LONDON SCHOOL OF HYGIENE & TROPICAL MEDICINE

Medical Statistics Department

SIMULATION IN THE HEALTH SERVICES WITH AN APPLICATION IN HOSPITAL WAITING LISTS

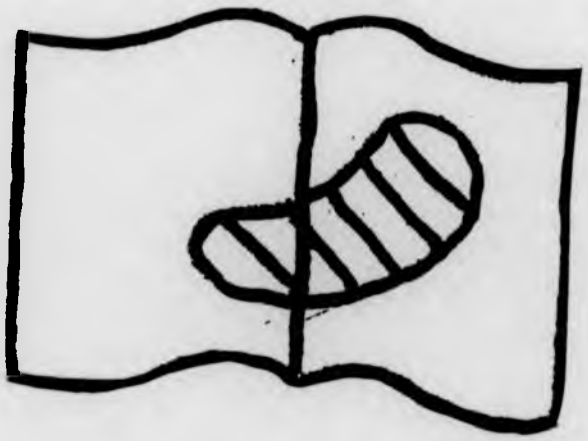
Thesis submitted to the University of London

for the degree of Doctor of Philosophy by

Erick Leonardo Suárez Pérez

May, 1986

Very Poor Quality print-



ABSTRACT

A description of simulation methodology has been presented in this thesis with emphasis on the generation of random variables, pseudo-random number generators and variance reduction techniques. Also shown is different applications of simulation within the health care services.

As an application of health care simulation, we simulated a hospital waiting list model, seen as a queueing process with the arrivals rate equal to the admission rate (ρ=1), to show the statistical effect of using a scoring system in the policies of admission. To study the effects in the waiting time the queuening theory concepts could not be applied when ρ=1, therefore simulation was used as an alternative procedure. From our simulation results, we could find an expression to the expected waiting time for any number of medical priorities defined in the waiting list, and we could develope a procedure to control the standard deviation of the waiting time. In general, the scoring sytem used in our simulation model was very useful to balance the waiting time distribution of those patients selected from a hospital waiting list which shown different levels of urgency for being hospitalized.

TO THE MEMORY OF MY FATHER

TO LIZETTE FOR HER LOVE AND

GREAT SUPPORT

CONTENTS

FIGURES

TABLES

GRAPHS

I) Simulation Process

1.1) Introduction

Planning resources in the health services is becoming an increasing complex task. New discoveries in the health care have produced changes in management procedures. These procedures have to be evaluated before their implementation. Simulation techniques have been used to study the effects of these procedures, especially when experiments in the real world are impossible or impractical to realize.

In this paper, we will define simulation as experimenting with a model (representation of a system) on a digital computer over time; if the increment of time is constant the simulation is called 'fixed-time incrementing' simulation and if time is variable, it is called 'next-event' simulation. The experimentation in simulation involves sampling from probability distributions in order to assign values to stochastic variables. This definition of simulation necessarily excludes such other "simulations" as:

a) simulations with a physical model such as a scale model or laboratory experiments with real people;
b) man-machine simulations which combine abstract models with the use of real people (e.g., gaming);
c) deterministic simulations, i.e., without the use of probability distributions or random numbers; these include econometric-type regression models;

d) so called Monte Carlo studies, which employ random numbers, but lack the time element of simulation.

The above definition of simulation stresses its experimental aspect, although it is clear that none of this experimenting can take place without the prerequisite system investigation (real or hypothetical system), model building, selection of an appropriate computer language, the generation of random or pseudo-random numbers, programming, program verification, and model validation. All these tasks must be completed first to have a model (i.e., the computer program) with which to experiment.

Naturally, simulation experiments are designed with certain objectives in mind. Some of the typical goals of simulation are: forecasting, estimation, comparing alternative competing systems, optimization, sensitivity analysis and answering "what-if" questions (i.e., how many hospital beds would be needed if the average length of stay in hospital were reduced?). Once the objectives have been specified, we can continue with the design and analysis of the simulation experiment as with any statistical analysis.

In this paper, we will describe some of the most important techniques used in simulation such as: generation of random variables (chapter II), pseudo-random number generators (chapter III) and variance reduction techniques (chapter IV). Also presented is a review of the recent publications of simulation in the health services (chapter V). In chapter VI, we give an introduction to Queueing Theory and its limitations where simulation is used

as an alternative.

A typical queueing problem within the health services is the long waiting time that patients have to wait before receiving medical care. This waiting time could produce serious deterioration in the patient health. This delay usually depends on the medical priority (degree of urgency to hospitalize any patient) and the actual waiting time of the patients. In chapter VII, we analyze the effects in the waiting time distribution when the policies of admission from a hospital waiting list are based on a system which combines in a single score the medical priority and the actual waiting time.

To conclude this chapter, we define some of the concepts needed in simulation such as: system, simulation model, simulation computer languages, verification & validation, outputs analysis and implementation & documentation.

1.2) System

A system is defined as a set of related entities called components or elements. For instance, a hospital can be considered as a system, with doctors, nurses, and patients as its components. The components have certain characteristics, called attributes, that have logical or numerical values, i.e., number of beds, number of surgical theatres, and so on. The attributes of the system elements define its state. For example, the number of patients in a hospital waiting list describe the system state.

The objective in studying one or several phenomena in terms of a system is to learn how change in state occurs, to predict change and control change. Most studies combine these objectives with varying emphasis. One particular combination of these objectives, called 'evaluation of alternatives', concerns the relationship between the input and output from a system, as depicted in the following figure:

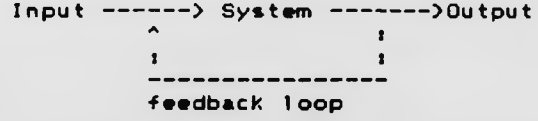


Figure 1.1 Graphical representation of a system

Input refers to an external stimulus to the system that induces changes in the system state. Output refers to measures of these state changes.

An example of evaluation of alternatives in the health services is as follows: Patients (inputs) are currently processed in the order of their arrival to a outpatients clinic (system), which has three physicians. The clinic plans to institute a new processing method that classifies a patient in group A if his expected consultation time is short, and in group B if the expected time is long. One physician will be assigned to process patients in group A on the first-come-first-served basis. The remaining two physicians will do likewise for patients in group B. The principal concern is the extent to which the new selection procedure will lead to a shorter mean waiting time for group A patients (output).

Systems can be classified in a variety of ways. These are natural or artificial, adaptive or non-adaptive and discrete or continuous. In a natural system there is no man intervention for the functioning of the system, while in the artificial one there is. An adaptive system reacts to changes in the environment, while in the non-adaptive there is no such reaction. A discrete system is one for which the state of the system changes only at a countable (or finite) number of points in time, on the other hand in a continuous system the state changes continuously with respect to the time.

1.3) Simulation Model

A model is a representation of a system. Models may be scaled physical objects (iconic models), mathematical equations and relationships (abstract models) and graphical representations (visual models). The usefulness of models has been demonstrated in describing, designing and analysing a system.

In this paper, we restrict our attention to the models which are called simulation models. A simulation model is defined as an abstract representation of a system which can be exercised in an experimental fashion on a digital computer over time. Simulation models are considered as a laboratory version of a system [28].

Often the boundaries of the system and of the model are rather arbitrarily defined. Most forces that impinge on the system must be neglected on a priori grounds to keep the model tractable, even when there is no rigorous proof that such neglect is justified. Inevitably, the model is better defined than the real system (see figure 1.2).

The building of simulation models is a complex process and in most cases is an art and a science. The modeling of a system is made easier if:

i) physical laws that pertain to the system are available;

ii) a pictorial or graphical representation can be made of the system;

iii) the variability of the system inputs, elements, and outputs is manageable.

Once a simulation model is developed, experiments can be performed. These experiments or simulations permit inferences to be drawn about systems

- without building them, if they are only proposed systems;
- without disturbing them, if they are operating systems that are costly or unsafe to experiment with;
- without destroying them, if the object of an experiment is to determine their limits of stress.

In this way, simulation models are used for design, procedural analysis and performance assessment in the area of the health services, as we will see in chapter V.

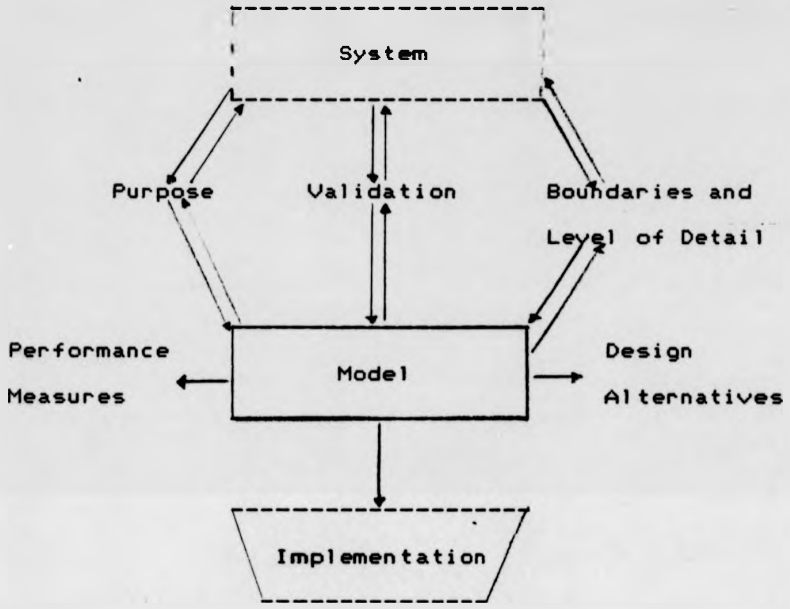


Figure 1.2 A model building approach for system analysis

1.4) Computer Simulation Languages

Simulation modeling assumes that we can describe a system in terms acceptable to a computer language. There have been such a great variety of these languages developed for simulation studies over the years that is nearly impossible to decide which language best fits any particular application. There were over 170 alone in 1972 and new ones are being developed every day [30].

The computer simulation languages can be classified as:

- General Purpose Simulation Languages
- General Purpose Languages

General purpose simulation languages are intended to permit the system analyst to concentrate his effort in the modelling of the system and to greatly simplify the process of computer programming of the model. Some of these languages are GPSS [11], SIMSCRIPT [36] and SIMULA [1].

SIMSCRIPT is considered more flexible for programming, but it assumes a prior knowledge of the computer language FORTRAN. GPSS can be learned very quickly with no prior knowledge of programming, but is less flexible and somewhat slower in execution than SIMSCRIPT ([22], p. 168). SIMULA is an extension of the programming language ALGOL 60.

General purpose languages such as FORTRAN [21], BASIC [20] and PASCAL [35] are used for simulation studies and for other types of studies. These languages are more flexible and can produce programs that are more efficient than the general purpose simulation languages.

There are also computer packages available for simulation studies such as CAPS [5], SMTBPC [18] and NAG [25]. CAPS is designed to run interactively and it conversationally prompts the user in a way that allows him to define a simulation model. SMTBPC contains the basic computational and statistical tools necessary to run a simulation experiment on an IBM PC-DOS. NAG is a set of Fortran subroutines that can be used for different mathematical purposes; for simulation, there are a good number of subroutines to generate random variables.

1.5) Verification and Validation

Two stages are needed to check that the simulation model represents the real system. These stages are:

- Verification.- Checking that the simulation program operates in the way that the model implementer thinks it does; that is, is the program free of errors and consistent with the simulation model?
- Validation.- Checking that the simulation model, correctly implemented, is a sufficiently close approximation to reality for the intended application.

The verification can be realized in different forms, some of these are:

- Manual verification of logic. Run the model for a short period by machine and by hand; then compare the results.
- Checking against known solutions. Adjust the model so that it represents a system with a known solution, and compare this with the model results.
- Sensitivity testing. Vary just one parameter while keeping all the others fixed, and check that the behaviour of the model is sensible.

When we create a model some problems in the validation may arise, because of the different interpretations that can be given to a specific system. It

is always necessary to restrict the boundaries of the model, ignoring everything outside that is not an explicit input, and neglect factors believed to be unimportant. Other types of approximations are:

- Distributional. Real-world probability distributions are frequently approximated by simple distributions, such as the normal or exponential.

- Independence. The model is frequently simplified if various componets (random variables) are assumed to be statistically independent.

- Stationarity. It simplifies matters to assume that parameters and other features of the system do not vary over time. This may be reasonable if it can be legitimately argued that any changes over the relatively limited period of interest are negligible.

When a simulation program produces nonsense, it is not always clear whether this is due to errors in the conceptual model, programming errors, or even the use of faulty data. A failed attempt at validation usually requests modification to the model, which implies changes in the computer program. This means that model construction, verification and validation often are in a dynamic feedback loop.

1.6) Outputs Analysis

Once the simulation model has been validated, the computer program will presumably be used to study the behaviour of the system which has been modelled, try out alternative system specifications and design, and aid in making recommendations and decisions. This section focuses on how outputs in a steady-state simulation should be analyzed to enable the drawing of valid, accurate and precise conclusions.

A steady-state simulation is one for which the measure of performance, θ, is defined as follows:

$$\lim_{n\to\infty} h(Y_1,..,Y_n) = \theta$$

where h is a function of the simulation outputs, Y_i's, i.e., the mean, the variance,... The quantity θ must be independent of the initial conditions of the simulation model. So, theoretically, one could initialize the simulation in any convenient way.

In a steady-state simulation two main questions have to be considered: how long the simulation should be run and how this run should be started in order to achieve independence between the initial conditions and the quantity θ. Various methods described below take different approaches to coping with these questions.

<u>The initial transient problem.</u> By definition of a limit, we can be sure that if we run the simulation long enough, $h(Y_1,...,Y_n)$ will be close to θ. This does not say

anything about the length of time we must wait for this to happen, and the duration of this "transient" or "warmup" period can depend largely on the way the run was initialized. The problem of this initial transience has been viewed as one of identifying the extent of the transient period, relative to some practical criteria. The usual procedure is then to delete (or truncate) this initial period from consideration, and use only the subsequent output measures, Y_i's [13].

The extent of the transient period is certainly influenced by the initial conditions. Thus, to reduce the amount of initial outputs that must be discarded, it is probably worth giving some thought to choosing initial conditions which at least appear to bear some rough resemblance to the anticipated steady-state conditions.

Replication Method. In this method, one observation, Y_j, is taken after a certain simulation time; this process is repeated several times to produce a sequence of Y_i's. This method has the advantage that the classical statistical analysis with i.i.d observations [23] can be used for the outputs analysis, given that different seeds (starting points in the pseudo-random number generator, chapter III) were used in each simulation run.

This method is the only one requiring more than one simulation run. It is pointed out that replication is the procedure appropriate for the analysis of terminating simulation; simulations which end when a pre-specified event occurs ([18], p. 245)

Regenerative Method. This method is based on a continued simulation run which is then partitioned into a series of consecutive, non-overlapping subsequences. These subsequences (epochs), each containing a varying number of observations and the demarcation point between adjacent epochs, are always in the same state (the regenerative point), e.g. empty-and-idle. This ensures the independence of the epochs ([16], pp. 297-300).

The evolution of the process between successive regeneration points is called a 'cycle', and what happens during a cycle is an i.i.d. replicate of what happens during any other cycle.

The regenerative method of confidence interval estimation in simulation has drawn much scholarly interest, mainly because it is a procedure with the sound underpinning of the renewal theory of stochastic process and thus lends itself well to general mathematical proofs. However, unless the process being simulated has this renewal property, the analyst may become frustrated in his search for regeneration points or suitable approximations [10].

Time Series Methods (i.e., autoregressive moving average and spectral analysis). A time series is a collection of observations (Y_t) made sequentially in time; for example, the mortality in the last ten years, and the number of patients hospitalized at midnight (daily census) during one

year. The objectives in a time series may be the prediction of the future based on knowledge of the past; it may be to obtain an understanding of the mechanism that generates the series; or a succinct description of the salient features of the series may simply be desired [4].

One of the advantages in time series methods is that they take into account the inherent serial autocorrelation in the outputs simulation. The disadvantage is the number of calculations involved in these methods. ([10], p. 247). An extensive description of these methods can be seen in Bratley et al ([2], pp. 81-102), Fishman ([9], pp. 262-309), Kelton [13] and Kleijnen [14].

One of the most important points to be made in this section is that statistical analysis should be an integral part of any simulation study. The lack of proper analysis leaves one with results that may be misleading and inaccurate. If one makes substantial efforts to validate, code and debug a complex simulation model, it seems worthwhile to make some effort to use the model effectively and interpret its outputs appropriately.

1.7) Implementation

The final stages in the simulation development process are the implementation of the simulation model and its use. No simulation project should be complete until its results are used in the decision-making process.

The success of the implementation task is largely dependent upon the degree to which the modeler has successfully performed the other activities in the simulation development process (see figure 1.3). If the model builder and the decision maker have worked closely together and they both understand the model and its outputs, then it is likely that the results of the projects will be implemented [33].

On the other hand, if the model formulation and the underlying assumptions are not effectively communicated, then it is more difficult to implement the recommendation from the simulation results, regardless of the validity of the simulation model.

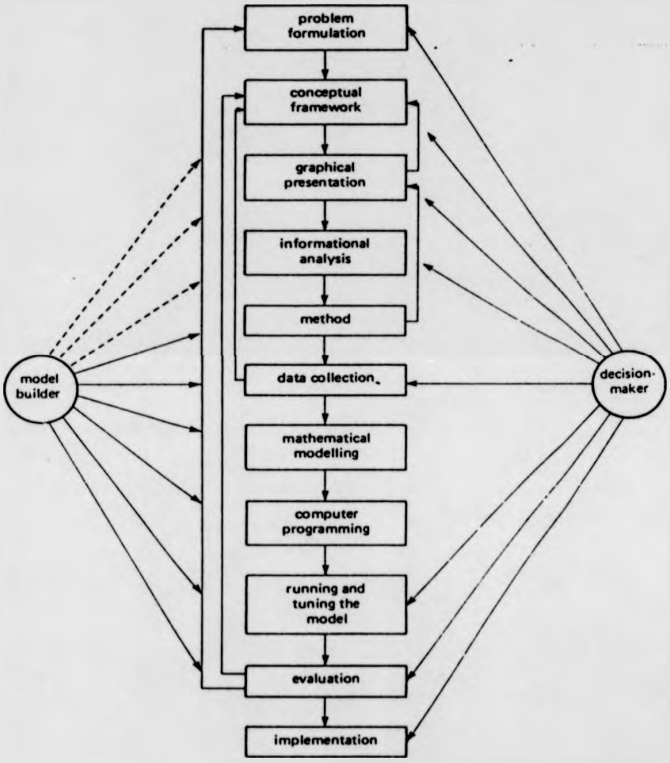


Figure 1.3 The model-building process, showing the mutual participation of the decision-maker and the model builder.

1.8) Historical Background

The earliest use of the concepts of simulation is usually credited to Count de Buffon, a french naturalist, who in 1777 stated what is now known as Buffon's needle problem: a needle of length 'a' is thrown at random on a plane covered with parallel lines that are all a distance 'd' apart ($a \leq d$); the probability that the needle intersects one of the lines is $p = 2a/\pi d$ (see Conolly [6], p. 185).

Buffon is supposed to have performed the experiment to calculate the value of π, by setting the observed frequency equal to the probability 'p'. This remarkable idea, the calculation of a deterministic quantity from the data of a random experiment, became a practical methodology with the advent of digital computers to carry out logical operations with artificially generated data at high speed. This technique was first used during the development of the atomic bomb for the numerical evaluation of integrals, where it was called the Monte Carlo Method [12].

The first application of the concept of simulation for the analysis of systems subject to stochastic demands was done at the outset of this century. In 1907, J.C.T. Baldwin described, in an engineering report for the American Telephone Company, what is probably the first modern simulation study. Baldwin's study used artificial data to simulate telephone traffic and determine the loads that could be handled by operators providing service with given

average delays. Baldwin called this technique the "throw down" method. Although it did not, of course, use a computer, it embodied the essential concepts of modern simulation studies ([7], p. 263).

REFERENCES

1 Birtwistle J., Dahl O., Myhrhaug B. and Nygaard K., Simula Begin. 2nd ed., Chartwell-Bratt Ltd., England, 1980.

2 Bratley P., Fox B. and Schrage L., A Guide to Simulation. Springer-Verlag, USA, 1983.

3 Bulgren W., Discrete System Simulation. Prentice-Hall Inc., USA, 1982.

4 Chatfield C., The Analysis of Time Series: Theory and Practice. Chapman and Hall, London, 1975.

5 Clementson A., Extended Control and Simulation Language: Users Manual, CLE. Com Ltd, Birmingham, G.B., 1982.

6 Conolly B., Techniques in Operational Research. Vol. 2, Ellis Horwood Ltd., G.B., 1981.

7 Cooper R., "Simulation of Queueing Models", Chapter 6, Introduction To Queueing Theory. 2nd ed., Edward Arnold, G.B., 1981, 281-307.

8 Doukidis G. and Paul R., "Research into Expert Systems to Aid Simulation Modeling", Journal of the Operational Research Society. April 1985; 36(4): 319-325.

9 Fishman G., Concepts and Methods in Discrete Event Digital Simulation, Wiley, USA, 1973.

10 Friedman L. and Friedman H., "Statistical Considerations in Computer Simulation: The State of the Art", Journal of Statistical Computation and Simulation. 1984; 19: 237-263.

11 Greenberg S., GPSS Primer, Wiley, USA, 1972.

12 Hammersley J. and Handscombs D., Monte Carlo Methods, Methuen & Co. Ltd., London, 1964.

13 Kelton W., "Simulation Analysis", 1983 Winter Simulation Conference Proceedings, IEEE, 1983, 159-168.

14 Kleijnen J., "Design and Analysis of Simulations: Practical Statistical Techniques", Simulation, March 1977; 28(3): 81-90.

15 Kobayashi H., Modeling and Analysis: An Introduction to System Performance Evaluation Methodology, Addison-Wesley, USA, 1978.

16 Law A. and Kelton D., Simulation, Modelling and Analysis, McGraw-Hill, USA, 1982.

17 Law A. and Kelton D., "Confidence Intervals for Steady-State Simulations: A Survey of Fixed Sample Size Procedures", Operations Research, Nov-Dec. 1984; 32(6): 1221-1239.

18 Lewis P. and Orav E., Notes: Simulation Methodology for Statisticians, Operations Analysts and Engineers, Short Course on Statistical Simulation Methodology, University of Birmingham, 29-30 May, 1985.

19 Lewis T. and Smith B., Computer Principles of Modeling and Simulation, Houghton Mifflin, USA, 1979.

20 Marateck S., Basic, 2nd ed., Academic Press, USA, 1982.

21 Meissner L. and Organick E., FORTRAN 77, Addison-Wesley, USA, 1980.

22 Mize J. and Cox J., Essentials of Simulation, Prentice-Hall, Inc., USA, 1968.

23 Mood A., Graybill F. and Boes D., Introduction to the Theory of Statistics, 3rd ed., McGraw-Hill Kogakusha, 1974.

24 Morgan B., Elements of Simulation, Chapman and Hall Ltd., G.B., 1984.

25 NAG, Fortran Mini Manual, Mark 10, Published and Printed by Numerical Algorithms Group Ltd., U.K.

26 Naylor T., "Computer Simulation Defined", Chapter 1, Computer Simulation Experiments with Models of Economic Systems, Wiley, USA, 1971.

27 Payne J., Introduction to Simulation: Programming Techniques and Methods of Analysis, McGraw-Hill, USA, 1982.

28 Pritsker A. and Pegden C., Introduction To Simulation and SLAM, Wiley, USA, 1979.

29 Rubinstein R., "Systems, Models, Simulation and the Monte Carlo Methods, Chapter 1, Simulation and the Monte Carlo Method, Wiley, USA, 1981.

30 Shannon R., "Introduction to Model Building", 1982 Winter Simulation Conference Proceedings, IEEE, 1982, 633-638.

31 Shannon R., "Simulation: An Overview", 1983 Winter Simulation Conference Proceedings, IEEE, 1983, 19-22.

32 Shigan E. and Gibbs R., Modeling Health Care System, Proceedings of a IIASA Workshop, March 28-29, 1977.

33 Shigan E., "Models of the Health System as a basis for Data Collection, Chapter 11, Information System for Health Services, Regional Office for Europe, World Health Organization, Copenhagen, 1980, 87-94.

34 Tilquin C., "Modeling Health Services Systems", Medical Care, 1976; XIV(3): 223-240.

35 Wilson I. and Addyman A., A Practical Introduction to Pascal, The MacMillan Press Ltd., G.B., 1978.

36 Wyman F., Simulation Modeling : A Guide to Using SIMSCRIPT, Wiley, USA, 1970.

II) Random Variate Generation

2.1) Introduction

A simulation based on random behaviour naturally requires a mechanism for generating a sequence of events where each sequence obeys a probability law governing a particular component of the random behaviour in question. The probability law may take many forms. One commonly encountered in simulation work assumes that events in the sequence are independent and identically distributed; for example, with Normal or Gamma distributions if the events are continuous random variables, and with Binomial or Poisson distributions if the events are discrete random variables. This chapter will describe a variety of methods for generating variables with random behaviour.

There are usually several alternative algorithms that can be used for generating random variables from a given distribution, and a number of factors should be considered when choosing which algorithm to use in a particular simulation. Law & Kelton ([26], p.241) made the following recommendations:

- Exactness. In this case, we should use an algorithm which results in random variables with exactly the desired distribution, within the limitations of machine accuracy. There are some approximate methods to generate random variables, for example, the sum of twelve uniform random numbers subtracted by six, $\Sigma_{i=1}^{12} U_i - 6$, is used to generate a normal (0,1) variable ([39], p. 89). Of course,

this approximation is not recommended because of the amount of uniform numbers needed to generate one sample. Fortunately, for most common distributions found in computer simulation languages (section 1.4) exact algorithms are now available, obviating the need to consider any approximate method.

- Efficiency. Given that we have the choice of alternative exact algorithms, this characteristic refers to the storage space and execution time required to generate a random variable. Some algorithms require storage of a large number of constants or of large tables, which could prove troublesome. As for execution time, there are really two factors. First, we hope that we can accomplish the generation of each random variable in a small amount of time; this is called 'marginal execution time'. Secondly, some algorithms have to do some initial computing to specify constants or tables that depend on the particular distribution; the requirement to do this is called the 'set-up time'. In most simulations, when we generate a large number of samples, the marginal time is likely to be more important than the set-up time.

In this chapter, we are going to survey some of the general procedures used to generate random variables (section 2.2). In addition, in section 2.3, algorithms are described for generating random variables from particular continuous and discretes distributions that have been useful

in simulation. Finally, in section 2.4, we describe a method to generate random variables when the distribution function is unknown.

As we shall see, the basic ingredient needed for every method of generating random variables from any distribution is a source of uniform random variables U(0,1). For this reason it is very important that a statistically reliable U(0,1) random numbers generator be available.

For convenience we refer to sampling from a particular distribution of a type of random variate by the word 'generation'. For example, exponential generation denotes sampling from a exponential distribution.

2.2) General Procedures

We start by describing some general procedures to generate random variables such as: Inverse Transform Method (for continuous and discrete distributions), Acceptance-Rejection Method, Forsythe's Method, Alias Method and the Composition Method.

2.2.1) Inverse Transform Method

2.2.1.1) Continuous Distributions

Let X be a random variable with cumulative probability distribution function (c.d.f.) $F_X(x) = Pr(X \leq x)$. Since $F_X(x)$ is a non-decreasing function, the inverse function $F^-_X(u)$ may be defined for any value of u between 0 and 1 as $F^-_X(u)$ is the smallest x that satisfies $F_X(x) \geq u$, (see figure 2.1), that is,

$$F^-_X(u) = \inf[x: F_X(x) \geq u], \quad 0 \leq u \leq 1$$

It has been shown ([34] & [39]) that if U is uniformly distributed over the interval (0,1), then $X = F^-_X(U)$ has the cumulative distribution function $F_X(x)$.

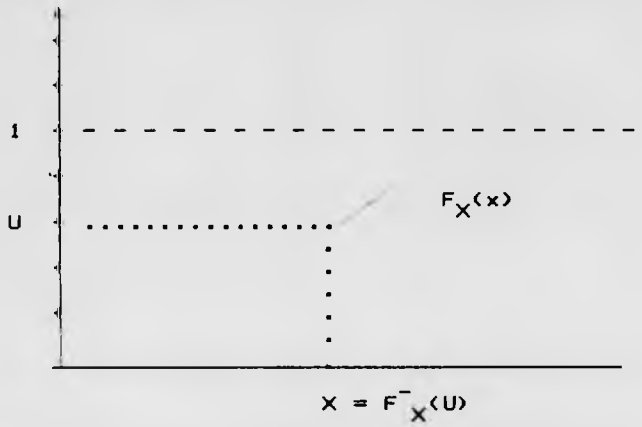


Figure 2.1.- Inverse probability integral transformation method

Therefore, to get a value, say x, of a random variable X, obtain a value, say u, of a variable U uniformly distributed over (0,1), compute $F^-_X(u)$ and set it equal to x. This procedure is presented in the following algorithm:

Algorithm IT-1

1) Generate U from U(0,1)
2) X <- $F^-_X(U)$
3) Deliver X

Example.- Generate a random variable from a uniform distribution U(a,b), that is,

$$f_X(x) = \begin{cases} 1/(b-a) & ; a \leq x \leq b \\ 0 & ; \text{otherwise} \end{cases}$$

and

$$F_X(x) = \begin{cases} 0 & ; x < a \\ (x-a)/(b-a) & ; a \leq x \leq b \\ 1 & ; x > b \end{cases}$$

Hence, applying the algorithm IT-1,

$$X = F^{-}_X(U) = a + (b-a) * U$$

To apply the inverse transform method, $F_X(x)$ must exist in a form for which the corresponding inverse transform can be found analytically. Distributions of this type are Exponential, Uniform, Weibull, Logistic and Cauchy. Unfortunately, for many probability distributions it is either impossible or extremely difficult to find the inverse transform, that is to solve

$$U = \int^{x}_{-\infty} f_X(x)\,dx$$

with respect to x.

Even in the case when $F^{-}_X(x)$ exists in an explicit form, this inverse transform method is not necessarily the most efficient method for generating random variates.

2.2.1.2) Discrete Distributions

Let X be a discrete r.v. with probability mass function (p.m.f.)

$$Pr[X = x_k] = P_k \quad ; \quad k=0,1,....$$

and with c.d.f.

$$q_k = Pr[X \leq k] = \Sigma_{i=0}^{k} P_i$$

then

$$Pr[q_{k-1} < U \leq q_k] = q_k - q_{k-1} = P_k$$

where U is from U(0,1). Thus

$$X = [x_k : q_{k-1} < U \leq q_k]$$

The above procedure is described in the following algorithm:

Algorithm IT-2

1) D <- P_0
2) B <- D
3) K <- 0
4) Generate U from U(0,1)
5) If U $\leq$ B (U$\leq q_k$) , deliver X = x_k
6) K <- K+1
7) D <- A_{k+1}*D ($A_{k+1}=P_{k+1}/P_k$)
8) B <- B + D ($q_{k+1}=q_k+P_{k+1}$)
9) Go to step 5

Table 2.1 represents the values of P_0 and A_{k+1} for some discrete distributions. In order to generate a random variable from a specified discrete distribution, we take the corresponding values of P_0 and A_{k+1} from table 2.1 and then run algorithm IT-2.

Table 2.1 Values of P_0 and A_{k+1} for some discrete distributions

Distributions P_k	P_0	$A_{k+1}=P_{k+1}/P_k$
a) Binomial $\frac{n!p^k(1-p)^{n-k}}{k!(n-k)!}$ $k=0,1,\ldots,n;\ p>0$	$(1-p)^n$	$\frac{(n-k)p}{(k+1)(1-p)}$
b) Poisson $\frac{e^{-\lambda}\lambda^k}{k!}$ $k=0,1,\ldots;\ \lambda>0$	$e^{-\lambda}$	$\lambda/(k+1)$
c) Geometric $p(1-p)^k$ $k=0,1,\ldots;\ p>0$	p	$(1-p)$

2.2.2) Acceptance-Rejection Method

The acceptance-rejection method consists of sampling a random variate from an appropriate distribution and subjecting it to a test to determine whether or not it will be acceptable for use. This method, first suggested by von Neumann [42], can be used when the p.d.f. $f_X(x)$ is known.

Let X to be generated from $f_X(x)$. To carry out the acceptance-rejection method, $f_X(x)$ is represented as

$$f_X(x) = C*h(x)*g(x) \quad \ldots\ldots\ldots (2.1)$$

where $C \geq 1$, h(x) is also a p.d.f. and $0 < g(x) < 1$. Then, we generate two random variates U and Y from U(0,1) and h(x) respectively, and test to see whether or not the inequality $U \leq g(Y)$ holds. If the inequality holds, then accept Y as the variate generated from $f_X(x)$. If the inequality does not hold, reject the pair U,Y and try again.

