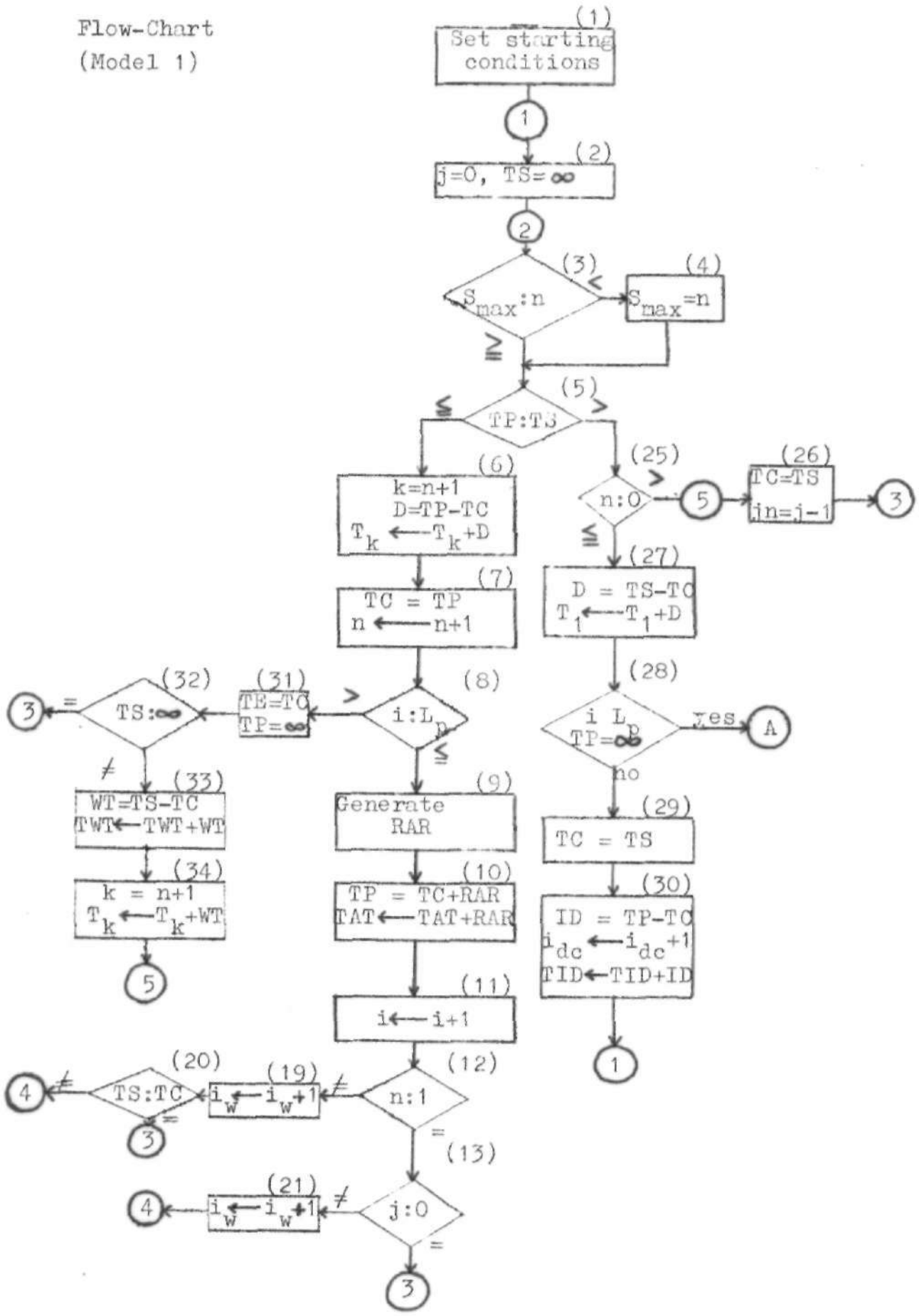
In the course of the simulation run the distinct time periods (T_k) that $\underline{k}$ units spend in the system are calculated every time an item enters,starts or ends the service (blocks 6,24,27 and 34 are provided for this purposes).

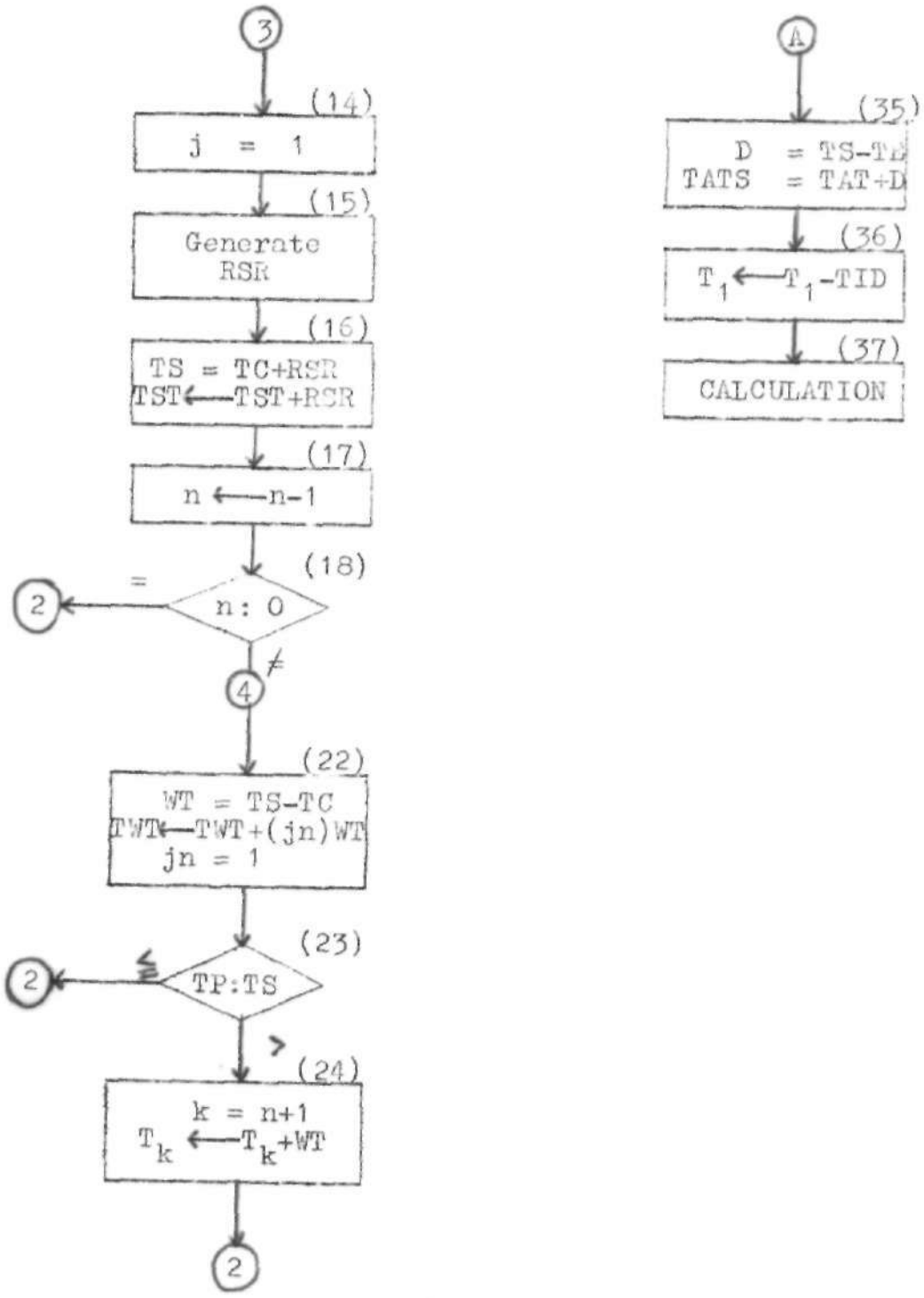
Before generating the next value of the arrival time the current value of TP is assigned to TC,and the new value of TP is obtained by adding the sampling value (RAR) to the previous value of TP.Then the number of items in the queue (n) and the number of arrivals (i) are increased by one (blocks 7,10,11).(When i = 1,RAR denotes the time interval between the first and second arrivals.)After this checks are made to decide whether to continue the process.If there are more than 1 waiting items or the service station is occupied,then the number of waiting units is increased by 1 (blocks 19,21),and a control is transferred to label 4 (possibly to 3) to calculate the current and the total waiting times (block 22).If n = 1 and the service station is idle (blocks 12,13), a service time (RSR) is generated,sampling from the distribution II by an external subroutine (block 15).This time the current number of waiting units (n) decreases by one (block 17).

Waiting time is calculated if n > 0.Then a prompt or indirect return to the front of the next run is made depending on the decision arrived at in block 23.

The process may be repeated or can be branched out at block 5 going on to block 25.It is obvious that new service time must be generated when there are waiting units in the system (n > 0),so the simulation carries on at label 3.Otherwise,the subroutine calculates the current

Flow-Chart
(Model 1)




3
(14)
j = 1
(15)
Generate RSR
(16)
TS = TC+RSR
TST←TST+RSR
(17)
n ←n-1
(18)
=
2
n : 0
≠
4
(22)
WT = TS-TC
TWT←TWT+(jn)WT
jn = 1
(23)
≤
2
TP:TS
>
(24)
k = n+1
T_k ←T_k+WT
2
A
(35)
D = TS-TL
TATS = TAT+D
(36)
T_1 ←T_1-TID
(37)
CALCULATION

and total idle times and increases the number of idle times (block 30). Every time one returns to label 1 TS < TP, all the arrivals but one have been serviced, so a large value is set to TS (to be larger than TP) and the process is repeated.

When the condition $i > L_p$ is true, the system may be in two states:

(1) All the units that have arrived - except the last - have been serviced, or

(2) There are one or more units waiting for service or a service is taking place.

According to these possibilities, the subroutine generates one more (the last) service time and finish-es the simulation run, or so many service times are generated as are necessary to service all the units waiting for service. By using blocks 31 and 32 the subroutine assures conditions for the appropriate end. Block 35 is used for calculating the total time that has elapsed between the start and the end of the simulation.

The final value of T_1 can be obtained by subtructing TID from it, since the time intervals for idle time appear twice, both in block 30 block 6.

Finally the subroutine executes the calculation by means of the relationships among the variables which have been defined peviously.

2.2.5 Check on the Validity

The validity of the model has been tested in the case when both the interarrival and service times follow negative exponential distribution with a given expected value. These distributions are most frequently encountered and theoretical results for such parameters are easily available. It is known that if the probability of a certain number of arrivals in unit time has Poisson distribution, the interarrival time distribution follows exponential distribution. The probability density function of the interarrival time of duration $\underline{t}$ is $\lambda e^{-\lambda t}$, where λ is the mean arrival rate, and $1/\lambda$ is the mean time between arrivals. The same density function has been applied for servicetime of duration $\underline{t}$ with parameter μ being the mean service rate and $1/\mu$ the mean service time per customer.

By using the notation introduced beforehand the corresponding theoretical values (supposing exponential distribution) can be calculated from the following identities when $\lambda/\mu < 1$, [64, p.131-133]:

$$E(k)_{theoretical} = \frac{\lambda}{\mu-\lambda}$$

$$E(n)_t = \frac{\lambda^2}{\mu(\mu-\lambda)}$$

$$E(w/w>0)_t = \frac{\mu}{\mu-\lambda}$$

$$E(T_{ws})_t = \frac{1}{\mu-\lambda}$$

$$E(WT)_t = \frac{\lambda}{\mu(\mu - \lambda)}$$

$$E(WT/WT>0)_t = \frac{1}{\mu - \lambda}$$

$$E(f_{occ.})_t = \lambda/\mu$$

$$E(f_{id.})_t = 1 - \lambda/\mu, \text{and}$$

the probability that there are $\underline{k}$ units in the system:

$$p_{k_t} = (1 - \lambda/\mu)(\lambda/\mu)^k \quad (k \geq 0).$$

The values of characteristic quantities of the model obtained by simulation using various values of the parameters λ and μ have been compared to the theoretical values.Later experiments were repeated with different starting values in the multiplicative congruential random number generators to test the effects of changes.Then the model has been tested by altering the values of L_p,that is the limit for the number of performances.By increasing the value of L_p the results are getting better.In each case the model provided good results.

The comparison of the values of some variables,and the probability distribution of the number of itmes,with a few varying values of the parameters λ and μ and with different values of L_p (200 and 1200) are listed in Table 2.2-1.

It is worth mentioning that the subroutine can be enlarged and tests can be performed to check the answers when the interarrival and service time distributions follow other probability distributions.

Comparison of the values of some variables and, probability distribution of the number of items in the system using the same starting conditions for both calculations ($\psi = \lambda/\mu$)

	ψ = .22			ψ = .4			ψ = .47			ψ = .65		
	Theoret-ical	Simulated L_p=200	L_p=1200	Theoret-ical	Simulated L_p=200	L_p=1200	Theoret-ical	Simulated L_p=200	L_p=1200	Theoret-icaI	Simulated L_p=200	L_p=1200
E(k)	.29	.31	.30	.67	.62	.70	.88	.80	.95	1.83	1.93	1.84
E(n)	.06	.06	.07	.27	.23	.30	.41	.35	.49	1.18	1.28	1.20
E(w/w > 0)	1.29	1.13	1.28	1.67	1.43	1.92	1.88	1.56	2.21	2.83	2.87	2.95
E(T_{we})	2.86	2.48	2.52	2.22	1.69	2.08	1.25	.96	1.24	4.35	4.56	4.22
E(WT)	.64	.20	.27	.89	.43	.72	.58	.33	.56	2.81	2.94	2.64
E(WT/WT > 0)	2.86	.81	1.21	2.22	1.09	1.83	1.25	.71	1.20	4.35	4.77	4.10
E(f_{id})	.78	.75	.77	.60	.61	.60	.53	.55	.54	.35	.36	.36
E($f_{occ.}$)	.22	.25	.23	.40	.39	.40	.47	.45	.46	.65	.64	.64
Probabilities n												
1	.7778	.7502	.7680	.6000	.6125	.6010	.5333	.5481	.5345	.3539	.3550	.3650
2	.1729	.1951	.1790	.2400	.2235	.2425	.2489	.2299	.2462	.2286	.1982	.2269
3	.0384	.0483	.0418	.0960	.1102	.0923	.1161	.1296	.1110	.1477	.1286	.1536
4	.0085	.0057	.0082	.0384	.0404	.0327	.0542	.0676	.0509	.0955	.1073	.0881
5	.0019	.0007	.0025	.0154	.0105	.0145	.0253	.0192	.0237	.0617	.0574	.0563
6	.0004	-	.0004	.0061	.0029	.0067	.0118	.0045	.0143	.0399	.0535	.0316
7	.0001	-	.0001	.0024	-	.0037	.0055	.0011	.0064	.0258	.0560	.0238
8				.0010	-	.0008	.0026	-	.0010	.0166	.0339	.0131
9				.0004	-	.0019	.0012	-	.0033	.0108	.0101	.0122
10				.0002	-	.0018	.0006	-	.0025	.0070	-	.0059
11				.0001	-	.0003	.0003	-	.0024	.0045	-	.0071
12						.0011	.0001	-	.0010	.0029	-	.0052
13						.0007	.0001	-	.0023	.0019	-	.0067
14									.0005	.0012	-	.0023
15										.0008	-	.0008
16										.0005	-	.0005
17										.0003	-	.0009
18										.0002	-	-
19										.0001	-	-
20										.0001	-	-

Table 2.2-1

2.3 Simulation of a Waiting Line With a Single Station and a Limited Number of Units (Model 2)

2.3.1 The Model

A birth and death process in which both arrival and service times are Poisson distributed is given by the system

$$\frac{d}{dt}p_k(t) = -(\lambda_k + \mu_k)p_k(t) + \lambda_{k-1}p_{k-1}(t) + \mu_{k+1}p_{k+1}(t) \qquad k > 0$$

$$\frac{d}{dt}p_o(t) = -\lambda_o p_o(t) + \mu_1 p_1(t)$$

where λ_k and μ_k are functions of $\underline{k}$ (the number of units in the system), constitutes the mathematical model of the waiting line in the case of a single station.

The source considered in the case of a limited number (m) of units has a special property: the rate of arrivals is proportional to the number of units in the source. The parameters of the birth and death process now are defined such that:

$$\left.\begin{aligned}\lambda_k &= m\lambda\\ \mu_k &= 0\end{aligned}\right\} k = 0$$

and

$$\left.\begin{aligned}\lambda_k &= (m-k)\lambda\\ \mu_k &= \mu\end{aligned}\right\} 0 < k \leq m$$

By these conditions we have the equations

$$p_o'(t) = -m\,\lambda p_o(t) + \mu p_1(t), \qquad k = 0$$

$$p_k'(t) = -[(m-k)\lambda + \mu]\,p_k(t) + (m-k+1)\lambda p_{k-1}(t) + \mu p_{k+1}(t),$$

when $0 < k < m$

$$p_m'(t) = -\mu p_m(t) + \lambda p_{m-1}(t), \qquad \text{when } k = m.$$

In a permanent system $p_k(t) = p_k$ $k = 0,1,\ldots,m$ and the former equations become

$$m\lambda p_o = \mu p_1$$

$$[(m-k)\lambda + \mu]p_k = (m-k+1)\lambda p_{k-1} + \mu p_{k+1}$$

$$\lambda p_{m-1} = \mu p_m.$$

The recursive formula

$$p_k = (m-k+1)\psi\, p_{k-1} \qquad 0 < k \leq m \qquad (2.3\text{-}1)$$

is obtained from the previous system ($\psi = \lambda/\mu$ being the traffic intensity).

From (2.3-1) we have

$$p_k = \frac{m!}{(m-k)!}\psi^k p_o \qquad 0 < k \leq m, \qquad (2.3\text{-}2)$$

and

$$p_o = \frac{1}{1 + \sum_{k=1}^{m}\frac{m!}{(m-k)!}\psi^k} \qquad (2.3\text{-}3)$$

By means of the equations (2.3-1) and (2.3-3) the following characteristics of the system can be derived:

Expected number of units in the system:

$$E(k) = m - \frac{1}{\psi}(1 - p_o) \qquad (2.3\text{-}4)$$

Expected number of units in the waiting line:

$$E(n) = m - \frac{1 + \psi}{\psi}(1 - p_o) \qquad (2.3\text{-}5)$$

Expected number of units not in the system:

$$m - E(k) = \frac{1 - p_o}{\psi} \qquad (2.3\text{-}6)$$

Expected waiting time in the system:

$$E(WT) = \frac{1}{\mu}\left(\frac{m}{1 - p_o} - \frac{1}{\psi}\right) \qquad (2.3\text{-}7)$$

Expected waiting time in the line:

$$E(WT/WT > 0) = \frac{1}{\mu}\left(\frac{m}{1 - p_o} - \frac{1 + \psi}{\psi}\right) \qquad (2.3\text{-}8)$$

Expected fraction of total time service station is idle:

$$E(f_{id}) = p_o \qquad (2.3\text{-}9)$$

The probability of waiting any time at all (expected fraction of total time service station is occupied):

$$E(f_{occ.}) = p(>0) = 1 - p_o \qquad (2.3\text{-}10)$$

[33,p.366-367].

2.3.2 The Flow Chart

The present simulation model is an extension of Model 1,with the restriction that the number of waiting units cannot arbitrarily increase. As soon as m takes up a predetermined value (m_p),no further unit is allowed to enter the system,until the next service starts;however a new arrival time generated by a stochastic variate generator occurs,possibly before the start of the next service.

The new variables of Model 2 are m,m_p and L_s,where

m: the number of maximum possible units in the system

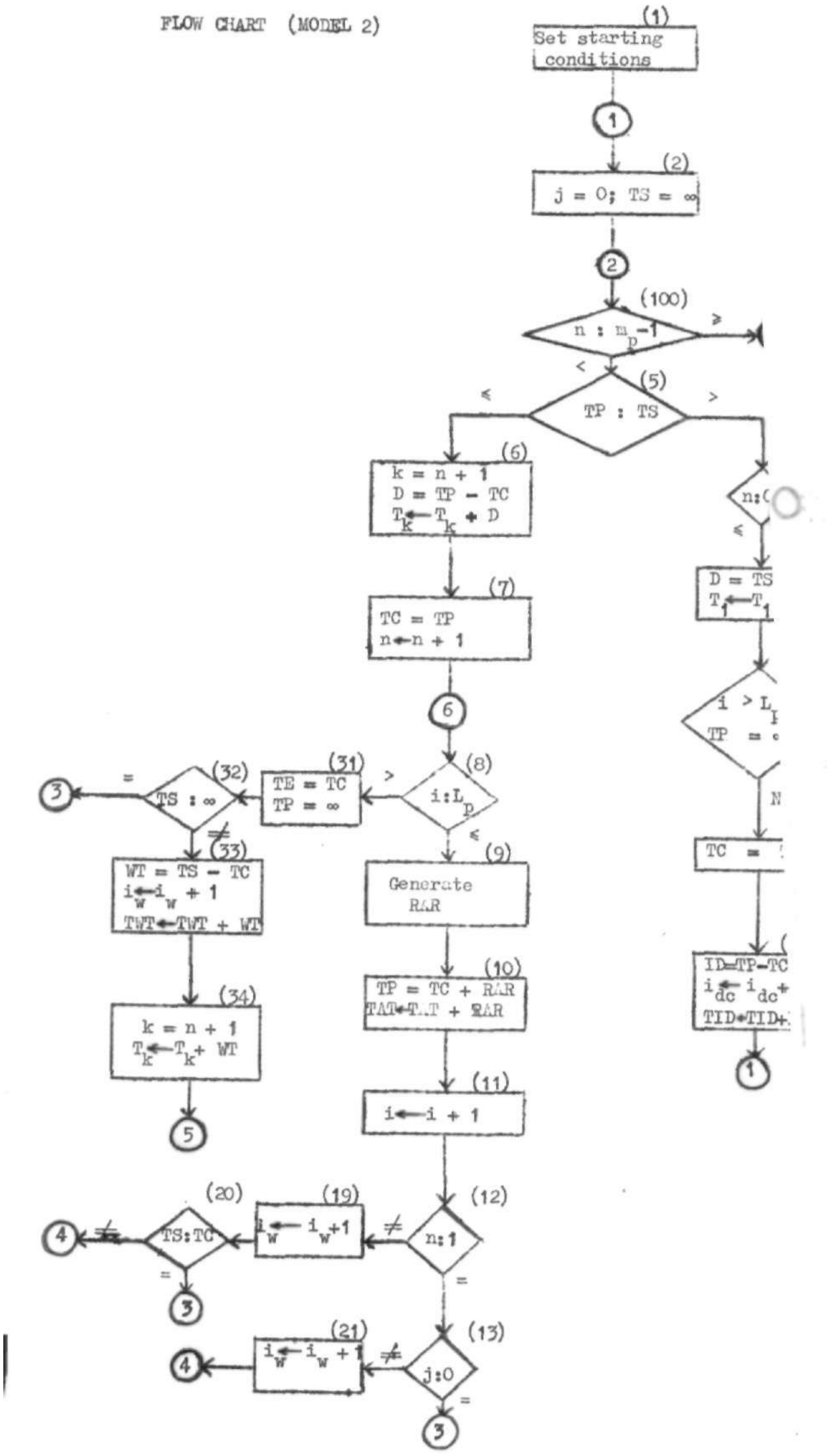
m_p:the number of maximum possible units in the queue ($m_p = m - 1$)

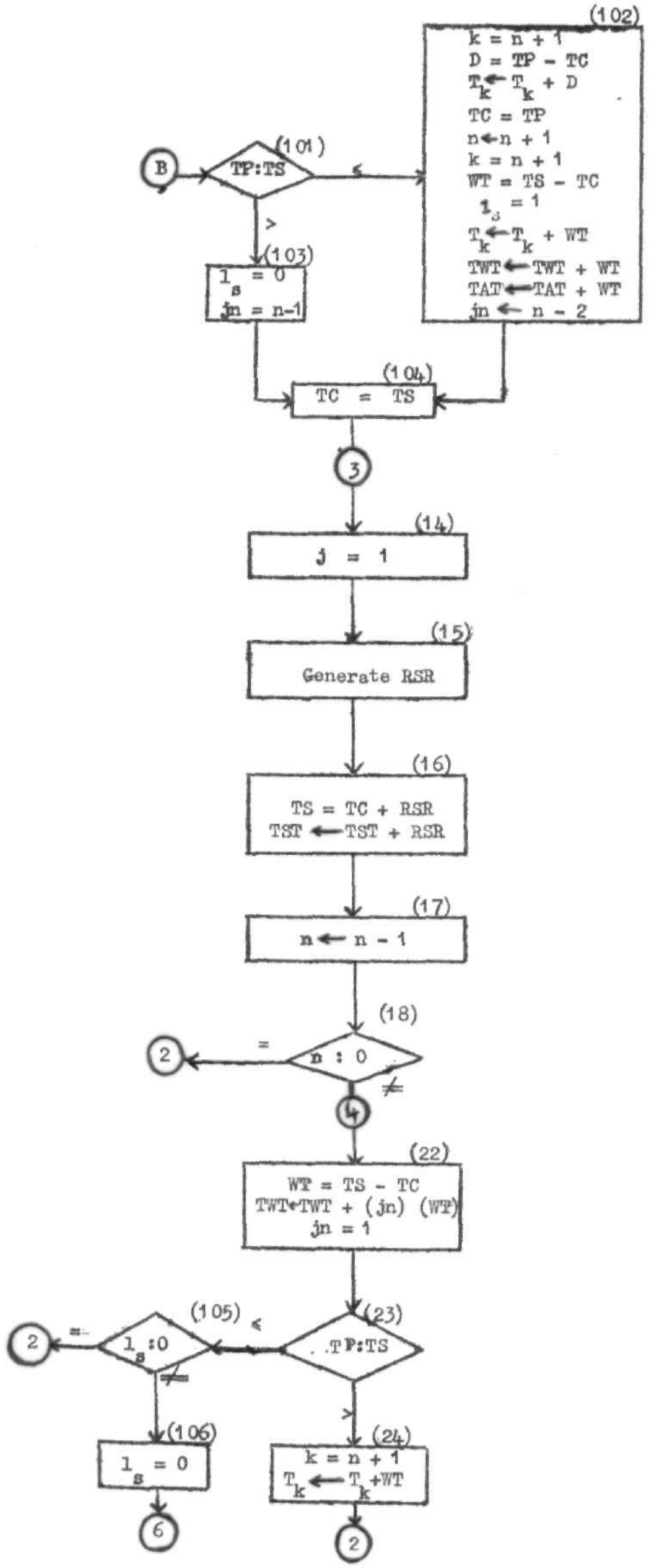
l_s: a marker to indicate whether the system is in state with maximum queue size (if there are m units in the queue then $l_s = 1$,otherwise $l_s = 0$).

Blocks 3 and 4 of Model 1 are unnecessary,nevertheless we keep the old numbering of the blocks (from Model 1) and the new blocks are numbered from 100 - 106.In addition,a new label (designated by 6) is placed between blocks 7 and 8.

The process branches at block 100, when $n = m_p - 1$.At this moment the system may be in two states according to the momentary values of TP and TS.When $TP_{current} \leq TS_{current}$ the process proceeds in block 102.The number of waiting units is increased by 1,that is it obtains the maximum

FLOW CHART (MODEL 2)




B
(101)
TP:TS
<
=
>
(102)
k = n + 1
D = TP - TC
$T_k \leftarrow T_k + D$
TC = TP
n ← n + 1
k = n + 1
WT = TS - TC
$1_s = 1$
$T_k \leftarrow T_k + WT$
TWT ← TWT + WT
TAT ← TAT + WT
jn ← n - 2
(103)
$1_s = 0$
jn = n-1
(104)
TC = TS
3
(14)
j = 1
(15)
Generate RSR
(16)
TS = TC + RSR
TST ← TST + RSR
(17)
n ← n - 1
(18)
n : 0
2
4
(22)
WT = TS - TC
TWT←TWT + (jn) (WT)
jn = 1
(105)
1_s:0
(23)
TP:TS
(106)
$1_s = 0$
(24)
k = n + 1
$T_k \leftarrow T_k$ +WT
6
2

value ($n = m_p$) and,based on the instructions of block 102,all characteristic data for every event will be calculated until the time $TS_{current}$ required for further calculations.The program continues at label 3 with generation of a new service time,and the start of the next interarrival time will be calculated from the recently generated service time.

Let us suppose that m machines work in a workshop,machine-breakdowns occur according to Poisson distribution,and one mechanic repairs broken down machines.In the critical case,when none of the machines works (1 machine is under repair and m_p machines are waiting for repair),the poisson distribution is left out of consideration until the current repair is finished.The next arrival time is generated from the time when at least 1 machine starts to work.

Every time the above "critical case" occurs ($m = m_p$) the number of arrivals obtained by simulation departs from the Poisson-rule to a certain extent.Considering the whole simulation only the distribution of arrivals between the "critical cases" can be taken into account as strictly Poisson distribution.The distribution taking place actually during the total simulation may be a better or worse approximation to it.Obviously, the results obtained by simulation reflect the real events,and the analitical formulae,inthis special case are only approximations to it.

When the event $n = m_p$ occurs,the process,according to the flow-chart, branches to label 6 (and not to label 2,since at this moment TC > TP and T_k would take up a false value in block 6).This branch is guaranteed by introduction of blocks 105 and 106,where there is a check on the value of l_s and it is set to 0 if it was 1 previously.On the contrary when

$$TP_{current} > TS_{current}$$

the relation $n < m_p$ is true,and the process carries on,generating new service time after passing blocks 101,103 and 104.

The remaining part of the flow-chart is self-explanatory.

Figure 2.3-1 displays a possible sequence of events of the simulation model by presenting current values of some essential variables when m=5. On this figure the first"critical case" ($n = m_p - 1$,and $TP \leq TS$) occurs when units 5 and 6 enter the system ($TP^{(5)}$, $TP^{(6)}$,$TP^{(7)}$),n takes up the predetermined maximum value (n=4) and the process (passing blocks 101 - 102 - 104) continues.After this sequence of events the next interarrival time is considered to be $TP^{(7)} - TS^{(2)}$,so the period $TS^{(2)} - TP^{(6)}$ is left out.

Another series of events can also be followed in the case of $n=m_p-1$, when $TP_{current} > TS_{current}$ (it is determined by $TP^{(4)}$,$TS^{(1)}$ and $TP^{(5)}$).

The analytical formulae necessary for calculation are given from (2.3-1) - (2.3-10) on page 38-39 (this chapter).The corresponding quantities of the system obtained by simulation are calculated by the same formulae as in the previous model,that is p_k,E(k),E(n),E(WT),E(WT/WT > 0),$E(f_{id.})$ and $E(f_{occ.})$,but now $k = 0,1,\ldots,m$ and $n = 0,1,\ldots,m_p$ (p.29-30).

In addition the expected number of units not in the system is

Simulation when $m_p = 4$

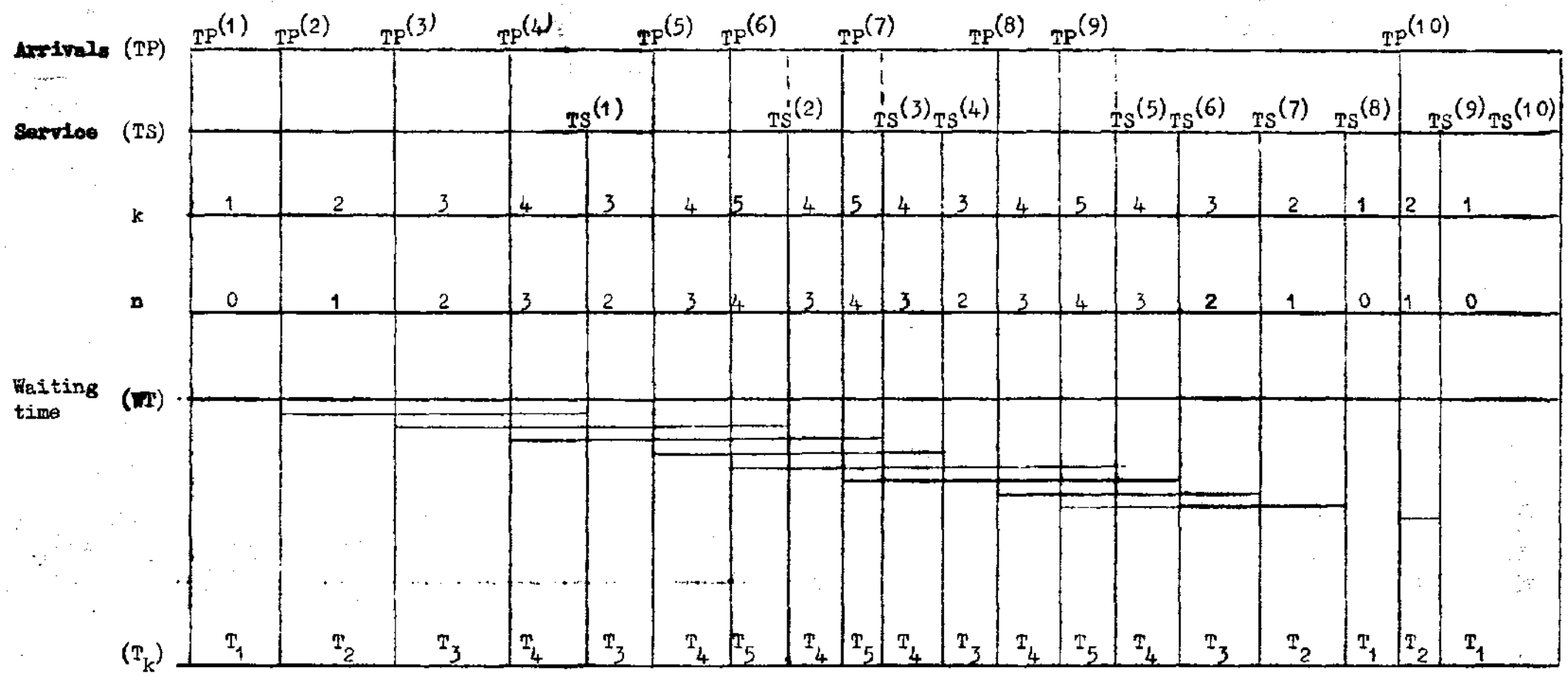


Figure 2.3-1

calculated as

$$m - \sum_{k=0}^{m} k p_k . \qquad (2.3\text{-}11)$$

2.3.3 Check on the Validity

The analytical formulae giving the base for comparison areas follows [33, p.109]

$$p_o = \frac{1}{1 + \sum_{k=1}^{m} \frac{m! \Psi^k}{(m-k)!}} \qquad (2.3\text{-}12)$$

and

$$p_k = \frac{m! \Psi^k}{(m-k)!} p_o , \qquad 0 < k \leq m \qquad (2.3\text{-}13)$$

(or the latter in recursive form is $p_k = (m-k+1)\Psi\, p_{k-1}$). These formulae can be easily derived. It follows from them that the theoretical probability that there are k units in the system, and the other theoretical results, depend upon $\Psi(=\lambda/\mu)$ and m. By proper selection of these parameters there would be short waiting times, and the probability that there are waiting units in the system in not very large. (Considering the former example with a workshop having m identical machines each operating independently of the other, one mechanic being available and supposing Poisson breakdowns and service times, breakdowns are rare and repairs do not take long). E.g. when $\Psi = .1$ ($\lambda = .1$, and $\mu = 1.$), and $m=5$ then $p_o = .5639, p_1 = .2820, p_2 = .1128, p_3 = .0338, p_4 = .0068, p_5 = .0007$ or in another case having the same Ψ and $m=10$ then $p_o = .2146, p_1 = .2146, p_2 = .2931, p_3 = .1545, p_4 = .1080, p_5 = .0649, p_6 = .0324, p_7 = .0130, p_8 = .0039, p_9 = .0008, p_{10} = .0001$.

When $\Psi > 1$, and either Ψ or m, or both of them are inceasing we are dealing with a congestion phenomenon, so the number of machines out of service increase considerably (that is the case with too many machines to be repaired by one mechanic, or frequent breakdawns and long repair-time), as the following table shows:

$m=5$

Ψ	p_o	p_1	p_2	p_3	p_4	p_5	$\sum_{k=2}^{5} p_k$
1.01	.0023	.0124	.0533	.1714	.3672	.3934	.9853
1.2	.0015	.0087	.0419	.1509	.3623	.4347	.9898
2.33	.0001	.0009	.0086	.0598	.2792	.6514	.9990

The probability that there is at least 1 unit in the waiting line is very high (last column), and the probability that the service station is idle is practically 0.

The expected waiting time in the system also increases, when $\Psi > 1$. The table below shows some values of E(WT) (following on the next page) with

different values of the parameters Ψ and m:

m \ Ψ	1.17	1.21	1.36	1.57	2.33
5	3.61	4.40	7.77	12.47	30.48
6	4.48	5.45	9.58	15.33	37.14
10	7.96	9.66	16.85	26.75	63.80

Complete simulation has been carried out for each combination of the values of the parameters Ψ, and m given below (with 3 different starting values for the stochastic-variate generators):

m = 5,6,7,8,9,10

λ	.1	.3	.5	.7	.9	.15	1.1	1.3	1.5	.3	.45
μ	1.0	1.2	1.4	1.6	1.8	.30	2.0	2.2	2.4	.4	.5
Ψ	.1	.25	.357	.44	.5	.5	.55	.59	.625	.75	.9

λ	.6	.75	.9	1.05	1.35	1.2	1.15	.95	.75	.55	.35
μ	.6	.70	.8	.9	1.15	1.0	.95	.75	.55	.35	.15
Ψ	1.0	1.07	1.13	1.17	1.17	1.2	1.21	1.267	1.36	1.57	2.33

On the comparison of the analytical and simulated values we have to consider the standpoint that the analytical probability distribution of the number of units being in the system has been gained from the hypothesis that the number of units that entered the system during a certain period follows Poisson distribution. Nevertheless in practice a rough approximation of the above mentioned distribution comes into being; namely, whenever the number of waiting units attains the critical value ($n = m_p$), new arrivals may not enter the system until n decreases by 1. However a new input unit may enter before this time, according to the Poisson arrival, whereas it may not happen on the basis of the conditions of the model, because the number of arrivals would be greater than the predetermined value of m_p. Owing to these reasons one cannot speak of exact Poisson arrival in practice during the actual operation of the model.

It follows from this fundamental fact that there is a great departure in many cases between the quantities calculated analytically and the results obtained by simulation, especially when $\Psi \leq 1$. One cannot expect as good an argument between the analytical and simulated results, as in the case of Model 1, where there was an excellent fitting between the corresponding quantities.

- 46 -

m = 5

	n	f_n	k	T_k	P_k^S	P_k^A		E(k)	E(n)	E(WT)	E(WT/WT>0)	EPθC
	-	-	0	14.1	.0731	.0016	*					
	0	45	1	13.4	.0694	.0094	A	4.15	3.15	3.61	2.74	.998
Ψ =1 .173	1	83	2	27.5	.1420	.0440						
	2	85	3	37.1	.1920	.1548	**					
	3	118	4	45.7	.2362	.3635	S	3.31	2.38	2.17	2.35	.927
	4	70	5	55.9	.2873	.4267						
	-	-	0	17.1	.0586	.0012						
	0	34	1	15.3	.0524	.0074	A	4.21	3.21	5.62	4.29	.999
Ψ=1 .267	1	75	2	36.6	.1251	.0372						
	2	91	3	56.8	.1944	.1415						
	3	125	4	71.0	.2431	.3585	S	3.49	2.55	3.58	3.84	.941
	4	76	5	95.4	.3264	.4342						
	-	-	0	13.3	.0464	.0008						
	0	30	1	17.7	.0451	.0058	A	4.27	3.27	7.77	5.95	.999
Ψ=1 .36	1	65	2	39.9	.1013	.0316						
	2	88	3	74.5	.1895	.1292						
	3	135	4	99.6	.2533	.3523	S	3.65	2.70	5.14	5.47	.954
	4	83	5	143.3	.3644	.4803						
	-	-	0	17.6	.0290	.0005						
	0	24	1	19.9	.0328	.0036	A	4.36	3.36	12.47	9.62	1.000
	1	49	2	44.7	.0737	.0227						
Ψ=1 .571	2	83	3	101.4	.1670	.1072						
	3	151	4	159.4	.2626	.3368	S	3.91	2.94	8.69	9.10	.971
	4	94	5	264.0	.4349	.5292						
	-	-	0	13.9	.0100	.0001						
	0	13	1	18.6	.0134	.0009	A	4.57	3.57	30.48	23.81	.990
Ψ=2.33	1	29	2	43.2	.0311	.0086						
	2	62	3	144.4	.1040	.0598						
	3	171	4	330.6	.2380	.2792	S	4.36	3.37	23.20	23.80	1.000
	4	126	5	838.2	.6035	.6514						
m = 7												
	-	-	0	10.7	.0547	.0000						
	0	26	1	8.6	.0450	.0002	A	6.15	5.15	5.35	4.48	.945
	1	44	2	12.4	.0652	.0016						
Ψ=1 .173	2	44	3	22.8	.1202	.0094						
	3	49	4	21.0	.1105	.0440						
	4	70	5	22.1	.1167	.1548						
	5	107	6	41.5	.2190	.3634	S	4.76	3.81	3.57	3.80	1.000
	6	61	7	51.0	.2687	.4266						
	-	-	0	10.6	.0176	.0000						
	0	14	1	9.8	.0163	.0000	A	6.36	5.36	18.18	15.33	1.000
	1	23	2	20.6	.0344	.0005						
Ψ=1 .571	2	30	3	41.1	.0685	.0036						
	3	49	4	65.7	.1095	.0227						
	4	68	5	65.6	.1093	.1072	S	5.56	4.58	13.64	14.13	.982
	5	129	6	133.8	.2230	.3368						
	6	88	7	202.7	.4213	.5292						
	-	-	0	8.7	.0063	.0000						
	0	5	1	9.4	.0068	.0000	A	6.57	5.57	43.81	37.14	1.000
	1	9	2	15.6	.0113	.0001						
Ψ=2.33	2	13	3	23.1	.0167	.0009						
	3	32	4	79.	.0571	.0086						
	4	68	5	124.3	.0898	.0598	S	6.22	5.23	36.19	36.74	.994
	5	159	6	300.4	.2171	.2792						
	6	115	7	823.2	.5949	.6514						

TABLE 2.3-1

* A : Analytical

** S : Simulated

Obviously the number of occurences of the critical case ($n = m_p$) is always less than the frequency of occurences of waiting lines with $m_p - 1$ units,no matter how the parameters λ, μ and m are selected. That is,when $n = m_p$ in the previous and next stage of the system,the number of waiting units has to be $n = m_p - 1$.

Since times are taken into account in the calculations (that is, times spent in line by $0,1,\ldots,m_p$ waiting units),when $\Psi > 1$ the period that i units spend in the waiting line is longer than the total waiting time of $i - 1$ units.Namely,

$$T_k > T_{k-1} \quad (k = 1,2,\ldots,m)$$

where T_k is the period spent by k units in the system (or the period spent by k-1 units in the waiting line) in the course of simulation.

Since $p_k = \frac{T_k}{TATS}$ by increasing the value of k, the simulated probabilities increase simultaneously as well.The analytical probabilities display the same tendency,but discrepances can be observed between the analytical and simulated probabilities.

Table 2.3-1 displays the above mentioned facts for a series of values of Ψ when $\Psi > 1$,namely

(1) even if $f_{m_p} < f_{m_p-1}$ yet $T_m > T_{m_p}$ in some cases,where f_{m_p} and f_{m_p-1} are the frequences of occurences of m_p and m_p-1 waiting units in the queue (that is m and m_p units in the system);

(2) $T_o < T_1 < \ldots < T_m$, so $p_o < p_1 < \ldots < p_m$ (however sometimes there is slight departure from this principle in the simulated results when Ψ is not far from 1);

(3) there is a comparison between some characteristics of the system calculated analytically and obtained by simulation.

When the traffic intensity is small and the number of waiting units does not reach the critical value m_p,simulation models 1 and 2 completely agree.Notwithstanding this,without changing the distributions of interarrival and service times the analytical values deviate significantly as regards Model 1 and 2.As soon as the constraint of prescribing a maximum value for the number of units in the system is neglected (Model 1),the simulated values fit prominently the analytical values.