The theory behind this method is based on the following theorem :

Theorem .- Let X be a random variable distributed with p.d.f. $f_X(x)$, which is represented as

$$f_X(x) = C*g(x)*h(x)$$

where $C \geq 1$, $0 \leq g(x) \leq 1$ and h(x) is also a p.d.f. Let U and Y be

distributed U(0,1) and h(x), respectively. Then

$$f_Y(x \mid U \leq g(Y)) = f_X(x)$$

(Proof, see ref [39] p. 46)

The efficiency of the acceptance-rejection method is determined by the inequality $U \leq g(Y)$. The probability that the inequality holds in each trial is obtained from:

$$Pr[\ U \leq g(Y)\ ] = \int Pr[\ U \leq g(Y \mid Y=x)\ ] * h(x)\,dx$$

where

$$Pr[\ U \leq g(Y \mid Y=x)\ ] = Pr[\ U \leq g(x)\ ] = g(x)$$

so,

$$Pr[\ U \leq g(Y)\ ] = \int g(x)h(x)\,dx$$

$$= \int [f_X(x)/C]\,dx$$

$$= 1/C$$

Then, the probability of success in each trial is $p = 1/C$. Since the trials are independent, the number of trials N, before a successful pair U, Y is found, has a geometric distribution:

$$P_N(n) = p(1-p)^n \quad ; n=0,1,....$$

with

$$E(N) = [\ 1 - 1/C\ ]/(1/C) = C - 1$$

Hence, $C = 1 + E(N)$ is defined as the expected number of pairs U & Y needed to get one sample from $f_X(x)$.

The algorithm for the acceptance-rejection method is as follows:

Algorithm AR-1

1) Generate U from U(0,1)
2) Generate Y from p.d.f. h(x)
3) If U≤g(Y), deliver Y as the variate generated from $f_X(x)$
4) Go to step 1

For this method to be of practical interest the following criteria must be used in selecting h(x):

- It should be easy to generate a random variable from h(x).
- The efficiency of the procedure 1/C should be large, that is, C should be close to 1.

Example.- Generate a random variable from the following distribution:

$$f_X(x) = 3*x^2 \quad ; 0 \leq x \leq 1$$

Let us describe $f_X(x)$ as the expression (2.1), assuming

$$h(x) = 2*x \qquad 0 \leq x \leq 1$$

$$g(x) = x \qquad 0 \leq x \leq 1$$

and

$$C = 3/2$$

To apply algorithm AR-1, we generate the variable Y from h(x); for this purpose, we use the algorithm IT-1. So,

$$\int_0^Y 2x\,dx = Y^2 = U_1 \rightarrow Y = U_1^{1/2} \quad ; \quad U_1 \text{ from } U(0,1)$$

Then, compare

$$U_2 \leq g(Y) = Y = U_1^{1/2} \quad ; \quad U_2 \text{ from } U(0,1)$$

If the condition holds, Y is taken as the variate generated from $f_X(x)$. If not, take another pair (U_1, U_2) and repeat the condition. The probability of success is $1/C = 2/3$ and the expected number of pairs U & Y needed per sample is $C = 3/2$.

2.2.3) Forsythe's Method

Forsythe's method is a rejection technique for sampling from a continuous distribution, where the original idea is attributed to von Neumann [42]. This method is useful for generating exponential random variates; it requires no complicated function such as log, square root, or sine.

General Algorithm.- Let $f(x) > 0$ be defined for all $X \geq 0$ and satisfy the first-order linear differential equation

$$f'(x) + b(x) * f(x) = 0 \quad \ldots\ldots\ldots (2.2)$$

$$0 \leq x < \infty \quad ; \quad b(x) \geq 0.$$

Let

$$B(x) = \int_0^x b(t)\,dt$$

and assume that

$$\beta = \int_0^\infty e^{-B(x)}\,dx$$

then

$$f(x) = (1/\beta) * e^{-B(x)}$$

is a unique solution of (2.2) with $\int_0^\infty f(x)dx = 1$. Hence $f(x)$ is the probability density distribution of non-negative random variables.

The procedure to generate a random variable X with the density distribution f(x) is divided into two stages. In the first stage of the method an interval is selected for x, and in the second stage the value of x is determined within the interval by the acceptance-rejection method. Three tables for this procedure are needed, (q_k), (d_k) and (r_k). q_k defines an extreme of K-th interval in which the range of values of X is divided:

$$\text{Range of } X = ([q_0,q_1), [q_1,q_2),\ldots,[q_{K-1},q_K])$$

The intervals are chosen to be as large as possible subject to the constraint

$$B(q_k) - B(q_{k-1}) \leq 1 \quad \ldots\ldots\ldots(2.3)$$

The definition of d_k and r_k are:

$$d_k = q_k - q_{k-1}$$

and

$$r_k = \int_0^{q_k} f(x)\,dx$$

For simplicity, the following function is defined

$$G_k(x) = B(q_{k-1} + x) - B(q_{k-1}) \quad ; \; k=1,2,\ldots,K$$

Now, to carry out Forsythe's method, the following algorithm is used:

Algorithm F-1

1) Set k <- 1. Generate a uniform deviate U.

2) If $U \leq r_k$, go to step 4

3) k <- k + 1, go to step 2

4) Generate another deviate U, set W <- U*d_k

5) Set t <- G_k(W)

6) Generate a sequence of independent samples U_1, U_2,....,U_n from U(0,1), where n is determined by the condition: t $\geq U_1 \geq \ldots\ldots \geq U_{n-1} < U_n$ [if $t<U_1$, then n=1].

If n is odd, deliver X=q_{k-1}+W, otherwise go to step 4.

Steps 1 to 3 determine which interval [q_{k-1}, q_k) the variable X will belong to. Steps 4 to 6 determine the value of X within the interval.

The algorithm F-1 is based on the following theorem:

Theorem.- Let t be a given number in [0,1). Generate independent [0,1) uniform variates $U_1,\ldots,U_n$. n is determined by the condition t $\geq U_1 \geq \ldots\ldots \geq U_{n-1} < U_n$ [n=1 if $t < U_1$]. Then

(a) the probability of n being an odd number is $P(t) = e^{-t}$

(b) the expected value of n, irrespective of whether n is odd or even, amounts to $E(t) = e^t$

(Proof, see Ahrens & Dieter [3] p. 928)

An important feature of this method is that it does not specify a unique algorithm, but rather a family of algorithms, subject to (2.3) being satisfied. The interval d_k can be chosen at will. A disadvantage of this method is that it requires tables of the constants q_k, d_k and r_k.

Examples of how q_k, d_k and r_k can be defined in two distributions are as follows:

Exponential Distribution.- If $b(x) = 1$ in (2.2), then $B(x) = x$ and $f(x) = e^{-x}$. For algorithm F-1, $q_k=k$, $d_k=1$, $r_k=1-e^{-k}$ and $G_k(x)=x$

Normal Distribution.- If $b(x)=x$ in (2.2), then $B(x)=x^2/2$ and $f(x)=(1/2\pi)^{\frac{1}{2}} * e^{-x^2/2}$, corresponding to half of the normal distribution. For the algorithm F-1, we have

$$q_0=0,\ q_1=1,.....,q_k=(2*k-1)^{\frac{1}{2}} \quad (k\geq 2)$$

and

$$d_1=1,\ d_2=3^{\frac{1}{2}}-1,....,d_k=(2k-1)^{\frac{1}{2}} - (2k-3)^{\frac{1}{2}}$$

Also

$$G_k(x) = x^2/2 + q_{k-1}*x$$

The value of r_k must be computed from the probability integral.

In 1976, Atkinson & Pearce [8] made a timing comparison to show the efficiency of Forsythe's method on two computers (Cyber 73-4 and IBM 360/65). They showed that this method generated Beta and Gamma variables appreciably faster than any previously published, except for the Gamma distribution with index α less than 0.1. For the generation of a Normal variable, Forsythe's method did not provide the best algorithm.

2.2.4) Alias Method

The alias method is a clever, new and fast method for generating random variables from an arbitrary discrete distribution. This method is due to Walker [43]. The method is related to rejection techniques but differs from them in that all samples comprising the input data contribute to the samples in the target distribution. A simple probabilistic proof that the method works for any discrete distribution with a finite number of outcomes can be found in Kronmal and Peterson [25].

Method

Suppose the random variable X is distributed over the integers 1,2,...,n with p(i)=Pr[X=i]. Let I be an integer uniformily distributed over 1,2,....,n<∞, i.e., Pr[I=j] = q = 1/n. The method consists of setting

$$X = \begin{cases} I & \text{with probability } R(I) \\ A(I) & \text{with probability } 1-R(I) \end{cases}$$

where A(I) is an alias. The functions R(I) & A(I) are chosen according to the following algorithm:

Algorithm AT-1

0. [Initialize sets H and L; ϕ denotes the empty set]: H <- ϕ, L <- ϕ.
1. For i = 1 to n:
   a) R(i) <- n*p(i);
   b) if R(i) > 1, then add i to H;
   c) if R(i) < 1, then add i to L.

2. a) If H = ϕ, stop;

   b) otherwise select an index j from L and an index k from H.

3. a) Set A(j) <- k;

   b) R(k) <- R(k) + R(j) - 1;

   c) if R(k) $\leq$ 1, remove k from H;

   d) if R(k) < 1, add k to L;

   e) remove j from L [and from further consideration].

4. Go to step 2

A Fortran subroutine of the above algorithm can be seen in Bratley, et al ([11], p. 300-301) and Walker [43]. The algorithm to generate a random variable using the alias method is as follows:

1) Set I <- $\lceil V \rceil$; V=nU , U ~ U(0,1) (thus I is from a discrete uniform on [1,n])

2) Set W <- I - V

3) If W $\leq$ R(I), deliver X = I ;
   otherwise deliver X = A(I)

It is remarkable that the number of operations required to generate a discrete variable, using Walker's alias method, are so few (in particular that only one comparison is needed) and that it does not depend on the discrete distribution specified, not even on the number of mass points of the distribution.

A timing comparison of the alias method can be seen in Peterson & Kronmal [37]. The authors show that this method is faster than the indexed-search method and Marsaglia table method (these two methods are not described in this paper). The experiment is realized with 50,000 samples generated from a Poisson variable in the computers CDC Cyber 170-750 and DEC-10.

2.2.5) Composition Method

The composition method is applied when the distribution function F, from which we wish to sample, can be expressed as a probability mixture of a finite number of distribution functions F_1, F_2,...F_k. We would hope to be able to sample from F_j's more easily than from the original F.

Specifically, it is assumed that for all x, F(x) can be written as

$$F_X(x) = \sum_{j=1}^{k} \alpha_j F_j(x)$$

where $k < \infty$, $\alpha_j \geq 0$, $\sum_{j=1}^{k} \alpha_j = 1$.
Equivalently, if X has density function $f_X(x)$ which can be written as

$$f_X(x) = \sum_{j=1}^{k} \alpha_j f_j(x)$$

where the f_j's are other densities, the composition method still applies; the discrete case is analogous. The general composition algorithm, then, is as follows:

1) Generate a positive random integer J such that $P(J = j) = \alpha_j$ for $j = 1, 2, \ldots, k$
2) Given that $J = j$, generate X from $f_j(x)$
3) Deliver X

Example.- Generate a random variable from

$$f_X(x) = (5/12)[1+(x-1)^4] \quad ; \quad 0 \leq x \leq 2$$

which can be written

$$f_X(x) = (5/6) * f_1(x) + (1/6) * f_2(x)$$

where

$$f_1(x) = 1/2 \quad ; \quad f_2(x) = (5/2) * (x-1)^4 \quad 0 \leq x \leq 2$$

therefore

$$X = \begin{cases} 2 * U_2 & ; \text{ if } U_1 < 5/6 \\ 1+(2 * U_2 - 1)^{(1/5)} & ; \text{ if } U_1 \geq 5/6 \end{cases}$$

The advantage of the composition method is that we can sometimes find a decomposition that assigns high probability α_i to p.d.f.'s from which sampling X is inexpensive and concomitantly assigns low probabilities α_i' to p.d.f.'s from which sampling X is expensive. For further information about this method see Ahrens & Dieter [2], Marsaglia [28] and Morgan ([34], p. 107-113).

2.3) Specific Procedures

Although most methods for generating random variables can be classified into one of the general approaches described in section 2.2, some techniques simply rely on special properties of the desired distribution function F of the random variable X. Frequently, the special property will take the form of representing X in terms of other random variables which are more easily generated. Since there is no general form of these techniques, we shall give examples for three continuous (Exponential, Normal and Gamma) and three discrete (Binomial, Poisson and Geometric) distributions.

2.3.1) Continuous Distributions

2.3.1.1) Exponential

An exponential distribution X has p.d.f.

$$f_X(x) = \begin{cases} (1/\beta) * \exp(-x/\beta) & ; \ 0 \leq x \leq \infty \ , \ \beta > 0 \\ 0 & ; \ \text{otherwise} \end{cases}$$

which we will denote by EXP(β).

Procedure E-1

By the inverse method

$$F_X(x) = 1 - e^{(-x/\beta)} = U$$

so,

$$X = -\beta * \ln(1-U)$$

since 1-U has the same distribution as U, we have

$$X = -\beta * \ln(U)$$

For sampling purposes it may be assumed that $\beta = 1$. If V is sampled from EXP(1), which is called the standard exponential distribution, then $X = \beta * V$ is from EXP(β), where V=-ln(U). The algorithm for this procedure is as follows:

Algorithm E-1

1) Generate U from U(0,1)

2) X <- $-\beta * \ln(U)$

3) Deliver X

Procedure E-2

Another procedure to generate a random variable with exponential distribution is based on the following proposition:

Proposition.- Let $U_1,\ldots,U_n,U_{n+1},\ldots,U_{2n-1}$ be independent uniformly distributed random variables, and let $U_{(1)},\ldots,U_{(n-1)}$ represent the order statistics corresponding to the random sample $U_{n+1},\ldots,U_{2n-1}$. Assume

$U_{(0)} = 0$ and $U_{(n)} = 1$; then the r.v.'s

$$Y_k = (U_{(k-1)} - U_{(k)}) * \ln \prod_{i=1}^{n} U_i \ , \quad k=1,\ldots,n$$

are independent and distributed EXP(1). (Proof in [39], p. 68) So, the algorithm for this procedure will be as follows:

Algorithm E-2

1) Generate 2n-1 uniformly distributed random variates $U_1,\ldots,U_n,U_{n+1},\ldots\ldots,U_{2n-1}$
2) Arrange the variates $U_{n+1},\ldots\ldots,U_{2n-1}$ in order of increasing magnitude, that is, define them to be the order statistics $U_{(1)},\ldots\ldots,U_{(n-1)}$
3) $Y_k \leftarrow (U_{(k-1)} - U_{(k)}) * \ln(\prod_{i=1}^{n} U_i)$, $k=1,\ldots,n$
4) Deliver Y_k, $k=1,\ldots,n$, as a random sample from EXP(1).

The advantage of algorithm E-2 is that it requires only one computation of $\ln\prod_{i=1}^{n} U_i$ for generating n exponential variates simultaneously. The disadvantage is that it needs 2n-1 uniform variates rather than n uniform variates for the inverse transform method. Additionally, this algorithm requires the order arrangement of the last n-1 uniform variates generated.

Alternative procedures for generating from EXP(β) without the use of logarithmic transformations can be seen in Ahrens and Dieter [2], Fishman [19] and Marsaglia [29].

2.3.1.2) Gamma

A random variable X has a gamma distribution if its p.d.f. is defined as

$$f_X(x) = \begin{cases} \dfrac{x^{(\alpha-1)} * e^{(-x/\beta)}}{\beta^{\alpha} * \Gamma(\alpha)} & ; \ 0 \leq x \leq \infty \ , \ \alpha, \beta > 0 \\ 0 \ , \quad \text{otherwise} \end{cases}$$

and is denoted by $G(\alpha,\beta)$. Note that for $\alpha=1$, $G(1,\beta)$ is $EXP(\beta)$.

Since the c.d.f. does not exist in explicit form for gamma distribution with $\alpha \neq 1$, the inverse transform method cannot be applied. Therefore, alternative methods of generating gamma variates must be considered.

Procedure G-1

One of the most important properties of the gamma distribution is the reproductive property; let X_i, $i=1,\ldots,n$, be a sequence of independent random variables from $G(\alpha_i,\beta)$, then $X=\Sigma_{i=1}^{n} X_i$ is from $G(\alpha,\beta)$ where $\alpha=\Sigma_{i=1}^{n}\alpha_i$.

Case α integer

If α is an integer, say $\alpha=m=\Sigma\alpha_i$, $i=1,\ldots,n$, a random variate from the gamma distribution $G(m,\beta)$ can be obtained by summing m independent exponential random

variates EXP(β), that is,

$$X = \beta * \Sigma_{i=1}^{m} [-\ln(U_i)]$$

$$= -\beta * \ln(\Pi_{i=1}^{m} U_i) \ldots\ldots\ldots (3.6.10)$$

In this case G(m,β) is called the Erlang distribution and is denoted by Er(m,β). The following algorithm describes generating r.v.'s from Er(m,β):

Algorithm G-1

1) X <- 0
2) Generate V from EXP(1)
3) X <- X + V
4) If m = 1, then
   4.1) X <- β*X,
   4.2) deliver X
5) m <- m - 1
6) Go to step 2

It is not difficult to see that the mean computation (CPU) time for generation from the Erlang distribution is an increasing linear function of m.

Case α is not an integer

For some time no exact method was known and approximate techniques were used. The most common method was the so-called probability switch method, which is based on the composition method.

Let $m=[\alpha]$ be the integral part of α, $(\alpha \geq 1)$, and let $\delta=\alpha-m$. With probability δ, generate a random variate from $G(m+1,\beta)$. With probability $1-\delta$, generate a random variate from $G(m,\beta)$;

$$G(\alpha,\beta) \approx \delta * G(m+1,\beta) + (1-\delta) * G(m,\beta) \quad ; \alpha \geq 1$$

This mixture of gamma variates with integral shape parameters will approximate the desired gamma distribution. This technique yields better results with higher values of α ([35], p. 88).

Procedure G-2

This procedure is due to Cheng [12] and describes gamma generation $G(\alpha,1)$ for $\alpha > 1$ with execution time asymptotically independent of α. The procedure is based on the acceptance-rejection method.

Let the density function of $G(\alpha,1)$

$$f_X(x) = \frac{x^{\alpha-1}e^{-x}}{\Gamma(\alpha)} \quad ; \quad x > 0$$

be represented by the product of the following functions:

$$h(x) = \begin{cases} \lambda\mu x^{\lambda-1}(\mu + x^{\lambda})^{-2} & , x \geq 0 \\ 0 & ; \text{ otherwise} \end{cases}$$

$$C = \frac{4\alpha^{\alpha}}{\Gamma(\alpha)\lambda e^{\alpha}}$$

$$g(x) = x^{\alpha-\lambda}(\mu + x^{\lambda})^2 e^{\alpha-x}/(4\alpha^{\alpha+\lambda})$$

where

$$\mu = \alpha^{\lambda} \quad , \quad \lambda = (2\alpha - 1)^{\frac{1}{2}}$$

C is a monotonically decreasing function of α. Selected values of C are given below

α	1	1.5	2	..	5	...	20
C	1.47	1.31	1.25		1.17		1.14

Using Sterling's approximation for $\Gamma(\alpha)$ ([1], p. 257) when α is large, it can be shown that C tends asymtotically to the value $2/\sqrt{\pi} \approx 1.13$ as α tends to infinity.

The distribution function corresponding to g(x) is

$$\Psi(x) = \frac{x^{\lambda}}{(\mu+x^{\lambda})}$$

To generate X from G(α,1), we sample x from $\Psi(x)$, which means substitute x from

$$\frac{x^{\lambda}}{(\mu+x^{\lambda})} = U_1 \qquad ; U_1 \text{ from } U(0,1)$$

$$\Rightarrow \quad \frac{x^{\lambda}}{\mu} = \frac{U_1}{(1-U_1)}$$

$$\Rightarrow \quad x = \alpha^{\lambda} e^{(1/\lambda)\ln((U_1)/(1-U_1))}$$

Then, we compare

$$U_2 \leq g(x^*) \qquad ; \quad U_2 \text{ from } U(0,1)$$

if the inequality holds, accept x^*, otherwise repeat the process. Cheng [12] recommends the following algorithm for this procedure:

Algorithm G-2

1) Sample U_1 and U_2 from $U(0,1)$
2) $V \leftarrow (1/\lambda) * \ln[U_1/(1-U_1)]$
3) $x \leftarrow \alpha e^V$
4) If $b + d*V - X \geq \ln(U_1^2 U_2)$
   deliver X
5) Go to step 1

where $b = \alpha - \ln 4$ and $d = \alpha + \lambda$.
Cheng [12] simplifies step 4 for a faster pretest of the general acceptance-rejection method as follows:

4') Set $Z = U_1^2 * U_2$, $Y = b + d*V - X$.
   If $Y + ((\ln 4.5) + 1) - 4.5*Z \geq 0$, deliver X.
   If $Y \geq \ln(Z)$, deliver X.

Cheng made a timing comparison of the basic method and the above modification. He shows good results in terms of speed for $\alpha > 1.5$. For further examples to generate a random variable from $G(\alpha,\beta)$ see Ahrens & Dieter [5], Cheng [14], Fishman [19] and Wallace [44].

2.3.1.3) Normal

A random variable X has a normal distribution if the p.d.f. is

$$f_X(x) = [1/(2\pi\sigma^2)^{1/2}]\exp[-(x-\mu)^2/(2\sigma^2)] \; ; \; -\infty<x<\infty$$

and is denoted by $N(\mu, \sigma^2)$. Here μ is the mean and σ^2 the variance. Since $X=\mu + Z\sigma$, where Z is from the standard normal distribution, N(0,1), we consider only generation of N(0,1). As the inverse of the normal distribution, $F^{-1}(x)$, does not have a closed-form expression, the inverse transform method cannot be applied; therefore, another method has to be employed.

Procedure N-1

This approach is due to Box and Muller [10]. The procedure is based on the following statement:
If U_1 & U_2 are independent random variates from U(0,1), then the variates

$$Z_1 = [-2\ln(U_1)]^{1/2}\cos(2\pi U_2)$$

$$Z_2 = [-2\ln(U_1)]^{1/2}\sin(2\pi U_2) \quad \ldots\ldots (2.4)$$

are independent standard normal deviates.

The algorithm for this procedure is as follows:

Algorithm N-1

1) Generate two independent random variates U_1 & U_2 from U(0,1)
2) Compute Z_1 & Z_2 simultaneously by substituting U_1 & U_2 in the equations (2.4)

Marsaglia & Bray [32] made an improvement to the above algorithm by eliminating the trigonometric calculations. This improvement, known as the polar method, is described in the following algorithm:

1) Generate U_1 and U_2 from U(0,1)
2) $V_i \leftarrow 2U_i - 1 \quad ; i = 1,2$
3) $W \leftarrow V^2_1 + V^2_2$
4) If $W > 1$, go to step 1
5) $Y \leftarrow [\,(-2\ln(W))/W\,]^{\frac{1}{2}}$
6) $X_1 \leftarrow V_1 * Y \quad , \quad X_2 \leftarrow V_2 * Y$
7) Deliver X_1 and X_2

The probability of rejecting the pair U_1 and U_2 is given by $1-\pi/4 \cong .215$. Atkinson and Pearce [8] show that the polar method is 30.7% faster than the Box and Muller method in a Cyber 73-14 computer machine and 8.6% faster in an IBM360/65. Ahrens and Dieter [2] experienced a 26.5% reduction in an IBM360/65.

The difference in percentages on the IBM's is due to the different pseudo-random generators; in Atkinson and Pearce simulation experiment takes on average 23μsec to

generate one pseudo random number, while in Ahrens & Dieter, 13μsec; so, apparently, the difference in both methods increases according to the speed for generating pseudo-random numbers.

Procedure N-2

This procedure is based on the acceptance-rejection method. Let the rv. X be distributed

$$f_X(x) = 2 * \frac{1}{\sqrt{2\pi}} \exp(-x^2/2) \quad ; x \geq 0$$

$$= \sqrt{2/\pi} \exp(-x^2/2)$$

which has the shape of the right side of the standard normal distribution. The function is multiplied by two to satisfy $\int_0^\infty f_X(x)dx=1$. Since the standard normal distribution is symmetrical about zero, we can assign a random sign to the r.v. generated from $f_X(x)$ and obtain a r.v. from N(0,1).

To generate a random variable, X, with $f_X(x)$, we apply the acceptance-rejection method as follows:

$$f_X(x) = C * h(x) * g(x)$$

where

$$h(x) = e^{-x}$$

$$C = \sqrt{2 * e/\pi}$$

and

$$g(x) = \exp[-(x-1)^2/2]$$

Then, the acceptance condition [U≤g(Y)] is checked, that is

$$U \leq \exp[-(Y-1)^2/2]$$

which is equivalent to

$$-\ln(U) \geq (Y-1)^2/2$$

where Y is from EXP(1).

Since -ln(U) is also from EXP(1), the last inequality can be written

$$V_2 \geq (V_1-1)^2/2$$

where both V_2=-ln(U) and V_1=Y are from EXP(1).

The efficiency of this procedure is equal to $1/C=\sqrt{\pi/2e}\cong 0.76$

The algorithm for the above procedure is as follows:

Algorithm N-2

1) Generate V_1 and V_2 from EXP(1)
2) If $V_2 \geq (V_1-1)^2/2$, go to step 1
3) Generate U from U(0,1)
4) If U ≥ 0.5, deliver $Z = -V_1$
5) Deliver $Z = V_1$

For further procedures to generate random variable from a normal distribution see Ahrens & Dieter [2], Kinderman & Ramage [23] and Marsaglia [31].

2.3.2) Discrete Distributions

2.3.2.1) Binomial

The probability mass function, p.m.f., of the binomial distribution is

$$f_R(r) = Pr[\ R=r\ ]$$

$$= \frac{n!}{(n-r)!r!}\ p^r(1-p)^{n-r} \qquad ;\ \begin{matrix} r=0,1,\ldots,n \\ p > 0 \end{matrix}$$

with $E(R) = n*p$ and $Var(R) = n*p*(1-p)$. The binomial distribution describes the number of successes in n independent trials, where p is the probability of success at any given trial.

To generate a random variable R, the following algorithm can be used:

Algorithm BI-1

1) Set R <- 0
2) Do the following n times:
    2.1) Generate U uniform on U(0,1)
    2.2) If U $\leq$ p , R <- R + 1
3) Deliver R

This algorithm will depend on the value assigned to n. For large n, the normal distribution can be considered. As n

increases the distribution of

$$Z = \frac{X - n*p + \frac{1}{2}}{[n*p*(1-p)]^{\frac{1}{2}}} \quad \ldots\ldots\ldots (2.5)$$

approaches N(0,1). (De Moivre-Laplace limit theorem, [18], p. 186).

To obtain a binomial variate we generate Z from N(0,1), solve (2.5) with respect to X, and round to non-negative integer, that is,

$$R = \max(0, [-0.5 + n*p + Z*(n*p*(1-p))^{\frac{1}{2}}])$$

where [*] denotes the integer part. If $np^{3/2} > 1.07$, the error in using the normal distribution function instead of the binomial never exceeds 0.05 for any r ([22], p. 132).

2.3.2.2) Poisson

The poisson distribution has a single parameter λ. The p.m.f. is

$$f_X(x) = e^{-\lambda}\lambda^x/x! \quad ; \quad x=0,1,\ldots$$

The mean and the variance are both λ. Because X has an infinite range, the inverse transform method (section 2.2.1.2) is very slow. An alternative method is to use the exponential distribution since the poisson is related to EXP(1), when the number of arrivals occurring in the interval $(0,\lambda)$ is a poisson with parameter λ.

The above relationship is used to derive a generating procedure. If (Y_i) is a sequence of independent exponentials, each with expectation 1, then we wish to find X such that the Xth arrival occurs before λ but the X+1st occurs after λ; that is

$$\Sigma_{i=1}^{X} Y_i \leq \lambda < \Sigma_{i=1}^{X+1} Y_i$$

Recalling how we generate exponentials, that is

$$-\Sigma_{i=1}^{X} \ln(U_i) \leq \lambda < -\Sigma_{i=1}^{X+1} \ln(U_i) ,$$

or equivalently

$$\prod_{i=1}^{X} U_i \geq e^{-\lambda} > \prod_{i=1}^{X+1} U_i$$

This gives the following algorithm:

Algorithm Po-1

1) Set X <- -1 , m <- exp(λ)

2) Repeat the following until m < 1,

2.1) Generate U uniform on U(0,1)

2.2) Set X <- X + 1, m <- mXU

3) Deliver X

When λ is large this procedure is slow. It is recomended that the normal distribution be used as we did for the binomial case (see Ahrens and Dieter [6]). Another method for generating poisson random variables can be seen in Atkinson [7].

2.3.2.3) Geometric

A random variable has the geometric distribution if the p.m.f. is equal to

$$f_X(x) = p*(1-p)^x \quad .....(2.6)$$

$$x=0,1,..... \quad ; \ 0 < p < 1$$

which is denoted by Ge(p). The mean and the variance for Ge(p) are $(1-p)/p$ and $(1-p)/p^2$, respectively. The geometric distribution describes the number of trials to the first success in a series of Bernoulli trials.

To generate a random variable with a geometric distribution, we use the relationship of this distribution with the exponential distribution. Let Y be from EXP(β), then

$$Pr[X < Y \leq X+1] = (1/\beta)\int_X^{X+1} e^{-Y/\beta} dy$$

$$= e^{-X/\beta}*(1-e^{-1/\beta}) \quad .....(2.7)$$

which is $Ge(p=1-e^{-1/\beta})$ for X=0,1,2,....

For $\beta = -1/\ln(1-p)$ (2.7) is identical to (2.6). Therefore,

$$X = \ln(U)/\ln(1-p) = - V/\ln(1-p)$$

where $V = -\ln(U)$ is a standard exponential variate. Hence to generate a random variable from Ge(p) we generate a random variable, say X, from EXP(β) with $\beta = - 1/\ln(1-p)$, and then we get the integer part [X]. This procedure is more efficient than the inverse transform method only for $p < 0.25$ ([39], p. 104).

2.4) Empirical Distribution

In some situations, rather than fit a theoretical distribution, we might want to use observed data to specify directly a distribution, from which samples are drawn during simulation. If this is the case, the distribution is called an empirical distribution.

For continuous random variables the type of empirical distribution that can be defined depends on whether we know the actual values of the individual original observations $X_1, X_2, \ldots, X_n$ rather than the frequency of X_i's which fall into each of several specified intervals. If the original data is available, we can define a continuous, piecewise linear distribution function F by first sorting the X_i's into increasing order. Let $X_{(i)}$ denote the ith smallest of the X_j's, so that

$$X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}.$$

Then F is given by

$$F(x) = \begin{cases} 0 & ; \text{ if } x < X_{(1)} \\ \dfrac{i-1}{(n-1)} + \dfrac{x - X_{(i)}}{(n-1)(X_{(i+1)} - X_{(i)})} & ; \text{ if } X_{(i)} \leq x < X_{(i+1)} \\ 1 & ; \text{ if } X_{(n)} \leq x \end{cases}$$

Figure 2.2 gives an illustration for n = 6. Note that F(x) rises most rapidly over those ranges of x in which the $X_{(i)}$'s are most densely distributed, as desired. Also, for each i, $F(X_{(i)}) = (i-1)/(n-1)$, which (for large n) is

approximately the proportion of the X_j's that are less than the $X_{(i)}$. However, one clear disadvantage of specifying this particular empirical distribution is that random variables generated from it during a simulation run can never be less than $X_{(1)}$ or more than $X_{(n)}$.

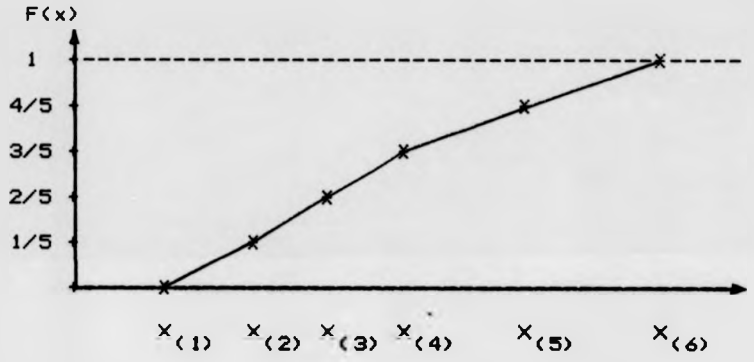


Figure 2.2 Continuous piecewise linear empirical distribution function

If, on the other hand, the data are grouped, a different approach must be taken. Suppose that the n X_j's are grouped into adjacent intervals $[a_0,a_1)$, $[a_1,a_2)$,..., $[a_{k-1},a_k)$ so that jth intervals contains n_j observations, where $n_1+n_2+\ldots+n_k = n$. A reasonable piecewise linear empirical distribution function G could be specified by first letting $G(a_0)=0$ and $G(a_j)=(n_1+n_2+\ldots+n_j)/n$ for

$j=1,2,..,K$. Then, interpolating linearly between the a_j's, we define

$$G(x) = \begin{cases} 0 & ; \text{ if } x < a_0 \\ G(a_{j-1}) + \dfrac{x-a_{j-1}}{a_j-a_{j-1}} * [G(a_j)-G(a_{j-1})] & ; \text{ if } a_{j-1} \leq x < a_j \\ 1 & ; \text{ if } a_k \leq x \end{cases}$$

Similar procedures can be applied for discrete data. For example, to generate a random variable from the empirical continuous distribution F(x) we have to find X from

$$F(x) = U \quad ; \quad U \text{ from } U(0,1)$$

so,

$$X = X_{(I)} + (P-I+1) * (X_{(I+1)} - X_{(I)})$$

where

$$P = (n-1) * U$$

and

$$I = [P] + 1$$

For further information about the generation of random variables from an empirical distribution see Bratley, et al ([11], pp. 137-139) and Law and Kelton ([26], pp. 176-177).