When $\Psi < 1$,congestion rarely occurs,or not at all.In these cases the probabilities obtained by simulation better fit the analytical values given in Model 1.

In Tables (2.3-2) and (2.3-3) the same sort of results as in the previous table in the case of $\Psi < 1$,completed by the analytical probabilities used for Model 1.As can be seen,the results obtained by simulation approximate the real situation much better.

If $\Psi < 1$ and $n = m_p$ does not occur then by increasing the number of iterations the results obtained by simulation approach better and better

m ≥ 6

Ψ	n	f_n	k	T_k	Probability (P_k)			
					Simulated		Analytical	
					L_p = 200	L_p =1200	Model 1	Model 2
.1	-	-	0	1733.1	.8876	.8982	.9000	.4845
	0	197	1	183	.0937	.0913	.0900	.2907
	1	200	2	29	.0148	.0089	.0090	.1454
	2	4	3	7.5	.0039	.0016	.0009	.0581
	3	0	4	0	--	--	.0001	.0174
	4	0	5	0	--	--	--	.0035
	5	0	6	0	--	--	--	.0004
.25	-	-	0	468.4	.7192	.7456	.7500	.1172
	0	179	1	118.2	.1815	.1872	.1875	.1757
	1	196	2	43.9	.0675	.0494	.0469	.2197
	2	22	3	16.8	.0257	.0146	.0117	.2197
	3	4	4	4.0	.0061	.0032	.0029	.1647
	4	0	5	-	--	--	.0008	.0824
	5	0	6	-	--	--	.0002	.0206
.357	-	-	0	234.8	.5997	.6366	.6429	.0417
	0	164	1	86.3	.2205	.2271	.2296	.0894
	1	188	2	38.4	.0980	.0867	.0820	.1596
	2	35	3	24	.0612	.0363	.0293	.2281
	3	12	4	6.4	.0163	.0112	.0105	
	4	2	5	1.7	.0043	.0021	.0037	
	5	0	6	0	--	--	.0013	
.437	-	-	0	143.2	.5109	.5551	.5625	.0203
	0	149	1	63.2	.2252	.2382	.2461	.0534
	1	177	2	35.3	.1258	.1164	.1077	.1167
	2	48	3	26.1	.0931	.0594	.0471	.2043
	3	23	4	9 .3	.0330	.0232	.0206	.2681
	4	4	5	3.4	.0120	.0074	.0090	.2346
	5	-	6	-	--	.0003	.0040	.1026

TABLE 2.3-2

Ψ		E(k)	E(n)	E(WT)	E(WT/WT>0)	EFθC
.1	Simulated L_p = 200	.14	.02	.07	.46	.11
	L_p = 1200	.11	.01	.03	.26	.10
	Analytical Model 1	.11	.01	.11	1.11	.10
	Model 2	.85	.33	1.64	.64	.52
.25	Simulated L_p = 200	.42	.14	.25	.80	.28
	B_p = 1200	.34	.09	.12	.44	.25
	Analytical Model 1	.33	.08	.28	1.11	.25
	Model 2	2.47	1.59	2.33	1.50	.88
.357	Simulated L_p = 200	.69	.29	.38	.92	.40
	L_p =1200	.56	.20	.22	.58	.36
	Analytical Model 1	.56	.20	.40	1.11	.36
	Model 2	3.32	2.36	2.47	1.76	.96
.437	Simulated L_p = 200	.95	.46	.46	.90	.49
	L_p =1200	.78	.34	.30	.66	.45
	Anal. Mod.	.78	.34	.49	1.11	.44
	Model 2	3.76	2.78	2.40	1.77	.98

Table 2.3-3

m	W		E(k)	E(n)	E(WT)	E(WT/ WT>0)	EFOC
6	.9	Simulated L_p = 200	1.81	1.06	2.10	3.15	.746
		Simulated L_p = 1200	2.48	1.69	3.85	5.09	.788
		Analytical	4.89	3.89	9.79	7.79	.999
	1.173	Sim. L_p=200	2.73	1.85	1.46	1.77	.886
		Sim. L_p=1200	3.44	2.54	2.28	2.60	.903
		Anal.	5.15	4.15	4.48	3.61	1.000
	1.36	Sim. L_p=200	3.44	2.49	3.99	4.27	.950
		Sim. L_p=1200	4.00	3.05	5.57	6.01	.947
		Anal.	5.27	4.27	9.58	7.76	1.000
	1.571	Sim. L_p=200	4.04	3.05	7.60	7.8	.985
		Sim. L_p=1200	4.45	3.47	9.80	10.3	.975
		Anal.	5.36	4.36	15.30	12.5	1.000
	2.33	Sim. L_p=200	5.00	4.00	23.4	23.4	1.000
		Sim. L_p=1200	5.23	4.23	27.5	27.7	.998
		Anal.	5.57	4.57	37.1	30.5	1.000
10	.9	L_p=200	2.00	1.25	2.52	3.57	.758
		L_p=1200	3.82	2.99	6.75	8.26	.834
		Anal.	8.89	7.89	17.78	15.78	1.000
	1.173	L_p=200	4.02	3.09	2.48	2.80	.925
		L_p=1200	6.15	5.20	4.60	4.92	.951
		Anal.	9.15	8.15	7.96	7.09	1.000
	1.36	L_p=200	5.95	4.97	8.0	8.1	.986
		L_p=1200	7.43	6.45	11.6	11.9	.984
		Anal.	9.27	8.27	16.8	15.0	1.000
	1.571	L_p=200	7.21	6.21	15.6	15.6	1.000
		L_p=1200	8.30	7.31	20.5	20.6	.996
		Anal.	9.36	8.36	26.8	23.9	1.000
	2.33	L_p=200	8.43	7.43	43.6	43.6	1.000
		L_p=1200	9.25	8.25	53.9	53.9	1.000
		Anal.	9.57	8.57	63.8	57.1	1.000

Table 2.3-4

the analytical values of Model 1 and move away from the analytical ones of Model 2.This tendency can be well perceived in Tables (2.3-2) and (2.3-3) where the basic characteristics of the system are registered when L_p = 200 and when L_p = 1200 to provide a basis for comparison with the analytical values of both of the models.

The situation changes when the event $n = m_p$ occurs whenever $\Psi > .5$. At this time by increasing the number of iterations the simulated statistics approach better the analytical results although one cannot speak of a really good fit due to the deviation between the simulation and theoretical models.

A conspicuous improvement can be observed when m keeps growing.

The same trend is shown in Table (2.3-4) with other characteristics of the system,namely,when L_p is growing the simulated results fit better the theoretical ones.

Considering the above model and results derived from it,there is no doubt that the real situation can better be approximated by simulation.As has been mentioned (in what has gone before) the method of simulation is worthy of application when its efficiency is superior to that of other techniques.

2.4 Simulation of a Waiting Line at Several Stations (Model 3)

2.4.1 The Model

The simulation of a waiting line system consisting of several service stations operating in parallel is discussed in this chapter.It is supposed that arrivals follow Poisson's law,the average rate of arrivals is λ ,and arrivals are admitted to the first vacant service station on a first-come,first-served basis.Each station has its own particular probability distribution for service time.

When an input-unit arrives in the system a check will be made whether all of the stations are occupied.If so,then this unit has to wait until a service is ended.When a service station becomes vacant before another unit arrives at the system,idle time occurs until the next unit arrives and enters the vacant station.

For the description of the model a flow-chart is used,where every instant and all characteristics of the simulation can be followed.The flow-chart consists of three parts:

(1) the multichannel model itself (FLOW CHART MODEL3,p.53-54),
(2) the random arrival and service time generators (these have been described in details in the Appendix,p.15-23),
(3) a separate"program flow-chart"(p.55-56) joined to the model in order to simulate some variables and then to use them for calculating the simulated statistics.

2.4.2 The Variables of the Model

L : the number of service stations,
L_p : " " " arrivals that have been simulated,
i : the current number of arrivals,
i_{ser}: " " " " generated service times,
i_w : the number of arrivals who wait,
i_{dc} : " " " occurences of idle time,
RAR : the random time interval between two successive arrivals,
RSR_h: " " service time at the h^{th} station,
ART_h: random time of the arrivals,
TS_h : complition time of the current service at the h^{th} station,
SID_h: total idle time at the h^{th} station,
SWT_h: " waiting " " " " " ,
$TRSR_h$: " service " " " " " ,
TST : " service time ($TST = \sum_{h=1}^{L} TRSR_h$),
TWT : total waiting time ($TWT = \sum_{h=1}^{L} SWT_h$),
TID : total idle time ($TID = \sum_{h=1}^{L} SID_h$),

KASHIF IBRAHIM
RECEIVED
DATE
A.B.U. ZARIA

TAT: arrival time of the unit which has been waiting the longest for service,

TSMIN: the minimum TS_h over all h ($h = 1,\ldots,L$),

WT_h: current waiting time at the h^{th} station,

DT_h: " idle " " " " " ,

TATS: total time of simulation.

2.4.3 The Flow Chart

At the start of the simulation the following relationships are given:

$$ART_1 = 0$$
$$TS_h = 0 \quad (h = 1,\ldots,L)$$
$$SID_h = 0 \quad (h = 1,\ldots,L)$$
$$SWT_h = 0 \quad (h = 1,\ldots,L)$$
$$TRSR_h = 0 \quad (h = 1,\ldots,L)$$
$$i = 1$$
$$i_{ser} = 0$$
$$i_w = 0$$
$$i_{dc} = L-1$$
$$TST = 0$$
$$TWT = 0$$
$$TID = 0$$

thus the first arrival enters the system at zero time.

After setting the starting conditions (block 1) as many arrival times are generated as there are service stations ($ART_1,\ldots,ART_L$), and simultaneously a service time is generated for each station ($RSR_1,\ldots,RSR_L$). Then the total service time that has occured until this instant is determined (blocks 2-12).

In block 13 the number of input units that has entered the system, and the number of generated service times are assigned to i and i_{ser}. The starting value of TAT necessary to the next value of TS is set in block 14.

By means of blocks from 15 to 22 the minimum of the momentary TS values is selected. Notations j and TSMIN are used to indicate the serial number of service station that becomes vacant first, and the minimum TS value, respectively.

Before the calculation of the current waiting [redacted] idle time and generating the next service time in blocks from 24 to 45

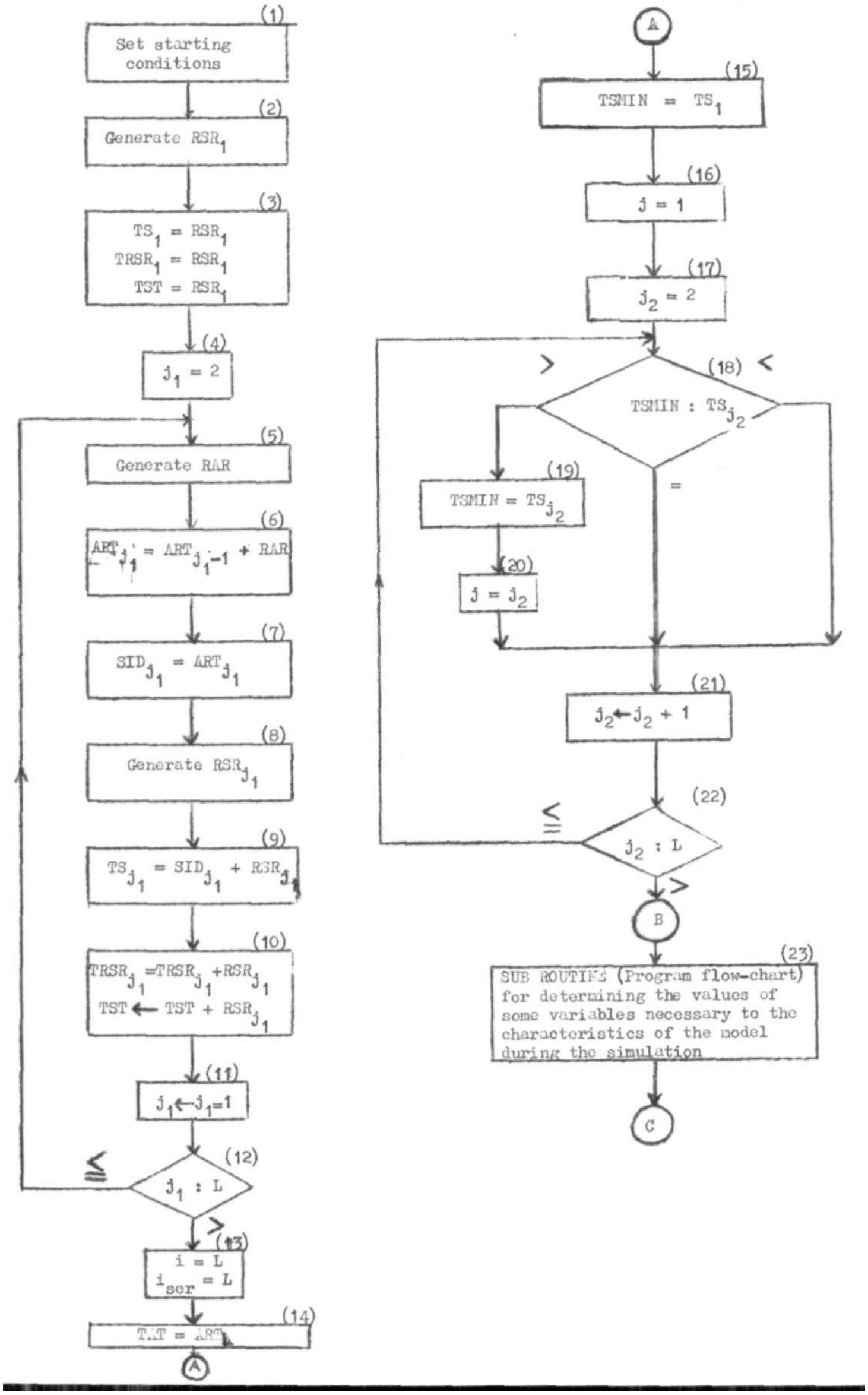
(i) the T_k quantities required for the simulated system -characteristics are determined,

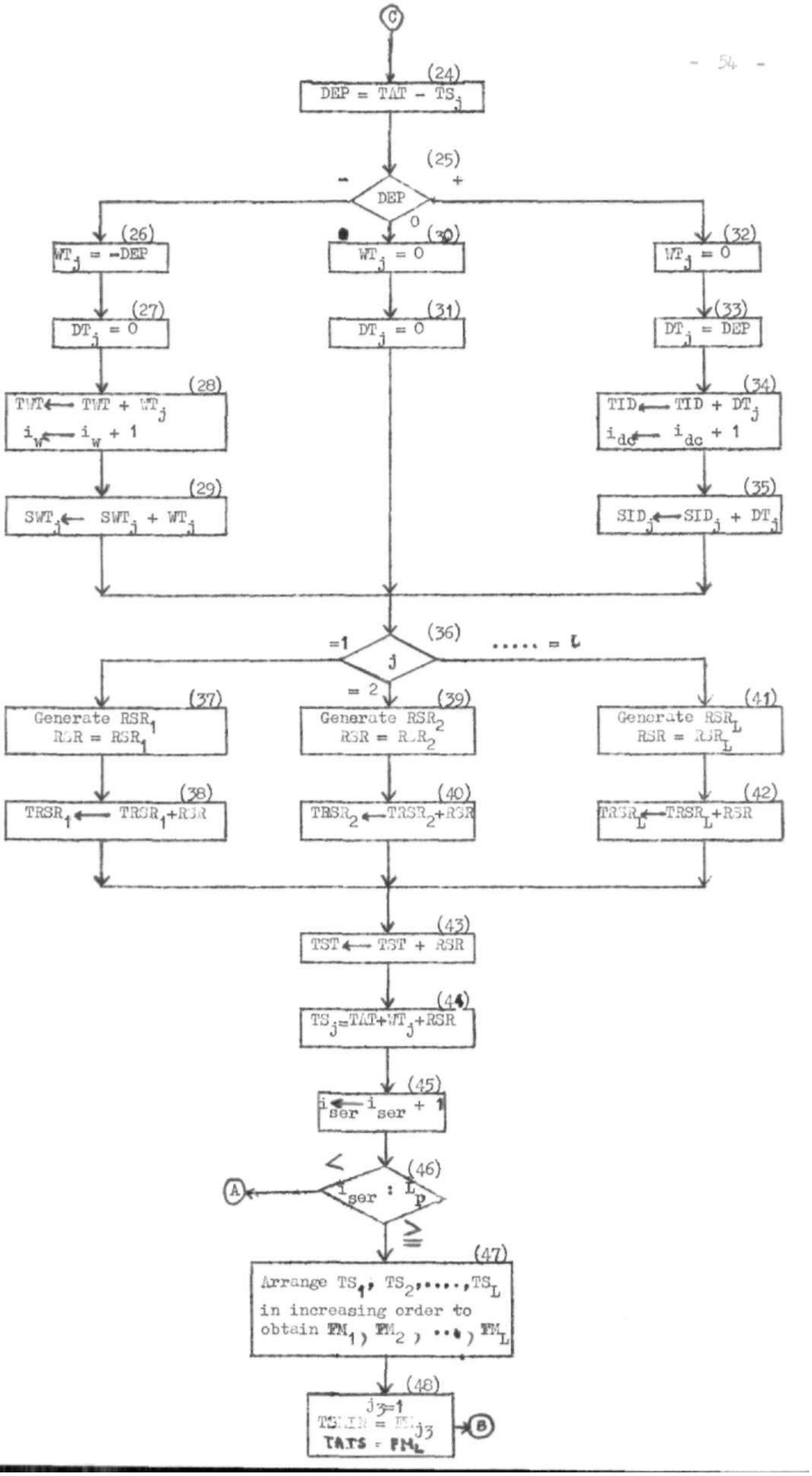
(ii) the values of the parameters varying during the simulation are set,

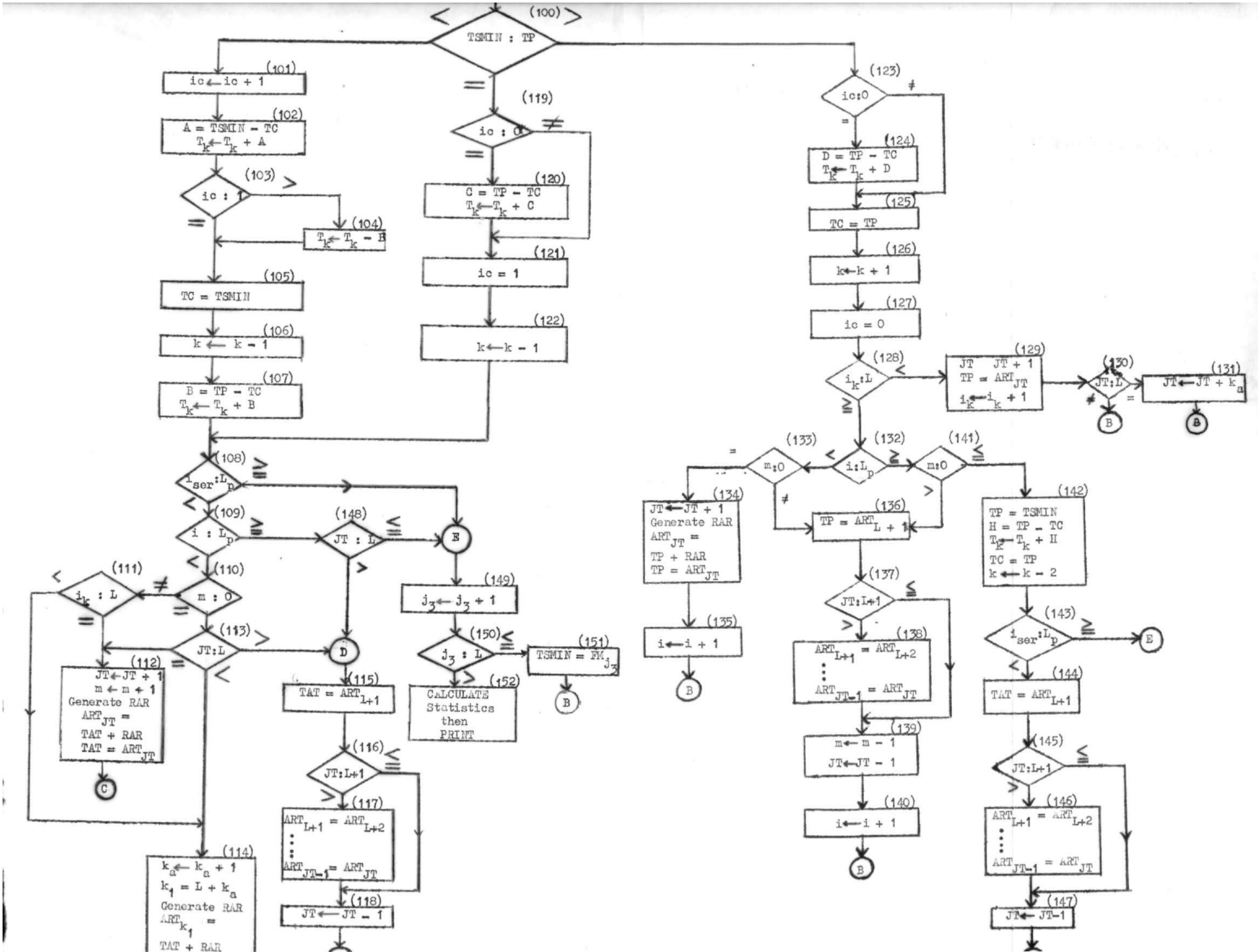
(iii) new arrival time is generated, and

(iv) a proper value is assigned to the next TAT by the "program - flow - chart" beginning at label E. At this moment $TAT = ART_{L+1}$.

FLOW CHART (MODEL 3)




C
(24)
DEP = TAT - TS_j
(25)
-
DEP
+
0
(26)
WT_j = -DEP
(30)
WT_j = 0
(32)
WT_j = 0
(27)
DT_j = 0
(31)
DT_j = 0
(33)
DT_j = DEP
(28)
TWT ← TWT + WT_j
i_w ← i_w + 1
(34)
TID ← TID + DT_j
i_dc ← i_dc + 1
(29)
SWT_j ← SWT_j + WT_j
(35)
SID_j ← SID_j + DT_j
(36)
j
=1
= 2
..... = L
(37)
Generate RSR_1
RSR = RSR_1
(39)
Generate RSR_2
RSR = RSR_2
(41)
Generate RSR_L
RSR = RSR_L
(38)
TRSR_1 ← TRSR_1+RSR
(40)
TRSR_2 ← TRSR_2+RSR
(42)
TRSR_L ← TRSR_L+RSR
(43)
TST ← TST + RSR
(44)
TS_j=TAT+WT_j+RSR
(45)
i_ser ← i_ser + 1
(46)
i_ser : L_p
<
≥
A
(47)
Arrange TS_1, TS_2,....,TS_L
in increasing order to
obtain FM_1, FM_2, ..., FM_L
(48)
j3=1
TSMIN = FM_j3
TATS = FM_L
B


(100)
TSMIN : TP
(101)
ic ← ic + 1
(102)
A = TSMIN - TC
$T_k \leftarrow T_k + A$
(103)
ic : 1
(104)
$T_k \leftarrow T_k - B$
(105)
TC = TSMIN
(106)
k ← k - 1
(107)
B = TP - TC
$T_k \leftarrow T_k + B$
(108)
$i_{ser} : L_p$
(109)
i : L_p
(110)
m : 0
(111)
i_k : L
(112)
JT ← JT + 1
m ← m + 1
Generate RAR
ART_{JT} =
TAT + RAR
TAT = ART_{JT}
C
(113)
JT:L
D
(114)
$k_a \leftarrow k_a + 1$
$k_1 = L + k_a$
Generate RAR
ART_{k_1} =
TAT + RAR
(115)
TAT = ART_{L+1}
(116)
JT:L+1
(117)
$ART_{L+1} = ART_{L+2}$
$ART_{JT-1} = ART_{JT}$
(118)
JT ← JT - 1
(119)
ic : 0
(120)
C = TP - TC
$T_k \leftarrow T_k + C$
(121)
ic = 1
(122)
k ← k - 1
(123)
ic:0
(124)
D = TP - TC
$T_k \leftarrow T_k + D$
(125)
TC = TP
(126)
k ← k + 1
(127)
ic = 0
(128)
i_k:L
(129)
JT JT + 1
TP = ART_{JT}
$i_k \leftarrow i_k + 1$
(130)
JT:L
B
(131)
JT ← JT + k_a
B
(132)
i:L_p
(133)
m:0
(134)
JT ← JT + 1
Generate RAR
ART_{JT} =
TP + RAR
TP = ART_{JT}
(135)
i ← i + 1
B
(136)
TP = ART_{L+1}
(137)
JT:L+1
(138)
$ART_{L+1} = ART_{L+2}$
$ART_{JT-1} = ART_{JT}$
(139)
m ← m - 1
JT ← JT - 1
(140)
i ← i + 1
B
(141)
m:0
(142)
TP = TSMIN
H = TP - TC
$T_k \leftarrow T_k + H$
TC = TP
k ← k - 2
(143)
$i_{ser} : L_p$
E
(144)
TAT = ART_{L+1}
(145)
JT:L+1
(146)
$ART_{L+1} = ART_{L+2}$
$ART_{JT-1} = ART_{JT}$
(147)
JT ← JT-1
(148)
JT : L
E
(149)
$j_3 \leftarrow j_3 + 1$
(150)
j_3 : L
(151)
TSMIN = FM_{j_3}
B
(152)
CALCULATE
Statistics
then
PRINT

After these assignments are performed the total and partial waiting and idle times at the separate service stations are calculated by the series of blocks beginning at label C.In block 24 we take the difference (DEP) between the total arrival time (TAT) that has been reached during the simulation and the ending time of the last service.Depending upon whether this difference is negative or positive,waiting or idle time is computed and added to either TWT and SWT_j or TID and SID_j,and the value of i_w or i_{dc} is increased by one,respectively (blocks 24-35).

Then a new service time is generated for the vacant station in one of the blocks 37,39,41.In block 44 the ending time of the recently generated service time is assigned to TS_j.The current number of generated service times is increased by 1 in block 45,then a check is made to decide whether to return to label A and repeat the process or to finish the simulation.If $i_{ser} \geq L_p$ the last service is about to take place at each station and the ending times of all these services $(TS_1,\ldots,TS_L)$ are arranged in increasing order of magnitude and are denoted by $FM_1,\ldots,FM_L$ (block 47),where FM_1 = minimum$(TS_1,\ldots,TS_L)$ and FM_L = maximum$(TS_1,\ldots,TS_L)$.It is obvious that the completion time of the simulation is FM_L which is assigned to TATS in block 48.Later on the simulated process does not return to this section of the flow-chart,but it ends by using the "program flow-chart" beginning at label B.

2.4.4 The Program FLOW CHART

As mentioned above a separate subroutine named "program flow-chart" is used for calculation.Its objective has been given in the previous chapter.

It is necessary to introduce some new variables for the purpose of computer program (some of them agree with that of Model 1),simultaneously "program-variables" are also used.These variables are as follows:

LSS:maximum number of units in the system during the simulation,

LS_{max}:maximum number of units in the waiting line during the simulation (LSS = LS_{max}+L),

TP:predicted time of the arrival being serviced next,

TC:current time that has been reached during the simulation,

k:current number of units in the system,

T_k:amount of time that k units spend in the system (k = 0,1,...,LSS),

JT:subscript for ART indicating the current number of arrivals (n = JT - L) that have entered the system but for which service times have not been assigned.Whenever a new arrival occurs JT is increased by 1,except at the starting phase of the simulation when not all the first L arrival times have been generated yet,

ic:indicator to set correct value for k when the current T_k is

determined (ic = 0 whenever the completion time of any service does not fall between two subsequent arrival times, otherwise ic ≥ 1),

k_a: counter used in the starting period of the simulation ($i \leq L$ and $JT < L$) indicating the number of new arrivals that have entered the system before the process passes the time ART_L,

i_k: counter in the starting period of the process ($i_k \leq L$ during the simulation),

m: counter indicating the number of TAT's that have been assigned (if m = 0 then TP = TAT, if $m \neq 0$ then TAT > TP and m=JT-L).

This subroutine is used when

(1) a service is ended,

(2) a new TP is to be defined, or

(3) both a service is finished and a TP is to be defined simultaneously. At the starting block (100) the program may branch out 3 possibly ways. When TSMIN > TP a new TP must be generated. At the outset of the simulation before all the T_k values have been collected until the time ART_L ($i_k < L$), the program branches to the right at block 128 (to blocks 129-131), but as soon as $i_k = L$ it will not return to this branch again.

According to the decision based on block 132 we can return again to label B through either the blocks 134-135 or 136-140 until all the units that must be simulated arrive inthe system ($i < L_p$). During this period ($i_k \geq L$ and $i < L_p$) whenever control is transferred to this branch a new value is assigned to TP and the simulation is carried out. Obviously the value of i is increased by 1 as well. It may turn out during the simulation that several units are waiting for service, but only the minimum of the arrival times is used here. For this reason i is increased only in these blocks (135 and 140) whereas generation of new arrival time may also occur in blocks 112 and 114.

If $i \geq L_p$ the block 142 is used to determine the current value of T_k. When all the service times have been generated ($i_{ser} \geq L_p$) a control is transferred to label E to perform the finishing period of the simulation (blocks 149-151). If $i_{ser} < L_p$ (block 143) then one return to B occurs (blocks 144-147). After the simulation has been completed the required simulated statistics and probabilities are calculated (by block 152).

Blocks 119 and 122 are formed for those unlikely cases when two random times coincide; that is TP = TSMIN. (The probability of such an event is theoretically equal to 0.)

In most of the cases the program branches to the left according to the condition TP < TSMIN. At this moment ic takes up a value greater than 0. If ic > 1 the block 104 is used; that is when there are at least 2 stations becoming vacant between the two subsequent random arrival times in question. Blocks 101-107 are self-explanatory. Decisions and branches are made according to the momentary value of i_{ser} and i.

When either of them is equal to L_p then the ending part of the program follows.

The counters m_i and JT are used to assign correct values of the following TAT or TP, or both. As soon as a service is finished and JT $\leq$ L a new arrival time must be generated by block 114 at the starting stage of the process ($i \leq L$) or by block 112 further on the simulation. (Since JT $\geq$ L when $i > L$, control will not be transferred to block 114 later on.) The purpose of both blocks is to determine the next correct value of TAT which is used to fix the ending time of the following service (by block 44). If there are some waiting units in the system, new arrival time is not generated but the above mentioned assignment occurs in block 115. After each of these boxes (112,114,115) control returns to label C in order to generate new service time.

Figures 2.4-1 and 2.4-2 are attached to the flow chart, where the variation of the most important variables of the model can be followed in two particular cases.

2.4.5 Application

To describe a waiting line at several stations analytically some hypotheses which are the usual ones for the most frequent kinds of delay phenomena must be assumed.

When a service station is free, the first customer in the line is served by that station. There is no customer preference for any station. The model has been tested in the case when all the stations have the same average rate of service μ, governed by the exponential distribution. Arrivals follow Poisson's law with average arrival rate of λ.

We assume we have a permanent system in which the probabilities p_k that there are k units in the system are independent of time. The traffic intensity per station is given again by $\Psi = \frac{\lambda}{\mu}$ and the restriction $\frac{\Psi}{L} < 1$ is adopted. Under the above conditions the characteristic quantities of the model [33, p.95-98 and 362-365] can be derived from the "equations of state".

The probability that a number of units k are in the system

$$p_k = \frac{\Psi}{k} p_{k-1} \qquad 1 \leq k < L \qquad (2.4\text{-}1)$$

and

$$p_k = \frac{\Psi}{L} p_{k-1} \qquad k \geq L \qquad (2.4\text{-}2)$$

where

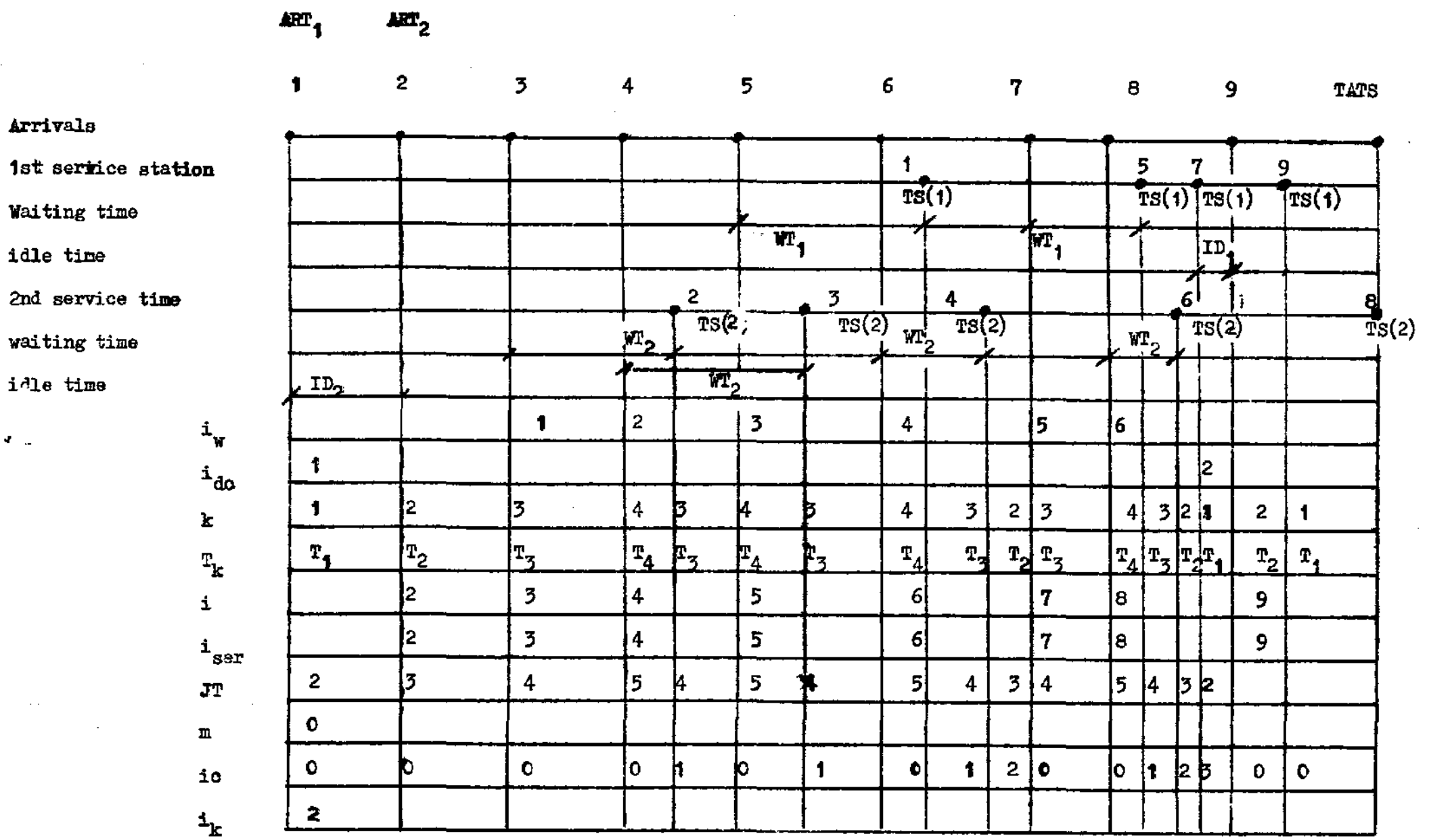
$$p_o = \frac{1}{\frac{\Psi^L}{L!\left(1 - \frac{\Psi}{L}\right)} + \sum_{s=0}^{L-1} \frac{\Psi^s}{s!}} \qquad (2.4\text{-}3)$$

and

$$\sum_{k=0}^{\infty} p_k = 1 .$$

Expected number of units in the waiting line

Simulation when $L = 2$ and $L_p = 9$



i_w			1	2		3		4			5	6					
i_{do}	1														2		
k	1	2	3	4	3	4	3	4	3	2	3	4	3	2	1	2	1
T_k	T_1	T_2	T_3	T_4	T_3	T_4	T_3	T_4	T_3	T_2	T_3	T_4	T_3	T_2	T_1	T_2	T_1
i		2	3	4		5		6			7	8				9	
i_{ser}		2	3	4		5		6			7	8				9	
JT	2	3	4	5	4	5	4	5	4	3	4	5	4	3	2		
m	0																
ic	0	0	0	0	1	0	1	0	1	2	0	0	1	2	3	0	0
i_k	2																

Figure 3.4-

Simulation when $L = 3$ and $L_p = 9$

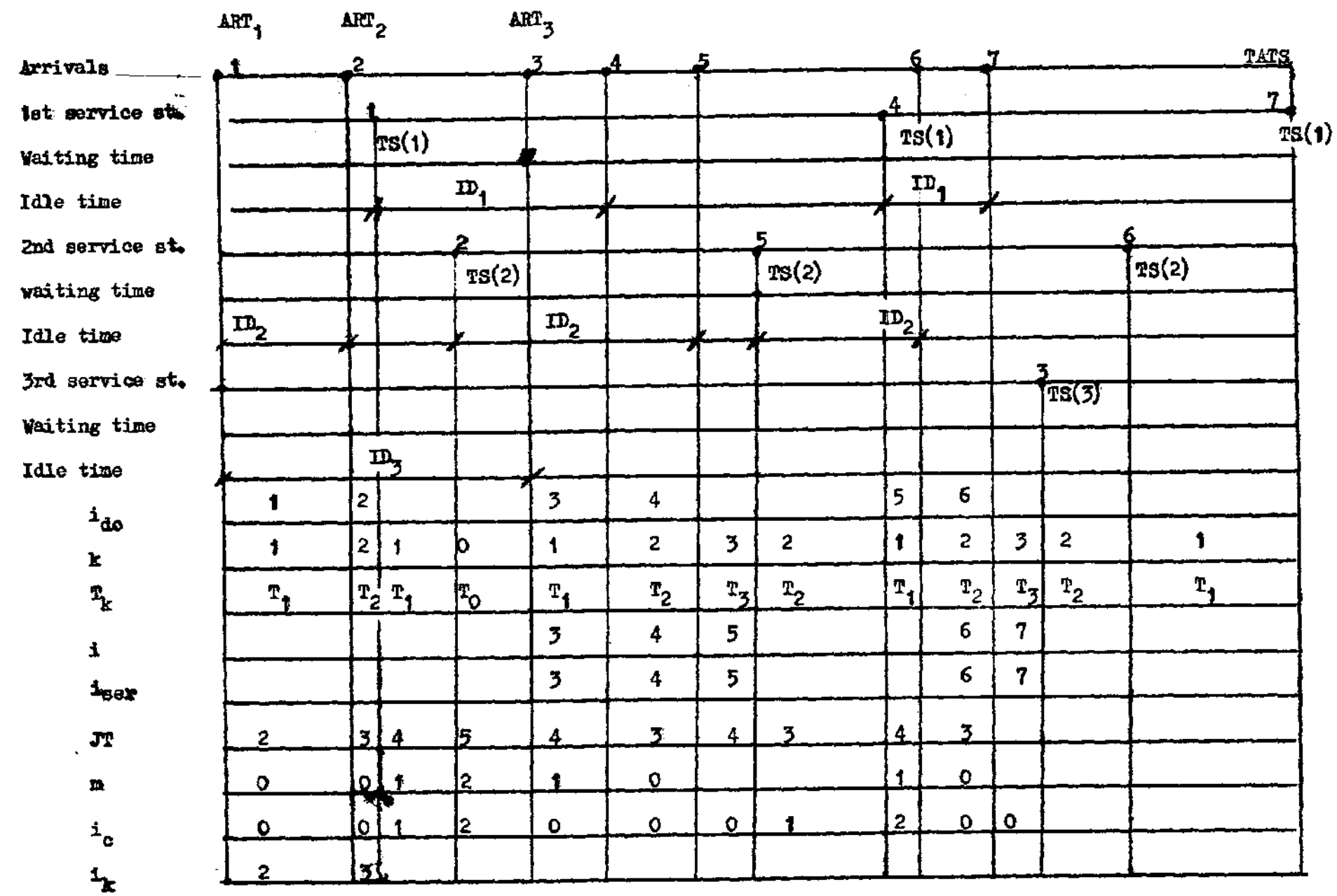


Figure .-2

$$E(n) = \frac{\Psi^{L+1}}{L.L!(1 - \frac{\Psi}{L})^2} p_o \qquad (2.4\text{-}4)$$

Expected number of units in the system

$$E(k) = E(n) + \Psi \qquad (2.4\text{-}5)$$

Expected number of idle stations

$$E(ID) = L - \Psi \qquad (2.4\text{-}6)$$

The probability of a delay

$$Pr(k \geq L) = \frac{\Psi^{L}}{L!(1 - \frac{\Psi}{L})} p_o \qquad (2.4\text{-}7)$$

Expected waiting time of an arrival

$$E(WT) = \frac{\Psi^{L}}{L.L!\mu(1 - \frac{\Psi}{L})^2} p_o \qquad (2.4\text{-}8)$$

Expected time an arrival spends in the system

$$E(T_{ws}) = E(WT) + \frac{1}{\mu} . \qquad (2.4\text{-}9)$$

The following formulae corresponding to the analytical solution use the notation introduced above:

the probability that a number of units $\underline{k}$ is in the system

$$p_k = \frac{T_k}{TATS} \quad (k = 0,1,\ldots,LSS),$$

the probability that a number of units $\underline{n}$ is in the queue

$$q_n = \frac{WA_n}{TATS} \quad (n = 0,1,\ldots,LS_{max})$$

where WA_n means the amount of time that $\underline{n}$ units spend in the system.

It is obvious that

$$\sum_{k=0}^{LSS} p_k = 1 \quad \text{and,} \quad \sum_{n=0}^{LS_{max}} q_n = 1,$$

$$WA_o = \sum_{k=0}^{L} T_k, WA_1 = T_{L+1},\ldots,WA_r = T_{L+r},\ldots,WA_{LS_{max}} = T_{LSS},$$

so

$$q_o = \frac{\sum_{k=0}^{L} T_k}{TATS},\ldots,q_r = \frac{T_{L+r}}{TATS},\ldots,q_{LS_{max}} = \frac{T_{LSS}}{TATS} .$$

$$E(k)_{simulated} = \sum_{k=0}^{LSS} kp_k$$

$$E(n)_{sim.} = \sum_{n=0}^{LS_{max}} nq_n$$

$$E(ID)_{sim.} = \sum_{k=0}^{L} (L - k)p_k$$

$$Pr(k \geq L)_{sim.} = \sum_{k=L}^{LSS} p_k$$

$$E(WT)_{sim.} = \frac{TWT}{L_p}$$

$$E(T_{ws})_{sim.} = \frac{TWT + TST}{L_p} .$$

Under the conditions given earlier these characteristics of the system have been simulated with varying values of Ψ and L.Simulation has been carried out with 32 different values of Ψ ,and simultaneously the number of service stations (L) has been selected with the restriction that $\Psi < L$.

For a fixed Ψ the model was tested with 4 or 5 different values of L.In addition,for every L the number of input units (L_p) was selected to be 300,then 1500.(It is to be noted that each simulation was repeated with different input parameters used in the stochastic-variate generators.)

In Tables from 2.4-1 to 2.4-4 with 3 different values (2.5; 4 and 9) of Ψ (out of 32) the simulated and corresponding theoretical probabilities appear in columns while the other quantities line by line. The simulated probability distribution fits the theoretical distribution very well,and the other simulated characteristics of the system agree with their theoretical counterparts.It can be seen that by increasing the number of simulated units (L_p) the results improve greatly.

For values of Ψ that are not included in the attached tables the simulated quantities agree with the analytical ones in each case,without exception.