2.5) Conclusions

We have seen that there is no unique way to generate random variables. There are several questions that have to be answered before starting to generate samples with specific random characteristics, such as:

- What sort of information is available to define the random behaviour of the model?
- Is the theoretical distribution known? Does the c.d.f. have a close-form expression?
- Is there already a procedure that generate the desired random variable?
- What is the average number of uniform random numbers required per sample?
- What is the computer speed to generate a single sample from a uniform random variable?
- What are the marginal execution and set-up times?
- Are there better procedures to generate the desired random variable?
- What computer languages are available which can generate the desired random variable? Which is the fastest?

Although the above questions have to be considered, it is important to say that their answers will depend very much on the conditions and complexities of the simulation model; we might allow ourselves to slow the speed to generate a random sample in order to concentrate our effors on the validation of the simulation model.

REFERENCES

1 Abramowitz M. and Stegun I., Handbook of Mathematical Functions, 9th ed., Dover Publications Inc., New York, 1970.

2 Ahrens J. and Dieter U., "Computer Methods for Sampling from the Exponential and Normal Distributions", Communications of the ACM, Oct. 1972; 15(10): 873-882.

3 Ahrens J. and Dieter U., "Extensions of Forsythe's Method for Random Sampling from the Normal Distribution", Mathematics of Computation, Oct. 1973; 27(124): 927-937.

4 Ahrens J. and Dieter U., "Computer Methods for Sampling from Gamma, Beta, Poisson and Binomial Distribtuions", Computing (Vienna), 1974; 12: 223-246.

5 Ahrens J. and Dieter U., "Generating Gamma Variates by a Modified Rejection Technique", Communications of the ACM, Jan. 1982; 25(1): 47-54.

6 Ahrens J. and Dieter U., "Computer Generation of Poisson Deviates from Modified Normal Distributions", ACM Transactions on Mathematical Software, June 1982; 8(2): 163-179.

7 Atkinson A., "Recent Developments in the Computer Generation of Poisson Random Variables", Applied Statistics, 1979; 28(3): 260-263.

8 Atkinson A. and Pearce M., "The Computer Generation of Beta, Gamma and Normal Random Variables", Journal of the Royal Statistical Society, Series A (General), 1976; 139(4): 431-460.

9 Atkinson A. and Whittaker J., "A Switching Algorithm for the Generation of Beta Random Variables with at Least One Parameter Less than 1", Journal of the Royal Statistitical Society, Series A (General), 1976; 139(4): 431-460.

10 Box G. and Muller M., "A Note on the Generation of Random Normal Deviates", Annals of Mathematical Statistics, 1958; 29: 610-611.

11 Bratley P., Fox B. and Schrage L., A Guide to Simulation, Springer-Verlag, USA, 1983.

12 Cheng R. "The Generation of Gamma Variables with Non-integral Shape Parameters", Applied Statistics, 1977; 26(1): 71-75.

13 Cheng R., "Generating Beta Variates with Nonintegral Shape Parameters", Communications of the ACM, April 1978; 21(4): 317-322.

14 Cheng R. and Feast G., "Some Simple Gamma Variate Generators", Applied Statistics, 1979; 28(3): 290-294.

15 Devroye L., "The Computer Generation of Poisson Random Variables", Computing (Vienna), 1981; 26: 197-207.

16 Devroye L., "A Note on Approximations in Random Variate Generation", Journal of the Statistical Computation and Simulation, 1982; 14(2): 149-158.

17 Devroye L., "On the Use of Probability Inequalities in Random Variate Generation", Journal of Statistical Computation and Simulation, 1984; 20(2): 91-100.

18 Feller W., An Introduction to Probability Theory and its Application, 3rd ed., Vol. 1, Wiley, USA, 1968.

19 Fishman G., Concepts and Methods in Discrete Events Digital Simulation, Wiley, USA, 1973.

20 Forsythe G., "Von Neumann's Comparison Method for Random Sampling from the Normal and Other Distributions", Mathematics of Computation, Oct. 1972; 26(120): 817-826.

21 Golder E. and Settle J., "The Box-Muller Method for Generating Pseudo-Random Normal Deviates", Applied Statistics, 1976; 25: 12-20.

22 Kendall M. and Stuart A., The Advanced Theory of Statistics, 4th ed., Vol. 3, Charles Griffin, G.B., 1977.

23 Kinderman A. and Ramage J., "Computer Generation of Normal Random Variables", Journal of the American Statistical Association, 1976; 71: 893-896.

24 Knuth D, The Art of Computer Programming, Addinson-Wesley, USA, 1969.

25 Kronmal R. and Peterson A., "On the Alias Method for Generating Random Variables from a Discrete Distribution", The American Statistician, Nov. 1979; 33(4): 214-218.

26 Law A. and Kelton D., "Generating Random Variables", Chapter 7, Simulation, Modeling and Analysis, McGraw-Hill, USA, 1982, 240-278.

27 Loukas S., "Simple Methods for Computer Generation of Bivariate Beta Random Variables", Journal of Statistical Computation and Simulation. 1984; 20(2): 145-152.

28 Marsaglia G., "Expressing a Random Variable in Terms of Uniform Random Variables", Annals of Mathematical Statistics. 1961; 32: 894-898.

29 Marsaglia G., "Generating Exponential Random Variables", Annals of Mathematical Statistics. 1961; 32: 899-900.

30 Marsaglia G., "Generating Discrete Random Variables in Computer", Communications of the ACM. 1963; 6: 37-38.

31 Marsaglia G., MacLaren M. and Bray T., "A Fast Procedure for Generating Normal Random Variables", Communications of the ACM. Jan. 1964; 7(7): 4-10.

32 Marsaglia G. and Bray T., "A Convenient Method for Generating Normal Variables", Society for Industrial and Applied Mathematics Review. 1964; 6: 260-264.

33 Mood A., Graybill F. and Boes D., Introduction to the Theory of Statistics. 3rd ed., McGraw-Hill, 1974.

34 Morgan B., Elements of Simulation. Chapman and Hall, G.B., 1984.

35 Naylor T., Balintfy J., Burdick D. and Chu K., Computer Simulation Techniques. Wiley, USA, 1966.

36 Neave H., "On Using the Box-Muller Transformation with Multiplicative Congruential Pseudo-Random Number Generators", Applied Statistics. 1973; 22: 92-97.

37 Peterson A. and Kronmal R., "Analytic Comparison of Three General-purpose Methods for the Computer Generation of Discrete Random Variables", Applied Statistics, 1983; 32(3): 276-286.

38 Ripley B., "Computer Generation of Random Variables: A Tutorial", International Statistical Review, 1983; 51: 301-319.

39 Rubinstein R., Simulation and the Monte Carlo Method, Wiley, USA, 1981.

40 Scott J., "Faster Methods for Random Sampling", Communications of the ACM, July 1984; 27(7): 703-718.

41 Tocher K., The Art of Simulation, Hodder and Stoughton, G.B., 1963.

42 Von Neumann J., "Various Techniques Used in Connection with Random Digits", Applied Mathematics, Journal of Research of the National Bureau of Standards, Series 3, 1951, 36-38.

43 Walker A., "An Efficient Method for Generating Discrete Random Variables with General Distributions", ACM Transactions on Mathematical Software, Sept. 1977; 3(3): 253-256.

44 Wallace N., "Computer Generation of Gamma Random Variates with Non-integral Shape Parameters", Communications of the ACM, Dec. 1974; 17(12): 691-695.

III) Pseudo-Random Numbers Generator

3.1) Introduction

Generating random numbers uniformly distributed in a specified interval is fundamental to simulation; every procedure discussed in chapter II for generating random variates transforms one or more uniform random numbers. Many techniques for generating random numbers have been suggested, tested and used in recent years [20]. Some of these are based on random phenomena, others on deterministic recurrence procedures.

Initially, manual methods were used, including such techniques as coin flipping, dice rolling, and roulette wheels. These methods were too slow for general use, and moreover, sequences generated by them could not be reproduced. Shortly, following the advent of the computer, it became possible to obtain some sort of random numbers. In 1955 the RAND Corporation [38] published a table of a million random digits that may be stored in the memory of a computer. The advantage of this method is that the sequence of random numbers will be reproducible; its main disadvantage is the problem of space in the computer memory.

In view of these difficulties, John von Neumann [46] suggested the mid-square method. This method takes the square of the preceding random number and extracts the middle digits. Suppose we wish to generate four digit integers and the last number generated was 8234. To obtain the next number in the sequence we square the last one and

use the middle four digits of the product. In this case the product is 67798756, so the next number is 7987. The next few numbers of the sequence are 7921, 7422, 0860. One of the main drawbacks of this method is that once a zero is encountered the sequence terminates, and that not all numbers are likely to occur (see Tocher [43], p. 74).

Another generator of random numbers is the Fibonacci method. This method adds two or more previous numbers together and then takes the remainder when this sum is divided by a number called the modulus; this procedure is called 'Additive Congruential'. If X_i is the i-th number generated and 'm' is the modulus, then the Fibonacci method is represented by

$$X_i \equiv (X_{i-1} + X_{i-2}) \bmod m \quad .....(3.1)$$

One of the weaknesses of this method is conspicuous serial correlation. For example, suppose that m=1000, X_0=1 and X_1=1; then the next sequence of numbers is 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 597, 584, 181, etc.

This section considers methods for generating a sequence of random fractions, i.e., real numbers U_n, uniformily distributed between zero and one. Since a computer can represent a real number with only finite accuracy, we shall present procedures to generate integer X_n between zero and some number m, and then the fraction $U_n = X_n/m$ which lies between zero and one.

3.2) Considerations in Pseudo-Random Number Generators

When a sequence of numbers that shows similar characteristics of a uniform variable is generated in a completely deterministic way, then the sequence is called a sequence of 'pseudo-random' numbers.

There are three main considerations that play influential roles in determining whether or not a particular source of numbers provides a sequence of pseudo-random numbers that are adequate for simulation. These numbers must pass a battery of statistical tests designed to reveal departures from independence and uniformity. For a truly random sequence $X_1, X_2, \ldots$ $(0 \leq X_i \leq m)$, the elements of any subsequence of these numbers are jointly independent and each has a uniform distribution over (0,m); failure to have this property can lead to severely misleading results in simulation.

The second significant property requires that the pseudo-random numbers contain enough digits to ensure that the generation of numbers on the interval (0,1) is sufficiently dense. This property usually depends on the length of words in the computer memory. Since most medium-size to large computers can store at least 31 binary digits, it is possible, at least in theory, to produce a fairly dense sampling on (0,1) of the 2^{31}=2,147,483,648 available numbers.

The third significant property concerns the efficiency with which a particular source produces pseudo-random numbers. The faster an algorithm produces a number, the more desirable that algorithm is. A minimal utilization of storage is also attractive, specially with the new generation of computers whose user charges often depend on space utilization as well as computing time.

Since these properties seldom, if ever, characterize any one method of producing pseudo-random numbers, some compromise must be made. It is generally agreed that the presence of sufficient independence and uniformity to preserve the integrity of the particular experiment under consideration should be the prevailing criterion in determining adequacy [11].

3.3) Linear Congruential Generators (LCG)

The most commonly used method for generating pseudo-random numbers is the Linear Congruential Generator (LCG). This generator produces a sequence of numbers according to some recursive formula; a new number is generated by the previous one. Although these processes are completely deterministic, it can be shown that the numbers generated by certain LCG's appear to be uniformily distributed and statistically independent.

The common representation of the LCG is a congruence relationship, expressed as:

$$X_{i+1} = (a*X_i + c)(\text{mod } m) \quad \ldots\ldots\ldots (3.2)$$

$$i=1,\ldots\ldots,n$$

where the multiplier a, the increment c, and the modulus m are non-negative integers. The modulo notation means that

$$X_{i+1} = a*X_i + c - m*k_i$$

where $k_i = [(a*X_i+c)/m]$ denotes the integer part in $(a*X_i+c)/m$. For example, let $a=c=X_0=7$ and $m=10$; then,

i	X_i	$X_{i+1}=(7*X_i+7)-[(7*X_i+7)/10]*10$	
1	7	6	
2	6	9	
3	9	0	
4	0	7	<- from here the sequence
:	:	:	starts to be repeated
:	:	:	

LCG can also be presented as the following sequence (see ref. [35]), p. 48):

$$X_1 \equiv a*X_0 + c \pmod m$$

$$X_2 \equiv a*X_1 + c = a^2*X_0 + (a + c)*c \pmod m$$

$$X_3 = a^3*X_0 + (a^2+a+1)*c = a^3*X_0 + c*\frac{(a^3-1)}{(a-1)} \pmod m$$

. . . . . .

$$X_i \equiv a^i*X_0 + c*\frac{(a^i-1)}{(a-1)} \pmod m$$

Because of the deterministic character of the sequence, the entire sequence recurs as soon as any number is repeated. It is said that the sequence "got into a loop"; that is, there is a cycle of numbers that is repeated endlessly. This property is common to all sequences having the general form $X_{i+1} = f(X_i)$. (See Knuth [24], p. 9)

The repeating cycle is called the 'period'; in the last example the period length was 4. When the length of the period is equal to m, it is say that the pseudo-random number generator has 'full period'.

Most computerized versions of the congruential generators employ a modulus $m = p^{\beta}$, where p denotes the number of numerals in the number system utilized by the computer and β denotes the number of digits in a word; for binary computers p = 2.

There are two reasons for choosing $m = p^{\beta}$. First, reduction modulo m is accomplished by truncating and retaining only the low order β digits and second, conversion to the unit interval (to obtain uniformally distributed variates) only involves moving the point to the left of the number, as we will see in section 3.3.4).

When $c = 0$, the generation of random numbers is little faster than it is when $c \neq 0$; the disadvantage when $c = 0$ is that the length of the period of the sequence is cut down, as we will see in section 3.3.2. The term 'Multiplicative Congruential Generator' is used to denote LCG with c=0, and 'Mixed Congruential Generator' when c≠0.

3.3.1) Mixed Congruential Generator ($c \neq 0$)

It can be shown that the generator defined in eq. (3.2) has a full period provided that the following conditions hold: ([24], p. 16)

i) c is relatively prime to m, that is, c and m have no common divisor

ii) $b = a - 1$ is a multiple of p, for every p dividing m

iii) b is a multiple of 4, if m is a multiple of 4

For a binary computer, where $m = 2^{\beta}$, it is shown that a full period is guaranteed when $a = 2^{r} + 1$, $r \geq 2$ and c is an odd number. Hull and Dobell [19] using a computer IBM 7090 show that good statistical results can be achieved when $m = 2^{35}$, $a = 2^{7} + 1$, and c=1.

When $m = 10^{\beta}$, in order to generate a sequence with a full period, 'c' must be a positive number not divisible by 2 or 5, and the multiplier 'a' must satisfy the condition $a \equiv 1 \pmod{20}$, or alternatively $a = 10^{r} + 1$, $r > 1$. Satisfactory statistical results have been achieved by choosing $a = 101$, $c = 1$ and $r \geq 4$. (see ref. [1])

3.3.2) Multiplicative Congruential Generator (c = 0)

The multiplicative congruential generator is represented by the following expression:

$$X_{i+1} = a*X_i \pmod{m} \ldots\ldots\ldots (3.3)$$

This generator can not achieve a full period, because 'c' does not satisfy the condition described in the previous section. However, it may be possible to achieve an acceptably long period, provided that X_0 is relatively prime to the modulus 'm' and that the multiplier 'a' meets certain congruence conditions. For example,

- if $m = 2^{\beta}$, $\beta \geq 4$, a period $h = 2^{\beta-2}$ is generated provided that X_0 is odd and $a \equiv 3 \bmod 8$ or $5 \bmod 8$ ([39], p.303). For instance, with $X_{i+1} \equiv 11X_i \bmod 2^4$, we generate a period $2^{4-2}=4$, because $11 \equiv 3 \bmod 8$ ($3=11-[11/8]*8$). An example of possible sequence with two starting points is as follows:

X_0	Sequence
3	1,11,9,3,1,11,9,3,.....
5	7,13,15,5,7,13,15,5,...

- if $m = 10^{\beta}$, $\beta \geq 5$, and X_0 is not a multiple of 2 or 5, the period $h=5*10^{\beta-2}$ is achieved when a = 200r ± s, where 'r' is any positive integer and 's' is any of the following 32 numbers: 3, 11, 19, 21, 27, 29, 37, 53, 59, 61, 67, 69, 77, 83, 91, 109, 117, 123, 131, 133, 139, 141, 147, 163, 171, 173, 179, 181, 187, 189, 197. (see ref. [24], p. 20)

It is possible to increase the length of the period to m-2 provided that the modulus is of the form $m=2^{\beta}-1$ and is a prime number. The condition for the multiplier 'a' is that it has to be prime to $m-1=2^{\beta}-2$ (the largest common divisor of 'a' and m-1 is unity). In this case 'a' is called a primitive root (see Tocher [43], p. 78). In Western and Miller [47] a list of primitive roots can be found for prime numbers $m \leq 999,961$.

Fuller [12] proves that with $m=2^{31}-1$ the multiplier $a=7^{n}$ (n is any integer greater than zero) is a primitive root. One common combination is $m=2^{31}-1 = 2147483647$ and $a=7^{5} = 16807$. This combination can be found in the APL system from IBM, the scientific library from IMSL, and in the SIMPL/I system ([4], p. 184).

3.3.2.1) Reverse Sequence

It is possible to generate a sequence of pseudo-random numbers in reverse order provided that the new multiplier is

$$a' = a^{L-1} \pmod{m}$$

where L is the cycle length. For example, consider the generator $X_{i+1} = 3X_i \pmod 7$. It generates the sequence 1, 3, 2, 6, 4, 5, 1; so, L=6. Using this information we calculate a',

$$\begin{aligned} a' &= 3^{6-1} \pmod 7 \\ &= 243 - [243/7]\times 7 \\ &= 5 \end{aligned}$$

Then, to generate the reverse sequence (1,5,4,6,2,3,1) we use the generator $X'_{i+1} \equiv 5 * X'_i \pmod 7$.

3.3.3) Serial Correlation of LCG

Greenberger [15] has shown that an approximation to the first step serial correlation, $Corr(X_{i+1}, X_i)$, in LCG is given by

$$\rho_{i+1} \approx \frac{1}{a} - \frac{6c}{am} * (1 - c/m) \pm \frac{a}{m}$$

The above formula is sometimes used to decide the values of the parameters in LCG which minimize ρ_{i+1} ([11], p. 175). For example, see the following combination of the constants a, c and m which produce a full period, but different correlations ([34], p. 60):

	a	c	m	ρ
i	$2^{34}+1$	1	2^{35}	0.25
ii	$2^{18}+1$	1	2^{35}	$<< 2^{-18}$

So, in this example, we would say that the second combination ($a=2^{18}+1$, $c=1$) is better than the first one, when $m=2^{35}$. However, as we will see in the section 3.4.2, this comparison of first step serial correlation will not be enough to guarantee a 'good' generator.

3.3.4) Defect of LCG

One of the defect of the multiplicative congruential generator is that if n-tuples $\pi_1=(u_1,\ldots,u_n)$, $\pi_2=(u_2,\ldots,u_{n+1})$, $\pi_3=(u_3,\ldots,u_{n+2})$,... of uniform variates $(u_i=X_i/m)$ produced by this generator are viewed as points in the unit cube of n-dimensions, then all the points will be found to lie in a relatively small number of parallel hyperplanes (see figure 3.1). Furthermore, there are many

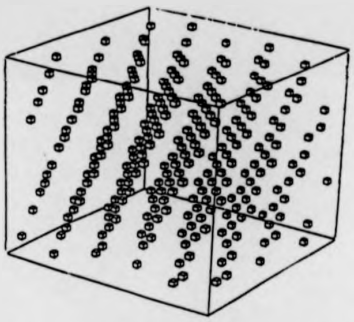


Figure 3.1 The tree-dimensional grid of triplets (u_j, u_{j+1}, u_{j+2})

systems of parallel hyperplanes which contain all the points. Marsaglia [33] shows an upper bound in the number of these hyperplanes by proving the following theorem:

Theorem. If $S_1, S_2,\ldots, S_n$ is any choice of integers such that

$$S_1+S_2a+S_3a^2+\ldots+S_na^{n-1}\equiv 0 \pmod m$$

then the points $\pi_i=(u_i,u_{i+1},\ldots,u_{i+n-1})$, i=1,2..., will lie

in the set of parallel hyperplanes defined by the equations

$$S_1 u_i + S_2 u_{i+1} + \ldots + S_n u_{i+n-1} = 0, \pm 1, \pm 2, \ldots$$

There are at most

$$|S_1| + |S_2| + \ldots + |S_n|$$

of these hyperplanes which intersect the unit n-cube, and there is always a choice of $S_1, S_2, \ldots, S_n$ such that all the points fall in fewer than $(n!m)^{1/n}$ hyperplanes.

Here is a table of $(n!m)^{1/n}$ for some values of m, power 2:

Table 3.1 .- Upper bound for the number of hyperplanes containing all the n-tuples

	n = 3	n = 5	n = 10
$m = 2^{16}$	73	23	13
$m = 2^{24}$	465	72	23
$m = 2^{32}$	2,953	220	41
$m = 2^{48}$	119,086	2,021	126

For example, in a binary computer with 32-bits words, $m=2^{32}$, fewer than 41 hyperplanes will contain all 10-tuples, fewer than 220 hyperplanes will contain all 5-tuples and fewer than 2,953 hyperplanes will contain all 3-tuples. Marsaglia states that similar results can be established for mixed congruential generators (see figure 3.2).

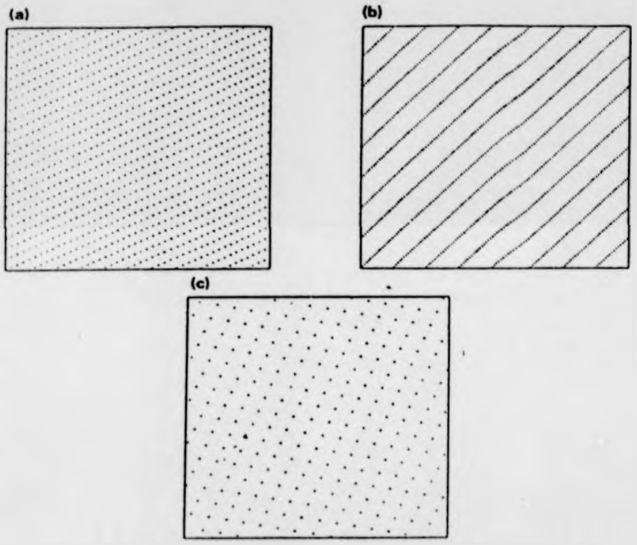


Figure 3.2 Lattices of all pairs (X_i, X_{i+1}) of a single period of congruential generators with modulus 1024 and (a) a = 45, c = 1 (b) a = 129, c = 1, and (c) a = 45, c = 0.

3.3.5) Linear Recursion mod 2 Generator

A linear congruential generator calculates X_i solely from X_{i-1}. Additive congruential generators (i.e, Fibonacci Method) use several previous values of X. Both of these methods are special cases of the general formula

$$X_i = (a_1X_{i-1}+a_2X_{i-2}+\ldots+a_nX_{i-n}+c) \bmod m \quad \ldots\ldots(3.4)$$

In this section we study the case when m=2 and c=0. Because X_i can equal only 0 or 1, such generators produce a bit stream (b_i). Furthermore, the only values that need to be considered for the a_j's are also 0 and 1. Thus b_i is obtained by adding modulo 2 to several of the preceding bits in the stream.

Modulo 2 addition is the exclusive-OR logical operation. This operation, denoted by XOR, makes the logical operation 'not-equal', NEQ, for example, when

A	B	A XOR B
1	1	0
1	0	1
0	1	1
0	0	0

One way of implementing the generator (3.4) with m=2 and c=0 is by using a Shift Register with Feedback (SRF) and a primitive polynomial h(x) of degree k; this sort of

polynomial will be defined later. As an example, take

$$h(x) = x^4 + x + 1$$

which has degree 4. This polynomial specifies a feedback shift register as shown in the following figure:

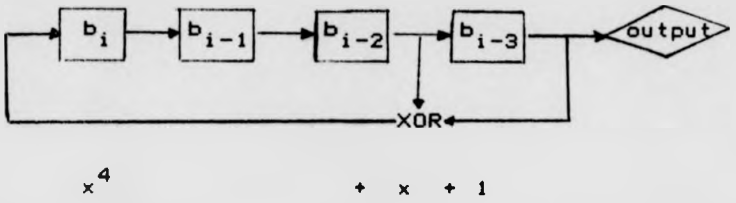


Figure 3.3.- Shift Register with Feedback (SRF)

Each box in figure 3.3 is a one-bit memory holding 0 or 1. At each iteration the register is shifted one place right, the boxes corresponding to the terms in h are added modulo 2, and the sum is fed back into the left-hand box. For example, if the shift register of figure 3.3 is initialized to the state 1000, its successive states and outputs are:

Table 3.2 Example of SRF

Iteration	States	Output
0	1 0 0 0	
1	0 1 0 0	0
2	0 0 1 0	0
3	1 0 0 1	0
4	1 1 0 0	1
5	0 1 1 0	0

XOR

J_0

6	1 0 1 1	0	J_1
7	0 1 0 1	1	
8	1 0 1 0	1	
9	1 1 0 1	0	J_2
10	1 1 1 0	1	
11	1 1 1 1	0	
12	0 1 1 1	1	
13	0 0 1 1	1	J_3
14	0 0 0 1	1	
15	1 0 0 0	1	
16	0 1 0 0	0	

Since each of the K boxes in figure 3.3 can hold 0 or 1, there are 2^K possible states for the shift register. Thus the sequence (b_i) must be periodic. Since the all-zero state generates only zero, the maximum possible period is 2^K-1.

Now we can define a primitive polynomial: h is a primitive polynomial if the shift register corresponding to h generates a sequence with period 2^K-1. In the previous example, all 15 possible non-zero states are achieved, so x^4+x+1 is a primitive polynomial. In Stahnke [41] a list of primitive polynomials of the form x^K+x^q+1 for $k \leq 168$ can be found.

Tausworthe [42] shows that if a sequence of n bits generated by SRF are considered as n-bit integers, then these integers are approximately uniformly distributed, and they do not have certain multidimensional non-uniformities associated with linear congruential generators. One way to generate such integers is to take k equal to the number of bits in the computer word (not counting the sign bit) and choose a primitive polynomial with only three terms, say

$$h(x) = x^k + x^q + 1$$

such that $k \geq 2q$. Now, if the computer can do full-word logical operations, components of the sequence (b_i) can generate k bits at a time as required, using only two shift and two exclusive-OR operations as follows:

Algorithm Ta-01

1) Y <- J (The integer J is formed by bits $b_{i+k-1}b_{i+k-2}\cdots b_i$)
2) Right shift Y by q bits, filling with zeros
3) Y <- J <- Y XOR J (the low-order bits of J have now been updated)
4) Left shift Y by k-q bits filling with zeros
5) J <- Y XOR J (J is now formed by bits $b_{i+2k-1}b_{i+2k-2}\cdots b_{i+k}$)
6) Deliver J, as the next required integer

For example, to form integers with four bits from the output bits of the table 3.2, the results from algorithm Ta-01 will be:

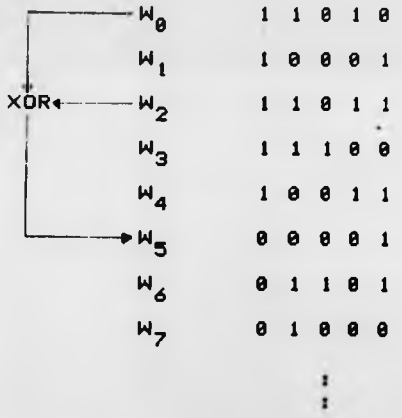
```
0001  The seed, J*0=1
0010  Shift q= 1 right ( <- )
----
0011  XOR
0000  Shift k-q= 3 left ( -> )
----
0011  XOR, J*1=3
0110  Shift q=1 right
----
0101  XOR
0000  Shift k-q= 3 left
----
0101  XOR, J*2=5
1010  Shift q=1 right
----
1111  XOR
0001  Shift k-q= 3 left
----
1110  XOR, J*3=14
1100  Shift q=1 right
----
0010  XOR
0000  Shift k-q= 3 left
----
0010  XOR, J*4= 2
```

* next pseudo-random number (decimals)

Lewis and Payne [28] suggest a refinement of the SRF algorithm. They use the logical operation XOR to a computer word W_i as follows: assuming a primitive polynomial x^k+x^q+1 such that 2^k-1 is a prime number, then the sequence of pseudo-random numbers, W_i, is calculated by the recursion

$$W_i = W_{i-k+q} \text{ XOR } W_{i-k}$$

For example, let the primitive polynomial be x^5+x^2+1, then the sequence of numbers is as follows:



This procedure is called the Generalized Feedback Shift Register (GFSR) method. The following algorithm can be used to implement GFSR:

Algorithm GFSR

1) Initialize W_1 to W_k; for example, using SRF algorithm
2) Initialize j <- k-q, i <- k
3) Set W_i <- W_j XOR W_i, and output W_i

4) Decrease j and i by 1. If j=0, set j <- k; if i=0, set i <- k

5) If enough pseudo-random integers have been generated, stop; otherwise, go to step 3

The advantage of the GFSR algorithm is that it is fast, easy to program, and offers large period length regardless of the word-size of the computer being used. Using k words of memory, a period of length 2^k-1 can be achieved. This algorithm is considered to be statistically qualified as a good pseudo-random number generator, if k and q are chosen properly ([4], p., 190).

3.3.6) Combination of Generators

Combining two sequences of pseudo-random numbers, (X_i) and (Y_i) to produce a third one, (Z_i) aims to reduce nonrandomness. If X_i and Y_i are distributed over the integers 0 and m-1, some suggestions are:

a) Set $Z_i = (X_i + Y_i) \bmod m$

b) Set $Z_i = X_i$ XOR Y_i

Wichman and Hill [49] report good results using essentially method a). Their method combine the following three generators

$$W_{i+1} = 171W_i \bmod 30269$$
$$X_{i+1} = 172X_i \bmod 30307$$
$$Y_{i+1} = 170Y_i \bmod 30323$$

to obtain the pseudo-random Z_{i+1} as follows:

$$Z_{i+1} = (W_{i+1}/30269 + X_{i+1}/30307 + Y_{i+1}/30323) \bmod 1$$

The period length of the above generator is 6.95×10^{12} [50]. The authors claim that this procedure is reasonably short, reasonably fast, machine independent, easily programmed in any language and statistically sound.

3.4) Testing

In this section, we describe some statistical tests for checking independence and uniformity of a sequence of pseudo-random numbers. As mentioned earlier, a sequence of pseudo-random numbers is completely deterministic, but insofar as it passes the set of statistical tests it may be treated as one of "truly" random numbers, that is, as a sample from U(0,1).

Knuth ([24], p. 38) divides tests of the supposedly independent and uniform U_i into two classes: Empirical Tests, in which a sample is taken and assessed without consideration of the way in which the numbers are generated, and Theoretical Tests which are based on the way in which the pseudo-random numbers are generated, and which do not require a sample. In this section, we describe some empirical tests and one theoretical test (the spectral test).

3.4.1) Empirical Tests

3.4.1.1) Chi-Square Test

The chi-square test (X^2 test) is perhaps the best known of all statistical tests, and it is a basic method which is used in connection with many other tests. The chi-square test applies in the following situation. Assume the event space (e.g., the possible values of the random number drawn) can be partitioned into n subsets (e.g., $U \leq 0.1$, $0.1 < U \leq 0.2$,...etc).

From a sample of M independent observations, let f_i be the number of outcomes falling into subset i. Let $\bar{f}_i$ be the expected number in the i-th subset under the hypothesized distribution. If the hypothesis is true, then as M increases, the statistic

$$\chi^2 = \Sigma_{i=1}^{n} \frac{(f_i - \bar{f}_i)^2}{\bar{f}_i}$$

has asymptotically the chi-square distribution with n-k-1 degrees of freedom, where K is the number of parameters estimated from the data; for testing the uniform distribution, k=0. If χ^2 is large, the hypothesized distribution is rejected (see Kendall & Stuart [22], pp. 419-442).

3.4.1.2) Kolmogorov-Smirnov Test

Like the chi-square test, this test allows one to make a statement about the probability of an observed sample being drawn from a specified distribution. It works only for continuous distributions. Unlike the chi-square test, not even the asymptotic distribution of the KS statistic is known analytically if any parameters are estimated from the data.