It is obvious that Model 3 can be applied to other,special,important cases as well.One of them might be the case where all the service stations do have a separate stochastic-variate generator for determining their service times from given average rates of services $\mu_1, \mu_2, \ldots, \mu_L$.A part of the simulated statistics is unchanged,but distinct statistics can be calculated at each service station.It is worth mentioning the expected waiting and idle times at each separate station. By introducing the formulae

$$\frac{SWT_h}{i_w^{(h)}} \quad \text{and} \quad \frac{SID_h}{i_{dc}^{(h)}} \qquad (h = 1,2,\ldots,L)$$

where $i_w^{(h)}$ and $i_{dc}^{(h)}$ are the number of waiting units and the number of occurences of idle times at the h^{th} station,respectively,and the unit costs of idle and waiting times are known,then the optimum number of service stations and hence the minimum cost of the system can be easily determined.

Numerical calculations have been carried out for the case mentioned. The same values of λ were used as in the first application,but different rates of service ($\mu_1, \mu_2, \ldots, \mu_L$) have been selected for each separate station in such a manner that their arithmetic mean agrees with μ ,the common average rate of service used previously.The model display excellent performance in this case as well.

Comparison of the values of probabilities and other characteristic quantities of Model 3 with varying values of parameters

Ψ = 2.5

L	k	Probability (P_k) simulated L_p =300	Probability (P_k) simulated L_p =1500	Probability (P_k) analytical		E(k)	E(n)	E(ID)	E(WT)	E(T_{ws})	Pr(k≥L)
					simulated						
3	0	.0617	.0462	.0449	L_p = 300	4.17	1.83	1.15	1.04	2.38	.63
	1	.1633	.1245	.1124							
	2	.1430	.1486	.1404	L_p =1500	4.70	2.24	.99	1.27	2.67	.68
	3	.1229	.1314	.1170							
	4	.1333	.1207	.0975	Analytical	6.01	3.51	.50	2.01	3.44	.70
	5	.0972	.0872	.0813							
	6	.0717	.0706	.0677							
	7	.0625	.0582	.0564							
	8	.0382	.0502	.0470							
	9	.0218	.0489	.0392							
	10	.0298	.0378	.0327							
	11	.0172	.0307	.0272							
	12	.0169	.0195	.0227							
	13	.0122	.0152	.0189							
	14	.0078	.0057	.0158							
	15	.0004	.0038	.0131							
4	0	.0926	.0704	.0737	Simulated						
	1	.2011	.1881	.1842							
	2	.2498	.2399	.2303	L_p = 300	2.79	.44	2.46	.25	1.59	.28
	3	.1724	.2021	.1919							
	4	.1022	.1171	.1199	L_p =1500	2.85	.37	2.35	.21	1.61	.30
	5	.0695	.0836	.0750							
	6	.0428	.0468	.0469	Analy.	3.03	.53	1.50	.31	1.73	.32
	7	.0290	.0262	.0293							
	8	.0174	.0162	.0183							
	9	.0060	.0060	.0114							
	10	.0137	.0035	.0071							
	11	.0004	.0002	.0045							
5	0	.1016	.0757	.0801	Simulated						
	1	.2101	.2040	.2003							
	2	.2762	.2591	.2503	L_p = 300	2.48	.12	3.59	.07	1.41	.12
	3	.1739	.2119	.2086							
	4	.1212	.1317	.1304	L_p =1500	2.55	.08	3.48	.05	1.45	.12
	5	.0583	.0677	.0652							
	6	.0260	.0272	.0326	Analy.	2.63	.13	2.50	.07	1.50	.13
	7	.0129	.0157	.0163							
	8	.0103	.0048	.0081							
	9	.0063	.0015	.0041							
	10	.0026	.0005	.0020							
6	0	.1029	.0769	.0816	Simulated						
	1	.2170	.2085	.2041							
	2	.2803	.2626	.2551	L_p = 300	2.39	.04	4.62	.02	1.36	.05
	3	.1736	.2142	.2126							
	4	.1220	.1319	.1328	L_p =1500	2.49	.02	4.51	.01	1.41	.04
	5	.0576	.0672	.0664							
	6	.0243	.0245	.0277	Analy.	2.53	.03	3.50	.02	1.45	.05
	7	.0094	.0104	.0115							
	8	.0089	.0031	.0048							
	9	.0040	.0008	.0020							

Table 2.4-1

$\bar{Y} = 4$

L	k	Probability (P_k) simulated L_p=300	Probability (P_k) simulated L_p=1500	Probability (P_k) analytical		E(k)	E(n)	E(ID)	E(WT)	$E(T_{ws})$	$Pr(k \geq L)$
6	0	.0127	.0134	.0167	L_p = 300	4.02	.33	3.16	.08	1.02	.25
	1	.1231	.0687	.0667							
	2	.1673	.1423	.1335	L_p = 1500	4.26	.32	2.89	.08	1.06	.26
	3	.1786	.1834	.1780							
	4	.1370	.1850	.1780	Analytical	4.57	.57	2.00	.14	1.14	.29
	5	.1358	.1462	.1424							
	6	.0969	.0959	.0949							
	7	.0543	.0720	.0633							
	8	.0413	.0511	.0422							
	9	.0299	.0261	.0281							
	10	.0141	.0101	.0187							
	11	.0090	.0043	.0125							
	12	-	.0010	.0083							
	13	-	.0004	.0056							
7	0	.0155	.0144	.0178	simulated						
	1	.1245	.0728	.0712							
	2	.1731	.1508	.1425	L_p = 300	3.80	.10	4.25	.03	.96	.11
	3	.1888	.1901	.1900							
	4	.1470	.1979	.1900	L_p =1500	4.03	.09	4.00	.02	1.00	.12
	5	.1476	.1610	.1520							
	6	.0926	.0928	.1013	Analytical	4.18	.18	3.00	.05	1.05	.14
	7	.0530	.0638	.0579							
	8	.0317	.0336	.0331							
	9	.0167	.0162	.0189							
	10	.0032	.0045	.0108							
	11	.0063	.0022	.0062							
8	0	.0173	.0147	.0182	simulated						
	1	.1230	.0760	.0727							
	2	.1739	.1493	.1453	L_p = 300	3.72	.03	5.29	.01	.95	.04
	3	.1957	.1973	.1937							
	4	.1535	.2000	.1937	L_p =1500	3.97	.03	5.04	.01	.99	.04
	5	.1398	.1589	.1550							
	6	.0958	.0918	.1033	Analytical	4.06	.06	4.00	.02	1.02	.06
	7	.0586	.0676	.0590							
	8	.0258	.0263	.0295							
	9	.0089	.0128	.0148							
	10	.0035	.0035	.0074							
	11	.0041	.0016	.0037							
9	0	.0173	.0149	.0183	simulated						
	1	.1230	.0762	.0731							
	2	.1739	.1492	.1462	L_p = 300	3.70	.01	6.30	.00	.94	.01
	3	.1973	.1991	.1949							
	4	.1569	.2015	.1949	L_p = 1500	3.94	.00	6.06	.00	.98	.02
	5	.1381	.1588	.1559							
	6	.0943	.0916	.1039	Analytical	4.02	.02	5.00	.01	1.01	.02
	7	.0592	.0670	.0594							
	8	.0266	.0257	.0297							
	9	.0089	.0119	.0132							
	10	.0024	.0036	.0059							
	11	.0021	.0004	.0026							

Table 2.4-2

$\Psi = 9$

L	k	Probability (P_k) simulated $L_p = 300$	Probability (P_k) simulated $L_p = 1500$	Probability (P_k) analytical
10	0	.0009	.0002	.0001
	1	.0493	.0073	.0006
	2	.0194	.0071	.0028
	3	.0481	.0189	.0085
	4	.0599	.0265	.0190
	5	.0598	.0367	.0342
	6	.0591	.0467	.0514
	7	.0503	.0661	.0660
	8	.0863	.0958	.0743
	9	.1006	.1011	.0743
	10	.0902	.0764	.0669
	11	.0859	.0684	.0602
	12	.0743	.0663	.0542
	13	.0746	.0554	.0488
	14	.0577	.0508	.0439
	15	.0338	.0531	.0395
	16	.0192	.0426	.0355
	17	.0150	.0262	.0320
	18	.0080	.0220	.0288
	19	.0059	.0212	.0259
	20	.0015	.0243	.0233
	21		.0180	.0210
	22		.0198	.0189
	23		.0210	.0170
	24		.0151	.0153
	25		.0052	.0138
	26		.0040	.0124
	27		.0013	.0112
	28		.0015	.0100
	29		.0008	.0090
11	0	.0009	.0002	.0001
	1	.0482	.0080	.0009
	2	.0222	.0102	.0040
	3	.0550	.0182	.0121
	4	.0564	.0313	.0272
	5	.0612	.0467	.0490
	6	.0660	.0714	.0735
	7	.0628	.0931	.0945
	8	.1173	.1210	.1063
	9	.1283	.1290	.1063
	10	.1194	.0999	.0957
	11	.0888	.0796	.0783
	12	.0494	.0558	.0640
	13	.0479	.0495	.0524
	14	.0343	.0434	.0429
	15	.0183	.0348	.0351
	16	.0108	.0216	.0287
	17	.0095	.0222	.0235
	18	.0033	.0258	.0192
	19		.0236	.0157
	20		.0103	.0129
	21		.0031	.0105
	22		.0013	.0086
	23			.0070

	E(k)	E(n)	E(ID)	E(WT)	$E(T_{WS})$	$P_r(k \geq L)$
simulated						
$L_p = 300$	8.95	1.21	2.88	.15	1.09	.47
$L_p = 1500$	11.74	3.01	1.76	.34	1.32	.59
Analytical	15.02	6.02	1.00	.67	1.67	.67
simulated						
$L_p = 300$	8.25	.46	4.03	.06	.99	.26
$L_p = 1500$	9.91	1.17	2.98	.13	1.11	.37
Analytical	10.94	1.94	2.00	.22	1.22	.43

Table 2.4-3

V = 9

L	k	PRobability (P_k)				E(k)	E(n)	E(ID)	E(WT)	$E(T_{ws})$	$P_r(k=L)$
		Simulated		Analytical	simulated						
		L_p = 300	L_p=1500								
12	0	.0009	.0007	.0001							
	1	.0492	.0075	.0010	L_p = 300	8.02	.20	5.08	.20	.96	.14
	2	.0207	.0104	.0046							
	3	.0541	.0183	.0137	L_p = 1500	9.22	.48	4.09	.05	1.03	.23
	4	.0578	.0319	.0308							
	5	.0645	.0525	.0555	Analytic	9.80	.80	3.00	.09	1.09	.27
	6	.0662	.0815	.0833							
	7	.0769	.1067	.1070							
	8	.1309	.1375	.1204							
	9	.1249	.1338	.1204							
	10	.1252	.1100	.1084							
	11	.0908	.0816	.0887							
	12	.0399	.0567	.0665							
	13	.0421	.0455	.0499							
	14	.0282	.0413	.0378							
	15	.0176	.0349	.0281							
	16	.0052	.0186	.0210							
	17	.0039	.0174	.0158							
	18	.0010	.0077	.0118							
	19	—	.0048	.0089							
	20	—	.0008	.0067							
13	0	.0009	.0007	.0001	simulated						
	1	.0492	.0079	.0011							
	2	.0207	.0111	.0048	L_p = 300	7.89	.07	6.14	.01	.95	.08
	3	.0557	.0189	.0144							
	4	.0577	.0317	.0325	L_p = 1500	8.92	.18	5.16	.02	1.00	.14
	5	.0657	.0531	.0584							
	6	.0707	.0876	.0876	Analytic	9.35	.35	4.00	.04	1.04	.16
	7	.0801	.1112	.1127							
	8	.1318	.1424	.1267							
	9	.1248	.1386	.1267							
	10	.1302	.1171	.1141							
	11	.0918	.0845	.0933							
	12	.0419	.0589	.0700							
	13	.0377	.0486	.0485							
	14	.0225	.0390	.0336							
	15	.0131	.0261	.0232							
	16	.0045	.0102	.0161							
	17	.0010	.0095	.0111							
	18	—	.0018	.0077							
	19	—	.0011	.0053							
14	0	.0009	.0007	.0001	simulated						
	1	.0492	.0081	.0011							
	2	.0207	.0116	.0049	L_p = 300	7.84	.02	7.17	.00	.94	.03
	3	.0557	.0191	.0147							
	4	.0580	.0316	.0332	L_p = 1500	8.81	.06	6.22	.01	.99	.07
	5	.0677	.0537	.0597							
	6	.0732	.0902	.0896							
	7	.0790	.1133	.1152	Analytic.	9.16	.16	5.00	.02	1.02	.09
	8	.1312	.1436	.1296							
	9	.1273	.1438	.1296							
	10	.1304	.1155	.1166							
	11	.0884	.0864	.0954							
	12	.0426	.0608	.0716							
	13	.0439	.0508	.0495							
	14	.0183	.0348	.0398							
	15	.0114	.0205	.0205							
	16	.0020	.0080	.0132							
	17	.0001	.0066	.0085							
	18	—	.0007	.0054							
	19	—	.0003	.0035							

2.5 Simulation of Waiting Lines in Series

A sequence of m service stations can be arranged in series such that a unit has to go through one station after another of the sequence before the whole service is finished. Some basic rules for sequential service lines are given by P.M.Morse [46, p.34-38], however the analytical discussion of the system in series is rather difficult. We confine ourselves here to some particular cases for which solution will be given by simulation.

The problem is what happens to a unit after it has completed service by the k^{th} station in the line. "If the $(k+1)^{st}$ station has already finished its previous unit, it is free and can take the next unit as it emerges from the k^{th} station." [46, p.34]. Let us suppose that the $(k+1)^{st}$ station is not free. Then we distinguish 3 cases. "The emerging unit could

(1) emerge from the k^{th} station and wait in a queue till the $(k+1)^{st}$ is free; or

(2) stay in the k^{th} station, blocking it from taking on another unit, even though it is idle and another unit may be waiting in the $(k-1)^{st}$ station, until the $(k+1)^{st}$ becomes free to take the waiting unit, thus unblocking the k^{th} station; or

(3) all units in service have to stay in their respective stations until all stations complete service on the units in them, when all units move at once, each into the next station, the one in the m^{th} station leaving the line and one from the queue moving into the first station (if there is no unit in the queue, all units wait until another unit arrives, when they all move)." [46, p.35]

A solution of the case 1 is given by T.H.Naylor [49, p.141-151]. We solve the problem mentioned in section 2 above for two particular cases (Models 4 and 5) and, in addition, the problem under the conditions given in section 3 (Model 6) by the method of simulation.

We assume in the case of section 2 (in Models 4 and 5) that each individual station has an exponential service time distribution each having the same mean rate μ, and that the arrivals follow Poisson-distribution with mean arrival rate λ.

2.5.-1 A two-station service line in which no queues ahead of station 1 or between the two stations are allowed (Model 4)

The Model

As a first application we simulate a two-station service line in which no queues ahead of station 1 or between the stations are allowed.A particular state of the system (of type 2) is given by specifying the states of the two stations.Station 2 can either be empty (0) or full (1),while station 1 can be empty (0) or full and working (1) or blocked (b) (because it has finished its service,but the second station is occupied).

According to this conditions one of the states (0,0),(1,0),(0,1),(1,1) and (b,1) occurs at a given time (the first number refers to the state of station 1,and the second to the state of station 2).If the probability functions are labelled with two subscripts corresponding to the states of the two stations,then their analytic description [33,p.381-382] is given as follows

$$P_{oo} = \frac{2}{F},\quad P_{o1} = \frac{2\Psi}{F},\quad P_{1o} = \frac{\Psi^2+2\Psi}{F},\quad P_{11} = P_{b1} = \frac{\Psi^2}{F},$$

where $\Psi = \lambda/\mu$ and

$$F = 3\Psi^2+4\Psi+2.$$

The expected number of units in the system is

$$E(k) = \frac{5\Psi^2+4\Psi}{F} = \begin{cases} 2\Psi - \frac{3}{2}\Psi^2 & (\Psi << 1) \\ 1 & (\Psi = 1) \\ \frac{5}{3} - \frac{32}{45\Psi} & (\Psi >> 1) \end{cases}$$

The expected number of busy service stations is

$$E(SB) = \frac{4\Psi^2+4\Psi}{F} = \begin{cases} 2\Psi - 2\Psi^2 & (\Psi << 1) \\ \frac{8}{9} & (\Psi = 1) \\ \frac{4}{3} - \frac{4}{9\Psi} & (\Psi >> 1) \end{cases}$$

The probability that a customer arrives at station 1 and cannot be served is

$$P_R = \frac{3\Psi^2+2\Psi}{F} = \begin{cases} \Psi - \frac{1}{2}\Psi^2 & (\Psi << 1) \\ \frac{5}{9} & (\Psi = 1) \\ 1 - \frac{2}{3\Psi} & (\Psi >> 1) \end{cases}$$

During the simulation the periods corresponding to the states (0,0), (1,0),(0,1),(1,1) and (b,1) are denoted by $T_{00}, T_{10}, T_{01}, T_{11}$ and T_{b1} and $T_{00}+T_{10}+T_{01}+T_{11}+T_{b1} = TATS$,
where TATS is the whole period of the simulation.The simulated probabilities can be calculated from these quantities as

$$p_{ik}^{(simulated)} = \frac{T_{ik}}{TATS} \quad (i,k = 0,1) \text{ and } p_{b1}^{(simulated)} = \frac{T_{b1}}{TATS}$$

The other corresponding statistics are

$$E(k)^{(s)} = p_{01}^{(s)} + p_{10}^{(s)} + 2(p_{11}^{(s)} + p_{b1}^{(s)})$$

$$E(SB)^{(s)} = p_{01}^{(s)} + p_{10}^{(s)} + p_{b1}^{(s)} + 2p_{11}^{(s)}$$

$$p_R^{(s)} = p_{10}^{(s)} + p_{b1}^{(s)} + p_{11}^{(s)*} \qquad {}^{*}(s) \text{ corresponds:"simulation"}$$

The variables used in the flow-chart are:

RAR:random time interval between two successive arrivals,

RSR_h:random service time at the h^{th}service station (h = 1,2),

TP :predicted time of the next arrival,

TS_h:time for the end of a single arrival's service at the h^{th}service station (h = 1,2),

i :the current number of arrivals that have joined the system,

L_p :the number of arrivals that have been simulated.

To produce a flow-chart for the purpose of simulation of a sequential service line under the above conditions one has to take into consideration firstly the possible transitions from one state of the system to another.These transitions are

from state (0,0) to state (1,0);
from state (1,0) to state (0,1);
from state (0,1) either to (1,1) or to (0,0);
from state (1,1) either to (1,0) or to (b,1); and
from state (b,1) to state (0,1).

In addition,five other transitions may happen with zero probabilities theoretically when at least two random events coincide in time.These are

from (1,0) to (1,1) when $TP^{(k+1)} = TS_1^{(k)}$ (the end of the service of unit $\underline{k}$ at station 1 coincide with the arrival of the $(k+1)^{th}$unit),

from (0,1) to (1,0) when $TP^{(k+1)} = TS_2^{(k)}$,

from (1,1) to (0,1) when $TS_1^{(k+1)} = TS_2^{(k)}$,

from (b,1) to (1,1) when $TP^{(k+1)} = TS_2^{(k-1)}$,

from (1,1) to (1,1) when $TP^{(k+1)} = TS_1^{(k)}$ and $TS_1^{(k)} = TS_2^{(k-1)}$.

There are two cases in the course of the simulation when a new arrival may not join the system:

(1) station 1 is occupied,

(2) the service of the k^{th} unit at station 1 has finished,but the $(k-1)^{st}$ unit at station 2 is still under service.

Arrivals met any of these conditions are refused and not served.

A series of all the possible "ordinary" transitions in Figure 2.5-1 and a series of the "unlikely" transitions in Figure 2.5-2 are shown on page 71 ,where the integers in the first three lines indicate the

serial numbers of the units.

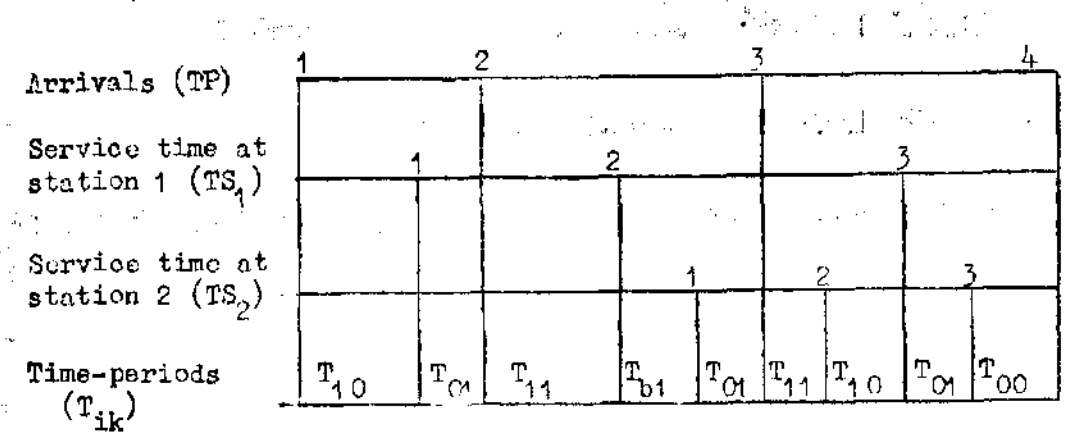


Figure 2.5-1

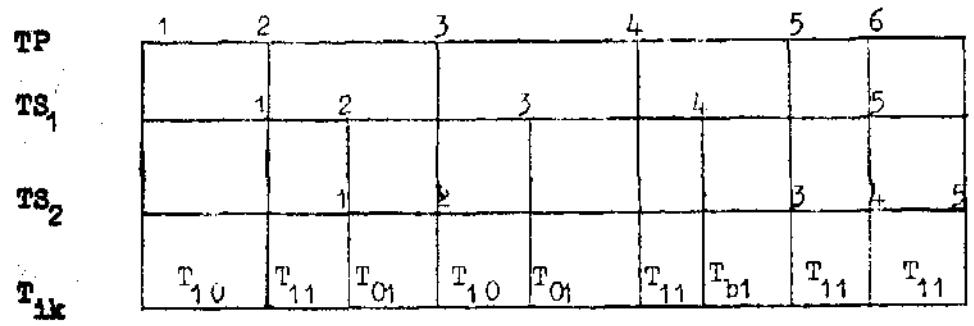


Figure 2.5-2

When the first arrival enters the system at 0 time $i = 0$, $T_{00} = 0$, $T_{10} = 0$, $T_{01} = 0$, $T_{11} = 0$ and $T_{b1} = 0$.