The procedure for this test is as follows: with n observations $X_1, X_2, \ldots, X_n$ order them, so that $X_{i+1} \geq X_i$ for $i=1,\ldots,n-1$. Compute

$$K_n^+ = \sqrt{n} \max_j [(j/n) - F(x_j)]$$

and

$$K_n^- = \sqrt{n} \max_j [F(x_j) - (j-1)/n]$$

These are respectively the maximum positive and negative differences between the empirical c.d.f. and the hypothesized c.d.f. Compare the K_n^+ and K_n^- with the values tabulated for example in Conover ([6], p. 397). If K_n^+ or K_n^- is too large or too small at the desired confidence level, then reject the hypothesis that the observations are drawn from F(x) (see Kendall & Stuart [22], pp. 452-460 and Knuth [24], pp. 45-52).

3.4.1.3) Empirical Tests Specifically for Uniform Random Number Sequences

The number of tests that can be applied to a sequence of random numbers is quite unlimited. Kendall and Babington [21] emphasized that the only limitation is human ingenuity; this implies that it will always be possible to develop a new test. In Knuth ([24], pp. 59-73) and Tocher ([43], pp. 43-49) several of these tests can be seen. Here we describe only two of them: the frequency test and the serial test.

Frequency Test

When a sequence of random numbers has a uniform distribution one of the first obvious requirements of a set of decimal digits is that each decimal digit shall occur with approximately equal frequency. The frequency test consists of recording the frequency of occurrence of each digit and comparing this with its expected frequency (one-tenth the sample size) by using a χ^2 test (section 3.411).

Serial Test

The frequency test merely tests the probability of the occurrence of each digit in a given position, but it does not exclude the possibility of serial correlation between digits in successive positions. Thus, the sequence

0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9

will satisfy the frequency test, but clearly it is not a random sequence. The serial test is used to check that no digit shall tend to be followed by any other digit. To make this test, we form a bivariate table showing the distribution of pairs of digits in the series, arranged in the rows according to the first digit, and in the columns according to the second digit; this procedure could be extended for more than two digit series. In all cells, we should get frequencies which are approximately equal in all cells [21]. This can be tested again by using χ^2 test.

3.4.2) The Spectral Test

The test was originally motivated by the consideration of non-random wave structure in the LCG. This test can be understood on much more straightforward geometrical grounds. Essentially, the spectral test is a way of measuring the n-dimensional uniformity of a complete cycle of the output generator.

The spectral test is related to the problem presented by Marsaglia [33] in relation to the maximum number of hyperplanes that contain the total number of pseudo-random numbers generated by the multiplicative congruential generator (section 3.4.4). This test determines the maximum distance between adjacent hyperplanes, the maximum being taken over all sets of covering hyperplanes (see figure 3.1). The larger this maximum, the worse the generator.

To show the spectral test in a practical way, we use an example presented in Bratley, et al ([4], p. 196). Let us consider a generator of the form $X_{i+1}=aX_i \bmod m$. Complete periods of the output for two specifications of this generator are listed below

Case $a = 7$, $m = 11$

$X_i = 1, 7, 5, 2, 3, 10, 4, 6, 9, 8$

Case $a = 6$, $m = 11$

$X_i = 1, 6, 3, 7, 9, 10, 5, 8, 4, 2$

Both cases satisfy one-dimensional uniformity as well as possible, i.e., every integer in the interval [1,10] is generated exactly once in a cycle. In two dimensions (X_{i+1}, X_i), uniformity collapses, see the following figure:

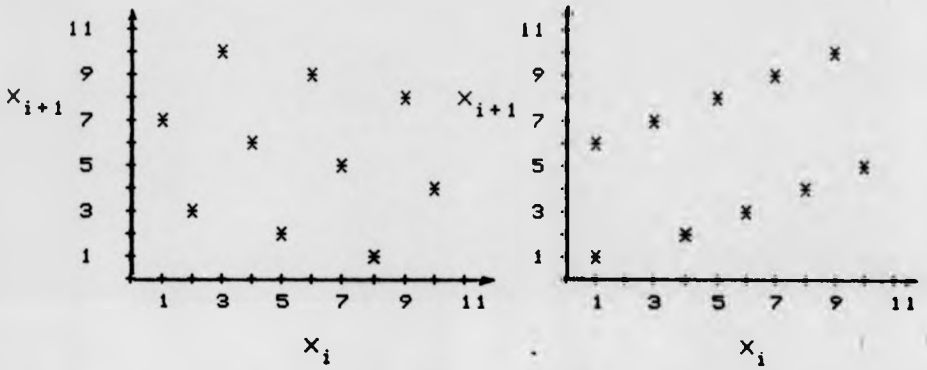


Figure 3.4 Plot of output pairs of two generators

In each case the points in 2-space can be covered by a family of parallel lines. For the generator $X_{i+1}=6X_i \bmod 11$, all the points can be covered by either two parallel lines of positive slope or five parallel lines of negative slope. For the generator $X_{i+1}=7X_i \bmod 11$, all points can be covered by either four lines of positive slope or three lines of negative slope. This confirms Marsaglia theorem (section 3.3.4) which says that there are at most $7 \approx (2!*11)^{\frac{1}{3}}$ hyperplanes that contain all the points in the plane.

In this example, the maximum distance for the first generator ($X_{i+1} \equiv 7X_i \bmod m$) is 3.48, which appears more random, while in the second one ($X_{i+1} \equiv 6X_i \bmod 11$) is 4.92 ([4], p. 197). So, we say that the first generator is better than the second one.

Algorithms to perform the spectral test can be found in Hoaglin & King [17] and Hopkins [18]. For a formal development of the spectral test see Knuth ([24], pp. 89-110).

3.5) Conclusions

We have seen that the linear congruential generator (LCG) is an easy way to generate pseudo-random numbers, and if its parameters are chosen properly this procedure can be used in simulation studies.

One of the disadvantage of LCG is the problem of non-uniformity presented in a space with k-dimensions (all the possible pseudo-random numbers generated by LCG can be covered by a number of hyperplanes). An alternative procedure is to use the Generalized Feedback Shift Register (GFSR) method.

The GFSR method does not have the problem of multidimensional non-uniformity associated with LCG and the period length depends on the memory capacity (if we have k words of memory, the maximum period length that can be generated is 2^k-1). This method is less easy to compute than the LCG, but due to its mentioned properties, its implementation is worthwhile in simulation studies.

REFERENCES

1 Allard J., Dobell A. and Hull T., "Mixed Congruential Random Number Generator for Decimal Machines", Journal of the ACM, 1963; 10: 131-141.

2 Atkinson A., "Test of Pseudo-Random Numbers", Applied Statistics, 1980; 29(2): 164-171.

3 Barnett V., "The Behaviour of Pseudo-Random Sequences Generated on Computers by the Multiplicative Congruential Method", Mathematics of Computation, 1962; 16: 63-69.

4 Bratley P., Fox B. and Schrange L., A Guide to Simulation, Springer-Verlag, USA, 1983.

5 Brown M. and Solomon H., "On Combining Pseudo-Random Number Generators", Annals of Statistics, 1979; 7: 691-695.

6 Conover W., Practical Non-Parametric Statistics, Wiley, USA, 1971.

7 Coveyou R., "Serial Correlation in the Generation of Pseudo-Random Numbers", Journal of the ACM, 1960; 7: 72-74.

8 Coveyou R. and MacPherson R., "Fourier Analysis of Uniform Random Generator", Journal of the ACM, 1967; 14: 100-119.

9 Dieter U. and Ahrens J., "An Exact Determination of Serial Correlations of Pseudo-Random Numbers", Numerische Mathematik, 1971; 17: 101-123.

10 Downham D. and Roberts F., "Multiplicative Congruential Pseudo-Random Number Generators", Computer Journal, 1967; 10: 74-77.

11 Fishman G., Concepts and Methods in Discrete Event Digital Simulation, Wiley, USA, 1973.

12 Fuller A., "The Period of Pseudo-Random Numbers Generated by Lehmer's Congruential Method", Computer Journal, 1976; 19(2): 173-177.

13 Golder E., "The Spectral Test for the Evaluation of Congruential Pseudo-Random Generators", Applied Statistics, 1976; 25: 173-180.

14 Gorenstein S., "Testing a Random Number Generator", Communications of the ACM, 1967; 10: 111-118.

15 Greenberger M., "An a Priori Determination of Serial Correlation in Computer Generated Random Numbers", Mathematics of Computation, 1961; 15: 383-389.

16 Hardy G. and Wright E., An Introduction to the Theory of Numbers, Oxford at the Clarendon Press, 2nd ed., G.B., 1945.

17 Hoaglin D. and King M., "A Remark on Algorithm AS 98: the Spectral Test for the Evaluation of Congruential Pseudo-Random Generators", Applied Statistics, 1978; 27: 375-377.

18 Hopkins T., "A Revised Algorithm for the Spectral Test", Applied Statistics, 1983; 32(3): 328-335.

19 Hull T. and Dobell A., "Mixed Congruential Random Number Generator for Binary Machines", Journal of the ACM, Jan. 1964; 11(1): 31-40.

20 Inoue H., Kumahora H. and Yoshizawa Y., "Random Numbers Generated by Physical Device", Applied Statistics, 1983; 32(2): 115-120.

21 Kendall M. and Babington S., "Randomness and Random Sampling Numbers", Journal of the Royal Statistical Society, 1938; 101: 147-166.

22 Kendall M. and Stuart A., The Advanced Theory of Statistics, 3rd ed., Vol. 2, Charles Griffin, 1960.

23 Kennedy W. and Gentle J., Chapter 6: Random Numbers, Generation, Tests and Applications, Statistical Computing, Marcel Dekker Inc., USA, 1980, 133-264.

24 Knuth D., The Art of Computer Programming, Seminumerical Algorithm, 2nd ed., Vol. 2, Addison-Wesley, USA, 1981.

25 Lehmer D., "Mathematical Methods in Large-Scale Computing Units", Proc. 2nd Symposium on large-scale digital calculating machinery (Harvard)- Annals of the Computation Laboratory of Harvard University, 26: 141-146.

26 Levene M. and Wolfowitz J., "The Covariance Matrix of Runs up and Down", The Annals of Mathematical Statistics, 1944; 15: 58-69.

27 Lewis P. and Orav E., "Notes in Simulation Methodology for Statisticians, Operations Analysts and Engineers", Short Course on Statistical Simulation Methodology, Department of Statistics, University of Birmingham, U.K., 29-30 May, 1985.

28 Lewis T. and Payne W., "Generalized Feedback Shift Register Pseudo-Random Number Algorithm", Journal of the ACM. 1973; 20: 456-468.

29 Lewis T. and Smith B., Computer Principles of Modeling and Simulation. Houghton Mifflin, USA, 1979.

30 Lilliefors H., "On the Kolmogorou-Smirnov Test for Normality with Mean and Variance Unknown", Journal of the American Statistical Association. 1967; 62: 399-402.

31 MacLaren M. and Marsaglia G., "Uniform Random Number Generators", Journal of the ACM. 1965; 12: 83-89.

32 Mann H. and Wald A., "On the Choice of the Number of Class Intervals in the Application of the Chi-Square Test", The Annals of Mathematical Statistics. 1942; 13: 306-317.

33 Marsaglia G., "Random Numbers Fall mainly in the Planes", Proceedings of the National Academy of Sciences. 1968; 61: 25-28.

34 Morgan B., "Generating Uniform Random Variables", Chapter 3, Elements of Simulation. Chapman and Hall, 1984, 51-76.

35 Naylor T., Balintty J., Burdick D. and Chu K., Computer Simulation Techniques. Wiley, USA, 1966.

36 Payne W., "Fortran Tausworthe Pseudo-Random Number Generator", Communications of the ACM. 1970; 13(1): 57.

37 Peskun P., "Theoretical Tests for Choosing the Parameters of the General Mixed Linear Congruential Pseudo-Random Number Generator", Journal of Statistical Computation and Simulation. 1980; 11(3&4): 281-305.

38 Rand Corporation, A Million Random Digits with 100,000 Normal Deviates. Free Press: Glencoe, ILL, USA, 1955.

39 Ripley B., "Computer Generation of Random Variables", International Statistics Review. 1983; 51: 301-319.

40 Rubinstein R., Simulation and the Monte Carlo Method. Wiley, USA, 1981.

41 Stannke W., "Primitive Binary Polynomials", Mathematics of Computation. 1973; 27: 977-980.

42 Tausworthe R., " Random Numbers Generated by Linear Recurrence Modulo Two", Mathematics of Computation. 1965; 19: 201-209.

43 Tocher K., The Art of Simulation. Hodder and Stoughton, G.B., 1963.

44 Tootill J., Robinson W. and Eagle D., "An Asymptotically Random Tausworthe Sequence", Journal of the ACM. 1973; 20: 469-481.

45 Tootill J., Robinson W. and Adams A., "The Runs Up-and-Down Performance of Tausworthe Pseudo-Random Numbers Generators", Journal of the ACM. 1971; 18: 381-399.

46 Von Neumann J., "Various Techniques Used in Connection with Random Digits", Applied Mathematics, Journal of Research of the National Bureau of Standards, Serie 12, 1951, 36-38.

47 Western A. and Miller J., "Tables of Indices and Primitive Roots", Royal Society Mathematical Tables, Vol. 9, University Press Cambridge, 1968.

48 Whittlesey J., "A Comparison of the Correlation Behaviour of Random Numbers Generators for the IBM-360", Communications of the ACM, 1968; 11: 641-644.

49 Wichman B. and Hill I., "An Efficient and Portable Pseudo-Random Number Generator", Applied Statistics, 1982; 31: 188-190.

50 Wichman B. and Hill I., "Correction: Algorithm AS183", Applied Statistics, 1984; 33(1): 123.

IV) Variance Reduction Techniques

4.1) Introduction

In a simulation experiment samples of many thousands of observations are often required to estimate some performance measure with sufficient accuracy. For large and complex simulations the requirement of sufficient accuracy may be very time-consuming. This accuracy can be measured in terms of the variance of the parameter to be estimated. This variance decreases as we increase the number of simulation runs, so that if σ^2 is the estimated variance for one run, then σ^2/n is the estimated variance for n independent runs.

Variance reduction techniques (VRT) are statistical procedures to reduce the estimated variance without the requirement to increase the number of simulation runs, n. In this chapter, we are going to describe some of the most commonly known VRT: Antithetic Variables (AV), Common Random Numbers (CRN) and Control Variables (CV). A full account of these techniques can be seen in Hammersley & Handscomb [12], James [18], Kleijnen [20], Law & Kelton [24] and Wilson [35].

4.2) Antithetic Variates

The antithetic variates (AV) technique tries to create negative correlation between observations, generating one observation from the random number $\{U\}$ and the other observation from its antithetic partner $\{(1-U)\}$. This technique is due to Hammersley and Morton [13] and was initially used in the Monte Carlo estimation of the value of an integral [11] & ([26], pp. 160-189).

Suppose we have a fixed policy and that in run r we use a sequence $(U^{(r)})$ of random numbers. Let the output be $X^{(i)}=T(U^{(i)})$. We want to estimate $E(X^{(i)})$. For example, with two runs our estimator is:

$$\bar{X} = (X^{(1)} + X^{(2)})/2$$

with

$$Var(\bar{X}) = (Var(X^{(1)}) + Var(X^{(2)}) + 2Cov(X^{(1)},X^{(2)}))/4$$

Hence the variance of $\bar{X}$ decreases if $X^{(1)}$ and $X^{(2)}$ are negatively correlated, or equivalently if $Cov(X^{(1)},X^{(2)})$ is negative. The strategy of negative correlation is to generate $X^{(i)}$ as follows:

$$X^{(1)} = T(U^{(1)})$$

and

$$X^{(2)} = T(1 - U^{(1)})$$

The success of AV will depend on how well the simulations are synchronized; this means that when the i-th pseudo random number is u_i in the first simulation, then the

i-th pseudo random number in the second simulation must be $1-u_i$.

To achieve a good synchronization, it is recommended to use those methods that generate random variables with exactly one pseudo-random number per sample, such as the inverse transform method (section 2.2.1) and the alias method (section 2.2.4). When there are several random events in the simulation model, the use of one pseudo-random number generator for every one of these events is recommended (see Kleijnen [20] and Zeigler [39]).

When more than two runs are considered, let us say R runs, the strategy adopted is to generate independent responses as follows:

$$Y^{(1)} = (X^{(1)} + X^{(2)})/2$$
$$Y^{(2)} = (X^{(3)} + X^{(4)})/2$$
$$\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Y^{(n)} = (X^{(R-1)} + X^{(R)})/2 \quad ; n = R/2$$

where the pair of X's within the Y's are

$$X^{(j)} = T(U^{(j)})$$

and

$$X^{(j+1)} = T(1 - U^{(j)})$$

Then the estimator of E(X) is

$$\bar{Y} = (Y^{(1)} + \ldots + Y^{(n)})/n$$

which is equal to

$$\bar{X} = (X^{(1)} + \ldots + X^{(R)})/R$$

The sample variance of $\bar{Y}$ is

$$s^2(\bar{Y}) = \Sigma_{i=1}^{n} (Y_i - \bar{Y})^2 / n(n-1)$$

which is an unbiased estimator of

$$\begin{aligned} Var(\bar{X}) = [&Var(X^{(1)}) + \ldots + Var(X^{(R)}) \\ &+ 2Cov(X^{(1)}, X^{(2)}) + \ldots + \\ &+ 2Cov(X^{(R-1)}, X^{(R)})]/R^2 \end{aligned}$$

where $Cov(X^{(j)}, X^{(j+1)}) < 0$. (See Arvidsen & Johnson [1]).

If the multiplicative congruential generator is applied in the simulation, i.e.,

$$(4.1) \ldots\ldots Z_i = aZ_{i-1} \pmod{m} \quad ; \; i=1,2,\ldots$$

$$u_i = Z_i/m$$

then, the antithetic random number $(1 - u_i)$ can be obtained simply by making the starting value Z_0^* "antithetic", i.e, if we take as starting value

$$Z_0^* = m - Z_0$$

in (4.1). Then, the resulting random numbers, say u^*, are antithetic, i.e., $u_i^* = 1 - u_i$. In this way no additional computing is needed ([20], p. 194). For further information about the theory behind AV techniques see Rubinstein, Samorodnitsky & Shaked [30] and Wilson [36].

4.3) Common Random Numbers

"Common Random Numbers (CRN) is the most widely recommended and used of the simulation variance reduction techniques" [15]. CRN attempts to improve the efficiency of response difference estimation by comparing alternative procedures in the same medium, for example by comparing two policies for reducing the waiting time in a hospital waiting list. Suppose X and Y are response measures for two alternatives policies. Then

$$Var(X-Y) = Var(X) + Var(Y) - 2Cov(X,Y) \quad \ldots\ldots\ldots \quad (4.2)$$

Hence the variance of the estimated difference is decreased if the covariance term in (4.2) can be made positive. Such a positive covariance is created by the use of the same random numbers, if we assume that both systems react to the stochastic input variables in the same direction. Although this is difficult to achieve in real-world experiments, it is possible in simulation [38].

To compare μ_X and μ_Y the expected responses of two policies, after N simulations of each policy, the following quantities are calculated

$$D_i = X_i - Y_i$$

$$\bar{X} = \Sigma_{i=1}^{N} X_i / N$$

$$\bar{Y} = \Sigma_{i=1}^{N} Y_i / N$$

$$\bar{D} = \Sigma_{i=1}^{N} D_i / N$$

and

$$S_D^2 = \Sigma\ (D_i - \bar{D})^2/(N-1)$$

Then, calculate

$$t_c = \frac{\bar{D}}{(S_D^2/N)^{1/2}}$$

and compare it with a t-distribution with N-1 degrees of freedom. If t_c is greater than $t_{\alpha,\ N-1}$, then reject the hypothesis $H_0: \mu_X = \mu_Y$. ([10], p. 31)

If we are interested in comparing the average response of more than two policies in a simulation system the problem becomes more difficult because of the correlation induced by CRN. One way of solving this problem is to generate different sequences of pseudo-random numbers and with each one of the sequences apply all the policies Then, to study the effect of these policies use the following linear model:

$$y_{ij} = \mu + \tau_i + \beta_j + \epsilon_{ij} \quad ; \begin{matrix} i=1,\ldots,p \\ j=1,\ldots,s \end{matrix}$$

where

y_{ij} : the output response for the simulator with policy 'i' and the sequence of pseudo-random numbers 'j'

μ : is an overall effect common to the particular factor or policy under study

τ_i : is the effect of the ith policy

β_j : is the effect due to the jth pseudo-random number sequence

ϵ_{ij} : is a random-error component

It is assumed that $s \geq p$, and ϵ_{ij} is a normal variable with

$$E(\epsilon_{ij}) = 0$$

$$E(\epsilon_{ij}^2) = \sigma_{ii}^2$$

$$E(\epsilon_{ij},\epsilon_{i'j}) = \sigma_{ii'} \quad ; \quad i \neq i'$$

$$E(\epsilon_{ij},\epsilon_{i'j'}) = 0 \quad ; \quad j \neq j'$$

These assumptions imply that observations (y_{ij}) from the same sequences of pseudo-random numbers are correlated and from different sequences are uncorrelated. Furthermore, the variance of the observations associated with policy 'i' is constant across the sequences.

To test $H_0: \tau_1 = \tau_2 = \ldots = \tau_p$ under the above assumptions the following procedure is applied: form a $(p-1)\times 1$ vector of paired differences within each pseudo-random stream, say,

$$\underline{Y}_j = \begin{bmatrix} y_{2j} - y_{1j} \\ y_{3j} - y_{1j} \\ \vdots \\ \vdots \\ y_{pj} - y_{1j} \end{bmatrix} \quad ; \quad j = 1,2,\ldots,s$$

Let

$$\bar{\underline{Y}} = (1/s)\ \Sigma_{j=1}^{s}\ \underline{Y}_j$$

be the sample mean vector of the $(\underline{Y}_j)$ and

$$S = [1/(s-1)] * \Sigma_{j=1}^{s} (\underline{Y}_j - \bar{\underline{Y}})(\underline{Y}_j - \bar{\underline{Y}})'$$

be the sample covariance matrix of the $(\underline{Y}_j)$. Then if H_0 is

true the quantity

$$F_0 = \frac{(s-p+1)s}{(s-1)(p-1)} \bar{Y}' S^{-1} \bar{Y}$$

is distributed as $F_{[\alpha;\ p-1,s-p+1]}$ from an F distribution ([10], p. 321). To test whether H_0 is true, we compare F_0 with the value of $F_{[\alpha;\ p-1,\ s-p+1]}$. Thus, if

$$F_0 > F_{[\alpha\ ;\ p-1,\ s-p+1]}$$

we conclude that H_0 is false with the probability of error α. In other words the mean responses of the policies are not all equal at the α significance level. An application of this procedure in an inventory problem can be seen in Heikes, et al [15].

4.4) Control Variables

The method of Control Variables (CV) attempts, like CRN and AV, to take advantage of correlation between certain random variables to obtain a variance reduction. Regression Analysis [7] is the basic statistical method used to develop this technique. Lavenberg & Welch [23] said that CV is one of the most promising of the VRT's that have been developed.

The procedure to reduce the variance in CV is as follows. Let μ be the unknown quantity to be estimated and let Y be an unbiased estimator of μ, $E(Y)=\mu$, derived from a single simulation. A random variable C is a control variable for Y if its expectation, μ_C, is known and if it is correlated with Y. The control variable C can be used to construct an unbiased estimator for μ which has a smaller variance than the estimator Y. For any constant b,

$$Y(b) = Y - b(C - \mu_C) \quad \ldots\ldots(4.3)$$

is also an unbiased estimator of μ, $E(Y(b)) = \mu$. Now

$$Var(Y(b)) = Var(Y) - 2bCov(Y,C) + b^2Var(C)$$

Hence if

$$2bCov(Y,C) > b^2Var(C)$$

Y(b) has a smaller variance than Y. The value of b which minimizes Var(Y(b)) is easily shown to be

$$\beta = Cov(Y,C)/Var(C)$$

and the resulting variance

$$Var(Y(b)) = (1 - \rho_{YC}^2) * Var(Y)$$

where ρ_{YC} is the correlation coefficient between Y and C. Hence the more correlated C is with the estimator, the greater the reduction in variance.

The model in (4.3) can be extended for more than one control variable. Lavenberg and Welch [23] describe this case by the following model:

$$Y(b_1,....,b_q) = Y + b_1(C_1-\mu_{C1}) + ...+b_q(C_q-\mu_{Cq})$$

where $E(C_i)=\mu_{Ci}$. To find the b_i's which minimize the variance of $Y(b_1,...,b_q)$ the techniques of multiple regression [7] are applied.

When the control variables C_i are generated from the same simulation as that from which the estimator Y was obtained, then the CV's are called 'internal' or 'concomitant' variables. Applications of this type of variable can be seen in Iglehart & Lewis [17], Lavenberg, Moeller and Saver [22], and Wilson & Pritsker [37]. When C_i's come from a second simulation using the same sequence of pseudo-random numbers the control variables are called 'external'; for applications of these types of variables see Burt, Gaver & Perlas [3], Gaver & Shelder [9], and Taaffe & Horn [32].

4.5) Conclusions

As we saw, the method of applying VRT depends on the aims of the simulation model. For example, if we want

- to compare different policies, we use CRN;
- to estimate E(Y), we use AV
- to estimate E(Y) by the relationship of certain variables related to Y, then we use CV.

Therefore, a full understanding of the way the model works is required for proper use of VRT. For a complex model, it is generally impossible to know beforehand how great a variance reduction might be realized, or whether or not the variance will be reduced at all in comparison with straightforward simulation. However, preliminary pilot runs could be made (if affordable) to compare the results of straightforward simulation. Another point is that some VRT themselves will increase computing cost, and this increase must be traded off against the potential gain in statistical efficiency.

REFERENCES

1 Arvidsen N. and Johnson T., "Variance Reduction Through Negative Correlation, A Simulation Study", Journal of Statistical Computation and Simulation, 1982; 15(2&3): 119-127.

2 Bratley P., "Variance Reduction", Chapter 2, A Guide to Simulation, Springer-Verlag, USA, 1983, 42-72.

3 Burt J., Gaver D. and Perlas M., "Simple Stochastic Networks: Some Problems and Procedures", Naval Research Logistics Quarterly, Dec. 1970; 17: 439-459.

4 Cheng R., "The Use of Antithetic Variates in Computer Simulations", Journal of the Operational Research Society, 1982; 33(1): 229-237.

5 Cheng R. and Feast G., "Control Variables with Known Mean and Variance", Journal of the Operational Research Society, 1980; 31: 51-56.

6 Cooley B. and Houck E., "On an Alternative Variance-Reduction Strategy for RSM Simulation Studies", Decision Sciences, The Journal of the American Institute for Decision Sciences, Jan. 1983; 14(1): 134-137.

7 Draper N. and Smith H., Applied Regression Analysis, Wiley, USA, 1966.

8 Fieller E. and Hartley H., "Sampling with Control Variables", Biometrika, 1954; 41: 494-501.

9 Gaver D. and Shedler G., "Control Variable Methods in the Simulation of a Model of a Multiprogrammed Computer System", Naval Research Logistics Quarterly, Dec. 1971; 18(4): 435-450.

10 Graybill F., An Introduction to Linear Statistical Models, Vol. 1, McGraw-Hill, 1961.

11 Halton J., "A Retrospective and Prospective Survey of the Monte Carlo Method", Society for Industrial and Applied Mathematics (S.I.A.M.) Review, 1970; 12: 1-63.

12 Hammersley J. and Handscomb D., "General Principles of the Monte Carlo Method", Chapter 5, Monte Carlo Method, Methven & Co. Ltd., G.B., 1964, 50-75.

13 Hammersley J. and Morton K., "A New Monte Carlo Technique: Antithetic Variates", Cambridge Philosophical Society Proceedings, 1956; 52: 449-475.

14 Hastings K., "Variance Reduction and Non-Normality", Biometrika, 1974; 61: 143-149.

15 Heikes R., Montgomery D. and Rardin R., "Using Common Random Numbers in Simulation Experiments - an Approach to Statistical Analysis", Simulation, Sept. 1976; 27(3): 81-85.

16 Hillier F. and Lieberman G., "Variance Reduction Techniques", Section 14.4.2, Introduction to Operational Research, Holden-Day Inc., USA, 1967, 453-462.

17 Iglehart D. and Lewis P., "Regenerative Simulation with Internal Controls", Journal of the ACM, 1979; 26: 271-282.

18 James B., "Variance Reduction Techniques", Journal of the Operational Research Society, 1985; 36(6): 525-530.

19 Joseph A., "A Criticism of the Monte Carlo Method as Applied to Mathematical Computations", Journal of the Royal Statistical Society, Series A, 1968; 131: 226-228.

20 Kleijnen J., "The Design and Analysis of Experiments", Chapter IV, Statistical Techniques in Simulation, Marcel Dekker Inc., USA, 1974, 105-285.

21 Kobayashi H., "Efficient Statistical Simulation", Chapter 4.9, Modeling and Analysis: An Introduction to System Performance Evaluation Methodology, Addison-Wesley, USA, 1978, 298-305.

22 Lavenberg S., Moeller T. and Saver C., "Concomitant Control Variables Applied to the Regenerative Simulation of Queueing Systems", Operations Research, Jan-Feb. 1979; 27(1): 134-160.

23 Lavenberg S. and Welch P., "A Perspective on the Use of Control Variables to Increase the Efficiency of Monte Carlo Simulations, Management Science, March 1981; 27(3): 322-335.

24 Law A. and Kelton D., "Variance-Reduction Techniques", Chapter 11, Simulation, Modeling and Analysis, McGraw-Hill, USA, 1982, 349-369.

25 Mood A., Graybill F. and Boes D., Introduction to the Theory of Statistics, 3rd ed., McGraw-Hill, 1974.

26 Morgan B., "Variance Reduction and Integral Estimation", Chapter 7, Elements of Simulation, Chapman and Hill, U.K. 1984, 160-189.

27 Moy W., "Variance Reduction", Chapter 10, Computer Simulation Experiments with Models of Economic Systems, Wiley, 1971, 269-289.

28 Rothery P., "The Use of Control Variates in Monte Carlo Estimation of Power", Applied Statistics, 1982; 31(2): 125-129.

29 Rubinstein R., "Variance Reduction Techniques", Chapter 4.3, Simulation and the Monte Carlo Method, Wiley, USA, 1981, 121-157.

30 Rubinstein R., Samorodnitsky.G. and Shaked M., "Antithetic Variates, Multivariate Dependence and Simulation of Stochastic Systems", Management Science, Jan. 1985; 31(1): 66-77.

31 Safizadeh H., "More on Variance-Reduction Strategies for RSM Simulation Studies", Decision Sciences, The Journal of the American Institute for Decision Sciences, Jan. 1983; 14(1): 138-139.

32 Taaffe M. and Horn S., "External Control Variance Reduction for Nonstationary Simulation", 1983 Winter Simulation Conference Proceedings, IEE, 1983, 341-343.

33 Tocher K., "Design of Simulation Experiments", Chapter 13, The Art of Simulation, Hodder and Stoughton Ltd., U.K., 1963, 171-182.

34 Wilson J., "Variance Reduction Techniques", 1982 Winter Simulation Conference Proceedings, IEE, 1982, 605-612.

35 Wilson J., "Variance Reduction: The Current State", Mathematics and Computers in Simulation, Feb. 1983; XXV(1): 55-59.

36 Wilson J., "Antithetic Sampling with Multivariate Inputs", American Journal of Mathematical and Management Sciences, 1983; 3(2): 121-144

37 Wilson J. and Pritsker A., "Variance Reduction in Queueing Simulation using Generalized Concomitant Variables", Journal of Statistical Computation and Simulation, 1984; 19: 129-153.

38 Wright R. and Ramsay T., "On the Effectiveness of Common Random Numbers", Management Science, July 1979; 25(7): 649-656.

39 Zeigler B., "A Cautionary Word about Antithetic Variates", Simulation Newsletter, No. 3, Summer 1979.

V) Health Care Simulations

5.1) Introduction

Simulation has been used in different areas (Industry, Government, Military, Education, Transport,..) to design, analyze and evaluate complex management systems. Because health care is an area with a great number of interacting complicated systems, simulation has been extensively applied in this area.

Health care is understood to be all those services (government, private and cooperative) provided to the population to improve or maintain a good standard in their living conditions, such as: hospital and clinical services, medical research, preventive medicine (family planning, screening programs, vaccinations, nutritional orientation, psycological therapy, community work, sports and recreation programs,...), social security programs (unemployment and child benefits, elderly pensions,..), provisions of schools, housing and employment with adequate salaries, and construction of green areas [9].

Health care simulations have concentrated mainly on solving problems related to admission control, emergency services, allocation and use of resources (doctors, nurses, beds, hospital theatres, ambulances) with limited cost, planning new facilities, developing new methods of organization, and in management training programmes.

One of the main problems in the analysis of health care systems is the great number of decision-makers that have to be considered (patients, doctors, administrators, nurses, radiologists,..), in addition to the fact that maybe not all of them have the same objectives in mind. For example, patients want to receive the best medical attention with the minimum waiting time, but doctors cannot provide immediate attention because usually his resources are limited, and the administrators want to use these resources to the full with the minimum cost.

When a complicated system is simulated the following methodological setbacks may appear:

- difficulties in defining the objectives of the study,
- difficulties in defining the simulation model,
- limited budget to run the computer program,
- the output results can be very difficult to interpret,
- and the final results may have a limited use.

Therefore, a compromise has to be made between the complexity of the model, the computer capacity, how easily the simulation outputs can be interpreted, and the generalization of the simulation results.

Despite the difficulties that could be presented by a simulation study, simulation has been increasingly used in a variety of problems. What has particularly stimulated the interest of health care simulations is the fact that:

- simulation models are easy to describe to non-modeling decision makers without resorting to the complication of analytical models,
- its versatility for the testing of alternative policies as an aid to decision making,
- and its value as a planning mechanism for evaluation of complex delivery system.

In the following sections we review different applications of simulation within the area of health care to describe in more detail what sort of problems has been tackled by the simulation analysts and their experiences to implement the simulation results. We start by describing the previous reviews in this area (section 5.2) and then the recent models published between 1983 and mid 1985.

5.2) Previous Review

One of the earliest problems dealt with by computer simulation research within health care was Bailey's Erlang Telephone model of hospital admission in 1954 [1]. Since then, an increasing number of simulation models have been developed for representing, analysing, and evaluating systems within the area of health care.

Three leading papers have been published about health care simulation: Roberts & England [17], Tunnicliffe [21] and Valinsky [24]. The main difference in these papers, apart from the volume of reviewed references, is the way the simulation model are classified. For example, Valinsky uses the following categories:

- Demand-oriented models (introduction of patients to various health care services)
- Resources-oriented models (involved ancillary services having both direct and indirect patient contact, as well as the simulation of emergency services)
- Design of healh care facilities
- National & Regional staff and facility planning
- Health Care education
- Communicable disease control

For every one of the above categories, a representative sample of health care simulation is selected for detailed treatment.

Tunnicliffe [21], firstly, ranks the health care problems by the number of people covered by the health care services as follows:

- Health care of population
- Health care within particular institution
- Health care of an individual patient

Then, within the above levels, the following categorization is presented:

Population	Institution/ Facility	Patient
i) Resources needed	A) Emergency care	1) Teaching models
ii) Cost implication	B) Size of facility	2) Medical Research
iii) Disease prevention and control	C) Organization of resources	
iv) Location of services	D) Delegation of tasks	
	E) Organization of patients	

Finally, representative health care simulations are described within Institution/Facility level. At the end of this paper, the author made a classification of different health services which can be useful in further health care simulation reviews.

In Roberts & England [17] the health care simulations are described by the relevant problem in the following way: admission/scheduling, appointments, facility size/design, material handling, manpower availibility, staff size, staff duties, cost, inventory, equipment, and transportation or logistics. The authors also show the health care simulations by field, site, computer language and type of publication.

Therefore, we see that there is still no uniform method of classifying health care simulation models. It would be beneficial to start to develop a unique classification that can be continuouly updated and made accessible to any one involved in health care simulation; actually, most of the health care simulation models are spread in different types of journal, such as, Statistical, Operational Research, Simulation, Epidemiology, and Health Services journals.

The recent and most extensive review of health care simulation is the one made by Roberts and England [17]. The authors reviewed 427 references published until 1980 and made a statistical description of these pubications (see table 5.1). They found that the main problems of study were: facilities size/design (23.3%), staff duties (21.7%), admissions/scheduling (13.%) and costs (10.7%). It is important to notice that the study of costs was not the main target of simulation; this result is considered as the principal difference with other areas of simulation, specially in industry.

The field of application was dominated by hospital administration (24.9%), followed by health system planning (13.7%) and primary care (10.7%). The recommendation of the World Health Organization in its program "Health for All by the year 2000" considered primary care to be the crucial point to develop (Declaration Alma-Ata, [10], pp. 21-22). Therefore, if we want to be in line with the international recommendations we should concentrate more of our work in this area.

Other areas which we consider should be tackled with more emphasis are Epidemiology and Public health because of the great number of people which can be benefited. In this area only 5% of publications were reported (see table 5.1). Among the treated problems were:

- Screening programs for early detection of hypertension,
- Alcoholism treatment,
- Mental health,
- Tuberculosis prevalence,
- Community narcotic control,
- Population control studies,
- Schistosomiasis in Egypt,
- Rubella epidemic,
- Influenza model,
- Yellow fever,
- Disaster planning to evaluate community's ability to cope with the after effects of nuclear attacks or any sort of natural or man made disaster.

The site of applications was mainly in hospitals (53.0%), including hospital outpatients. The amount of simulation works in hospitals shows why they are considered the most complex institutions in health care systems. Most of the hospital models were seen as a collection of interactive queueing systems where patients and personnel queue to demand health facilities, and were simulated mainly with the purpose of finding the most efficient policy for managing the distribution of limited resources.

Other results were in terms of the computer languages, where Fortran (26.7%) and GPSS (26.4%) are the leading ones, followed by SIMSCRIPT (9.9%) and DYNAMO (9.2%). Despite the dominance of special purpose simulation languages over the general purpose languages, the authors show that after 1975 there has been a balance between these types of languages, where Fortran and GPSS continued to be the most frequently used languages. We consider that the preference to use one or other language depends basically on the software available, how easy the language is to be learned, and how easily it can be adapted to the particular needs of the simulation experiment.

The main type of publications used for health care simulations were in health care journals (22.7%), conference proceedings (18.2%) and method journals (17.3). An interesting result from this classification was that a great number of papers were unpublished (28.1%), which we consider is significant information lost that could be useful in future research.

Table 5.1

Results presented by Roberts & England [17] from 427 publications of health care simulations made until 1980 by problem, site, field, language and publication.

Problem	%
1) Facility size/design	23.3
2) Staff size	21.7
3) Admissions/scheduling	13.0
4) Cost	10.7
5) Staff duties	8.8
6) Appointments	7.0
7) Manpower availability	5.8
8) Others	10.0

Field	%
1) Hospital administration	24.9
2) Health systems planning	13.7
3) Primary care	10.7
4) Emergency care	7.2
5) Education	6.2
6) Epidemiology/Public health	5.1
7) Radiology	4.4
8) Dentistry	4.2
9) Surgery/Recovery	3.8
10) Blood banks	3.3
11) Others	16.5

Site	%
1) Hospitals	44.0
2) Regional/National	13.0
3) Solo and group practice	11.5
4) Community	9.2
5) Hospital outpatient	8.5
6) Others	13.8

Language	%
1) Fortran	26.7
2) GPSS	26.4
3) SIMSCRIPT	9.9
4) DYNAMO	9.2
5) Not specified	5.1
6) Others	22.7

Publication	%
1) Health care Journal	22.7
2) Conference Proceedings	18.2
3) Method Journal	17.3
4) Unpublished papers	28.1
5) Others	13.7

The main conclusions presented by Robert & England [17] concerning the future of simulation in health care were as follows:

1) Simulation will become institutionalized in hospitals and used routinely as a means of operational analysis.
2) Simulation will be depended upon increasingly as a means of health planning.
3) By expanding and extending simulation to a broader range of problems in health care, simulation will become a more accepted methodology for problem solving.
4) Simulation in health care will follow applications elsewhere by being used more frequently with analytical methods and be examined more thoroughly by statistical analysis of output.

Valinsky [24] presents his conclusions in terms of the problems faced by the health care simulation researchers in the simulation process as follows:

i) Understanding the system.- Only a few of the many simulation studies yield evidence of having had close cooperation between the researchers and decision-makers, therefore, there were difficulties in understanding the system.

ii) Choice of relevant variables.- There were no rules governing the determination of what and how many variables are to be included; thus the formulating process may be regarded as much as an art as a science.

iii) Design of computer program.- Although many computer languages are available to meet a variety of program needs, the choice of a specific language to express the simulation model was frequently determined by what languages the analyst knows and is most comfortable with.

iv) Gathering adequate information.- It was found that many investigators were concerned with the time, cost and difficulty involved in obtaining accurate and sufficient data.

v) Implementation problem.- Despite the increasing number of publications, there was only a minimal acceptance and use made of the simulation results.

We consider that the most critical problem to overcome in health care simulations is the one of implementation. This is because, despite the fact that simulation analysts could provide technical support in the management process, the final decisions are not taken by them. What's required is greater collaboration between analysts and decision makers.

Tunnicliffe [22], from his review of 200 health care simulations, where he found that only 16 of them were implemented, made the following recommendations for a successful implementation:

- Education of health service personnel in the way their system works, encouraging them to take a broader view of health care and making them aware of O.R. techniques available to them
- Tackling the problems which have been carefully selected so that implementation is most likely to follow.

In the following section we describe in more detail the latest publications of health care simulation (1983 to mid 1985) in order that we can see what have been the recent experiences in the implementation of the simulation results.

5.3) Recent Publications

5.3.1) Introduction

In this section, we describe the recent works in the area of health care simulations found in 25 different periodical journals related to epidemiology, publich health, health services, medicine, statistics, operational research and simulation, published from 1983 to mid 1985 and available in English libraries. The publications found are classified according to the problem tackled.

5.3.2) Publications

- Bed Usage -

One of the most important resources in a hospital are beds. The reason is that most of the resources in the hospital (doctors, nurses, laboratory technicians, administrators, operating theatres, blood banks, cooks, cleaners,..) have to be available for the bed occupancy. Therefore, bed planning is a difficult task because of the great number of decision-makers that need to be considered. However, simulation has been shown to be a useful tool for this sort of planning.

One of the health simulation analysts dealing with bed planning is Dumas ([4] & [5]). He simulates a model, written in SIMSCRIPT II.5, that represents the effect of different bed allocations in a New York hospital. This hospital routinely uses three related measures of effectiveness in assesing bed utilization: patient days,

average daily census, and percent occupancy. During the simulation process a new measure was found, the misplaced patients (patients which are not allocated to the right bed, in terms of their sex and medical needs). Using this measure a new bed configuration was produced in the hospital, involving substantial reallocation of beds among the existing services. In this simulation work a close participation is mentioned with the hospital personnel since the start of the simulation.

- Emergency Services -

Another important area within the health care is the emergency system. A critical component of this system is a responsive and well-managed ambulance service. Uyeno & Seeberg [23] construct a simulation model, written in GPSS V, to balance the distribution of this service in an area of 937 square miles in Vancouver, Canada. Their objective was to determine the best place to locate six ambulances - two paramedic (not handling transfer calls) and four regulars- to produce the minimum response time for the following types of calls: paramedic, emergency, regular and transfer. From the simulation results four strategic points were found with minimum response time; three ambulances were allocated to the most heavily congested point, and one ambulance each to the rest of the points. When these results were implemented, the hospital manager found that even with a substantial increase in demand in the order of 40% to 50%, the system would be able to handle the demand without deterioration in service. This paper also mentions good cooperation with the

management personnel in the whole simulation process.

Another paper which is related to the hospital emergency department is the one presented by Carroll and Oi-Lu [2]. The authors simulate the total service time in an emergency department generated by three types of patients (critical, urgent, and others) in order to determine under four different conditions what personnel would be required in the department in three different time schedules (7:00-15:00, 15:00-23:00, 23:00-7:00). Using a regression equation, the total service per type of patient is predicted by age, arrival time, hospitalization, and use of haematology, microbiology and X-ray departments. The simulation model was written in GPSS and the outputs were analyzed with the statistical package SPSS. One of the simulation results show that to cope with a fifty percent service extension in the department, another examination room would be required, the number of nurses should be increased by 50% in the peak hours (7-23 hrs), and no increment in doctors would be needed. The consistency of this simulation result will depend on how valid the regression model is under different conditions.

- Scheduling Surgical Suite -

A surgical suite consists of a set of rooms equipped to handle many types of surgery, such as: orthopaedic, ophthamology, cardiology, and cosmetic surgery. The rooms are booked in advance and not all have the same equipment. One of the problems for scheduling these sorts of rooms is their low occupancy rate which is mainly attributed to the variability in the surgery time.

Jones, Sahney and Kutuglu [13] design an interactive model, written in Slam II, to tackle the above problem in a surgical suite with 8 rooms to handle 12 types of surgery. Five policies were simulated and the main results show that a 90% occupancy is a plausible goal, which we consider a substantial gain after the authors have reported an average of 60% in the modern hospitals. The question is to see whether this simulation result can be extended to any surgical suite with minimum modifications or, can it only be used in this particular surgical unit.

- Allocation of Human Resources -

Simulation has also been used in the allocation of human resources. One of the papers relevant to this area is the one by Nelson & Ravindra [15]. They simulate a queueing model, written in SLAM, to evaluate different staffing configurations in a screening program to minimize the size of the staff while still meeting the guidelines for the tasks assigments. The objective of the screening program was to reduce the risk of cardiovascular disease among United

States Air Forces personnel. To show a pictorial representation of the simulation results, the computer languages GANS (Graphical Analysis of Network Structure) was used. The simulation results demonstrated that the medical technicians involved in the screening program should work in parallel rather than in series (see section 6.7). This paper does not stated how successful the implementation of this recomendation was.

- Ancillary Workloads -

Once a patient has been hospitalized several services from different departments will be needed for his treatment. Some of these department are: Bacteriology/Microbiology Lab, Biochemistry Lab, Immunology Lab, Haematology Lab, Nuclear Medicine, Radiology, Physical Therapy, Blood Bank. All the services produced by these departments are identified as the Ancillary Workloads.

The demand for ancillary workloads is also a very important characteristic in hospital planning. Hancock and Walter [11] present a simulation model to determine the number of procedures that would be performed in nineteen ancillary departments on a day of the week basis for the year 1990 based on information of 1976.. The authors assumed that the length of stay (LOS) will be less than in 1976, and that the number of services required per patient will be the same. To simulate the demand for the ancillary services, the Admission Scheduling and Control System Simulator (ASCSS) was used. The model that determines the flow into, out of ,

and between each service was written in PL/1 and assembly language. The simulation results demonstrated that if it is required that LOS be reduced, the same intensive services provided during the week (Monday-Friday) have to be also for the the weekends, which for this particular simulation meant that the staff in weekends should be nearly 75% of the staff during the week. The question is to find how similar will the conditions in the hospital be in 1990 in relation to 1976 to validate this simulation results.

- Medical Treatment -

Simulation also has been useful in the analysis of medical care decisions. The simulation language, written in Fortran SLN (Simulation of Logical Networks) was designed specially to facilitate the modelling of these decisions.

Roberts and Klein [18] describe the applications of SLN in the evaluation of the actual therapeutic protocols for End-Stage-Renal disease, Chronic Stable Angina, Renal Artery Stenosis, Hypertension, and Hypercholesterolomia. The authors, based on the simulation results, mention several recommendations for the treatment of each disease, in terms of survival and cost benefits, but do not say what has been the impact of these recommendations.

Another simulation model, written in Pascal, used to determine the best medical treatment in terms of resources effect was designed by Davis [3] for the Wessex Renal Unit in England. Patients with kidney failure can receive one of the following long-term treatments:

- Haemodialysis,
- Transplant Surgery,
- Continuous Ambulatory Peritoneal Dialysis (CAPD).

Patients start their treatment with either Haemodialysis or CAPD. The author compares the resources effect when patients start with CAPD (policy A) instead of Haemodialysis (policy B). He shows that the policies are different in terms of dialysis machine usage (policy B higher) but not differences in bed occupancy. It would be interesting to combine these simulation results with the benefits that the patients would receive in each policy.

- Planning a OBS/GYN Clinic -

Planning an Obstetric/Gynaecology clinic is a difficult task because of the irregular demand, the immediate medical attention required, and the short average length of stay. A model, written in GESIM (General Simulation) which is a version of GPSS, to deal with this sort of problem was designed by Mahachek and Knabe [14]. The authors's objective was to determine the effect of a plan to combine physically (and organizationally) OBS and GYN outpatient clinics. Although the simulation results show

that under the present space conditions (examination and waiting rooms) the clinics could not be combined, there was such a substantial gain in understanding the system by the management personnel that an extraordinarily detailed space requirement was generated, and time allowed for further detailed simulation.

- Birth Control Policies -

A model for testing the implication of different control policies in the growth of the population in India up to the year 2000 was designed by Patil, Janahanlal and Ghista [16]. Approximately 500 equations were used to represent this model. These equations are solved by using the program CSMP (Continuous System Modeling Program) on an IBM 370/155. By combining low birth rates (3% between 1975-1980, 4% between 1980-1990, and 2.73% between 1990-2001), increase in marriage age for women from 15 to 20, and the quality of life (indicated by nutritional level and per capita income) the authors show different alternatives in the population growth. The problem in this type of simulation is to show how valid the 500 equations used in this model will be for the year 2000.

- Management Training -

In the training of management personnel involved in the area of health care, the first step to be considered is the proper understanding of the functioning of the systems within this area. When we refer to health care systems at National or Regional levels, the complexity to understand these system increses, because of the number of decision-makers involved and the number of interactions to be considered within these systems [20]. Despite this possible complexity, simulation has been successfully used for describing these types of systems.

The Centre for Health Services Management at Leicester Polytechnic in England has been developing a series of health care simulation models over the last ten years. The latest one, which was written in Fortran, designed by Waller, Burdell and Brough [25], is used as an aid to management training and the development of planning skills in the National Health Service (NHS) in Great Britain. This model is used as part of a management decision exercise that simulates the operation planning process of different district management teams. This exercise has been incorporated into the training programmes of short and long duration for all senior levels and categories of staff in the Health Services.

The main purposes of the above model is to provide greater insight into the working of the NHS planning system and into the way in which managerial planning decisions

affect the use of staff, money and other resources. An additional purpose is to show the part computer simulation can play in aiding the management decision-making process. As an example of this model, the authors presented the simulation results in terms of cost effects in a specific health district with five specialities under different conditions (waiting list, length of stay, bed occupancy,..). In general we found this model very useful, not just for its educational qualities, but also for the level of applicability (National, Regional and district levels). It would be interesting to determine how valid this model would be in another country with different conditions, such as México, where the health services are run, basically, by three government enterprises, in addition to private practice of medicine [12].

5.3.3) Conclusions

We see from this review how simulation has continued to be applied in a variety of fields within the health care services, for example, bed usage, emergency services, scheduling, allocation of resources, and medical treatments. Also, we notice that these applications were made for different purposes, such as: solving specific problems, developing new management methods, and for management training. However, we found some implementation

problems: only 3 out of the 11 reviewed references reported successful implementation (Duma [4] & [5], Uyeno & Seeberg [23], and Waller et al [25]). Two basic characteristics were found in those simulations with successful implementation:

- the simulation analysts were part of the institution where the simulation was taking place
- the management personnel were involved in the simulation process from its starting stages.

Therefore, we consider that a program to promote the potential benefits of health care simulations should be developed (conferences, seminars, or optional academic lectures to students in areas related to Health problems) in order that the simulation analysts can be accepted as permanent staff of the health care institutions, and the management personnel of these institutions can see the simulation techniques as a plausible alternative for planning, training, and solving problems within the health care services.

REFERENCES

1 Bailey N., "Queueing for Medical care", Applied Statistics, 1954; 3: 137-145.

2 Carroll J. and Wu O., "Simulation of Urban Hospital Emergency Room Activities", Simulation Series, the Society for Computer Simulation [SCS], Jan. 1983; 11(2): 20-24.

3 Davis R., "An Interactive Simulation in the Health Service", Journal of the Operational Research Society, July 1985; 36(7): 597-606.

4 Dumas B., "Hospital Bed Utilization: An Implemented Simulation Approach to Adjusting and Maintaining Appropriate Levels", Health Services Research, April 1985; 20(1): 43-61.

5 Dumas B., "Simulation Modeling for Hospital Bed Planning", Simulation, Aug. 1984; 43(2): 69-78.

6 England W. and Roberts S., "Applications of Computer Simulaton in Health Care", 1978 Winter Simulation Conference Proceedings, IEE, 1978, 665-677.

7 Fries B., "Bibliography of Operations Research in Health-Care Systems", Operations Research, Sept-Oct. 1976; 24(5): 801-813.

8 Fries B., "Bibliography of Operations Research in Health-Care Systems: An Update", Operations Research, March-April 1979; 27(2): 408-419.

9 Ghista D., "Towards a Better Health Care System", Modeling and Simulation, 1983; 14(3): 1049-1050.

10 Halfdan M., "The Meaning of Health for All by the year 2000", World Health Forum. WHO, 1981; 2(1): 5-22.

11 Hancock W. and Walter P., "The Use of Admissions Simulation to Stabilize Ancillary Workloads", Simulation. Aug. 1984; 43(2): 88-94.

12 Lopez-Acuña D., "Health Services in México", Journal of Public Health Policy. March 1980: 83-95.

13 Jones A., Sahney V. and Kurtoglu A., "A Discrete Event Simulation for the Management of Surgical Suite Scheduling", 16th Annual Simulation Symposium. IEEE, 1983, 263-278.

14 Manachek A. and Knabe T., "Computer Simulation of Patient Flow in Obstetrical/Gynecology Clinics", Simulation. Aug. 1984; 43(2): 95-101.

15 Nelson B. and Ravindran A., "A Hybrid Graphical-Simulation Analysis of a Health System Application", Simulation. May 1985; 44(5): 219-224.

16 Patil M., Janahanlal P. and Ghista D., "Mathematical Simulation of Impact of Birth Control Policies on Indian Population System", Simulation. Sept. 1983; 41(3): 103-117.

17 Roberts S. and England W., "Survey of the Application of Simulation to Health Care", Simulation Series. The Society for Computer Simulation [SCS], 1981; 10(1).

18 Roberts S. and Klein R., "Simulation of Medical Decisions: Application of SLN", Simulation. Nov. 1984; 43(5): 234-241.

19 Swain R., Health Systems Analysis. Grid Publishing Inc., Ohio, 1981.

20 Tilquin C., "Modeling Health Services Systems", Medical Care. March 1976; 14(3): 223-240.

21 Tunniclife W., "A Review of Operational problems tackled by Computer Simulation in Health Care Facilities", Health and Social Services Journal. July 1980: 873-80.

22 Tunniclife W., "Implementation of Computer Simulation Projects in Health Care", Journal of the Operational Research Society. 1981; 32(2): 825-832.

23 Uyeno D. and Seeberg C., "A Practical Methodology for Ambulance Location", Simulation. Aug. 1984; 43(2): 79-87.

24 Valinsky D., Simulation, Chapter 8, Operational Research in Health Care, The Johns Hopkins University Press, Baltimore & London, 1975, 114-176.

25 Waller A., Burdett F. and Brough R., "An Interactive Stochastic Simulation Model of a Health Service District", Journal of the Operational Research Society. 1984; 35(4): 297-302.

VI) Queueing Theory

6.1) Introduction

A queueing system can be described as customers arriving for service, waiting for service (if the service is not immediately available) and leaving the system after being served. Such a basic system can be schematically shown as in figure 6.1. Although any queueing system may be diagrammed in this manner, it should be clear that a reasonably accurate representation of such a system would require a detailed characterization of the underlying process ([10] , [15]).

Figure 6.1 Schematic diagram of a Queueing process.

The term 'customer' is used in a general sense and does not imply necessarily a human customer. For example, a customer could be an aeroplane waiting to take off, or a computer program waiting to be run on a time-sharing basis.

A queueing process is defined by the following six characteristics:

i) Arrivals pattern of customers

ii) Service pattern of servers

iii) Queue discipline

iv) System capacity

v) Number of service channels

vi) Number of service stages

In most cases these characteristics provide an adequate description of a queueing system.

6.2) Arrivals Pattern of Customers

The arrivals pattern or input to a queueing system is often measured in terms of the average number of arrivals per unit of time -mean arrival rate- or by the average time between successive arrivals -mean interarrivals time-. If there is uncertainty in the arrivals pattern (often referred to as random, probabilistic or stochastic), then these mean values only provide measures of the central tendency for the input process and further characterization is required in the form of the probability distribution associated with this random process.

It is also neccesary to know the reaction of a customer upon entering the system. A customer may decide to wait regardless of how long the queue becomes, or if the queue is too long to suit him may decide not to enter it. If a customer decides not to enter the queue upon arrival, he is said to have 'balked'. On the other hand, a customer may enter the queue, but after a time lose patience and decide to leave. In this case he is said to have 'reneged'. In the event that there are two or more parallel waiting lines,

customers may switch from one to another, that is 'jockey' for position. These three situations are all examples of queues with 'impatient customers'.

Another factor of interest concerning the input process is the possibility that arrivals come in batches instead of one at a time. In the event that more than one arrival can enter the system simultaneously the input is said to occur in 'bulk' or 'batches'. In the bulk-arrival situation, not only may the time between succesive arrivals of the batches be probabilistic but also the number of customers in a batch (the batch size) [6].

One final factor to be considered in the arrivals pattern is how time affects this pattern. An arrivals pattern that does not change with time is called a 'stationary' arrivals pattern. One that is not time-independent in the sense described above is called 'non-stationary'.

The easiest arrival pattern to treat mathematically is 'purely random', i.e., the probability of an arrival in an interval (t,t+dt) is λdt, where λ is a constant. This arrival pattern is described by the Poisson distribution, whose p.d.f. is

$$f_N(n) = (\lambda t)^n e^{-\lambda t}/n! \quad ; \quad n=0,1,2,...$$

$f_N(n)$ gives the probability of there being n arrivals during a time interval of length t, where λt is the average number

of arrivals in t. If t is considered to be unit time, then the distribution becomes

$$f_N(n) = \lambda^n e^{-\lambda}/n! \qquad ; \quad n=0,1,2,...$$

where λ is the average number of arrivals/unit time. When the arrival pattern is a Poisson distribution, it has been shown that the probability density fuction (p.d.f) of the interval between an arrival and the next subsequent arrival is the exponential distribution ($f_X(x)=\lambda e^{-\lambda x}$; $x \geq 0$). ([10], p. 7-10).

6.3) Service Pattern of Servers

Many of the characteristics of the arrivals pattern can be used for the service pattern. For example, service patterns can also be described by a rate (number of customers served per unit of time) or as a time (time required to service a customer). However, one important difference exists between service and arrivals. When one speaks about service rate or service time, these terms are conditional on the fact that the system is not empty; that is, there is someone in the system requiring service. If the system is empty, the service facility is idle. Service may also be single or batch. In the single case one server gives service to one customer and in the batch case one server simultaneously gives service to several customers (for example, people boarding a train, or a computer with parallel processing). Applications of bulk service can be seen in Bailey [3], Chaudhry [6] and Downton [11].

A great mathematical simplification of results can be found when the p.d.f. of service-time is given by the exponential curve

$$h_T(t) = \mu e^{-\mu t} \quad ; \quad t \geq 0$$

Now, if the random variable T is a service-time,

$$Pr(T \geq to) = \int_{to}^{\infty} \mu e^{-\mu t} dt = e^{-\mu to}$$

From this result we can calculate the probability that the service of a customer is completed in an interval dt, given that the service has been in progress for a time to. This is the conditional probability

$$Pr(to \leq T \leq to+dt \mid T \geq to) = \frac{Pr(to \leq T \leq to+dt)}{Pr(T \geq to)}$$

$$= e^{-\mu to} \times (1-e^{-\mu dt}) / e^{-\mu to}$$

$$= (1 - e^{-\mu dt})$$

$$\cong \mu dt$$

In other words the probability that the service is completed in a small element of time is constant, independent of how long service has been in progress; this is called the memoryless or Markovian property of the exponential distribution. Thus service can be treated as if it were a completely random operation ([10], p. 20).

6.4) Queue Discipline

Queue discipline refers to the manner in which customers are selected for service when a queue has formed. This decision may be based on any or all of the following:

a) Some measure related to the relative arrival for those customers in the queue.

b) Some measure (exact value, estimate, pdf) of the service to be received or so far received.

c) Some function of group membership

Examples of queueing disciplines that depend only upon the arrival time are: First-Come-First-Served (FCFS), Last-Come-First-Served (LCFS), and service in random order (SIRO). Discrimination on the basis of service time only may take the following form: Shortest-Job-First (SJF), Longest-Job-First (LJF), similar rules based on average and so on. Order of service may be based on an externally imposed priority class structure; when this is the case the queueing discipline is called 'Priority Queueing Discipline'.

An example of priority queueing discipline is when the customers are assigned some sort of initial score on arrival to the system, then their respective scores are incremented in relation to the amount of time that they have spent in the system, with selection based on the highest cumulated scores. When this is the case, the discipline is called 'Dynamic Priority Discipline'. This type of discipline was introduced by Jackson [16]. A formal development of this discipline can be seen in Kleinrock [22].

Another type of priority queueing discipline is the alternating-priority (zero-switch) queueing discipline. Under this discipline, assuming a queue with two classes of customers, the server attends one class until no one is present in that class, at which point he switches instantaneously to the other class and serves members of that class until no one is present in that class, and so forth ([33] and [34]).

When some sort of priority discipline is used in the system and more than one customer can be found in one type of priority, it has also to be specified what sort of policy will be followed within each priority.

There are two general situations in priority queueing disciplines. In the first, which is called 'pre-emptive', the customer with the highest priority is allowed to enter the service immediately, even if a customer with lower priority is already in service when the higher priority customer enters the system [14]. In the second general priority situation, called the 'non-preemptive' case, the higher priority customer goes to the head of the queue but cannot get into service until the customer presently in service is dealt with, even though this customer has a lower priority.

6.5) System Capacity

In some queueing processes there is a physical limitation to the amount of waiting room, so that when the line reaches a certain length, no further customers are allowed to enter the system until space becomes available by a service completion. These are referred to as finite queueing situations; that is, there is a finite limit to the maximum queue size.

6.6) Number of Service Channels

The number of service channels refers to the number of parallel service stations which can service customers simultaneously. Figure 6.1 depicts a single channel system, while figure 6.2 shows two variations of a multichannel system; the first variation (a) has only one queue for all the channels and the second one (b) has one queue for every channel.

Figure 6.2 Multichannel Queueing System.

6.7) Stage of Service

A queueing system could have a single stage of service or it may have several stages (multistage service). An example of a multistage service queueing discipline would be a physical examination procedure, where each patient must proceed through several stages (see figure 6.3) such as medical history; ear, nose and throat examination; blood test; electrocardiogram; eye examination; and so on.

Figure 6.3 Multistage Queueing System

6.8) Mathematical Approach

When we construct a mathematical model of a system, the underlying motivation is to evaluate some measure of performance. One of the relevant measures in a queueing system will be a measure of congestion. The simplest measure of congestion is the 'traffic intensity' which is represented by the simbol ρ. The value of ρ indicates the mean fraction of the time the queueing system is used. In the special case of a single-server queue the traffic intensity is calculated by

$$\rho = \frac{1/\mu}{1/\lambda} = \frac{\lambda}{\mu}$$

$1/\mu$: Mean service time of a single customer

$1/\lambda$; Mean interval between successive individual customers

For a queue system with M servers in paralell (see figure 6.2.a), the traffic intensity is

$$\rho = \frac{\lambda}{\mu * M}$$

The unit of traffic intensity is sometimes called erlang, out of deference to A. K. Erlang, for his pioneering work in queueing theory ([10], p. 40)

When the traffic intensity is greater than one, $\rho > 1$, customers are arriving faster than the servers can handle, therefore an increasing queue will develop.

If $\rho < 1$ then, on average, the servers are able to deal with more than one customer's requirements before the next customer arrives and we should therefore expect the servers to cope satisfactorily with their task.

Now, the problem is to find what sort of behaviour we would expect when $\rho = 1$. Under this condition, on average the time between successive customers ($1/\lambda$) is equal to the average service time ($1/\mu$); in other words, on average when one customer arrives another is being served. Therefore, we expect that the mean number of customers in the queue should be zero. However, to show a formal demostration of this is quite difficult, as we will see in the next section.