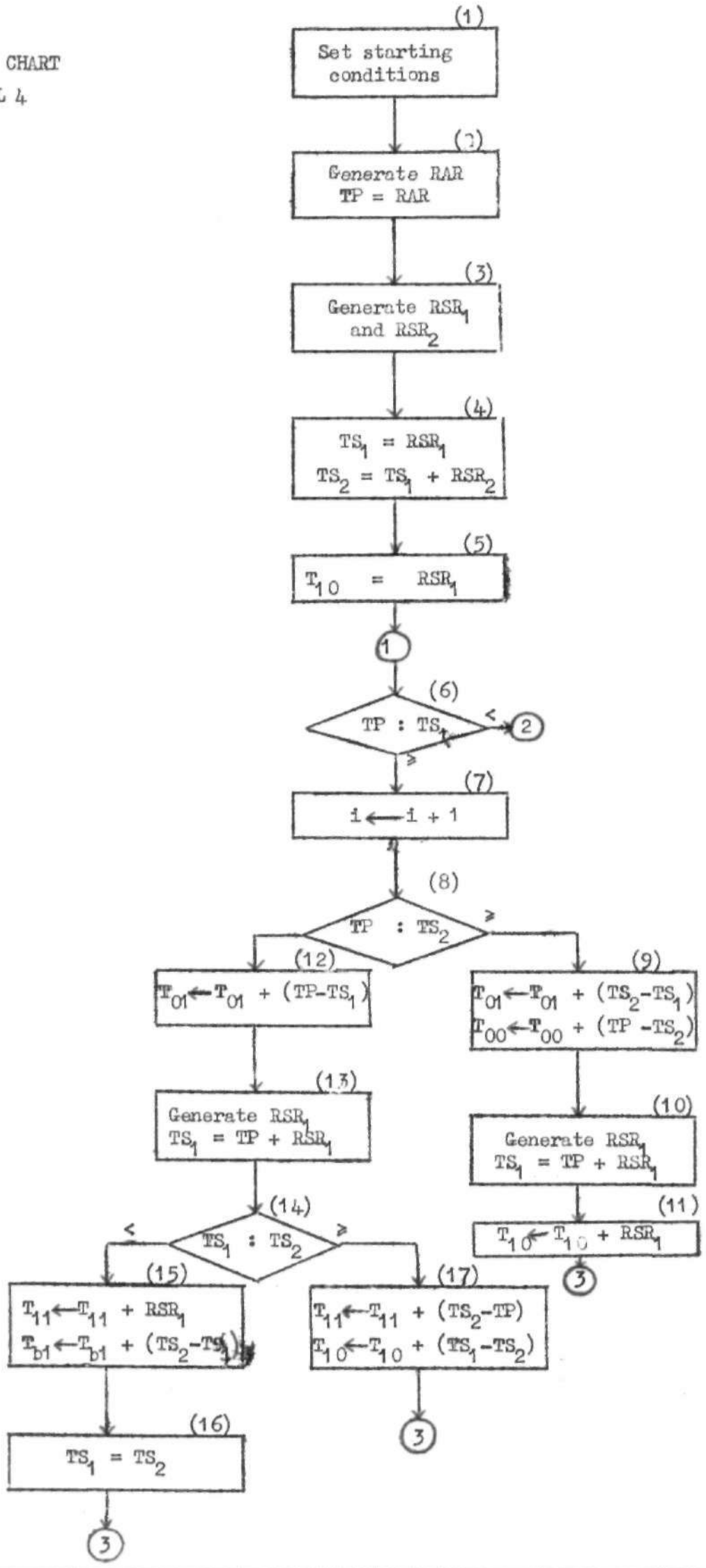
The Flow Chart

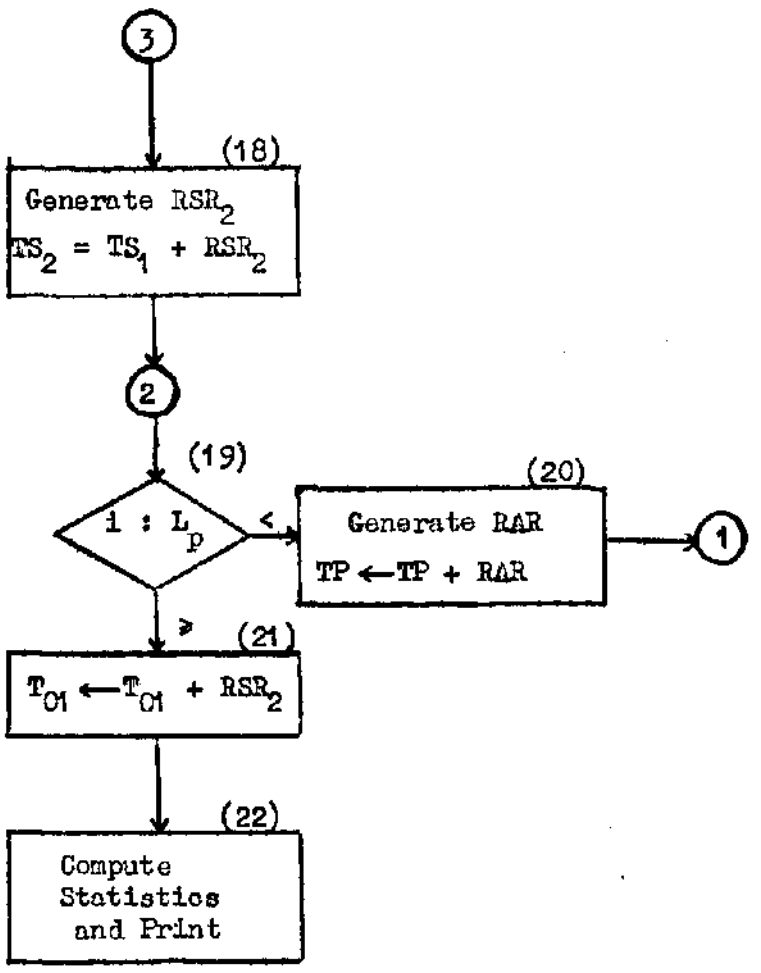
In block 2 the period between the arrivals of the 1^{st} and 2^{nd} units is generated and, this interarrival time is assigned to TP. Then random service times are generated for both stations, and ending times for the first services (TS_1, TS_2) are designated (blocks 3 and 4). At the outset the system takes up the state (1,0).

Block 6 decides whether a new arrival may join the system or not. Each time when $TP < TS_1$ the arrival is rejected from the system, a new interarrival time and TP are generated in block 20, then control is transferred to label 1 until $TP \geq TS_1$. Units rejected are not taken into account further on, but the number of arrivals accepted is increased by 1 (block 7).

The decision in block 8 is necessary to calculate the appropriate periods that have taken place so far. Then a new service time is generated (block 10 or 13). If $TP < TS_2$ another decision must be made (by block 14) before a new value for TS_2 is assigned. Obviously, the event $TS_1 < TS_2$ occurs when the $(k+1)^{st}$ unit has finished its service at station 1, but unit $\underline{k}$ is under service at station 2, and each time if $Ts_1^{(k+1)} < TS_2^{(k)}$ a value for $T_{b1} = TS_2^{(k)} - TS_1^{(k+1)}$ is calculated. This time a temporary value is assigned to TS_1 (block 16), which is not an ending time for service at station 1, but is required to determine

FLOW CHART
MODEL 4




3
(18)
Generate RSR_2
$TS_2 = TS_1 + RSR_2$
2
(19)
$1 : L_p$
<
(20)
Generate RAR
TP ← TP + RAR
1
≥
(21)
$T_{01} \leftarrow T_{01} + RSR_2$
(22)
Compute
Statistics
and Print

the correct value for the period T_{01} in the next step.

After the performance of any of the 3 main branches of the process (determined by blocks 6,8 and 14) a control is transferred to label 3 to generate a new ending time for service at station 2 (block 18).

The process described above is repeated until $i = L_p$ then the last period of the simulation is taking place (block 21), and the required statistics are computed.

2.5-2 A two-station system with a waiting line no longer than one unit before the second station but none before the first (Model 5)

The Model

In this model we also consider a two-station system, and allow a waiting line no longer then one unit before the second station, but none before the first. The allowed states are (0,0),(1,0),(0,1),(1,1),(0,2), (1,2) and (b,2), where the state (b,2) occurs when the queue for station 2 is filled, the service at station 1 has finished and so station 1 is blocked.

The probabilities to consider are

$$P_{00} = \frac{\Psi+4}{G},\quad P_{01} = \frac{\Psi^2+4\Psi}{G},\quad P_{02} = \frac{2\Psi^2}{G},\quad P_{10} = \frac{\Psi^3+3\Psi^2+4}{G},$$

$$P_{11} = \frac{\Psi^3+2\Psi^2}{G},\quad P_{12} = P_{b2} = \frac{\Psi^3}{G} \quad \text{and} \quad \Psi = \lambda/\mu \ ;$$

$$G = 4\Psi^3+8\Psi^2+9\Psi+4.$$

The quantities

$$L(k) = \frac{9\Psi^3+12\Psi^2+8\Psi}{G} = \begin{cases} 2\Psi & (\Psi << 1) \\ \frac{29}{25} & (\Psi = 1) \\ \frac{9}{4} & (\Psi >> 1) \end{cases}$$

$$E(SB) = \frac{6\Psi^3+10\Psi^2+8\Psi}{G} = \begin{cases} 2\Psi & (\Psi << 1) \\ \frac{24}{25} & (\Psi = 1) \\ \frac{3}{2} & (\Psi >> 1) \end{cases}$$

$$P_R = \frac{4\Psi^3+5\Psi^2+4\Psi}{G} = \begin{cases} \Psi-\Psi^2 & (\Psi << 1) \\ \frac{13}{25} & (\Psi = 1) \\ 1-\frac{3}{4\Psi} & (\Psi >> 1) \end{cases}$$

are the expected number of units in the system, the expected number of busy stations, and the fraction of times the first station is occupied, preventing an arrival from entering, respectively. The fraction of time that station 2 is busy is $T_2 = \frac{1}{2} E(SB)$, [46, p.37-38].

Using the same notation as in the previous model the simulated probabilities and other statistics are

$$p_{ik}^{(s)} = \frac{T_{ik}}{TATS} \quad (i,k = 0,1) \text{ and } p_{j2}^{(s)} = \frac{T_{j2}}{TATS} \quad (j = 0,1,b)$$

$$E(k)^{(s)} = p_{10}^{(s)} + p_{01}^{(s)} + 2(p_{11}^{(s)} + p_{02}^{(s)}) + 3(p_{12}^{(s)} + p_{b2}^{(s)})$$

$$E(B)^{(s)} = p_{10}^{(s)} + p_{01}^{(s)} + 2(p_{11}^{(s)} + p_{02}^{(s)} + p_{12}^{(s)} + p_{b2}^{(s)})$$

$$p_R^{(s)} = p_{10}^{(s)} + p_{11}^{(s)} + p_{12}^{(s)} + p_{b2}^{(s)} .$$

Transitions

The possible transitions between states of this model are

from state (0,0) to state (1,0)
from (1,0) to (0,1)
from (0,1) either to (1,1) or to (0,0)
from (1,1) either to (1,0) or to (0,2)
from (1,2) either to (b,2) or to (1,1)
from (0,2) either to (1,2) or to (0,1)
from (b,2) to (0,2).

Besides, the following transitions might happen theoretically with a probability of zero

from (1,0) to (1,1) when $TP^{(k+1)} = TS_1^{(k)}$ (the $(k+1)^{st}$ unit arrives in the system at the very moment when the service of the k^{th} unit ends at station 2 and its service starts at station 2),

from (0,1) to (1,0) when $TP^{(k+1)} = TS_2^{(k)}$

from (1,1) to (1,2) when $TP^{(k+1)} = TS_1^{(k)}$ and $TS_1^{(k)} < TS_2^{(k-1)}$

from (1,1) to (1,1) when $TP^{(k+1)} = TS_1^{(k)}$ and $TS_1^{(k)} = TS_2^{(k-1)}$

from (1,1) to (0,1) when $TS_1^{(k)} = TS_2^{(k-1)}$

from (1,2) to (0,2) when $TS_1^{(k)} = TS_2^{(k-2)}$

from (1,2) to (1,2) when $TP^{(k+1)} = TS_1^{(k)}$ and $TS_1^{(k)} = TS_2^{(k-2)}$

from (0,2) to (1,1) when $TP^{(k+1)} = TS_2^{(k-1)}$

from (b,2) to (1,2) when $TP^{(k+1)} = TS_2^{(k-2)}$.

A series of the transitions of the states including the all possible "regular" cases and, another series of states with "unlikely" cases are displayd in Figures 2.5-3 and 2.5-4, respectively.

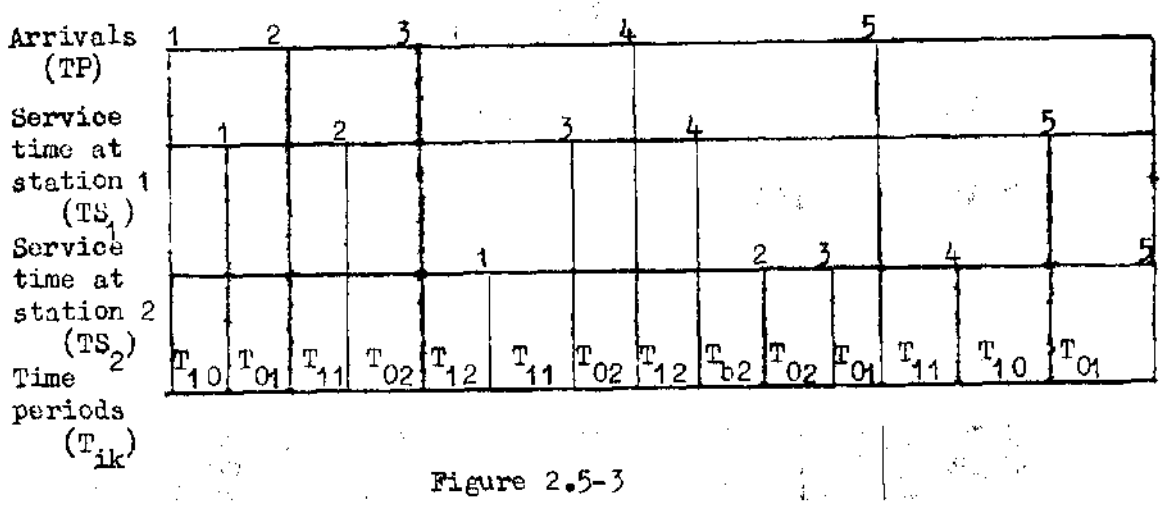


Figure 2.5-3

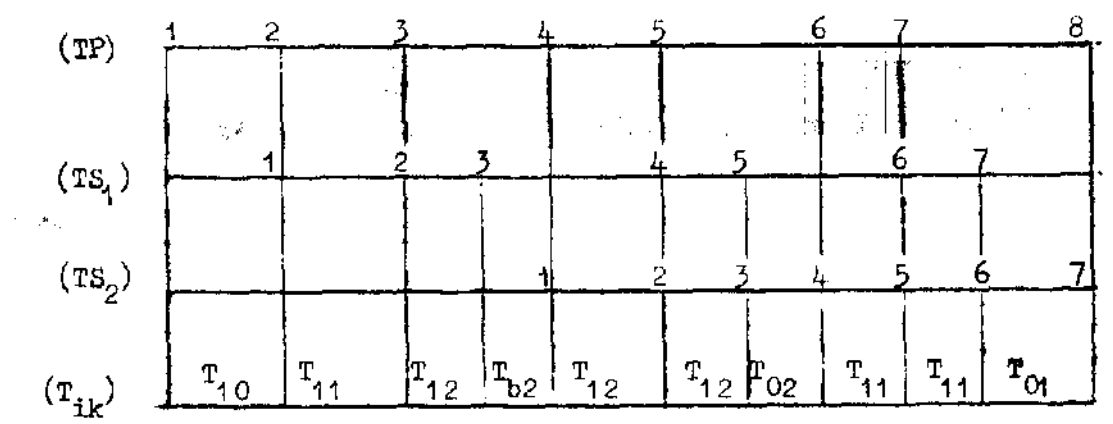


Figure 2.5-4

The Flow Chart

The flow-chart of Model 5 agrees with that of Model 4 at the outset with the slight modification that the generation of new arrival times is carried out separately in blocks 8 and 22 (and in a later section of the flow-chart in block 35 as well).

There is an indication by the equation TP = ∞ (blocks 25 and 38) that a large value is assigned to TP if i = L_p, to direct the process (at block 26) to the finishing part of the simulation (label 5).

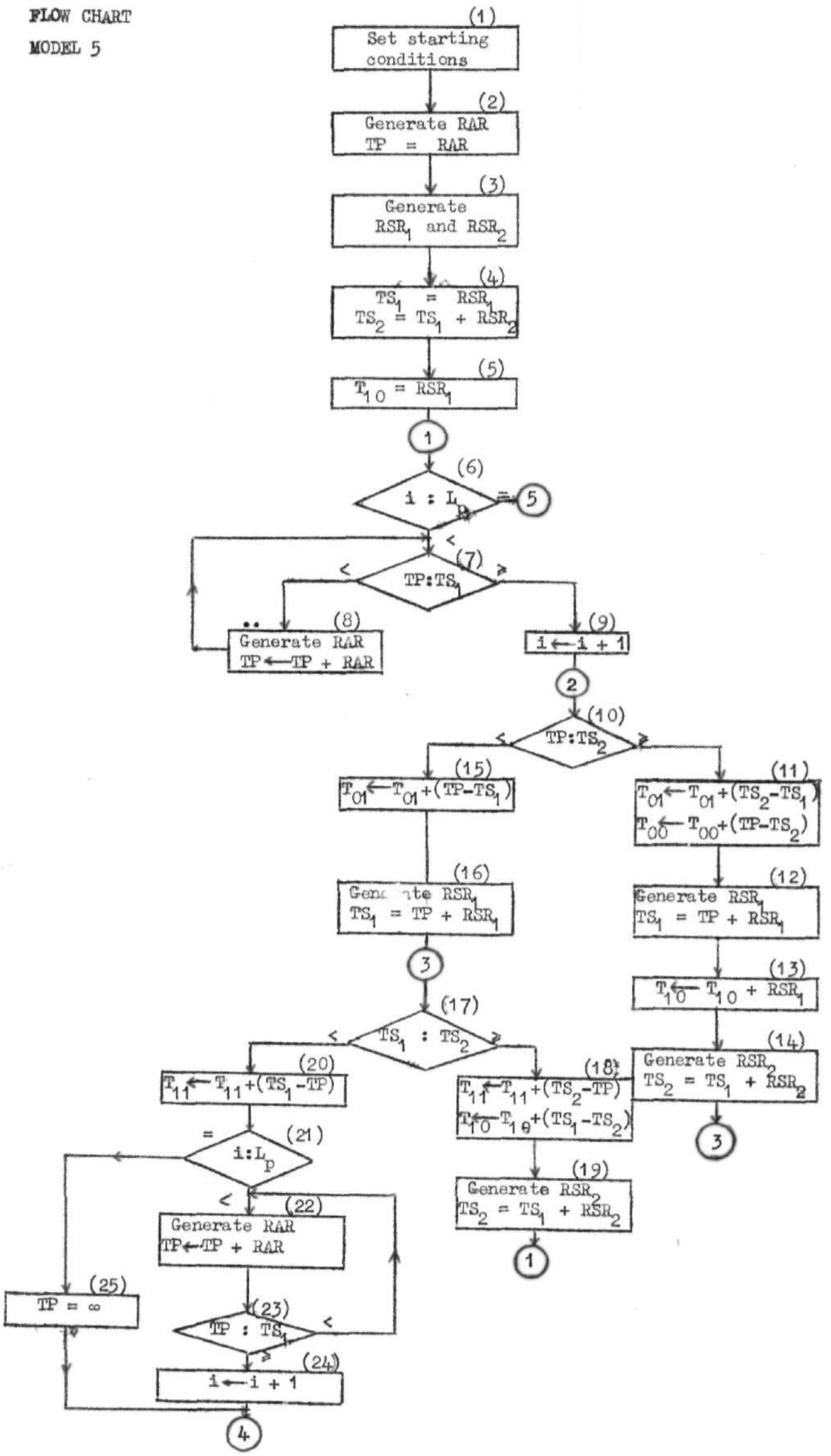
In the periods when the states (1,2),(0,2),(b,2) do not occur, the blocks from 1-25 are used (as in Model 4). In the further part of this flow-chart the extension and repetition of Model 4 appear by the introduction of two new blocks (26 and 29) which are similar to the decisions of blocks 10 and 17. The role of these decisions differ from that of blocks 10 and 17 in that there is one more unit in the system now, so the periods T_{02}, T_{12} and T_{b2} required for the calculations can be determined if we also take account of these decisions.