6.8.1) The single Server Queue

Assume Poisson arrivals, with mean λ per unit of time, and exponentially distributed service time with mean $1/\mu$. Let 'j' be the number of people in the system, and $(q_j(t))$, $j=0,1,2\ldots$ be the probability distribution of the number of customers in the system (i.e., in the queue or being served) at time t, i.e.,

$$q_j(t) \equiv \Pr(j \text{ in the system at time } t)$$

Then $(q_j(t+dt))$ and $(q_j(t))$ are related in the following manner:

$$\Pr(j=0 \text{ at } t+dt) = \Pr(j=0 \text{ at } t)\times\Pr(\text{no arrival in } dt) + \Pr(j=1 \text{ at } t)\times\Pr(\text{a service is completed in } dt)$$

from sections 6.2 and 6.3, this probability can be written as

$$q_0(t+dt) = q_0(t)(1-\lambda dt) + q_1(t)\mu dt$$

So,

$$\lim_{dt\to 0} \frac{q_0(t+dt) - q_0(t)}{dt} = \mu q_1(t) - \lambda q_0(t) = q_0'(t)$$

When $\rho=\lambda/\mu < 1$ and $t\to\infty$, then $q_0'(t)\to 0$, $q_0(t)\to q_0$, $q_1(t)\to q_1$. Hence

$$\mu q_1 - \lambda q_0 = 0$$

Similarly, when j=1,2,..., it can be shown that

$$\begin{aligned} \mu q_{j+1} - \lambda q_j &= \mu q_j - \lambda q_{j-1} \\ &= \mu q_{j-1} - \lambda q_{j-2} \\ &\quad \cdots\cdots\cdots \\ &= \mu q_1 - \lambda q_0 \\ &= 0 \end{aligned}$$

So,

$$\begin{aligned} q_j &= (\lambda/\mu) q_{j-1} \\ &= \rho * q_{j-1} \\ &= \rho^j * q_0 \qquad ; \quad j=0,1,2,\ldots \end{aligned}$$

Now as

$$\Sigma_{j=0}^{\infty} q_j = q_0 \Sigma_0^{\infty} \rho^j = 1$$

and

$$\Sigma_0^{\infty} \rho^j = 1/(1-\rho)$$

then

$$q_0 = 1 - \rho$$

which is the probability that the server is idle, and

$$q_j = \rho^j * (1 - \rho) \quad \ldots\ldots\ldots (6.1)$$

which is the probability of having j customers in the system, if and only if $\rho < 1$. In general, for the analysis of any queueing systems, $\rho < 1$ is the main condition to use the Queueing theory concepts. If this condition is not satisfied, then simulation is used as an alternative procedure [15].

When $\rho < 1$ some basic results can be presented for the particular case of a single server. For example,

(i) Expected number in the system

$$E(\text{number in the system}) = \Sigma_0^{\infty} j*q_j$$
$$= (1-\rho)\Sigma_j j*\rho^j$$
$$= \rho/(1-\rho)$$

(ii) Expected number being served

$$0*q_0 + 1*(1-q_0) = \rho$$

(iii) Expected number in the queue

$$Q = \rho/(1-\rho) - \rho = \rho^2/(1-\rho)$$

(iv) Expected queueing time

$$D = Q/\lambda = \rho/(\mu-\lambda)$$

(v) Expected Waiting Time

$$E(\text{waiting time}) = D + \text{service time}$$
$$= \rho/(\mu-\lambda) + 1/\mu$$
$$= 1/(\mu-\lambda)$$

The distribution of eq. (6.1) is a standard distribution in probability theory, called the Geometric distribution. Its mean is $\rho/(1-\rho)$ and its variance is $\rho/(1-\rho)^2$. The probability of finding more than n customers in the queue is

$$\Pr[N > n] = q_{n+1} + q_{n+2} + q_{n+3} + \ldots\ldots$$
$$= (1-\rho)\rho^{n+1}\Sigma_{i=0}^{\infty}\rho^i$$
$$= (1-\rho)\rho^{n+1}*[1/(1-\rho)]$$
$$= \rho^{n+1}$$

Thus with low values of traffic intensity long queues are extremely unlikely. Table 6.1 shows dependence of various characteristics of the queue upon the traffic intensity. It will be noticed that the reduction of the mean queue size, and of the probability of very long queues, necessarily raises the proportion of time which the server spends idle. This illustrates a general feature of queueing systems, namely that finding the most economical arrangement of the system usually involves compromising between ensuring full use of resources and minimizing delays to customers.

Now, a first insight into the behaviour of the queueing system with $\rho=1$ can be seen in table 6.1. From this table, we notice that the expected number in the queue tends to infinity as ρ gets close to one. This behaviour contradicts our previous statement in which we said that the expected number of customers should be zero. Which is the correct one? Assuming the approach based on table 6.1 is correct, is this a realistic situation? Is it possible to allow that customer queue increase in this form, especially in a system like a hospital waiting list where different medical priorities are involved in the list? If so, what sort of control mechanism can be implemented to avoid an infinite increase? What would be the waiting time distribution?

In chapter VII, we simulated a model which represents a hospital waiting list, seen as a queueing system with $\rho=1$, in order to answer the above questions. We consider that simulation was the only alternative for answering these questions. With no medical priorities in the waiting list some of these questions can be answered by using Little's formula; this formula is described in the next section. The results from this formula were quite useful to validate our simulation model.

Table 6.1

Dependence of various characteristics of the queue upon the traffic intensity; single server with random arrivals and exponetial service-times

Traffic intensity ρ	Probability server free $1-\rho$	Expected number in the system $\rho/(1-\rho)$	Probability of more than four customers in queue ρ^5
0.1	0.9	0.111	0.00001
0.2	0.8	0.250	0.0003
0.3	0.7	0.429	0.002
0.4	0.6	0.667	0.010
0.5	0.5	1.000	0.031
0.6	0.4	1.500	0.078
0.7	0.3	2.333	0.168
0.8	0.2	4.000	0.328
0.9	0.1	9.000	0.590
0.99	0.01	99.000	0.951
0.999	0.001	999.000	0.995

* when $\rho \approx 1$, the system is said to be in heavy traffic [19].

6.9) Little's Formula: $L = \lambda \bar{W}$

Probably the simplest and yet the most important formula that is used in queueing analysis is the formula $L=\lambda\bar{W}$, known as Little's formula. This formula states:

Time-average number of units in the system (L)	=	Arrival rate (λ)	×	Average delay ($\bar{W}$)

The proof of this formula can be seen in Little [25].

Consider a queueing system as shown in fig. 6.4. Let $0 < t_1 < t_2 < \ldots$ be the arrival times of customers to the system numbered in the order of arrival. Let a counting process

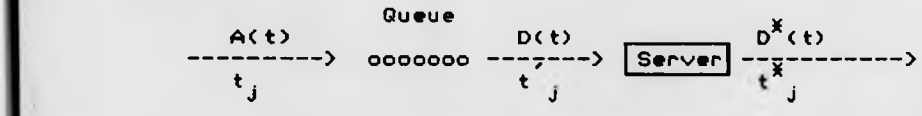


Figure 6.4 A queueing system and its counting process A(t) D(t), and $D^*(t)$

A(t) represents, for each t, the cumulative number of arrivals up to time t:

$$A(t) = \text{number of } t_j \leq t$$

This is a step function that increases by one at each time t_j, as shown in fig. 6.5. If we assume the FCFS queue discipline, then the order in which the customers leave the queue is the same as the order in which they arrive. In

other words, if we denote by t'_j the departure time (from the queue) of customer j, then

$$0 < t'_1 < t'_2 < \ldots.$$

When a customer arrives at the system to find that the server is idle, then the customer enters the service inmediately, i.e., $t_j = t'_j$. We define another counting process

$$D(t) = \text{number of } t'_j \leq t$$

that is, the cumulative number of customers who enter service by time t (fig. 6.5). At any time t, $A(t) - D(t) = Q(t)$ represents the queue length at any time t.

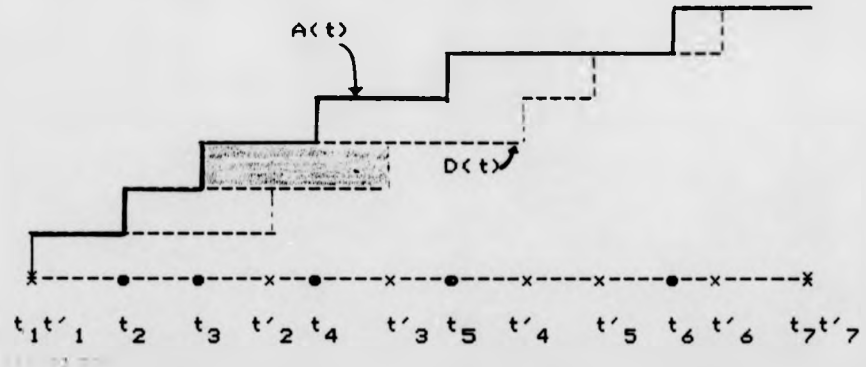


Figure 6.5 Typical behaviour of the counting process A(t) and D(t)

In the case FCFS discipline, $t'_j - t_j$, the time that the jth customer spends in the queue, is equal to the horizontal distance (or equivalently the area of a horizontal strip) between the curves A(t) and D(t); for example, in figure 6.5, the shaded area $t'_3 - t_3$. Now by defining the point of

reference (or time origin) t=0 for which A(t)=D(t) (in other words, the server is idle at t=0), then by choosing another instant τ such that $A(\tau)=D(\tau)$, we define the following variables:

$n(\tau) = A(\tau) - A(0)$ = the total number of arrivals during the period $(0,\tau)$

$\lambda(\tau) = n(\tau)/\tau$ = the mean arrival rate during $(0,\tau)$

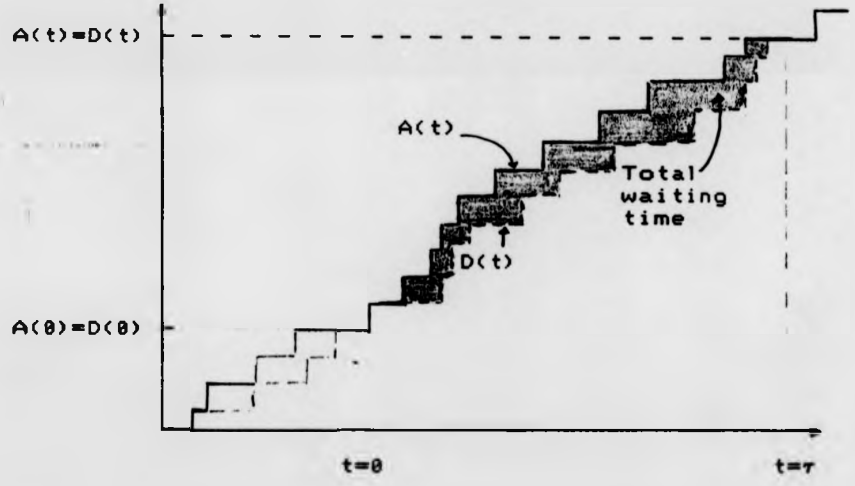


Figure 6.6 The total waiting time

The entire shaded area of fig. 6.6 can be decomposed into $n(\tau)$ separate horizontal strips of the type illustrated in fig. 6.5. The average length of the horizontal strip is

$\bar{W}(\tau) = \Sigma^{n(\tau)}_{j=1} W_j / n(\tau)$ = The mean waiting time per customer in the period $(0,\tau)$

Since the quantity $Q(t) = A(t) - D(t)$ represents the queue size function, its time average,

$$\bar{Q}(\tau) = \int_0^\tau Q(t)\,dt/\tau \quad \ldots\ldots\ldots(6.2),$$

is the mean queue size over the period $(0,\tau)$. Since each strip has the width of unity, the sum of the strip lengths is equal to the sum of the strip areas. Therefore, we have

$$\sum_{j=1}^{n(\tau)} W_j = \int_0^\tau Q(t)\,dt \quad \ldots\ldots\ldots(6.3)$$

From eq.'s (6.2) and (6.3) we have

$$\bar{Q}(\tau) = \sum_{j=1}^{n(\tau)} W_j/\tau \quad \ldots\ldots\ldots(6.4)$$

By substituting the definitions of $\lambda(\tau)$ and $\bar{W}(\tau)$ into the right-hand side of (6.4), we get

$$\bar{Q}(\tau) = \lambda(\tau) * \bar{W}(\tau)$$

If the limits

$$\lambda = \lim_{\tau \to \infty} \lambda(\tau)$$

and

$$\bar{W} = \lim_{\tau \to \infty} \bar{W}(\tau)$$

exist, then a limit for $\bar{Q}(\tau)$ also exists, defined by

$$L = \lim_{\tau \to \infty} \bar{Q}(\tau)$$

and the three limits must satisfy the relationship

$$L = \lambda * \bar{W}$$

The proof of the above results can be seen in Cooper ([9], pp. 178-185). Kobayashi ([24], p. 121-122) shows that Little's formula is also true for any queue discipline. For further information and recent applications of Little's formula see Ramalhoto, Amaral and Cochito [30].

REFERENCES

1 Ackoff R. and Sasieni M., "Queuing Problems", Chapter 10, Fundamentals of Operational Research, Wiley, USA, 1968, 248-274.

2 Aczel M., "The Effect of Introducing Priorities", Operations Research, Sept-Oct. 1960; 8(5): 730-733.

3 Bailey N., "On Queueing Processes with Bulk Service", Journal of the Royal Statistical Society, Series B: 16, 1954, 80-87.

4 Bailey N., "A Note on Equalising the Mean Waiting Time of Succesive Customers in a Finite Queue", Journal of the Royal Statistical Society, Series B: 17, 1955, 262-263.

5 Balachandran K., "Purchasing Priorities in Queues", Management Science, Jan. 1972; 18(5): 319-326.

6 Chaudhry M. and Templeton J., A First Course in Bulk Queues, Wiley, USA, 1983.

7 Cobham A., "Priority Assignment in Waiting Line Problems", Operations Research, 1954; 2: 70-76.

8 Conolly B., Queueing Systems (Lecture Notes), Ellis Horwood Ltd., 1975.

9 Cooper K., Introduction to Queueing Theory, North Holland, 1981.

10 Cox D. and Smith W., Queues, Methuen and Co. Ltd., G.B., 1968.

11 Downton F., "On Limiting Distribution Arising in Bulk Services Queues", Journal of the Royal Statistical Society, Series B: 18, 1956, 265-274.

12 Downton F., " Waiting Time in Bulk Service Queues", Journal of the Royal Statistical Society. Series B: 17, 1955, 256-261.

13 Eilon S., "A Simpler Proof of L = λ * W", Operations Research. 1969; 17: 915-917.

14 Gaver D., "A Waiting Line with Interrupted Service, including Priorities", Journal of the Royal Statistical Society. Series B: 24, 1962, 73-90.

15 Gross D. and Harris C., Fundamentals of Queueing Theory. Wiley, USA, 1974.

16 Jackson J., "Some Problems in Queueing with Dynamic Priorities", Naval Research Logistics Quarterly. Sept. 1960; 7(3): 235-249.

17 Jackson J., "Queues with Dynamic Priority Discipline", Management Science. 1962; 8: 18-34.

18 Jewell W., "A Simple Proof of: L = λ * W", Operations Research. 1967; 15: 1109-1116.

19 Kingman J., "On Queues in Heavy Traffic", Journal of the Royal Statistical Society. Series B: 24, 1962, 383-392.

20 Kingman J., Queueing Theory. "Approximations for Queues in Heavy Traffic", In R. Cruon (ed), London, English University Press, 1967.

21 Kleinrock L., "A Conservation Law for a wide class of Queueing Disciplines, Naval Research Logistics Quarterly. 1965; 12: 181-192.

22 Kleinrock L., "Priority Queueing", Chapter 3, Queueing System. Vol. II, Wiley, USA, 1976, 106-155.

23 Kleinrock L. and Finkelstein, "Time Dependent Priority Queues", Operations Research, 1967; 15: 104-116.

24 Kobayashi H., "Basic Queueing Analysis", Chapter 3, Modeling and Analysis: An Introduction to System Performance Evaluation Methodology, Addison-Wesley, USA, 1978, 95-220.

25 Little J., "A Proof for the Queueing Formula: L = λ ⁎ W", Operations Research, 1961; 9: 383-387.

26 Miller R., "A Contribution to Theory of Bulk Queues", Journal of the Royal Statistical Society, Series B: 21, 1959, 320-327.

27 Miller R., "Priority Queues", Ann. Math. Statistics, 1960; 31(1).

28 Miller W, "A Survey of the Theoretical Behaviour of Queues", United Kindom Atomic Energy Authority, 1966.

29 Minh D. and Sorli R., "Simulating the GI/G/1 Queues in Heavy Traffic",Operations Research, Sept-Oct. 1983; 31(5): 966-971.

30 Ramalhoto M., Amaral J. and Cochito M., "A Survey of J. Little's Formula", International Statistical Review, 1983; 51: 255-278.

31 Saaty L., "Résumé of Queueing Theory", Chapter 11, Mathematical Methods of Operations Research, MaGraw-Hill, Japan, 1959, 331-374.

32 Shrage L., "An Alternative Proof of a Conservation Law for the Queue G/G/1", Operations Research, 1970; 18: 185-187.

33 Sikes J., " Simplified Analysis of an Alternating Priorities Queueing Model with set up Times", Operations Research, 1970; 18: 1182-1192.

34 Stidham S., "Regenerative Processes in the Theory of Queues, with Applications to the Alternating-Priority Queues", Adv. Appl. Prob., 1972; 4: 542-577.

35 Tambouratzis D., "On a Property of the Variance of the Waiting time of a Queue", Journal of Applied Probability, 1968; 5: 702-703.

36 Wishart D., "Queueing Systems in which the Discipline is Last Come, Last Served", Operations Research, Sept-Oct. 1960; 8: 591-599.

37 Wolf R., "Work Conserving Priorities", Journal of Applied Probability, 1970; 7: 327-337.

VII) Simulation of a Hospital Waiting List Model

7.1) Introduction

The modern general hospital is complicated technically, socially and architecturally, and has parts that are comparable with a factory, a school and a department store. A hospital is an organic system, in which any part may interact with any other part, so that the consequence of a decision to change something cannot readily be foreseen. When a postulated change is expensive, irreversible or risky, or when there is disagreement as to whether it is desirable, it is understandable that there is often reluctance to make a decision for change because of the unknown interaction involved. However, simulation can be used in the analysis of those situations in which the costs and the risks are too high to be carried out in the real world ([41], p. 1).

A simple representation of the hospital system is presented in figure 7.1. This figure shows the route by which patients pass into and out of the hospital. It merely illustrates the main flows and delays through appointment and waiting list. At its simplest, the size of the waiting list depends on the demand for treatment and the rate at which patients can be treated. When demand is greater than throughput, a queue is inevitable. In reality the demand is affected by a host of complex factors such as morbidity, public expectation, seasonal influences, referral policies of GP's and consultants, technological developments

and so on. But these factors, in general, are not controllable and hospital managers must concern themselves with hospital throughput [32].

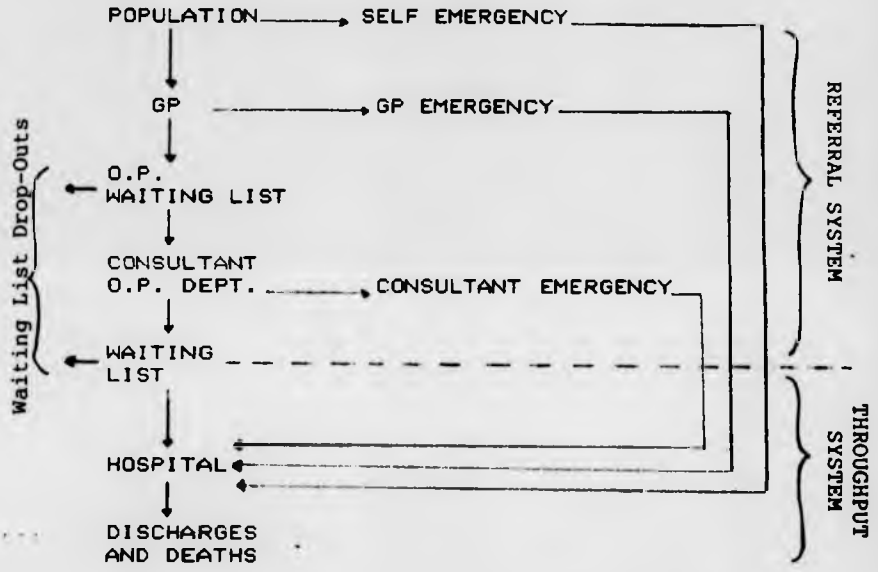


Figure 7.1 Flow diagram of patients entering hospital

In Great Britain there are three principal sources of information relating to the waiting list. The Annual Hospital Returns (Form SH3) which registers, for each speciality in each hospital, the number of patients awaiting treatment at 31 December each year since 1949. This head count yields no information about waiting time. The Hospital In-Patient Enquiry (HIPE) has provided a 10% sample of

discharges from hospitals which contains some information about waiting times since 1958. The Department of Health and Social Security, DHSS, commenced collection of in-patient waiting time information on a district basis in 1975 (Form SHB 203).

From 1964 to 1971 HIPE reported that the mean waiting time fluctuated between 13.7 and 14.8 weeks. During the same period information from SH3 indicated that there were between 8.9 to 9.5 patients on waiting lists per 1000 population [19]. Snaith [57] reported that 96% of those patients waiting for treatment were in surgical specialities; with 26% in general surgery, 16% in traumatic and orthopaedic surgery, 18% in otorhinolaryngology and 14% in gynaecology as the principal categories.

Despite the periodical collection of data from waiting lists, some care has to be taken in interpreting these data. For many reasons waiting lists can give inflated figures. Admission to another hospital, patient recovery, death, willingness to put up with the complaint and other reasons are all encountered. In a study reported by Yates [64] about the replies of 450 letters sent to patients who had been the longest on a waiting list showed that 22% wanted to be taken off the list, 20% did not reply and only 46% were actually available for admission.

The DHSS in Great Britain has distinguished two types of medical priorities in a hospital waiting lists: Urgent and Non-Urgent cases. The DHSS has deemed that the

Urgent cases should wait for no more than a month and the Non-Urgent cases for no more than a year. Despite this recomendation it has been reported that between 1977 and 1979 the proportion of urgent cases that had to wait for more than a month in all surgical specialities rose from two thirds to three quarters [18]. Yeates [65] made a study in various cities in England and reported big differences in the percentages of urgent patients waiting for more than one month on an Urology in-patient waiting list; the range was from 26.7% to 84.4%.

The increase in the waiting time for urgent cases can vary from hospital to hospital. Therefore the courses of action also have to be different. When it is decided to reduce the length of the list by increasing resources as a means of decreasing the waiting time, some care has to be taken because the increment of resources does not always lead to a reduction in the length of the list.

Frost [30] & [31] showed that a 1% increase in consultants produces an increment in the length of the waiting list. He demonstrated that after two years the increment in the list would be .78% and after five years .95%. Frost concluded that the waiting lists exist because consultants are able to control their own work load.

Weightman [63] states that an admission from a waiting list represents a compromise between competing responsibilities. Firstly, there is the duty to each patient requiring surgical treatment to arrange his admission

according to his medical needs taking into consideration the likely course of his disease if admission is delayed, the frequency and severity of distressing symptoms and the patient's ability in the interim to work or to run a household. On the other side of the balance it is encumbent upon medical and administrative staff to utilize the available resources to the full.

Considering the above statement, we have defined a model which represents the process of admission from a hospital waiting list. One of our purposes is to analyze the behaviour of the waiting time distribution when the hospital waiting list model is represented as a queueing system with $\rho=1$; under the condition ($\rho=1$) we assume that the resources are used to their maximum capacity, because the supply of medical service is matched to the demand of this service ($1/\lambda=1/\mu$). Another purpose is to show the statistical effects in the waiting time distribution for every medical priority in the list, when the medical needs of the patients are taken into consideration to define the policies of admission.

7.2) General Objectives

The general objectives of this chapter is to analyze the process of admission from a hospital waiting list, seen as a queueing process, with the following characteristics:

- queueing discipline based on a Dynamic Priority Discipline
- the queue discipline is non-preemptive
- the arrivals and the admissions rates are equal ($\rho=1$)
- the arrivals and admissions patterns are in batches and are stationary patterns
- the arrivals intervals are constant and the number of arrivals per unit of time is a random variable
- there are no 'impatient customers'

The problem is to find out how the waiting time for each medical priority is affected by different conditions in a hospital waiting list; the condition $\rho=1$ justifies the use of simulation to solve this problem (chapter VI).

7.3) Model of a Hospital Waiting List

The design of a general model for a hospital waiting list is a difficult task because of the different types of demands and resources involved in each hospital. However, some general characteristics could be identified on the waiting list. These characteristics are:

- Arrivals.- A certain number of new patients are placed on the waiting list every week. Each new patient is assigned some type of priority to define his need for being hospitalized.
- Admissions.- A number of patients are withdrawn from the waiting list for hospitalization every week.
- Waiting Patients.- After the arrivals and admissions a number of patients stay on the waiting list.

Using the above characteristics, we defined our model of a hospital waiting list as follows:

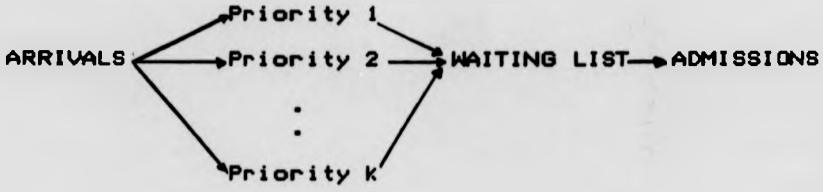


Figure 7.2 Model of a Hospital Waiting List

Now, to know the admissions procedures in our model, it is neccesary to specify the policies of admission or set of rules that define the type and number of patients to be hospitalized from the waiting list.

The problem in defining the policies of admission in a hospital waiting list is that usually there are patients who require to be hospitalized more urgently than others, but at the same time there are patients who have been waiting for a long time and for humanitarian grounds have to be admitted after a certain number of weeks. The principle that we used to define the policies of admission in the simulation model was the following:

" There is a maximum number of waiting weeks for every different type of patient on the waiting list; if a patient has been waiting more than his maximum, he becomes the next potential admission. When there is more than one potential admission, FCFS discipline is applied. The most urgent cases have preference in case of a tie. "

Then to apply the the above principle we need to develop a mechanism that combines the medical priority and the waiting time. The mechanism we used was based on a scoring system. This system is defined in the next section.

7.3.1) Scoring System

The policy of admission applied in our hospital waiting list model was based on a scoring system, which is defined by the following steps:

a) According to the medical priority each patient at the time of arrival to the waiting list is assigned an Initial Score. We identified this initial score by the symbol IS_K; K refers to the medical priority. The most urgent cases have higher IS than the less urgent cases,

$$IS_K \geq IS_{K+1}$$

(the lower the index K the higher the priority)

b) The score of every patient is incremented by one unit every waiting week,

$$S_{k,t} = S_{k,t-1} + 1 \quad ; \quad S_{k,0} = IS_K$$

($S_{k,t}$: Score after t weeks on the list for patient with priority k)

c) Patients with the highest scores are withdrawn from the waiting list for admission.

d) The most urgent cases have preference in the event of a tie in the scores.

What we are really proposing in the Scoring System is to artificially change the waiting time at the arrival point and then select those patients with the lohgest artificial waiting time, giving preference for urgent cases in the event of a tie. In other words, we use the queueing

discipline First-Come-First-Served (FCFS) with artificial waiting times. Applications of this type of system can be seen in Fordyce & Phillips [28], Luckman & Murray [42] and Phoenix [52].

7.3.2) Assumptions

To study the effects of the Scoring System we made the following assumptions in the hospital waiting list model:

i) The total number of arrivals per week, Arr_i, is an independent identically distributed (iid) random variable, with expected value MU and standard deviation SD.

ii) The total number of admissions per week, ADM, is constant and is equal to MU,

$$ADM = MU$$

which implys $\rho=1$.

iii) Two types of medical priorities are identified on the hospital waiting list:

- Urgent cases
- Non-Urgent cases

iv) The proportion of urgent arrivals each week follows a binomial distribution with fixed p.

v) The initial score for Urgent cases is greater than zero; for Non-Urgent cases it is always zero. Because of this assumption, we will use IS

without any index to indicate the initial score for urgent cases.

vi) The medical priority cannot be changed.

vii) The proportion of lost patients is zero.

viii) The proportion of patients who reject the admission is zero.

7.3.3) Mathematical Model

The total length of the waiting list, symbolized by Q, can be represented by the following expression after 't' simulated weeks:

$$Q^{(t)} = Q0 + \sum (Arr_i - ADM)$$

Q0 : Initial Population, the value of Q at the beginning of the simulation

Arr_i: Total number of arrivals in the i-th week

ADM : Total number of admissions per week (constant)

Then,

$$E(Q^{(t)}) = Q0 + t*(MU-ADM)$$
$$= Q0 \quad ; ADM = MU$$

and the variance,

$$Var(Q^{(t)}) = Var(\sum(Arr_i - ADM))$$
$$= t*Var(Arr_i - ADM)$$
$$= t*Var(Arr_i)$$

So, a non-stationary process is produced in $Var(Q^{(t)})$, as its value depends on the time. If $Var(Arr_i)=0$, then the non-

stationary problem will not present itself; in these conditions Q is always equal to Q0, a situation difficult to find in a real waiting list. Therefore, to tackle the non-stationary problem and at the same time have conditions close to a real waiting list, we impose some control in the arrivals pattern to maintain Q very close to Q0. The procedure was as follows:

- If Q was above a certain limit, then the admissions were increased.
- If Q was below another limit, then the admissions were decreased.

As a consequence of this control, it was neccessary to introduce the following variables in our simulation model:

LE.- The distance around Q0 that defines the normal limits within which it is required to maintain Q:
  - Lower Limit: Q0-LE
  - Upper Limit: Q0+LE

  (these limits will be referred as the 'normal limits' for Q)

AD.- Adjustment in the admissions to be applied when Q is out of its normal limits.

We could have adjusted the arrivals instead of the admissions, but we considered that the hospital managers have more control over the admissions than over the arrivals. Also we considered that control in the arrivals would mean transfering the problem to another hospital; this would be satisfactory for the patients if they would receive

immediate medical attention, but if not, the problem of waiting time would continue.

Now, due to the random factor involved in our model of a hospital waiting list, some care has to be taken when Q is out of its normal limits because this could be just a temporary fluctuation (See figure 7.3). For this possible fluctuation, we decided to introduce a delay to identify the need for adjustment in the admissions. The definition of this delay was: when Q is out of its normal limits during a certain number of consecutive weeks, then we decided that an adjustment in the admissions was needed. This delay was identified by the symbol D1.

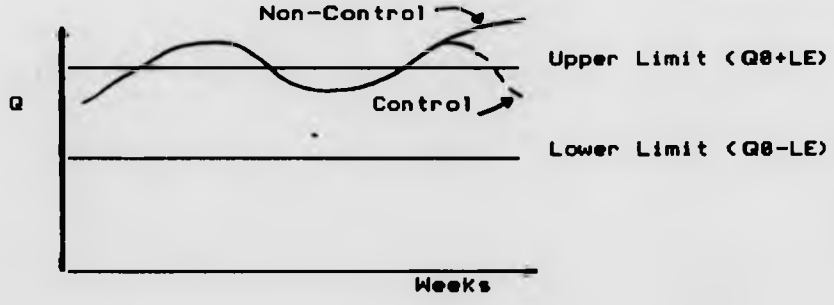


Figure 7.3 Q against the time (weeks)

Because of the uncertainties in deciding the adjustment in the admissions, it was necessary to define a second delay, D2, independent of D1. This delay, D2, will

define the time when the adjustment in the admissions will be operated; in a real situation D2 allows the hospital manager time for planning the changes in the number of admissions.

7.3.4) Admission Pattern

In summary the general rules for the admissions pattern are:

a) If Q is between Q0 ± LE, then the total number of admissions will be equal to the expected number of arrivals.

b) If Q is not between Q0 ± LE during D1 consecutives weeks, then the total number of admissions will start to be modified in in D2 weeks' times. The criteria for the adjustment in the admissions is:

   b.1) If Q > Q0 + LE, then the total number of admissions will be equal to MU + AD.

   b.2) If Q < Q0 - LE, then the total number of admissions will be equal to MU - AD

c) When the adjustment in the admissions is being applied and Q fell within its normal limits in the previous week, then the adjustment will last for the next D2 weeks. The same rule is applied when Q jumps from one of its normal limit to the other.

7.3.5) Measures of Effectiveness

To analyze how the admissions pattern and the scoring system are affecting the behaviour of the hospital waiting list model, we made the following calculations:

i) The arithmetic mean of the waiting times of those patients admitted in one year (52 weeks) for each priority. We will refer to this mean by the symbol W_i ; i = 1 Urgent cases, i = 2 Non-Urgent cases. W_i is calculated with the real waiting time and not with the artificial waiting time involved in the Scoring System.

ii) The arithmetic mean of the number of patients waiting every week during one year (52 weeks) for each priority. We will refer to this mean by the symbol Q_i ; i = 1 Urgent cases, i = 2 Non-Urgent cases.

So, W_i and Q_i summarize the waiting times and size of the waiting list that occur during year in our simulation model.

7.3.6) Consequences

As urgent and non-urgent cases receive almost the same treatment once they become potential admissions, we expect the following results when the initial score for urgent cases is equal to zero, IS=0:

a) No difference in the mean waiting time between priorities,

$$\bar{W}_1 \cong \bar{W}_2$$

b) The mean number of patients on the list per priority on the list should be proportional to the arrivals rate:

- $\bar{Q}_1$ = Q0 * PU
- $\bar{Q}_2$ = Q0 * (1-PU)

(PU : Proportion of Urgent Arrivals)

These results were useful in correcting and validating the computer program that simulated the hospital waiting list model.

7.4) Variables

In this section we present a summary of the variables defined in the model of a hospital waiting list. These variables are:

1) Q0 : Initial population on the waiting list including urgent and non-urgent cases.

2) MU : Expected number of arrivals per week, including urgent and non-urgent cases.

3) IS : Initial Score assigned to urgent cases at the time of their arrival on the waiting list.

4) PU : Expected proportion of urgent arrivals per week.

5) SD : Standard Deviation in the arrivals pattern.

6) DIST : Statistical distribution for the arrivals pattern.

7) LE : Distance around Q0 to define the 'normal limits' for the total length of the list,
- Lower Limit Q0 - LE
- Upper Limit Q0 + LE

8) AD : Adjustment in the admissions when Q is out of its 'normal limits',
- If Q < Q0 - LE, then
  Admissions = MU - AD
- If Q > Q0 + LE, then
  Admissions = MU + AD

9) D1 : Delay one. Number of consecutive weeks that Q is out of its 'normal limits'. This delay defines whether AD is necessary.

10) D2 : Delay two. Number of weeks to start AD, once it is decided there is need for the adjustment in the admissions.

11) W_i : Arithmetic Mean waiting time for the admissions during one year in each medical priority.

12) Q_i : Arithmetic Mean number of patients per week on the waiting list during one year in each medical priority.

7.5) Computer Program

Due to the local computer facilities available, the hospital waiting list model was simulated with the computer language FORTRAN 77 [45]. The computer program consisted of four main subroutines:

Subroutine	Task
i) INITIA	Set initial conditions
ii) ADMISS	Define the total number of admissions
iii) PRIORY	Allocate the medical priorities
iv) SELECT	Select patients with the highest scores

The arrivals patterns were generated using NAG subroutines [48]. A normal and uniform distribution were used in the arrivals pattern. With the normal distribution the following algorithm was applied:

1) x <- G05DDF(A , B) ; A indicates the mean and B the standard deviation

2) ARRIV <- NINT (x)

G05DDF(A,B) is a NAG subroutine that generates samples from a normal distribution using the algorithm of Brent [13], which is based on Forsythe's method (section 2.2.3).

To generate the total number of arrivals with a Uniform or Rectangular distribution, U (a , b), the

following algorithm was used:

1) u_i <- G05CAF(u_{i-1})

2) ARRIV = a + INT[((b-a) + 0.999)*u_i]

G05CAF is a NAG subroutine which returns a pseudo-random number with a Uniform distribution between zero and one, U(0,1). This subroutine uses the following multiplicative congruential generator (section 3.3.1)

$$X_i \equiv 13^{13} * X_{i-1} \quad (\text{MOD } 2^{59})$$

where $X_0=2n+1$, n is a positive integer.
Because $13^{13} \not\equiv 3$ or 5 (mod 8), the above generator does not reach its maximum possible period, which is 2^{57} (see section 3.3.2). However, we decided to use it to save programming time.

The PRIORY subroutine (allocation of priorities) is described in the following algorithm:

1) For J = 1 to ARRIV do

   1.1) u_k <- G05CAF (u_{k-1})

   1.2) If u_k < PU, URG <- URG + 1

2) NOURG <- ARRIV - URG

3) Return

ARRIV : Total number of arrivals (Urg's and Non-Urg's cases)

PU: Expected proportion of Urgent arrivals

URG: Number of Urgent arrivals

NOURG: Number of Non-Urgent arrivals

The algorithm for ADMISS subroutine (admissions pattern) was:

1) If Q < Q0-LE, or, Q > Q0+LE, go to step 4
2) ADMS <- MU , unless an adjustment in the admissions has been programmed for this week.
3) Return
4) If Q > Q0+LE, go to step 8
5) Q < Q0-LE for 'D1' consecutive weeks? If not, go to step 2
6) In 'D2' weeks, ADMS <- MU-AD (adjustment in the admissions)
7) Go to step 2
8) Q > Q0+LE for 'D1' consecutive weeks? If not, go to step 2
9) In 'D2' weeks, ADMS <- MU+AD (adjustment in the admissions)
10) Go to step 2

For the SELECT subroutine (highest score) the algorithm was:

1) For j=1 to ADMS do
    1.1) Select the patient with the highest score
    1.2) Withdraw from the waiting list the selected patient
    1.3) Register the waiting time if the number of weeks simulated is greater than 50
2) Return

In SELECT subroutine, the variable 'ADMS' is provided by the subroutine ADMIS.

So, the algorithm for the main program was:

1) Call INITIA
2) For j=1 to WEEKS do
   2.1) Call ADMISS
   2.2) x <- G05DDF(A,B)
   2.3) ARRIV <- NINT(x)
   2.4) Call PRIORY
   2.5) Call SELECT
3) Compute W_i & Q_i
4) Print W_i & Q_i
5) Stop

WEEKS: simulation time (weeks)
- arrivals from a normal distribution

The outputs of the program were W_i and Q_i; these values were calculated after the simulation time was greater than 50 weeks. The generated data were later statistically analyzed with the packages GENSTAT [53] and GLIM [49].

```
                HOSPITAL WAITING LIST
                FORTRAN PROGRAM

                - MAIN PROGRAM -

      INTEGER WAIT(201,2),ICLEN(3),ICADM(3),CLEN,CADM,NYEAR,REPL
      INTEGER AD(2),SEED,IS(2),IP,ISDI,IISI,IPI(3),QO,YEAR
      INTEGER ISD(3),IIS(3),ADMS,ARRIV,LAG,Q(2),WEEKS,TIME
      INTEGER ID1(3),ID2(3),D1,D2
      REAL MU,SD,PUA,MEAN(2),SUM(2),LEN(2)
      DOUBLE PRECISION G05DDF,G05CAF
       COMMON Q,WAIT,LAG,PUA
      COMMON /BL1/ AD,SUM
      COMMON /BL3/ IS
C     ** READING DATA **
C
      DATA (ICLEN(J),J=1,3),(ICADM(K),K=1,3)/10,10,5,1,1,1/
      DATA (IPI(J),J=1,3),(ISD(K),K=1,3)/50,50,20,6,6,6/
      DATA (IIS(J),J=1,3)/10,10,5/
      DATA (ID1(J),J=1,3),(ID2(K),K=1,3) /3,3,3,1,1,1/
      DATA NYEAR,SEED/1,4876/
      CALL G05CBF(SEED)
C     ** DEFINING CONDITIONS **
C
      DO 140 QO=200,300,50
       DO 140 IMU=20,30,5
      MU=FLOAT(IMU)
      DO 140 ISDI=ISD(1),ISD(2),ISD(3)
      SD=FLOAT(ISDI)
      DO 140 CLEN=ICLEN(1),ICLEN(2),ICLEN(3)
      DO 140 CADM=ICADM(1),ICADM(2),ICADM(3)
      DO 140 IP=IPI(1),IPI(2),IPI(3)
      PUA=FLOAT(IP)/100.
      DO 140 IISI=IIS(1),IIS(2),IIS(3)
      IS(1)=IISI
C     **  DELAYS  **
C
       DO 140 D1=ID1(1),ID1(2);ID1(3)
       DO 140 D2=ID2(1),ID2(2),ID2(3)
C     ** REPLICATIONS **
C
      DO 140 REPL=1,20
C     ** SETTING INITIAL CONDITIONS **
C
      CALL INITIA(QO,MU,TIME,WEEKS,NYEAR,YEAR,CLEN)

C
C     ** START SIMULATION **
C
      DO 140 I=2,WEEKS
C     ** DEFINING ADMISSIONS **
C
          ADMS=NINT(MU)
      IF((SD.GT.O.).AND.(CADM.GT.O))THEN
          CALL ADMISS(ADMS,CADM,Q(1)+Q(2),D1,D2)
         END IF
C     ** GENERATING ARRIVALS **
C
      ARRIV=NINT(MU)
      IF(SD.GT.O.O)ARRIV=NINT(G05DDF(MU,SD))
      IF(ARRIV.LE.O)GOTO 20
C     ** ALLOCATING MEDICAL PRIORITIES **
C
      CALL PRIORY(ARRIV,PUA,WAIT(I,1))
      WAIT(I,2)=ARRIV-WAIT(I,1)
   20 IF(ADMS.LE.O)GOTO 40
```

```
C      ** SELECTING THE HIGHEST SCORES **
C
       CALL SELECT(ADMS,I)
C
   40  IF(I.GT.LAG+1)THEN
       DO 50 K=1,2
 50    LEN(K)=LEN(K)+FLOAT(Q(K))/52
       END IF
C
       IF(I.EQ.TIME)THEN
        TIME=TIME+52
        YEAR=YEAR+1
       DO 60 K=1,2
 60    IF(AD(K).GT.0)MEAN(K)=SUM(K)/AD(K)
C
C      ** PRINT WAITING TIME AND LENGTH (MEANS) **
C
       WRITE(6,125)(MEAN(K),K=1,2),(LEN(L),L=1,2),PUA,MU,SD,QO,
      +IS(1),CADM,CLEN,YEAR,D1,D2
 125   FORMAT(2X,7(F6.2,1X),7(I3,1X))
       DO 70 K=1,2
       SUM(K)=0.0
       LEN(K)=0.0
        AD(K)=0
 70    AD(K)=0
       END IF
 140   CONTINUE
C
C      ** END SIMULATION **
C
        STOP
        END
```

```
C     ** INITIALIZATION OF DATA **
      SUBROUTINE INITIA(QO,MU,TI,WE,NYE,YEA,CL)
      INTEGER WAIT(201,2),Q(2),IS(2),QO,LAG,TI,WE,NYE,CL,YEA
      INTEGER LI,LS,CTA(4),CTB(4)
      REAL DUMMY,PUA,PP,MU
       COMMON Q,WAIT,LAG,PUA
      COMMON /BL2/ LI,LS,CTA,CTB
      COMMON /BL3/ IS
      DO 8 J=1,4
      CTA(J)=0
    8 CTB(J)=0
      LAG=50
      YEA=0
      TI=103
      WE=51+NYE*52
      LI=QO-CL
      LS=QO+CL
      PP=PUA*(1-PUA)
      DUMMY=PUA*QO-(PP*IS(1)*MU)
      IF(DUMMY.LT.0.)DUMMY=0.0
      Q(1)=NINT(DUMMY)
      Q(2)=QO-Q(1)
      DO 10 K=1,2
      DO 10 I=1,201
   10 WAIT(I,K)=0
      DO 13 K=1,2
   13 WAIT(1,K)=Q(K)
      RETURN
      END
C     ** DEFINING ADMISSIONS **
        SUBROUTINE ADMISS(NA,CAD,LENG,D1,D2)
        INTEGER NA,CAD,LI,LS,LENG,D1,D2,CTA(4),CTB(4)
        COMMON /BL2/ LI,LS,CTA,CTB
         IF(D2.EQ.0)THEN
           IF(LENG.GT.LS)NA=NA+CAD
           IF(LENG.LT.LI)NA=NA-CAD
          ELSE
         IF(CTA(2).EQ.1)THEN
            CTA(3)=CTA(3)+1
            IF(CTA(3).GE.D2)NA=NA+CAD
           ELSE
            IF(CTA(3).NE.0)THEN
             CTA(4)=CTA(4)+1
             IF(CTA(4).LT.D2)THEN
               NA=NA+CAD
              ELSE
               CTA(3)=0
               CTA(4)=0
              END IF
           END IF
         END IF
         IF(CTB(2).EQ.1)THEN
            CTB(3)=CTB(3)+1
            IF(CTB(3).GE.D2)NA=NA-CAD
           ELSE
            IF(CTB(3).NE.0)THEN
             CTB(4)=CTB(4)+1
             IF(CTB(4).LT.D2)THEN
               NA=NA-CAD
              ELSE
               CTB(3)=0
               CTB(4)=0
              END IF
           END IF
           END IF
```

```
          IF((LENG.LT.LI).OR.(LENG.GT.LS))THEN
            IF(LENG.GT.LS)THEN
           IF(CTA(2).EQ.0)THEN
                IF(D1.EQ.0)THEN
                  CTA(2)=1
                  CTB(2)=0
                ELSE
              CTA(1)=CTA(1)+1
               IF(CTA(1).EQ.D1)THEN
                 CTA(2)=1
                 CTB(2)=0
               END IF
             END IF
            END IF
           END IF
           IF(LENG.LT.LI)THEN
            IF(CTB(2).EQ.0)THEN
            IF(D1.EQ.0)THEN
               CTB(2)=1
               CTA(2)=0
             ELSE
           CTB(1)=CTB(1)+1
               IF(CTB(1).EQ.D1)THEN
               CTB(2)=1
               CTA(2)=0
               END IF
               END IF
               END IF
            END IF
           ELSE
               CTB(1)=0
               CTA(1)=0
               CTB(2)=0
               CTA(2)=0
           END IF
           END IF
           RETURN
           END
C      ** GENERATING PRIORITIES **
       SUBROUTINE PRIORY(AR,P,URG)
       INTEGER URG,Q(2),AR
       REAL P
       DOUBLE PRECISION RN,G05CAF
       COMMON Q
       DO 70 J=1,AR
       RN=G05CAF(RN)
 70    IF(RN.LE.P)URG=URG+1
       Q(1)=Q(1)+URG
       Q(2)=Q(2)+(AR-URG)
       RETURN
       END
C
```

```
C      ** SELECTING PATIENTS **
       SUBROUTINE SELECT(ADMS,I)
       INTEGER Q(2),WAIT(201,2),LAG,I,ADMS,AD(2),SCORE
       INTEGER JX,KX,IS(2),MAXS
       REAL SUM(2)
       COMMON Q,WAIT,LAG
       COMMON /BL1/ AD,SUM
       COMMON /BL3/ IS
       DO 120 IX=1,ADMS
       JX=0
       KX=0
       MAXS=0
       DO 110 K=1,2
       IF(Q(K).EQ.0)GOTO 110
       JI=1
        IF(I.GT.50)JI=I-50
       DO 100 J=JI,I
       IF(WAIT(J,K).EQ.0)GOTO 100
       SCORE=(I-J)+IS(K)
       IF(SCORE.LE.MAXS)GOTO 100
       MAXS=SCORE
       JX=J
       KX=K
100    CONTINUE
 110   CONTINUE
       IF((JX.GT.0).AND.(KX.GT.0))THEN
       Q(KX)=Q(KX)-1
       WAIT(JX,KX)=WAIT(JX,KX)-1
        IF(I.GT.LAG+1)THEN
        SUM(KX)=SUM(KX)+FLOAT(I-JX)
        AD(KX)=AD(KX)+1
       END IF
       END IF
120    CONTINUE
       RETURN
       END
```

7.6) Statistical Results

7.6.1) Q in Control and Non-Control Conditions

7.6.1.1) Introduction

We started the analysis of the hospital waiting list model by comparing the effects in W_i & Q_i when control and non-control conditions were applied in the admissions pattern. The model was simulated for 310 weeks (five years plus 50 weeks). After 50 weeks of simulation we start to calculate W_i and Q_i for every simulated year, so, for every run we have five W_i's. Every run was repeated 30 times, which implys 5X30=150 observations from our simulation model. The conditions in which the model was run were:

- Q0 = 200 (initial population, urgent and non-urgent cases)
- MU = 20 (average number of arrivals per week)
- SD = 6 (standard deviation used in the arrivals distribution)
- PU = 0.5 (expected proportion of urgent arrivals per week)
- IS = 10 (initial advantage given to urgent cases, weeks)
- Arrivals ~ N(MU , SD)

The values assigned to the control variables were:

- AD = 1 (adjustment in the admissions)
- LE = 10 (distance around Q0 to define normal limits for Q)

- D1 = 3 (consecutive weeks Q is out of its normal limits)

- D2 = 1 (weeks left to start adjustment in the admissions)

At the end of every simulated year we calculated the following statistics:

$$\bar{W}_i = \frac{\sum_{j=1}^{30} W_{i(j)}}{30} \quad \text{(j no. of replications)}$$

$$\bar{Q}_i = \frac{\sum_{j=1}^{30} Q_{i(j)}}{30}$$

$$s^2_{Wi} = \frac{\sum_{j=1}^{30} (W_{i(j)} - \bar{W}_i)^2}{29}$$

and

$$s^2_{Qi} = \frac{\sum_{j=1}^{30} (Q_{i(j)} - \bar{Q}_i)^2}{29}$$

Then we built the confidence intervals (assuming normality) to estimate $E(W_i)$ & $E(Q_i)$ per year in control and non-control conditions.

7.6.1.2) Results

We found that $\bar{W}_i$ & $\bar{Q}_i$ did not show any significant change during the five simulated years, either in control and non-control conditions (see graphs 7.1 and 7.2). However, in terms of s_{Wi} and s_{Qi}, when there were no control conditions these statistics increase with time (non-stationary process), but in control conditions behaviour

through time was very constant. In response to these results, we carried on the analysis of the simulation model using only control conditions and one year in the simulation time.

7.6.2) $\bar{W}_i$ & $\bar{Q}_i$ by DIST, Q0, MU, PU and IS

7.6.2.1) Introduction

In this section we analyzed how the variables DIST, Q0, MU, PU,and IS affect W_i & Q_i with specific values for the control conditions. For this purpose we decided to use the following linear model:

$$y = \mu + \alpha_j + \beta_k + \delta_l + \phi_m + \tau_n + \sum \text{Interactions} + \varepsilon \quad ...(I)$$

y: W_i, Q_i

μ : General mean

α_j : Effect of Q0

β_k : Effect of MU

δ_l : Effect of PU

ϕ_m : Effect of IS

τ_n : Effect of DIST

ε : Experimental error. Random variable with

$E(\varepsilon) = 0$; $E(\varepsilon^2) = \sigma^2$

$E(\varepsilon.\varepsilon') = 0$

The specific conditions in the simulation model were:

Q0: 200 , 300

MU: 20 , 30

PU: 0.50 , 0.70

IS: 5 , 10

DIST: Normal and Uniform distribution

The above variables will be referred to as factors and their corresponding values as levels.

When the normal distribution was used to generate the arrivals per week the standard deviation was six,

SD = 6

To define the range of the Uniform distribution the following criteria were applied:

- If MU = 20, then the arrivals distribution was U [10 , 30]
- If MU = 30, then the arrivals distribution was U [20 , 40]

(In both cases the standard deviations were approximately six; we tried to keep the same variability as in the normal distribution)

The control variables were assigned the following values:

- AD = 1 (adjustment in the admissions)
- LE = 10 (distance around Q0 to define normal limits)
- D1 = 3 (delay number one to define the need for adjustment in the admissions)
- D2 = 1 (delay number two to start adjustment in the admissions)

The number of replications defined in this simulation experiment was twenty, so the total number of observations was 20X32 = 640 units (2^5=32 indicates the total number of combinations of the five factors).

Analysis of Variance (ANOVA) was used to analyze which effects in the linear model were different from zero [37]. The statistical package used for this purpose was GENSTAT [53].

7.6.2.2) ANOVA results

In this simulation we found that the effect of using different distributions in the arrivals pattern did not show any significant change in $\bar{W}_i$ & $\bar{Q}_i$ (see table 7.6). We found significant interactions between Q0 & MU and PU & IS; see graph 7.3 & 7.4. In general the behaviour presented in these graphs is as follows:

- When IS was increased,
  $| \bar{Y}(PU=0.7) - \bar{Y}(PU=0.5) |$ increases
  where $Y = W_i$ and Q_i
- When Q0/MU is constant, $\bar{W}_i$ & $\bar{Q}_i$ are not affected.

When we compared the mean waiting time between priorities, $\bar{W}_2 - \bar{W}_1$, we found that this difference was constant, and equal to IS plus 0.5,

$$\bar{W}_2 - \bar{W}_1 \equiv IS + 0.5$$

When we repeated the ANOVA with W_2-W_1 as the experimental unit, we found that the only significant factor was IS. This means that the only difference in the mean waiting time is explained by the initial advantage given to the urgent cases, IS. This confirmed that both priorities were treated almost equally in the selection procedures once IS was assigned. The 0.50 in the above expression is explained by the preference that Urgent cases had in the case of a tie.

7.6.3) $E(W_i)$ & $E(Q_i)$, approach by Little's Equation

7.6.3.1) Introduction

In this section we concentrate our attention on the expected values of W_i & Q_i. We develop an approximation to $E(W_i)$ & $E(Q_i)$ by using our previous results and Little's formula (see section 6.10). To begin with, we introduce our notation in this formula

$$Q0 = MU * E(W)$$

or,

$$E(W) = Q0/MU \quad \ldots\ldots\ldots\ldots\ldots\ldots (7.1)$$

With the same principle of Little's Equation, we can obtain an expression for each priority in the waiting list as follows:

$$E(Q_1) = (MU*PU) * E(W_1) \quad \ldots\ldots\ldots\ldots (7.2)$$

and

$$E(Q_2) = [MU*(1-PU)] * E(W_2) \quad \ldots\ldots\ldots (7.3)$$

Hence, to calculate $E(Q_1)$ we need to find $E(W_1)$. To solve

this problem, firstly we split Q by the sum Q_1+Q_2 so,

$$E(Q_1) + E(Q_2) = Q0 \quad \ldots\ldots\ldots\ldots\ldots\ldots (7.4)$$

Secondly, we divide this expression by MU,

$$E(Q_2)/MU + E(Q_1)/MU = Q0/MU$$

This expression could be written as below (the components of the left side were multiplied by one):

$$(1-PU)*E(Q_2)/(MU*(1-PU) + PU*E(Q_1)/(MU*PU) = Q0/MU$$

Then, using equations (7.1), (7.2) & (7.3) we obtained the following relationship:

$$(1-PU)*E(W2) + PU*E(W_1) = E(W) \quad \ldots\ldots (7.5)$$

Now, based on our previous results (section 7.6.2.2), we assumed as true the following relationship:

$$E(W_2) - E(W_1) = IS + 0.50 \quad \ldots\ldots\ldots (7.6)$$

Finally, combining the expressions (7.5) & (7.6), we have a system of simultaneous linear equations with two variables $E(W_1)$ & $E(W_2)$. Solving this sytem we found the following approximations to the expected waiting time per priority:

$$E(W_1) = E(W) - (1-PU)*IS' \quad \ldots\ldots\ldots (7.7)$$

and

$$E(W_2) = E(W) + PU*IS' \quad \ldots\ldots\ldots\ldots\ldots (7.8)$$

$$(IS' = IS + 0.50)$$

With eq.'s (7.7) & (7.8) substituted in eq.'s (7.2) & (7.3) we could get an approach to the expected number of patients waiting on the list:

a) Urgent cases

$$E(Q_1) = [E(W) - (1-PU)*IS']*(MU*PU)$$

$$= [Q0 - (1-PU)*(IS'*MU)]*PU$$

b) Non-Urgent cases

$$E(Q_2) = [E(W) + PU*IS']*[MU*(1-PU)]$$

$$= [Q0 + MU*PU*IS']*(1-PU)$$

$$= Q0 - E(Q_1)$$

($E(Q_1)$ and $E(Q_2)$ were the values used to define the initial number of patients per priority in the simulation model)

7.6.3.2) [$\bar{W}_i$ vs $E(W_i)$] & [$\bar{Q}_i$ vs $E(Q_i)$]

When we compared the means from our simulated data against the approximations to the expected values, $E(W_i)$ & $E(Q_i)$, we found no real difference between them (see table 7.7). This result was confirmed with a X^2 test for Goodness of Fit (section 3.4.1.1). So, on average the waiting time and the length of the list per priority can be well explained by a combination of Q0, MU, PU & IS, using the specific control conditions defined in this simulation.

7.6.3.3) Error distributions by Q0, MU, PU and IS

To show the behaviour of the simulated data against the approximation of the expected values, we present in graphs 7.5, 7.6, 7.7 and 7.8 the distribution of

$$(W_i - E(W_i)) \quad vs \quad E(W_i)$$

and

$$(Q_i - E(Q_i)) \quad vs \quad E(Q_i)$$

To compare the errors distribution $[W_i - E(W_i)]$ and $[Q_i - E(Q_i)]$, we calculated the corresponding standard deviations around the expected values,

$$SD_y = [\sum(Obs - Exp)^2/n-1)]^{1/2} ,$$
$$(y = W_i, Q_i)$$

For this purpose we carried out an analysis of variance to determine whether Q0, MU, PU and IS had some effect on SD_y. Our results did not show evidence to reject the hypothesis that the factors effects were equal to zero (see table 7.9).

Therefore, we could say that the variability in W_i & Q_i was not affected by Q0, MU, PU and IS.

7.6.3.4) Maximum value for IS

Using the results from the previous section, we could find the range of values in IS which have some effect on the waiting time. As

$$E(W_i) \geq 0$$

which means, for urgent cases,

$$E(W) - (1 - PU) * IS' \geq 0$$

$$IS' = IS + 0.50$$

then, the maximum value for IS which satisfied the above relationship is:

$$IS_{max} = E(W)/(1-PU) - 0.50$$

From this point the mean waiting time for urgent cases will not be affected and will be equal to zero. This limit will also provide information about the maximum waiting time (mean) for the non-urgent cases and also the maximum difference in the mean waiting time between priorities.

7.6.4) $\overline{SD}_{(Wi,\ Qi)}$ by SD, AD, LE, D1 & D2

7.6.4.1) Introduction

The main objective in this section is to explore the effects in the variability of our simulation model when SD, AD, LE, D1 & D2 take different values and Q0, MU, PU, IS are constant.

To reach our objective we carried out another simulation of the hospital waiting list model with the following characteristics:

- Standard deviation in the arrivals patterns:
  SD = 3 , 6
- Distance around Q0 to define the normal limits in Q (Patients):
  LE = 10 , 20
- Adjustment in the admissions when Q is out of its normal limits (Patients):
  AD = 1 , 2
- Delay number one. Number of consecutive weeks Q is out of its normal limits to decide the need for adjustment in the admissions:
  D1 = 4 , 8
- Delay number two. Number of weeks left to start the adjustment in the admissions:
  D2 = 2 , 4

The constant conditions were:

- Q0 = 200,
- MU = 20,
- PU = 0.50
- IS = 10
- Arrivals ~ N(20 , SD)

The number of replications was 10. So, the total number of possible factor combinations was $2^5 = 32$.

To measure the variability in the simulated model we calculated the standard deviations around the expected values for every five replications, either in W_i & Q_i. Therefore, two standard deviations were calculated in each factor level. So the total number of standard deviations was:

$$2 \times 32 = 64 \text{ Standard Deviations}$$

We split the observations of W_i & Q_i to provide clearer information about the behaviour of the standard deviations in the ANOVA table.

The symbols used to identify the standard deviations were:

- SD_{Wi} : Standard deviation around $E(W_i)$. (n = 5)
- SD_{Qi} : Standard deviation around $E(Q_i)$. (n = 5)

For simplicity we did not introduce in the above symbols the indexes to identify under which conditions they were

generated. (It is important to remember that we use the symbol "SD" without any index to refer to the standard deviations in the arrivals pattern).

Before analysing the behaviour of the standard deviations around the expected values, we validated the simulated data by checking that the means $\bar{W}_i$ & $\bar{Q}_i$ were not affected by SD, AD, LE, D1 & D2.

We have seen in the previous results that $\bar{W}_i$ & $\bar{Q}_i$ were well explained by Q0, MU, PU and IS. Since in this simulation they were constant, the means should not be affected. We expected these means to be equal to (using our approximation to the expected values):

- $\bar{W}_1 \cong 200/20 - (1-0.50)*(10.5) = 4.75$
- $\bar{W}_2 \cong \bar{W}_1 + 10.5 = 15.25$
- $\bar{Q}_1 \cong \bar{W}_1*(20)*(0.50) = 47.5$
- $\bar{Q}_2 \cong 200 - \bar{Q}_1 = 152.5$

Having validated the simulated data, we studied the behaviour of $\overline{SD}_{Wi}$ & $\overline{SD}_{Qi}$ under the new conditions in our simulation model. We used an ANOVA table to find out which factors produced significant effects in the standard deviations.

Finally, to have an approach to $\overline{SD}_{Wi,\ Qi}$ under different conditions, we fitted the following regression model:

$$\overline{SD}_y = \alpha + \sum \beta_j * (f - \bar{f})$$

y: W_i , Q_i

f: Significant factors in ANOVA.

The analysis of variance was carried out with the computer package GENSTAT and the regression model with the GLIM package.

7.6.4.2) $\bar{W}_i$ & $\bar{Q}_i$ by SD, AD, LE, D1 and D2

We found that the means $\bar{W}_i$ & $\bar{Q}_i$ were not affected by SD, AD, LE, D1 and D2. We did not find evidence from the simulated data to reject the hypothesis that the main effects from SD, AD, LE, D1 & D2 and their respective interactions were all equal to zero (see table 7.14) These results confirmed that Q0, MU, PU and IS were the only factors in our simulation model which explain $\bar{W}_i$ & $\bar{Q}_i$.

7.6.4.3) $\overline{SD}_{Wi}$ & $\overline{SD}_{Qi}$ by SD, AD, LE, D1 and D2

We found that the only significant differences in $\overline{SD}_{Wi}$ were presented in SD, AD & LE. For $\overline{SD}_{Qi}$ the significant differences were with SD and LE (see table 7.19).

It was interesting to find that the two types of delays, D1 & D2, did not produce any significant effect in the standard deviations distribution. This result will allow the hospital

manager to anticipate and prepare the use and non-use of resources, knowing that the variability in the distribution of W_i & Q_i will not be affected.

In general, we found that the behaviour of $\overline{\overline{SD}}_{Wi}$ in relation to LE, AD and SD was as follows (see graph 7.9):

- $\overline{\overline{SD}}_{Wi}$ increases when SD & LE are increased, and decreases when AD is incresed.

The above behaviour is explained by the relationship that exists between Q0 and the factors SD, LE & AD. The factors SD & LE ensured that Q fell within a certain distance from Q0, while AD attempted to close this distance. These patterns could be represented as follows:

i) if SD or LE $\uparrow \rightarrow |\ Q - Q0\ | \uparrow \rightarrow \overline{\overline{SD}}_{Wi} \uparrow$

ii) if AD $\uparrow \rightarrow |\ Q - Q0\ | \downarrow \rightarrow \overline{\overline{SD}}_{Wi} \downarrow$

From graph 7.9 we did not note any difference between $\overline{\overline{SD}}_{W1}$ & $\overline{\overline{SD}}_{W2}$. To check this we carried out a t-test and the results did not show any significant result. This confirmed again that both priorities were treated equally in our simulation model.

7.6.4.4) Regression Model in $\overline{\overline{SD}}_{Wi}$

As we found that there was no significant difference between $\overline{\overline{SD}}_{W1}$ & $\overline{\overline{SD}}_{W2}$, we decided to merge these standard deviations to have only one representative of the variability in the waiting time. The merged standard

deviation was identified by the symbol $\overline{SD}_W$.

Using $\overline{SD}_W$ as our dependent variable we fitted a regression model in terms of SD, LE & AD, and found the following model (See table 7.22):

$$(7.9)\ldots\ldots \overline{SD}_W = 0.69+0.11*(SD-4.5)+0.05*(LE-15)-0.32*(AD-1.5)$$

So, with the above regression model we could have an approximation to the variability in our simulation model with different values in SD, LE and AD.

7.6.4.5) Limits for LE & AD

Because LE & AD can take any positive number, we looked for those in which the variability of the simulation model is really affected. To do this, we assumed the following expression: .

$$\overline{SD}_W \leq \overline{SD}_W^{(n-c)} \ldots\ldots\ldots 7.9.1$$

$\overline{SD}_W^{(n-c)}$: The mean of the standard deviations in the waiting time for both priorities when no control is applied in the admission pattern; the admissions per week is constant.

To find $\overline{\overline{SD}}_W^{(n-c)}$ we repeated the simulation experiment with no control conditions and we found the following regression model:

$$(7.10)\ldots\ldots \overline{\overline{SD}}_W^{(n-c)} = 1.25 + 0.50 * (SD - 4.5)$$

Using eq.'s (7.9) & (7.10) in 7.9.1 we obtained the following relationship:

$$(7.11)\ldots\ldots LE - 6.4 * AD \leq 16.6 + 7.8 * (SD - 4.5)$$

$$[\ LE,\ AD\ \&\ SD \geq 0\ ]$$

So, $\overline{\overline{SD}}_W$ will be affected by LE & AD when the expression (7.11) is satisfied.

7.6.4.6) Regression model in $\overline{\overline{SD}}_{Qi}$

We found that the only significant factors to explain $\overline{\overline{SD}}_{Qi}$ were SD & LE (See Table 7.24). Also, we found significant differences in the mean of the standard deviations in the length of the list between priorities,

$$\overline{\overline{SD}}_{Q_1} \neq \overline{\overline{SD}}_{Q_2}$$

As a consequence of the above results, we decided to fit a regression model for each priority. The regression results were (See table 7.25):

$$\overline{\overline{SD}}_{Q1} = 8.5 + 1.33*(SD-4.5) + 0.45*(LE-15)$$

and

$$\overline{\overline{SD}}_{Q2} = 10.0 + 2.00*(SD-4.5) + 0.45*(LE-15)$$

[the constant terms and the coefficients for (SD-4.5) were significantly different]

Therefore, to know the variability in Q_i when more than two priorities are defined on the hospital waiting list model, it would be necessary to repeat the simulation experiment. An alternative procedure is to use Little's formula as follows:

i) We get a confidence interval for the expected waiting time per priority assuming normality,

$$(\hat{Wi}_{Inf} , \hat{Wi}_{Sup})$$

Inf. Limit: $E(W_i) - Z_\alpha * \overline{\overline{SD}}_W$

Sup. Limit: $E(W_i) + Z_\alpha * \overline{\overline{SD}}_W$

$Z_\alpha \sim N(0 , 1)$

ii) then the limits for the waiting time are multiplied by the arrivals rate for each priority,

$$(\hat{Qi}_{Inf} = \lambda_i * \hat{Wi}_{Inf} , \hat{Qi}_{Sup} = \lambda_i * \hat{Wi}_{Sup})$$

λ_i: Expected number of arrivals with priority "i".