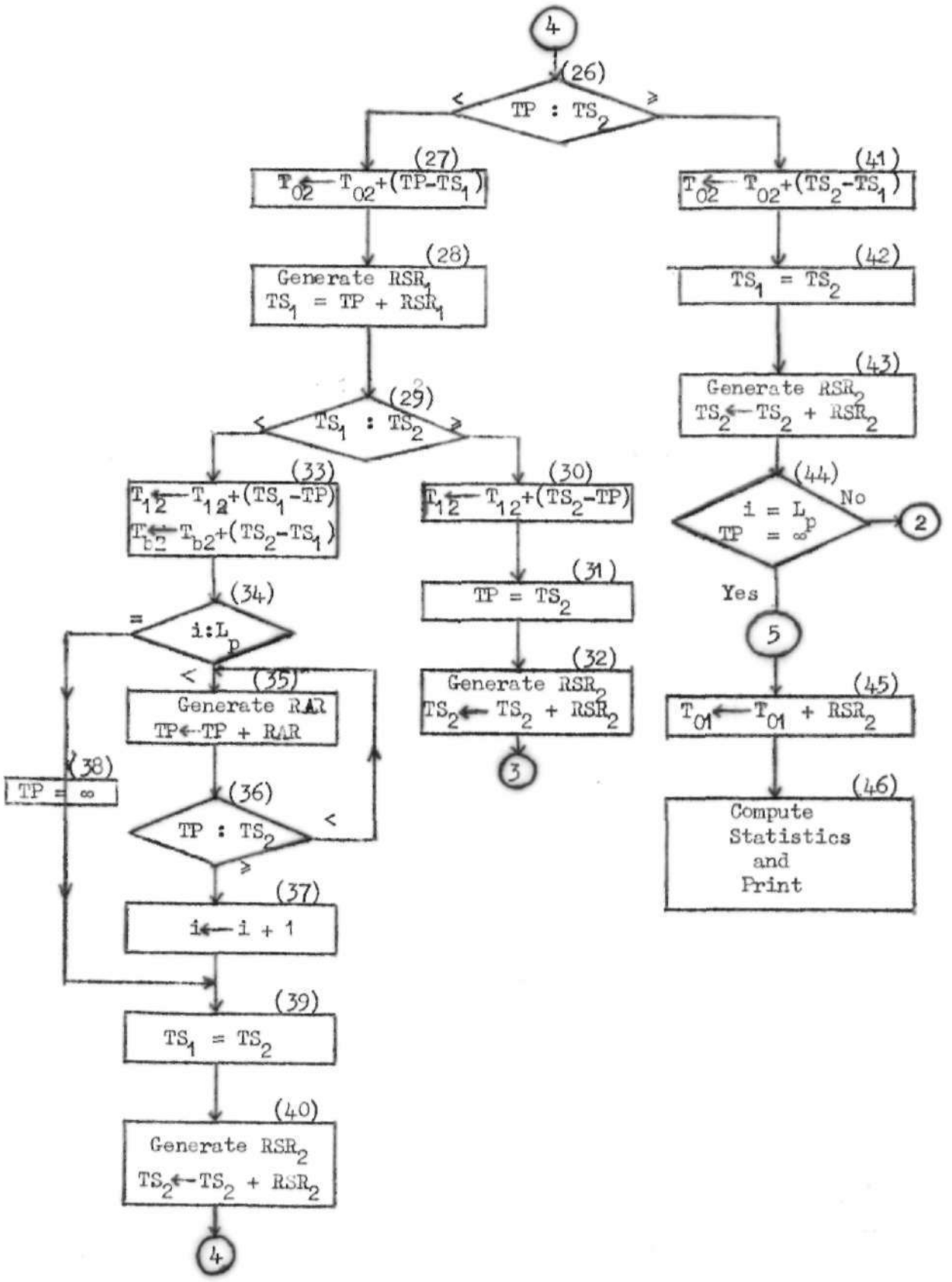
Generation of service times occurs if either station 1 is empty and a new arrival has entered the system or a service has finished at station 2 and there has been a waiting unit between the two stations.

Three new blocks (31, 39 and 42) are introduced now, similar to block 16 of Model 4, to make use of the branches created so far. By means of these assignments the enlargement of the flow-chart can be

FLOW CHART

MODEL 5




4
(26)
TP : TS2
(27)
T02 ← T02+(TP-TS1)
(28)
Generate RSR1
TS1 = TP + RSR1
(29)
TS1 : TS2
(33)
T12 ← T12+(TS1-TP)
Tb2 ← Tb2+(TS2-TS1)
(34)
i:Lp
(35)
Generate RAR
TP ← TP + RAR
(36)
TP : TS2
(37)
i ← i + 1
(38)
TP = ∞
(39)
TS1 = TS2
(40)
Generate RSR2
TS2 ← TS2 + RSR2
4
(30)
T12 ← T12+(TS2-TP)
(31)
TP = TS2
(32)
Generate RSR2
TS2 ← TS2 + RSR2
3
(41)
T02 ← T02+(TS2-TS1)
(42)
TS1 = TS2
(43)
Generate RSR2
TS2 ← TS2 + RSR2
(44)
i = Lp
TP = ∞
No
2
Yes
5
(45)
T01 ← T01 + RSR2
(46)
Compute
Statistics
and
Print

avoided.Every time when a transition from state (1,2) into (1,1) occurs TS_1 is assigned to TP (block 31),and when transition from (0,2) into (0,1) happens block 42 is used to assign TS_2 to TS_1.In addition, after each (b,2) state TS_1 takes up the value of TS_2 (block 39).

The process described above is repeated until all the units to be simulated have entered the system and have been served.

The flow-chart can be applied for the "unlikely" cases as well.This time the differences $|TP - TS_1|$, $|TP - TS_2|$ and $|TS_1 - TS_2|$are all equal to zero (which are used for calculation of T_{ik} in blocks 11,15, 18,20,27,30,33 and 41).

Numerical Results

Calculation of the simulated quantities of both of Model 4 and Model 5 has been carried out with varying values of the parameters λ and μ .Tables 2.5-1 and 2.5-2 display a comparison between the simulated and analytical quantities with different values of Ψ for Model 4 and 5,respectively.

Simulation was performed in each particular case for 300 and then 1500 arrivals.It can be seen from the results that by increasing the number of iterations the simulated quantities fit the analytical ones much better.

For values of the parameters not included in the above tables the corresponding quantities also agree well,without exception.

2.5-3 Simulation of Waiting Lines in Series Under Special Conditions (Model 6)

The Model

It is assumed that each separate service station (process) has its own service rate without specifying particular probability distribution for either the interarrival or the service times.

In the formulation of the mathematical model for sequential service line defined in section 3 on page 68,the following notations are used:

m:the number of service stations,

RAR_i:the time interval between the arrival of the i^{th} unit and the $(i-1)^{st}$unit,where $i = 2,\ldots,L_p$.

RSR_{ih}:the service time for the i^{th} unit at the h^{th} station,where $i = 1,\ldots,L_p$ and $h = 1,\ldots,m$.

TS_{i0}:the arrival time of the i^{th} unit in the system,where $i = 1,\ldots,L_p$.

TS_{ih}:ending time of service of the i^{th}unit at the h^{th} station where $i = 1,\ldots,L_p$ and $h = 1,\ldots,m$.

WT_{ih}:the amount of time the i^{th} unit spends waiting to enter the h^{th} station,where $i = 1,\ldots,L_p$ and $h = 1,\ldots,m$.

Comparison of the the quantities of Model 4 obtained by simulation to the analytical results

	Ψ = .3			Ψ = .5			Ψ = 1.0			Ψ = 4.5		
	Simulated		Analytical	Simulated		Analytical	Simulated		Analytical	Simulated		Analytical
	Lp =300	L_p =1500		L_p = 300	L_p = 1500		L_p = 300	L_p = 1500		L_p = 300	L_p = 1500	
P_{00}	.5377	.5693	.5764	.3959	.4153	.4211	.2057	.2166	.2222	.0305	.0264	.0248
P_{10}	.2008	.2004	.1988	.2433	.2618	.2632	.3050	.3300	.3333	.3113	.3542	.3622
P_{01}	.2003	.1779	.1729	.2184	.2114	.2105	.2347	.2243	.2222	.1092	.1142	.1115
P_{11}	.0256	.0267	.0259	.0692	.0565	.0526	.1186	.1172	.1111	.2585	.2566	.2508
P_{b1}	.0350	.0256	.0259	.0726	.0548	.0526	.1349	.1118	.1111	.2891	.2484	.2508
E(k)	.52	.48	.48	.75	.70	.68	1.05	1.01	1.00	1.52	1.48	1.48
E(SB)	.49	.46	.45	.67	.64	.63	.91	.90	.89	1.23	1.23	1.23
P_R	.2615	.2527	.2507	.3850	.3732	.3684	.5586	.5590	.5556	.8589	.8591	.8638

Table 2.5-1

Comparison of the quantities of Model 5 obtained by simulation to the analytical results.

	Ψ = .3			Ψ = .5			Ψ = 1.0			Ψ = 4.5		
	Simulated		Analytical	Simulated		Analytical	Simulated		Analytical	Simulated		Analytical
	L_p = 300	L_p = 1500		L_p = 300	L_p = 1500		L_p = 300	L_p = 1500		L_p = 300	L_p = 1500	
P_{00}	.5445	.5680	.5712	.3726	.4012	.4091	.1706	.1933	.2000	.0122	.0143	.0149
P_{10}	.1895	.1979	.1989	.2339	.2602	.2614	.2885	.3172	.3200	.2992	.2669	.2975
P_{01}	.1938	.1743	.1714	.2160	.2087	.2045	.1868	.2019	.2000	.0699	.0628	.0670
P_{11}	.0345	.0286	.0275	.0759	.0599	.0568	.1359	.1278	.1200	.2348	.2345	.2305
P_{02}	.0310	.0252	.0239	.0676	.0462	.0455	.1049	.0864	.0800	.0716	.0744	.0709
P_{12}	.0026	.0026	.0036	.0152	.0104	.0114	.0563	.0381	.0400	.1461	.1699	.1596
P_{b2}	.0036	.0034	.0036	.0179	.0132	.0114	.0559	.0350	.0400	.1645	.1769	.1596
E(k)	.53	.50	.49	.84	.75	.74	1.29	1.17	1.16	1.91	1.99	1.92
E(SB)	.53	.49	.46	.80	.73	.66	1.18	1.09	.96	1.60	1.64	1.38
P_R	.2302	.2325	.2335	.3429	.3437	.3409	.5366	.5182	.5200	.8447	.8482	.8472

Table 2.3-2

IDT_{ih}: the amount of time the h^{th} station remains idle while waiting for the i^{th} unit to arrive, where $i = 1,\ldots,L_p$ and $h = 1,\ldots,m$.

Under the given circumstances a new series of services may start immediately after each station has finished service on the respective unit, otherwise service does not start at either station until the next arrival enters the system.

It follows that waiting time as well as idle time may happen at the same station simultaneously. Waiting time occurs when either a unit has arrived in the system but cannot be serviced yet at the 1^{st} station, or service on a unit at the h^{th} station has finished but the same unit cannot enter the $(h+1)^{st}$ station for service.

Idle time occurs when a service has finished at a station but the next arrival has not arrived yet or cannot enter this station.

The whole simulation can be split up three parts, namely:

(1) <u>starting period</u>: from 0 time until the first unit starts its service at station <u>m</u>,

(2) <u>operating period</u>: the time that elapses between the end of the starting period and the time that the last unit starts service at the first station,

(3) <u>finishing period</u>: the rest of the simulation.

During the "starting" period waiting and idle times can be calculated for subsequent arrivals as follows:

$$WT_{i,1} = TSMAX - TS_{i,0}$$
$$WT_{i-1,2} = TSMAX - TS_{i-1,1}$$
$$\vdots$$
$$WT_{1,i} = TSMAX - TS_{1,i-1}$$

and

$$IDT_{i,1} = TSMAX - TS_{i-1,1}$$
$$IDT_{i-1,2} = TSMAX - TS_{i-2,2}$$
$$\vdots$$
$$IDT_{1,i} = TSMAX - TS_{0,i}$$

where $i = 2,\ldots,m$ and $TSMAX = \text{maximum}(TS_{i,0}, TS_{i-1,1}, \ldots, TS_{1,i-1})$ and, all of the quantities $TS_{0,2},\ldots,TS_{0,m}$ that appear in the calculation of $IDT_{1,2},\ldots,IDT_{1,m}$ are equal to zero. It is obvious that

$$WT_{i-1,2} = IDT_{i,1}$$
$$WT_{i-2,3} = IDT_{i-1,2}$$
$$\vdots$$
$$WT_{1,i} = IDT_{2,i-1}.$$

Summarizing what has been said before

$$WT_{i-1,l+1} = TSMAX - TS_{i-1,l} \qquad (2.5\text{-}1)$$

and

$$IDT_{i-1,l+1} = TSMAX - TS_{i-(l+1),l+1} \qquad (2.5\text{-}2)$$

where for each subsequent $\underline{i}$ $(i = 2,\ldots,m)$ $l = 0,1,\ldots,i-1$.

The "operating" and "finishing" periods can be discussed together. The formulae for computing waiting and idle times are almost the same as above,but a slight modification has to be introduced,namely

$$WT_{i-1,l+n+1} = TSMAX - TS_{i-1,l+n} \qquad (2.5\text{-}3)$$

and

$$IDT_{i-1,l+n+1} = TSMAX - TS_{i-(l+1),l+n+1} \qquad (2.5\text{-}4)$$

where for each subsequent $\underline{i}$ $(m < i \leq L_p)$ $l = 0,1,\ldots,m-n-1$.

In these formulae $\underline{n}$ means the number of stations where the service of the last unit has finished or will finish before the present time of TSMAX.The value of $\underline{n}$ may vary from 0 to $\underline{m}$,and obviously $n = 0$ in the [redacted] "operating" period of the simulation.It can be seen that WT_{ih} and IDT_{ih} are obtained as in the "starting" period, but now these quantities must be calculated for each station(nevertheless not for all stations in the "starting" period).In the"finishing" period (when $i = L_p$) $n > 0$,so waiting and idle times don't occur at stations where all services have finished.

By the introduction of another variable,$\underline{k}$ formulae (2.5-3) and (2.5-4) can be expanded to the whole simulation and uniform formulae can be created for all the three periods,that is

$$WT_{i-1,k+l+1} = TSMAX - TS_{i-1,k+l} \qquad (2.5\text{-}5)$$

and

$$IDT_{i-1,k+l+1} = TSMAX - TS_{i-(l+1),k+l+1} \qquad (2.5\text{-}6)$$

where for each subsequent $\underline{i}$ $(i = 2,\ldots,m)$ $l = 0,1,\ldots,m-n-1$ with the restriction that the values of $\underline{n}$ and $\underline{k}$ vary during the different periods of the simulation as given in the Table (2.5-1) below

variable \ period	starting	operating	finishing
n	m - i	0	varies from 1 to m
k	0	0	n

Table. 2.5-1

After the end of the simulated process total waiting and idle times can be calculated at each station.If we denote these quantities by WT_h and IDT_h,respectively,then

$$WT_h = \sum_{i=1}^{L_p} WT_{ih} \qquad h = 1,\ldots,m$$

and

$$IDT_h = \sum_{i=0}^{L_p} IDT_{ih} \qquad h = 1,\ldots,m,$$

where IDT_{0h} occurs when the last unit at the h^{th} station has finished its service and $IDT_{0R} = TS_{L_p m} - TS_{L_p h}$. (It is supposed that $IDT_{11} = 0$.)

The Flow Chart

The simulation starts at zero time, so that $TS_{1,0} = 0$, and the first step is carried out in blocks 1 to 5 by setting the starting conditions, generating the first service time and assigning the required values to TS_{11} and TSMAX.

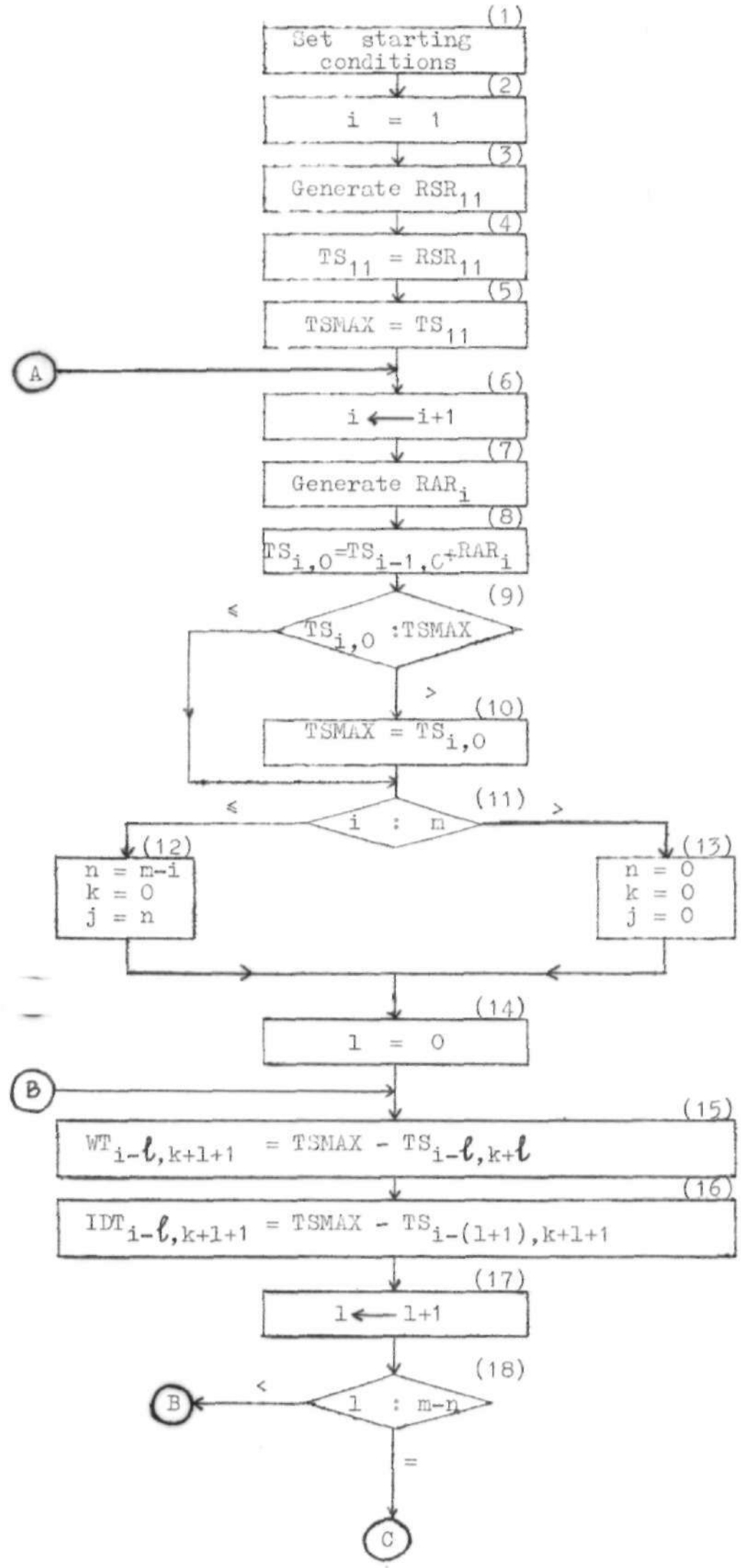
Each time a new unit arrives the value of i is increased by 1, a new interarrival time is generated and the arrival time of the i^{th} unit is determined (blocks 6-8). If this time ($TS_{i,0}$) is greater than TSMAX, the maximum of the momentary $TS_{i,h}$ values, then $TS_{i,0}$ will be assigned to TSMAX.

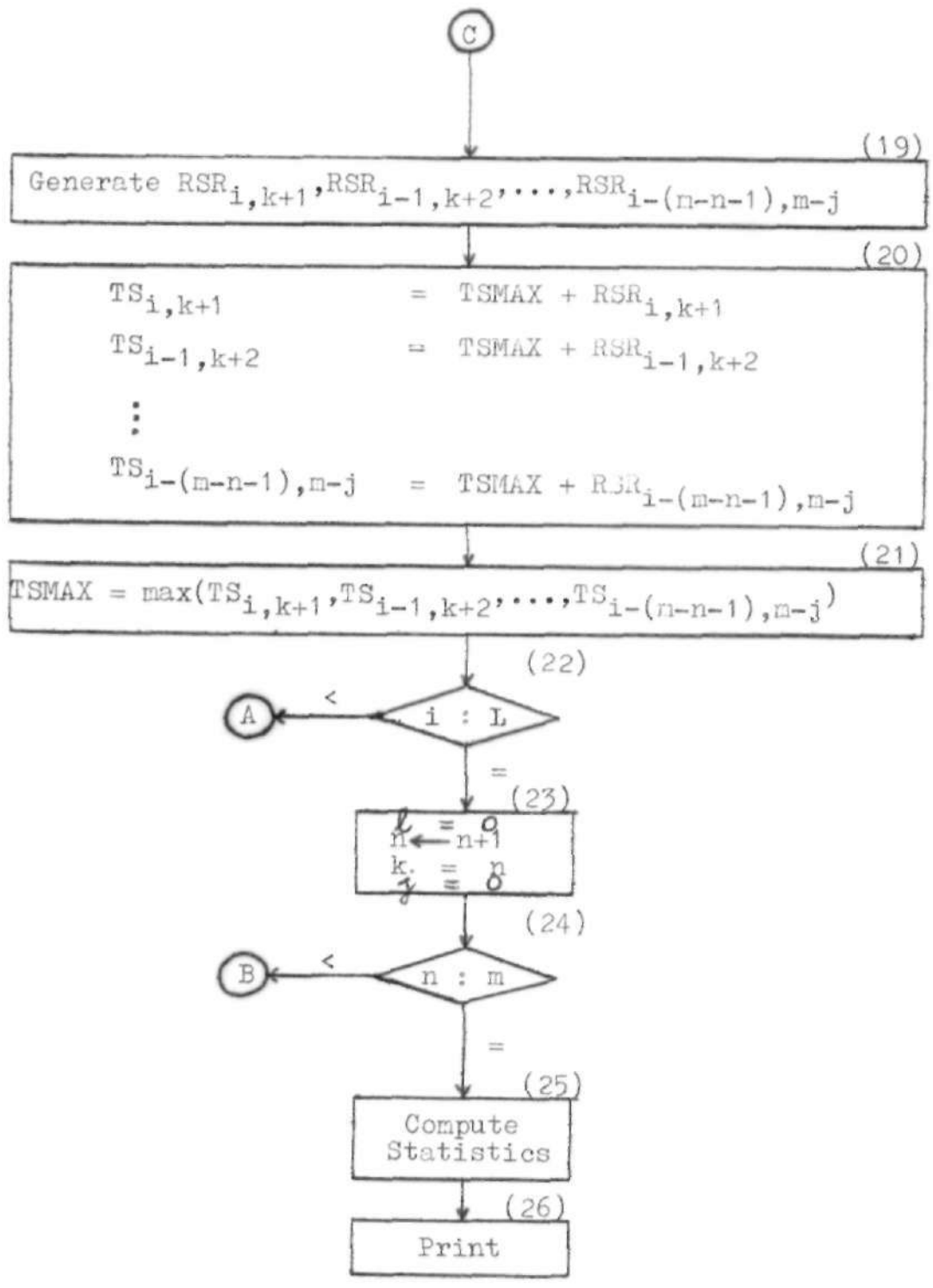
We can decide by block 11 whether the simulation has been in the starting period ($i < m$) or not. Depending on this decision adequate values are assigned to n and k in accordance with Table (2.5-1). At the same time j also takes up a value having importance later, since j is an index to determine the next series of service times and ending times of services in blocks 19 and 20. In the "starting" period service times have to be generated only for the first i stations ($i < m$) which can be realized by taking $j = m - i$. In the "operating" period $j = 0$, as all stations are operating. During the "finishing" period also $j = 0$, and another index k decides on what stations have not yet finished all of the services and for what stations new service times must be generated (block 19).

Beginning from label B a loop has been made (blocks 14-18) to calculate the current waiting and idle times using formulae (2.5-5) and (2.5-6). This loop is repeated until the current waiting and idle times are calculated at each station. During the whole simulation WT_{ih} and IDT_{ih} are computed for all possible combinations of i and h ($i = 1,\ldots,L_p$ and $h = 1,\ldots,m$) except WT_{11} and IDT_{11}. Both of WT_{11} and IDT_{11} are equal to 0, and these assignments are given in block 1 among the starting conditions. In the calculation of $IDT_{12},\ldots,IDT_{1m}$ by formula (2.5-6) in block 16 quantities $TS_{02},\ldots,TS_{0m}$ appear as well, but $TS_{02} = 0,\ldots,TS_{0m} = 0$ as mentioned above. These values are also assigned in block 1.

The maximum of the recently generated TS_{ih} is determined and denoted by TSMAX in block 21.

According to the decision of block 22 we either return to label A (in the "starting" and "operating" periods) or start to carry out the "finishing" period if all the units have entered the system. In the later case we begin to count the number of stations (n) where each service has finished or ends before the present TSMAX time, then we assign the value of n to k (block 23).


(1)
Set starting conditions
(2)
i = 1
(3)
Generate RSR11
(4)
TS11 = RSR11
(5)
TSMAX = TS11
A
(6)
i ← i+1
(7)
Generate RARi
(8)
TSi,0=TSi-1,C+RARi
(9)
TSi,0 :TSMAX
≤
>
(10)
TSMAX = TSi,0
(11)
i : n
≤
>
(12)
n = m-i
k = 0
j = n
(13)
n = 0
k = 0
j = 0
(14)
l = 0
B
(15)
WTi-ℓ,k+l+1 = TSMAX - TSi-ℓ,k+ℓ
(16)
IDTi-ℓ,k+l+1 = TSMAX - TSi-(l+1),k+l+1
(17)
l ← l+1
(18)
l : m-n
<
=
B
C


C
(19)
Generate $RSR_{i,k+1}, RSR_{i-1,k+2}, \ldots, RSR_{i-(n-n-1),m-j}$
(20)
$TS_{i,k+1} = TSMAX + RSR_{i,k+1}$
$TS_{i-1,k+2} = TSMAX + RSR_{i-1,k+2}$
$TS_{i-(m-n-1),m-j} = TSMAX + RSR_{i-(m-n-1),m-j}$
(21)
$TSMAX = \max(TS_{i,k+1}, TS_{i-1,k+2}, \ldots, TS_{i-(n-n-1),m-j})$
(22)
A
<
i : L
=
(23)
n ← n+1
k = n
j = 0
(24)
B
<
n : m
=
(25)
Compute Statistics
(26)
Print

Until the last service at station m starts (n < m) control is transferred to label B to repeat the process described above.The simulation ends at time $TS_{L_p,m}$ when the last unit has serviced at station m.

Finally,required statistics can be calculated and results printed.

PART 3

POSTSCRIPT

3.1 Computer Simulation

Increasing speed and decreasing cost of electronic computers - as well as the development of programming languages particularly suitable for simulations - have resulted in a dramatic increase in the number of computer simulations in recent years.Computer simulation has become popular as a tool for analyzing the behavior of complex systems.The growth in simulation activity is reflected in the literature of fields such as operations research,statistics,computer science,engineering,economics,business administration etc.Simulation extends from specific activities (for example waiting lines,inventory control,production scheduling etc.) to simulations of entire corporations and even the economy of a country.As a tool for testing the behavior of particular deterministic or stochastic systems,simulation has achieved a noteworthy record in only a short period of time.

3.2 Definition

As regards simulation,it can be defined "...as a numerical technique for conducting experiments with certain types of mathematical models which describe the behavior of a complex system on a digital computer over extended periods of time."[48,p.2]

The fact that simulation is a numerical technique implies that it is a technique of "last resort" to be used when analytical techniques are not available for obtaining solutions to a given model.Being a technique of last resort by no means implies that simulation will be of limited usefulness in the sciences,since a great number of problems give rise to mathematical models for which standard analytical techniques don't exist for finding solutions.For many problems in different fields of the sciences,simulation may be the only technique that is available to the analyst.

In some cases,even if one suspects that an analytical solution may exist to a particular model,it may be less costly in terms of the time of the analyst and of the computer time to run a simulation.

A computer simulation is an experiment.This means that it is possible to perform controlled,laboratory-like experiments only by using mathematical model programmed into a computer.The only difference between a simulation experiment and a "real world" experiment is that,with simulation,the experiment is conducted with a model of the real system rather than with the real system.

Although a computer is not a necessary tool for carrying out a simulation experiment with a mathematical model,it certainly speeds up the process by eliminating computational drudgery and reducing the

probability of error.For these reasons the thesis has concentrated only on computer simulation experiments.

With computer simulation one can conduct experiments with a model at a particular point in time.This is achieved by replicating a given simulation run,that is,by changing one or more of the conditions under which the simulation is being conducted.Experiments can also be conducted over extended periods of time.Such computer simulation results when we simply extend the length of a given simulation run over the indicated time without changing any of the conditions under which the simulation is being run.These two concepts raise several difficult methodological problems.Two of them are:(1) how many times must a particular simulation experiment be replicated in order to achieve a given level of statistical precision? (2) how long must one run a simulation so that any statistical inferences made about the behavior of the system will not be influenced by the initial conditions of the system? Naylor [48 and 49],Chorafas [8], Gordon [21],Mize and Cox [45],Maisel and Ghugnoli [39],Venikov [74],Abraham and Tiller [1] and others discuss these problems in details.

Most simulation experiments imply random variables and are called stochastic simulations over which decision makers can exercise little or no control.By including stochastic variables in the model,a simulation experiment can be used to make inferences about the overall behavior of the system in question,that is based on the probability distribution of these stochastic variables.

3.3 Model

The starting point of any computer simulation experiment is the making of a model of the system to be simulated.A conceptual framework that has been developed by Hunter [29] is given here.

Let U denote some endogenous[1] variable of the system that we wish to study,and let $\underline{V}$ denote the k variables (exogenous[2] and policy variables) that are thought to influence U according to the functional relationship

$$U = \Omega(\underline{V}) \qquad (3.-1)$$

A special case of (3.-1) is the simple linear model

$$U = \sum_{i=1}^{k} \alpha_i V_i \qquad (3.-2)$$

---

[1] "Endogenous variables are the dependent or output variables of the system and are generated from the interaction of the system's exogenous and status variable according to the system's operating characteristics."

[2] "Exogenous variables are the independent or input variables of the model and are assumed to have been predetermined and given independently of the system being modeled."[49,p.10-11.]

where the α_i's are parameters.If experimentation were possible,one could vary the $\underline{V}$,observe U,determine the estimates of the parameters $\underline{\alpha}$,and theninterpret the fitted model

$$\hat{U} = \sum_{i=1}^{k} \hat{\alpha}_i V_i \qquad (3.-3)$$

where $\hat{\alpha}_i$ and $\hat{U}$ denote estimates of α_i and U,respectively.

Sometimes it is impossible or impractical to perform controlled experiments with some models.However,it may be possible to perform a type of quasi experiment with a model of the system through the use of computer simulation techniques.In our case,U could be simulated by varying either,or both,the $\underline{\alpha}$ and $\underline{V}$,if experimentation were impossible.

Model (3.-3) is too simple and it could probably be solved by direct analytical techniques instead of using simulation.But by adding a random variable ρ,the model becomes more realistic

$$U = \sum_{i=1}^{k} \alpha_i V_i + \rho$$

where the probability density function of ρ is given by $f(\rho,\underline{\pi})$ and $\underline{\pi}$ represents the parameters of the distribution.

Further complexity can be achieved by introducing the transformations $g(U)$ and $h(V_i)$.Some of these transformations might involve nonlinearities as well as additional parameters.Additional random variables θ_j,each entering with its own weight λ_j and each with its own distribution and parameters $\Phi(\theta_j,\underline{\pi}_j)$,also could be introduced into the model.A time dependence,denoted by the subscript t,could also be applied.In general,the model would then be specified as follows:

$$g(U_t) = \sum_{i=1}^{k} \alpha_i h(V_{it}) + \sum_{j=1}^{m} \lambda_j \Phi(\theta_{jt},\underline{\pi}_j) + f(\rho,\underline{\pi}) \qquad (3.-4)$$

Dummy variables,consisting of ones and zeros,could be used to represent the presence or absence of certain variables at certain times, and to identify blocks of variables that are used together.This model can also be developed by imposing costraints on the variables and parameters,by modifying the additivity aspects or by introducing other variables.Obviously,at this point we would have a model that cannot be solved by analytical techniques.So we would have to resort to simulation as a mode of analysis.

3.4 Flow Chart.Languages

The model concept is translated into a flow-chart and we program and check out the model for the appropriate computer.

Flow-charting is a software technique which when properly applied is an indispensable aid in the logical development of a model.A flow-chart is an orderly representation of a process,that is,a graphic

illustration in which activities are defined and their interrelationship illustrated.There are several conventions used for flow-charts,which describe the successive steps of calculation,but we have adopted a system involving minimal conventions which is general and not oriented towards any particular computer,

A flow-chart can be considered to be a framework for writing a program on any suitable high-level or appropriate simulation language.A number of simulation languages has been developed that are aimed at simplifying the task of writting simulation programs for a variety of different types of models and systems.Some of these languages are GPSS,SIMON, SIMSCRIPT,SIMULA,MONTECODE,CSL,ESP, GSP, GASP,DYNAMO,OPS,SPL,AS, SMOOTH (1973), GSL (1973),PDEL and its extended version PDEL-ID (1973), RSSL (1974) etc.(several of them were made for simulation of computer systems).

"The simulation languages that are available today differ considerably in the extent to which they can render the simulation procedures more or less automatic... Reductions in programming time achieved through the use of simulation languages are usually associated with increases in computer running time and computer costs.In the final analysis the decision whether to use a particular simulation language rests primarily on economic considerations..."[49,p.240].

Since some of the above conditions were not available,the general - purpose language,FORTRAN IV,has been used.Obviously,the programming language itself doesn't influence or determine the results in any stochastic process executed by computer simulation.

3.5 Advantages

There is a great number of reasons for using the simulation approach. "The major advantage of a simulation is that it permits study of the real system without actual modification of that system in any way..." Computer simulation "permits controlled experimentation with

(1) consideration of many factors,
(2) manipulation of many individual units,
(3) ability to consider alternative policies;and
(4) little or no disturbance of the actual system...

... effective training tool,provides operational insight,may dispel operational myths,may make middle management more effective"[39,p.5]. Detailed discussion about the advantages (and disadvantages) of simulation can be found in the literature e.g. [49,p.4-9], [48,p.6-10], [1,p.1-22],[41,p.10-11] ,[39,p.4-6],[74,p.1-8,18-22,28-42],[73,p.1-5] etc.

* * *

Queueing theory is able to make far-reaching investigation in many fields of science and economy,and has become common practice to describe complex,"inter-connected" systems.However,the theory is not capable of offering a solution to many problems and simulation is frequently used to surmount emerging difficulties.There are plenty of queueing models that can be evaluated by simulation.Various techniques for simulating waiting lines of varying complexity exist and the extension to more complex systems can be envisaged.

Numerous articles associated with computer simulation of waiting lines can be found in such journals as Simulation,Operations Research, Management Science etc.Additional papers on simulation are common at national meetings of the relevant professional socioties.A substantial number of books have also been published dealing with this topic. Only a limited number of queueing models has been simulated in this dissertation.Since none of the currently available numerical techniques is suitable for evaluating all queueing models,in many cases,simulation techniques play increasing role.The application of these techniques may be the objective of future research.

REFERENCES

1.Abraham,F.F.,Tiller,V.A.:An Introduction to Computer Simulation in Applied Science,Plenum Press,New York,1972

2.Ashour,S.,Jha,R.D.:Numerical Transient-state Solutions of Queueing Systems, SIMULATION Vol.21,No.4.Oct,1973

3.Bailey,T.J.:The Elements of Stochastic Processes.With Application to the Natural Sciences, J.Wiley,1964

4.Black,J.R.,Even,J.C.Jr.:Computer Solutions of Queueing Models, SIMULATION,Vol.21,No.4,Oct,1973

5.Bhat,U.N.:Elements of Applied Stochastic Processes, J.Wiley,1972

6.Blake,K.,Gordon,G.:Systems Simulation with Digital Computers, IBM Systems Journal,Vol.3.No.1.,1964

7.Buxton,J.N.:Simulation Programming Languages, North-Holland Publishing Company,Amsterdam,1968

8.Chorafas,D.N.:Systems and Simulation, Academic Press,New York,1965

9.Churchman,C.W.,Ackoff,R.L.,Arnoff,E.L.:Introduction to Operations Research, J.Wiley,1957

10.Clarke,A.B.,Disney,R.L.:Probability and Random Processes for Engineers and Scientists,J.Wiley,1970

11.Chu,K.,Naylor,T.H.:Two Alternative Methods for Simulating Waiting Line Models, Journ.of Industrial Engineering, Nov-Dec,1965

12.Coveyou,R.R.:Serial Correlation in the Generation of Pseudo-random Numbers,Journ.of Assoc.for Comp.Mach. 7.1960

13.Davies,N.R.:Decision Tables in Discrete-system Simulation, SIMULATIO,Vol.22 No.2,Febr,1974

14.Dieter,U.:Pseudo-random Numbers:The Exact Distribution of Pairs, Mathematics of Computation,Vol.25.No.116,Oct,1971

15.Dutton,J.M.,Starbuck,W.H.:Computer Simulation of Human Behavior,J.Wiley,1971

16.Dwass,M.:Probability and Statistics, W.A.Benjamin,Inc.New-York,1970

17.Feller,Wm.:An Introduction to Probability Theory and its Applications, J.Wiley,1957

18.Good,I.J.:The Serial Test for Sampling Numbers and Other Tests for Randomness,Proc.Camb.Phil.Soc.,49:2,1953,276-284

19.Good,I.J.:On the Serial Test for Random Sequences, Annals.Maths.Stat. 28,1957, 262-264

20.Golden,D.G.,Schoeffler,J.D.:Problems in the Implementation of a Combined Continuous-Discrete Simulation Language, SIMULATION,Vol.20,No.2,Febr,1973

21.Gordon,G.:System Simulation, Prentice-Hall,Englewood Cliffs,1969

22.Greenberger,M.:Notes on a New Pseudo-Random Number Generator, J.Assoc.Comp.Mach.,8,1961,p.163-167.

23.Gruenberger,Jaffray,G.:Problems for Computer Solution, J.Wiley,1965.

24.Hammersley,J.M.,Handscomb,D.C.:Monte Carlo Methods, J.Wiley,1964.

25.Hilborn,R.:A Control System for FORTRAN Simulation Programming, SIMULATION,Vol.20.No.5,May,1973.

26.Hoffmann de Visme,G.F.A.:A Program Using Random Numbers of Arbitrary Probability Distribution to Simulate the Flow of Patients Through a Hospital Casualty Ward, SIMULATION,Vol.21.No.1,July 1973.

27.Hull,T.E.,Dobell,A.R.:Random Number Generators, SIAM Review 4,1962,p.230-254.

28.Hull,T.E.,Dobel,A.R.:Mixed Random Number Generators for Binary Machines, J.Association for Comp.Mach. Vol.11,1,1964,p.31-40.

29.Hunter,J.S.,Naylor,T.H.:Experimental Design for Computer Simulation Experiments, Management Science,XVI, March,1970.

30.Hutchison,D.W.:A New Uniform Pseudo-Random Number Generator, Communications of ACM,9:6,June,1966,p.432-433.

31.Jackson,J.R.:Simulation of Queues with Dynamic Priorities, Management Sciences Research Project,Research Report No.71,Univ.of California,Los Angeles,1961.

32.Juncosa,M.L.:Random Number Generation on the BRL High-Speed Computing Machines,Report No.855,Ballistic Lab.Aberdeen Proving Ground,Maryland,1953.

33.Kaufmann,A.:Methods and Models of Operations Research, Englewood Cliffs,N.J.,Prentice-Hall,1963.

34.Kendall,D.G.,Harding,E.F.:Stochastic Analysis, J.Wiley,1973

35.Koldenhof,E.A.:Basics on Computer Simulation, IAG - Journal,Vol.4,No.3.1971.

36.Lee,A.M.:Applied Queueing Theory, MacMillan Co.of Canada Ltd.,St Martin's Press Inc.,1968.

37.Lehmer,D.H.:Mathematical Methods in Large-Scale Computing Units, Annals Computer Laboratory,Harward Univ., XXVI,1951.

38.MacLaren,M.D.,Marsaglia,G.:Uniform Random Number Generators, Journal of the ACM,XII.No.1,1965.

39.Maisel,H.,Gnugnoli,G.: Simulation of Discrete Stochastic Systems, Chicago:Sci.Res.Assoc.,1972.

40.Manetsh,T.J.:A Generalized Simulation Approach to Agricultural Sector Analysis with Special Reference to Nigeria, Michigan State University,East Lansing,

Michigan,Nov,1971.
41.Martin,F.F.:Computer Modelling and Simulation,
J.Wiley,1968.
42.McCracken,D.D.:A Guide to FORTRAN IV Programming,
J.Wiley,1965.
43.McLeod,J.:Simulation:From Art to Science for Society,
SIMULATION,Vol.21.No.6,Dec,1973.
44.McLeod,J.:Simulation Today - from Fuzz to Fact,
SIMULATION,Vol.20.,No.3,March,1973
45.Mize,J.H.,Cox,J.G.:Essentials of Simulation,
Prentice-Hall International,Inc.,1968.
46.Morse,P.M.:Queues,Inventories and Maintenance,
J.Wiley,1958.
47.Muller,Mervin,E.: A Comparison of Methods for Generating
Normal Deviates on Digital Computers,
Journal of the ACM,VI,1959,p.376-383.
48.Naylor,T.H.:Computer Simulation Experiments with Models
of Economic Systems,J.Wiley,1971.
49.Naylor,T.H.,Balintfy,J.L.,Burdick,D.S.,Chu,K.:Computer
Simulation Techniques,J.Wiley,1966.
50.Neumann,J.:Various Techniques in Connection with Random
Digits,Monte Carlo Method ,National Bureau of Stand-
ards,Applied Mathematics Series 12,1951.
51.Newell,G.F.:Applications of Queueing Theory,
Chapman & Hall,1971.
52.Ore,O.:Number Theory and Its History,
McGraw-Hill Book Co.,1948.
53.Panico,J.A.:Queueing Theory,
Prentice-Hall,Inc.,1969.
54.Papoulis :Probability,Random Variables and Stochastic
Processes,McGraw-Hill
55.Phipps,T.E.,Jr.:Machine Repair as a Waiting-Line Problem,
Operations Research,IV,1956,p.76-85.
56.Polge,R.J.,Holliday,E.H.,Bhagavan,B.K.:Generation of a
Pseudo-Random Set with Desired Correlation and
Probability Distribution,
SIMULATION,Vol.20.No.5.May,1973.
57.Prekopa,A.:Valoszinusegelmelet,
Muszaki Konyvkiado,Budapest,1962.
58.Pritsker,A.A.B.:Application of Multichannel Queueing Results
to the Analysis of Conveyor Systems,Journal of
Industrial Engineering,17:1,Jan,1966.
59.Ramberg,J.S.,Schmeiser,B.W.:An Approximate Method for
Generating Asymetric Random Variables,Communications
of ACM,Vol.17.No.2.February,1974.
60.Reitman,J.:Computer Simulation Applications,
J.Wiley,1971.

61.Rossmiller,G.E.,Caroll,T.W.,Kim,S.G.,Kim,Y.S.,Manetsch,T.J.,
Kim,D.H.,Johnson,G.L.:Korean Agricultural Sector Analysis
and Recommended Development Strategies 1971-1985,
Michigan State Univ.,East Lansing,Michigan,1972.
62.Saaty,T.L.:Mathematical Methods of Operations Research,
McGraw-Hill Book Co.,Inc.,1959.
63.Saaty,T.:Elements of Queueing Theory,
McGraw-Hill Book Co.,Inc.,1961.
64.Sasieni,M.Yaspan,A.Friedman,L.:Operations Research Methods
and Problems,J.Wiley,1959.
65.Sohnle,R.C.,Tartar,J.,Sampson,J.R.:Requirements for Inter-
active Simulation System,
SIMULATION,Vol.20.No.5.,May,1973.
66.Srejgyer,Ju.A.:Monte Carlo Modszerek,
Muszaki Konyvkiado,Budapest,1965.
67.Stockmal,F.:Calculations with Pseudo-Random Numbers,
ACM Journal,Vol.11,1,1964.
68.Taha,H.A.:Operations Research,
MacMillan Co.New-York,1971.
69.Teichroew,D.:Distribution Sampling with High Speed Computers
Ph.D.Thesis,University of North Carolina,1953.
70.Teichroew,D.,Lubin,J.F.:Computer Simulation Discussion of
the Technique and Comparison of Languageg,
Communications of the ACM,9:10,Oct,1966.
71.Tocher,K.D.:The Application of Automatic Computers to Sampl-
ing Experiments, Journal of the Royal Statistical
Society, B 16,1954.
72.Tocher,K.D.:Review of Simulation Languages,
Operational Research Quarterly,XVI,June,1965.
73.Tocher,K.D.:The Art of Simulation,
D.Van Nostrand Co.Inc.Princeton,N.J.,1967.
74.Venikov,V.A.:Theory of Similarity and Simulation,
MacDonald and Scientific,London,1969.
75.Wagner,H.M.:Principles of Operations Research,
Prentice-Hall International,Inc.,1969.