Because both priorities were treated equally once they were assigned IS, we expected that $\overline{\overline{SD}}_W$ would be statistically the same for all priorities defined in the hospital waiting list model.

7.6.5) $E(W_i)$ with $i > 2$

Now our problem is to find the expected waiting time when more than two priorities are defined in our simulation model. In this section we present an approximation to solve this problem. We assumed initially three priorities, and then we made the generalization for any number of priorities.

With three priorities in our simulation model, we expected the following relationships to be satisfied:

$$E(W_2) - E(W_1) = (IS_1 - IS_2) + 0.50 \quad \ldots\ldots\ldots (7.12)$$

$$E(W_3) - E(W_2) = (IS_2 - IS_3) + 0.50 \quad \ldots\ldots\ldots (7.13)$$

[$IS_K \geq IS_{K+1}$; patients with priority "K" have higher priority than those with priority "K+1"]

Extending expression (7.5) from section 7.6.3.1, we obtained the following expression:

$$(1-P_1-P_2)*E(W_3) + P_2*E(W_2) + P_1*E(W_1) = E(W) \ldots\ldots (7.14)$$

Combining expressions (7.12), (7.13) & (7.14) we have a system of simultaneous equations with three variables $E(W_1)$, $E(W_2)$ and $E(W_3)$. Solving this system, we obtain the following approximations:

$$E(W_1) = E(W) - [IS''*(1-P_1-P_2)+IS'*(1-P_1)] \ldots\ldots (7.15)$$

$$E(W_2) = E(W_1) + IS' \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7.16)$$

$$E(W_3) = E(W_2) + IS'' \quad \text{.....(7.17)}$$

$$IS' = (IS_1 - IS_2) + 0.50$$

$$IS'' = (IS_2 - IS_3) + 0.50$$

From the above expressions, we generalize the expected waiting time for any number of priorities on the hospital waiting list model as follows:

$$E(W_1) = E(W) - [\sum_{h=1}^{k-1} IS_h' * (1 - \sum_{j=1}^{h} P_j)] \quad \text{.....(7.18)}$$

$$E(W_{h+1}) - E(W_h) = IS_h' \quad \text{.....(7.19)}$$

$$IS_h' = (IS_h - IS_{h+1}) + 0.50$$

$$E(W) = Q0/MU$$

k: No. of priorities on the waiting list

P_j: Expected proportion of arrivals with priority "j"

IS_h: Initial score given to patients with priority "h"

To find the maximum value for IS_1 for the expression (7.18), we have to consider the following relationships:

i) $E(W_1) \geq 0$

ii) $IS_1 \geq IS_2 \geq \ldots\ldots \geq IS_k \geq 0$

Now, assuming that all IS_j (j=2,k-1) take their maximum possible value, which is IS_1, and that IS_k (initial score

for the lowest priority) is equal to zero,

a) $IS_j = IS_1$; $j = 2,(k-1)$

b) $IS_k = 0$

then, under these conditions, the difference in the expected waiting time will be 0.50, except for the last two lowest priorities, in which case it would be:

$$E(W_{k-1}) - E(W_k) = IS_1 + 0.5$$

Using eq. 7.18 with the above assumptions, $E(W_1)$ would be:

$$E(W_1) = E(W) - [\sum_{i=1}^{k-2}(0.5)(1-\sum_{j=1}^{1}P_j) + (IS_1+0.5)(1-\sum_{j=1}^{k-1}P_j)]$$

So, the maximum value for IS_1 that satisfies $E(W_1)\geq 0$ is

$$IS_{1(max)} = \frac{E(W) - \sum_{i=1}^{k-2}(0.5)(1-\sum_{j=1}^{1}P_j)}{1 - \sum_{j=1}^{k-1}P_j} - 0.50$$

If IS_k (the score of the lowest priority) is equal to IS_1, then there is no need to know the maximum value for IS_1 because all the priorities on the list would receive the same treatment. If $0 < IS_k < IS_1$, then the maximum value for IS_1 would be the maximum value for IS_1 when $IS_k=0$ plus IS_k,

- If $0 < IS_k < IS_1$, then

  $IS^*_{1(max)} = IS_{1(max)} + IS_k$

  ($IS_{1(max)}$, when $IS_k=0$)

All the expressions that we obtained for a general case in the expected waiting time are consistent with our simulation results, where two medical priorities were defined in the hospital waiting list model.

7.7) Summary

By using simulation, we could evaluate the behaviour of the waiting time per priority produced when a scoring system was used in the policies of admission, using a queueing model (chapter VI) with the arrivals rate equal to the admission rates, which means, with the traffic intensity equal to one ($\rho=1$).

We considered that having $\rho=1$ in the simulation model meant we could have a more realistic representation of a waiting list, in terms of its total length, because if $\rho < 1$, the expected length would tend to zero, and if $\rho > 1$, it would tend to infinity; its increment or decrement would depend on the difference between arrivals and admissions rates.

By analytical means (section 7.3.1) we demonstrated that if we use $\rho=1$, the expected total length of the list, referred to by the symbol Q, would be equal to its initial value at the start of the simulation (Q0) and its variance would increase with the time, a non-stationary process.

When we simulated our model with $\rho=1$, we found that the distribution of the waiting time and the length of the list for urgent cases were also affected by the time; their variances increased with the time (see graphs 7.1 & 7.2).

One way to solve the non-stationary problem in our simulation model was to have a constant number of total arrivals, this implied that the total length of the list would be always equal to its initial value, Q0, and the only random effect in the model would be the allocation of medical priorities, which were assumed to follow a binomial distribution. Under these results we checked that the computer program (section 7.5) was running according to what we expected.

Then, in order to develop more accurate results in the variability of the simulation model with random arrivals and constant admissions (the expected number of total arrivals per week was the number of admission per week), we experimented with the model, placing an upper and lower limits for Q. These limits were defined to be very close to Q0 to avoid the non-stationary problem in Q presented in the model with random arrivals.

The procedure used to implement the above limits was as follows: if Q is above its upper limit the total number of admissions were increased, if it was below the lower limit, the admissions were decreased.

To operate the above procedure in a flexible way, two types of delays were defined in the simulation model; the first one to decide the need for changing the admissions number, and the second one, independent of the first delay, to start the modification in the number of admissions.

So, the new admissions pattern would depend on the upper and lower limits for Q, the increment or decrement in the admissions when Q is out its limits, and the two types of delays to decide and initiate the changing in the admissions.

Because of the number of conditional operations that had to be considered in the new admission pattern, we found the flexibility of Fortan 77 very useful (see section 7.5). In any simulation purpose languages (SIMSCRIPT, GPSS, DYNAMO,..), we consider that the programming of this admission pattern would have been much more difficult, despite the fact that some of them may accept Fortran subroutines.

To check that the model was running as we expected with the new admissions pattern, we displayed the results in Q and in the number of admissions week by week, during 100 simulated weeks.

The validation of the model was based on the comparison of the mean waiting time per priority. We assumed initially two types of medical priorities (Urgent and Non-Urgent cases). We expected not to have any significant difference in the average waiting time, when no priority

distinctions were made in the policies of admission. We would have liked to compare these results with some real system in order to get better validation of the simulation model. However the risks involved changing the policies of admission in a hospital were too high for this academic research. Therefore, we constrained our validation to logical and mathematical comparisons.

Once the computer program was validated, the model was run under different values for the following variables: Q0, MU, IS, PU, DIST, LE, AD, D1, and D2.

One of the important results of this simulation was that we could control the variability in the waiting time, measured by the standard deviation, by maintaining the total length of the list within certain limits. A regresion equation in terms of the limits defined for Q, the increments (+ or -) for the admissions, and the standard deviation in the number of arrivals was used to explain this variability (section 7.6.4)

Another important result was that we could develop a formula for the expected waiting time for any number of priorities in the waiting list, with the traffic intensity equal to one (see section 7.6.5); this formula was a combination of the initial value for Q, the total arrivals rate, the proportion of arrivals for each priority, and the

initial score given to each priority. This formula was the combination of the following preliminary results:

- the simulation results showed that the difference in the mean waiting time was determined by the initial score given to each priority
- the expected waiting time, when the list does not have priorities, which is the case when our model was defined with IS=0, can be found by using Little's equation (section 6.10)

The other simulation results were that the type of distribution used to generate the total number of arrivals and the two types of delay in the admission pattern did not show any significant effect on the waiting time distribution.

7.8) Conclusion of the Experiment

We have seen the statistical results of a simulation model that represents the process of admission from a hospital waiting list when a scoring system is used in the policies of admission. Also we show one way to tackle the problem of non-stationarity in the variability of the length of the list when the arrival rate is equal to the admission rate. On the other hand, we saw how the simulation results can be combined with analytical methods to produce a formula for the expected waiting time.

Despite the above results, the question still remains of how consistent these results would be if they were applied to a real waiting list with conditions similar to our simulation model. Three facts suggest that our results could be consistent with the real world:

- the first one is the fact that the scoring system to decide what sort of priority can be assigned to each priority is very flexible,
- the second one, based on the result that the two types of delays were not significant, is that there is enough time to decide the need for changing the number of admissions (4-8 weeks) and to start this change (2-4 weeks), without affecting the waiting time distribution,

- and the third one is that the expected value and the variance of the arrivals distribution were the significant factors in the waiting time distribution, and not the type of distribution associated with the arrivals pattern.

One of the limitations in our hospital waiting list model was the assumption that the proportion of lost patients (P_{lost}) was zero. We saw in the introduction of this chapter that there were a number of lost patients in list with long waiting list (see ref. [4]). To adjust our simulation result to this possible situation in our model ($P_{lost} \neq 0$), we have to change Q0 to $Q0 \times (1-P_{lost})$.

The other limitation was in relation to the medical priority. We considered that the assumption that the priority did not change with the time in the waiting list is a difficult one to satisfy in a real waiting list. If the priority increases with the waiting time, the general procedure is to use an expression of the following form (see references [18], [42] & [52]:

SCORE = $\Psi(\xi, t)$

$\Psi(\xi,t)$: function of some biological, social and economical factors of the patients (ξ) and the waiting time (t).

Then, the same principle that we used in our simulation model is applied. The difficulties in this sort of function Ψ is the definition of the factors ξ, and the validity of combining the factors ξ in a single expression.

In our simulation model, we used a very simple scoring formula ($\Psi(\underline{f},t)=IS+t$) to have full control of the parameter IS and to be able to concentrate our analysis on the waiting time distribution per priority when the traffic intensity was equal to one.

Therefore, in the simulation analysis of a scoring system that defines the policies of admission from a hospital waiting list, a compromise has to be made between the assumptions of the simulation model, the control mechanism (Ψ), the complexity of evaluating the effects of the paramenters involved in Ψ, and the objective of the simulation experiment.

Table 7.1

$\bar{W}_1$ & $\bar{Q}_1$ in control and non-control conditions during five years

Characteristics in the simulation model*:

Q0 = 200 ; MU = 20 ; PU = 0.50 ; IS = 10 ;

SD = 6 ; AD = 1 ; LE = 10 ; D1 = 3 ; D2 = 1

Year	Control				Non-Control			
	W_1		Q_1		W_1		Q_1	
	Mean	s	Mean	s	Mean	s	Mean	s
1	4.78	0.9	48.1	11.5	4.73	1.6	47.9	18.8
2	4.65	1.4	46.8	15.2	4.58	3.2	46.4	32.9
3	4.87	0.9	49.2	10.8	4.81	3.8	48.7	39.0
4	4.67	1.5	47.0	16.1	4.74	4.3	47.7	44.3
5	4.32	1.5	43.0	15.1	4.54	4.9	45.6	50.4

- The means were calculated with n = 30.
- s: Sample Standard Deviations

* Normal Distribution was used in the arrivals pattern

Table 7.2

ANOVA in W_1 by DIST, Q0, MU, PU and IS

*** Analysis of Variance ***

Variate: W_1

Source of Variation	DF	SS	MS	VR
DIST	1	0.1302	0.1302	0.202
Q0	1	2778.6799	2778.6799	4313.215
MU	1	2595.5212	2595.5212	4028.907
IS	1	643.9058	643.9058	999.505
PU	1	386.7749	386.7749	600.373
DIST.Q0	1	0.0010	0.0010	0.002
DIST.MU	1	0.9083	0.9083	1.410
Q0.MU	1	133.4348	133.4348	207.125
DIST.IS	1	0.0691	0.0691	0.107
Q0.IS	1	0.0304	0.0304	0.047
MU.IS	1	0.9820	0.9820	1.524
DIST.PU	1	1.1946	1.1946	1.854
Q0.PU	1	0.6169	0.6169	0.958
MU.PU	1	0.3739	0.3739	0.580
IS.PU	1	69.5968	69.5968	108.032
DIST.Q0.MU	1	0.2187	0.2187	0.339
DIST.Q0.IS	1	0.0318	0.0318	0.049
DIST.MU.IS	1	0.0835	0.0835	0.130
Q0.MU.IS	1	1.7925	1.7925	2.782
DIST.Q0.PU	1	0.2337	0.2337	0.363
DIST.MU.PU	1	0.0542	0.0542	0.084
Q0.MU.PU	1	2.9417	2.9417	4.566
DIST.IS.PU	1	0.3672	0.3672	0.570
Q0.IS.PU	1	0.4791	0.4791	0.744
MU.IS.PU	1	0.1069	0.1069	0.166
RESIDUAL	614	395.5540	0.6442	
GRAND TOTAL	639	7014.0273		

GRAND MEAN 7.157
TOTAL NUMBER OF OBSERVATIONS 640

Table 7.3

ANOVA in W_2 by DIST, Q0, MU, PU and IS

*** Analysis of Variance ***

Variate: W_2

Source of Variation	DF	SS	MS	VR
DIST	1	0.2349	0.2349	0.333
Q0	1	2808.9707	2808.9707	3980.202
MU	1	2595.1548	2595.1548	3677.233
IS	1	1424.0820	1424.0820	2017.868
PU	1	382.2629	382.2629	541.652
DIST.Q0	1	0.0187	0.0187	0.027
DIST.MU	1	0.7062	0.7062	1.001
Q0.MU	1	133.3338	133.3338	188.929
DIST.IS	1	0.0544	0.0544	0.077
Q0.IS	1	0.0053	0.0053	0.007
MU.IS	1	1.3432	1.3432	1.903
DIST.PU	1	1.2461	1.2461	1.766
Q0.PU	1	0.7604	0.7604	1.077
MU.PU	1	0.7659	0.7659	1.085
IS.PU	1	68.1596	68.1596	96.579
DIST.Q0.MU	1	0.1434	0.1434	0.203
DIST.Q0.IS	1	0.0975	0.0975	0.138
DIST.MU.IS	1	0.0228	0.0228	0.032
Q0.MU.IS	1	2.0341	2.0341	2.882
DIST.Q0.PU	1	0.1550	0.1550	0.220
DIST.MU.PU	1	0.1210	0.1210	0.171
Q0.MU.PU	1	3.3902	3.3902	4.804
DIST.IS.PU	1	0.7673	0.7673	1.087
Q0.IS.PU	1	0.5966	0.5966	0.845
MU.IS.PU	1	0.2568	0.2568	0.364
RESIDUAL	614	433.3218	0.7057	
GRAND TOTAL	639	7857.9453		

GRAND MEAN 15.155
TOTAL NUMBER OF OBSERVATIONS 640

Table 7.4

ANOVA in Q_1 by DIST, Q0, MU, PU and IS

*** Analysis of Variance ***

Variate: Q_1

Source of Variation	DF	SS	MS	VR
DIST	1	149.6	149.6	0.786
Q0	1	575116.9	575116.9	3024.012
MU	1	44187.2	44187.2	232.340
IS	1	123526.4	123526.4	649.512
PU	1	517017.2	517017.2	2718.519
DIST.Q0	1	9.0	9.0	0.047
DIST.MU	1	234.6	234.6	1.234
Q0.MU	1	397.1	397.1	2.088
DIST.IS	1	79.2	79.2	0.416
Q0.IS	1	4.4	4.4	0.023
MU.IS	1	3750.2	3750.2	19.719
DIST.PU	1	378.3	378.3	1.989
Q0.PU	1	16964.7	16964.7	89.202
MU.PU	1	1013.6	1013.6	5.330
IS.PU	1	4697.7	4697.7	24.701
DIST.Q0.MU	1	13.5	13.5	0.071
DIST.Q0.IS	1	56.3	56.3	0.296
DIST.MU.IS	1	114.3	114.3	0.601
Q0.MU.IS	1	822.0	822.0	4.322
DIST.Q0.PU	1	67.3	67.3	0.354
DIST.MU.PU	1	17.7	17.7	0.093
Q0.MU.PU	1	742.5	742.5	3.904
DIST.IS.PU	1	573.5	573.5	3.016
Q0.IS.PU	1	109.6	109.6	0.577
MU.IS.PU	1	524.9	524.9	2.760
RESIDUAL	614	116772.6	190.2	
GRAND TOTAL	639	1407330.0		

GRAND MEAN 103.25
TOTAL NUMBER OF OBSERVATIONS 640

Table 7.5

ANOVA in Q_2 by DIST, Q0, MU, PU and IS

*** Analysis of Variance ***

Variate: Q_2

Source of Variation	DF	SS	MS	VR
DIST	1	117.3	117.3	0.803
Q0	1	250888.1	250888.1	1718.202
MU	1	71738.3	71738.3	491.298
IS	1	123890.2	123890.2	848.459
PU	1	551632.6	551632.6	3777.845
DIST.Q0	1	2.9	2.9	0.020
DIST.MU	1	30.2	30.2	0.207
Q0.MU	1	1.2	1.2	0.008
DIST.IS	1	15.0	15.0	0.103
Q0.IS	1	57.6	57.6	0.395
MU.IS	1	8906.0	8906.0	60.993
DIST.PU	1	103.5	103.5	0.709
Q0.PU	1	17607.0	17607.0	120.581
MU.PU	1	484.4	484.4	3.318
IS.PU	1	15.5	15.5	0.106
DIST.Q0.MU	1	155.2	155.2	1.063
DIST.Q0.IS	1	316.6	316.6	2.168
DIST.MU.IS	1	2.7	2.7	0.018
Q0.MU.IS	1	475.3	475.3	3.255
DIST.Q0.PU	1	144.9	144.9	0.993
DIST.MU.PU	1	1.9	1.9	0.013
Q0.MU.PU	1	272.4	272.4	1.865
DIST.IS.PU	1	108.2	108.2	0.741
Q0.IS.PU	1	32.9	32.9	0.225
MU.IS.PU	1	49.8	49.8	0.341
RESIDUAL	614	89654.9	146.0	
GRAND TOTAL	639	1116703.0		

GRAND MEAN 145.74
TOTAL NUMBER OF OBSERVATIONS 640

Table 7.6

Analysis of residual sums of squares when Q0, MU, PU and IS were used to explain $\bar{W}_i$ & $\bar{Q}_i$

Characteristics in the simulation model:

SD = 6 ; AD = 1 ; LE = 10 ; D1 = 3 ; D2 = 1

Units	Model I		Model II		$\frac{(SS_2-SS_1)*f_1}{SS_1*(f_2-f_1)}$
	f_1	SS_1	f_2	SS_2	
W_1	614	395.6	633	406.3 (1)	< 1
W_2	614	433.3	633	446.0 (1)	< 1
Q_1	614	116,772.6	632	122,080.0 (2)	1.6*
Q_2	614	89,654.9	633	92,042.4 (3)	<1

* p > 0.05

f_i : degrees of freedom

SS_i : Residual sums of squares

Model I $(W_i ; Q_i) = \mu+\alpha_j+\beta_k+\phi_l+\tau_m+\eta_n+$ $\sum_{1 \& 2 \text{ order}}$ interactions $+ \varepsilon$

Model II (Using significant effects) :

(1) $W_i = \mu+\alpha_j+\beta_k+\phi_l+\tau_m+(\alpha.\beta)_{jk}+(\phi.\tau)_{lm}+\varepsilon$

(2) $Q_1 = \mu+\alpha_j+\beta_k+\phi_l+\tau_m+(\alpha.\phi)_{jl}+(\beta.\tau)_{km}+(\phi.\tau)_{lm}+\varepsilon$

(3) $Q_2 = \mu+\alpha_j+\beta_k+\phi_l+\tau_m+(\alpha.\phi)_{jl}+(\beta.\tau)_{km}+\varepsilon$

μ	General mean	ϕ_l	Effect of PU
α_j	Effect of Q0	τ_m	Effect of IS
β_k	Effect of MU	η_n	Effect of DIST
		ε	Experimental error

Source: Tables 7.2, 7.3, 7.4 and 7.5

Table 7.7

Observed and Expected $\bar{W}_i$ & $\bar{Q}_i$ (1)

- Control Variables: AD = 1 , LE = 10
  D1 = 3 , D2 = 1
- SD = 6

Conditions				$\bar{W}_i$ (n=40)		$\bar{Q}_i$ (n=40)	
Q0	MU	PU	IS	$\bar{W}_1$	$\bar{W}_2$	$\bar{Q}_1$	$\bar{Q}_2$
			5	7.4 (7.3)	12.9 (12.8)	73.7 (73.0)	129.7 (127.0)
		.5					
	20		10	4.6 (4.8)	15.1 (15.3)	46.0 (48.0)	148.7 (152.0)
			5	8.1 (8.4)	13.6 (13.9)	112.9 (117.6)	81.7 (82.4)
		.7					
200			10	6.5 (6.9)	16.9 (17.4)	90.6 (96.6)	101.2 (103.4)
		.5	5	3.9 (3.9)	9.4 (9.4)	58.5 (58.5)	141.2 (141.5)
	30		10	1.5 (1.4)	11.9 (11.9)	21.8 (21.0)	180.6 (179.0)
		.7	5	4.9 (5.0)	10.4 (10.5)	103.2 (105.0)	94.3 (95.0)
			10	3.8 (3.5)	14.3 (14.0)	79.5 (73.0)	130.2 (127.0)
			5	12.3 (12.3)	17.7 (17.8)	121.6 (123.0)	176.4 (177.0)
		.5					
	20		10	9.5 (9.8)	20.0 (20.3)	96.3 (98.0)	199.8 (210.0)
			5	13.2 (13.4)	18.7 (18.9)	183.9 (187.6)	112.2 (112.4)
		.7					
300			10	11.9 (11.9)	22.4 (22.4)	167.5 (166.6)	131.7 (133.4)
			5	7.4 (7.3)	12.9 (12.8)	111.9 (109.5)	196.9 (198.5)
		.5					
	30		10	4.6 (4.8)	15.1 (15.3)	68.9 (72.0)	227.6 (228.0)
			5	8.2 (8.4)	13.7 (13.9)	171.4 (176.4)	122.4 (123.6)
		.7					
			10	6.9 (6.9)	17.4 (17.4)	144.3 (144.9)	157.6 (155.1)
$\chi^2 = \Sigma(O-E)^2/E$				0.105 *	0.045 *	2.16 *	1.07 *

* Non-Significant; $p > 0.05$

(1) The expected values are between brackets

Table 7.8

$SD_{(Wi;Qi)}$ by Q0, MU, PU & IS

(n=40)

- Control Variables: AD = 1 , LE = 10
  D1 = 3 , D2 = 1
- SD = 6

Conditions Q0	MU	PU	IS	SD_{W_1}	SD_{W_2}	SD_{Q_1}	SD_{Q_2}
			5	0.84	0.84	11.9	13.8
		.5					
	20		10	0.88	0.88	10.2	13.2
			5	1.00	1.00	16.9	8.8
		.7					
200			10	1.64	1.8	24.8	15.6
			5	0.83	0.98	13.8	16.9
		.5					
	30		10	0.59	0.65	9.3	13.2
			5	0.64	0.67	15.6	7.5
		.7					
			10	0.86	0.84	20.8	11.6
			5	0.67	0.69	8.6	9.7
		.5					
	20		10	1.14	1.1	14.8	10.2
			5	0.58	0.59	12.4	10.8
		.7					
300			10	0.61	0.58	12.6	8.2
			5	0.41	0.39	9.5	12.5
		.5					
	30		10	0.46	0.45	8.8	12.3
			5	0.51	0.49	13.8	9.8
		.7					
			10	0.47	0.50	12.1	10.2
	MEAN			0.76	0.78	13.4	11.9

Table 7.9

Analysis of residual sums of squares when Q0, MU, PU and IS were used to explain $\overline{SD}_{Wi}$ & $\overline{SD}_{Qi}$

Characteristics in the simulation model:

SD = 6 ; AD = 1 ; LE = 10 ; D1 = 3 ; D2 = 1

Units	Model I		Model II		$\frac{(SS_2-SS_1)*f_1}{SS_1*(f_2-f_1)}$
	f_1	SS_1	f_2	SS_2	
SD_{W1}	5	0.33	15	1.48	1.73 *
SD_{W2}	5	0.50	15	1.87	1.35 *
SD_{Q1}	5	54.90	15	285.50	2.10 *
SD_{Q2}	5	62.80	15	147.80	< 1 *

* p > 0.05

f_i : degrees of freedom

SS_i : Residual sums of squares

Model I $(SD_{Wi} ; SD_{Qi}) = \mu+\alpha_j+\beta_k+\phi_l+\tau_m+$ Σ interactions $+ \varepsilon$ (1 & 2 order)

Model II $(SD_{Wi} ; SD_{Qi}) = \mu+\varepsilon$

μ General mean

α_j Effect of Q0

β_k Effect of MU

ϕ_l Effect of PU

τ_m Effect of IS

ε Experimental error

Source: Tables 7.8

Table 7.10

ANOVA in W_1 by SD, AD, LE, D1 and D2

*** Analysis of Variance ***

Variate: W_1

Source of Variation	DF	SS	MS	VR
SD	1	4.0500	4.0500	7.432
AD	1	0.3050	0.3050	0.560
LE	1	1.6762	1.6762	3.076
D1	1	0.2611	0.2611	0.479
D2	1	0.0583	0.0583	0.107
SD.AD	1	0.7315	0.7315	1.342
SD.LE	1	2.1092	2.1092	3.871
AD.LE	1	0.0600	0.0600	0.110
SD.D1	1	0.0281	0.0281	0.052
AD.D1	1	0.3976	0.3976	0.730
LE.D1	1	0.4590	0.4590	0.842
SD.D2	1	0.3188	0.3188	0.585
AD.D2	1	0.0495	0.0495	0.091
LE.D2	1	0.0104	0.0104	0.019
D1.D2	1	2.8426	2.8426	5.216
SD.AD.LE	1	0.0551	0.0551	0.101
SD.AD.D1	1	0.5040	0.5040	0.925
SD.LE.D1	1	0.0099	0.0099	0.018
AD.LE.D1	1	0.1193	0.1193	0.219
SD.AD.D2	1	0.2398	0.2398	0.440
SD.LE.D2	1	0.1479	0.1479	0.271
AD.LE.D2	1	1.2802	1.2802	2.349
SD.D1.D2	1	0.8632	0.8632	1.584
AD.D1.D2	1	0.3524	0.3524	0.647
LE.D1.D2	1	1.5263	1.5263	2.801
RESIDUAL	294	160.2091	0.5449	
GRAND TOTAL	319	178.6647		

GRAND MEAN 4.713
TOTAL NUMBER OF OBSERVATIONS 320

Table 7.11

ANOVA in W_2 by SD, AD, LE, D1 and D2

*** Analysis of Variance ***

Variate: W_2

Source of Variation	DF	SS	MS	VR
SD	1	3.6061	3.6061	6.663
AD	1	0.4720	0.4720	0.872
LE	1	1.4540	1.4540	2.686
D1	1	0.1308	0.1308	0.242
D2	1	0.0254	0.0254	0.047
SD.AD	1	0.7615	0.7615	1.407
SD.LE	1	2.0464	2.0464	3.781
AD.LE	1	0.0144	0.0144	0.027
SD.D1	1	0.0016	0.0016	0.003
AD.D1	1	0.4418	0.4418	0.816
LE.D1	1	0.6239	0.6239	1.153
SD.D2	1	0.3505	0.3505	0.647
AD.D2	1	0.0119	0.0119	0.022
LE.D2	1	0.0170	0.0170	0.031
D1.D2	1	2.8671	2.8671	5.297
SD.AD.LE	1	0.0705	0.0705	0.130
SD.AD.D1	1	0.5080	0.5080	0.939
SD.LE.D1	1	0.0001	0.0001	0.000
AD.LE.D1	1	0.0458	0.0458	0.085
SD.AD.D2	1	0.3829	0.3829	0.708
SD.LE.D2	1	0.0875	0.0875	0.162
AD.LE.D2	1	1.3637	1.3637	2.520
SD.D1.D2	1	0.8894	0.8894	1.643
AD.D1.D2	1	0.4969	0.4969	0.918
LE.D1.D2	1	2.1239	2.1239	3.924
RESIDUAL	294	159.1271	0.5412	
GRAND TOTAL	319	177.9201		

GRAND MEAN 15.205
TOTAL NUMBER OF OBSERVATIONS 320

Table 7.12

ANOVA in Q_1 by SD, AD, LE, D1 and D2

*** Analysis of Variance ***

Variate: Q_1

Source of Variation	DF	SS	MS	VR
SD	1	452.30	452.30	6.261
AD	1	34.51	34.51	0.478
LE	1	187.95	187.95	2.602
D1	1	35.31	35.31	0.489
D2	1	4.29	4.29	0.059
SD.AD	1	82.60	82.60	1.143
SD.LE	1	239.19	239.19	3.311
AD.LE	1	7.51	7.51	0.104
SD.D1	1	3.32	3.32	0.046
AD.D1	1	31.48	31.48	0.436
LE.D1	1	39.62	39.62	0.548
SD.D2	1	16.99	16.99	0.235
AD.D2	1	9.82	9.82	0.136
LE.D2	1	0.06	0.06	0.001
D1.D2	1	283.28	283.28	3.921
SD.AD.LE	1	10.38	10.38	0.144
SD.AD.D1	1	81.22	81.22	1.124
SD.LE.D1	1	0.00	0.00	0.000
AD.LE.D1	1	2.67	2.67	0.037
SD.AD.D2	1	1.60	1.60	0.022
SD.LE.D2	1	13.04	13.04	0.181
AD.LE.D2	1	189.23	189.23	2.620
SD.D1.D2	1	134.97	134.97	1.868
AD.D1.D2	1	7.21	7.21	0.100
LE.D1.D2	1	160.32	160.32	2.219
RESIDUAL	294	21238.66	72.24	
GRAND TOTAL	319	23267.50		

GRAND MEAN 47.23
TOTAL NUMBER OF OBSERVATIONS 320

Table 7.13

ANOVA in Q_2 by SD, AD, LE, D1 and D2

*** Analysis of Variance ***

Variate: Q_2

Source of Variation	DF	SS	MS	VR
SD	1	175.5	175.5	1.447
AD	1	43.2	43.2	0.356
LE	1	1.5	1.5	0.012
D1	1	164.8	164.8	1.359
D2	1	33.7	33.7	0.278
SD.AD	1	112.2	112.2	0.925
SD.LE	1	82.7	82.7	0.682
AD.LE	1	1.7	1.7	0.014
SD.D1	1	76.7	76.7	0.632
AD.D1	1	450.3	450.3	3.713
LE.D1	1	1.8	1.8	0.015
SD.D2	1	74.9	74.9	0.618
AD.D2	1	93.4	93.4	0.770
LE.D2	1	4.7	4.7	0.038
D1.D2	1	558.1	558.1	4.603
SD.AD.LE	1	70.3	70.3	0.580
SD.AD.D1	1	163.7	163.7	1.350
SD.LE.D1	1	11.1	11.1	0.091
AD.LE.D1	1	16.2	16.2	0.134
SD.AD.D2	1	45.1	45.1	0.372
SD.LE.D2	1	208.1	208.1	1.716
AD.LE.D2	1	99.6	99.6	0.821
SD.D1.D2	1	352.5	352.5	2.907
AD.D1.D2	1	102.2	102.2	0.842
LE.D1.D2	1	258.3	258.3	2.130
RESIDUAL	294	35649.4	121.3	
GRAND TOTAL	319	38851.6		

GRAND MEAN	151.74
TOTAL NUMBER OF OBSERVATIONS	320

Table 7.14

Analysis of residual sums of squares when control variables were used to explain $\bar{W}_i$ & $\bar{Q}_i$

Characteristics in the simulation model:

Q0 = 200 ; MU = 20 ; PU = 0.50 ; IS = 10

Units	Model I		Model II		$\frac{(SS_2-SS_1)*f_1}{SS_1*(f_2-f_1)}$
	f_1	SS_1	f_2	SS_2	
W_1	294	160.29	319	178.67	1.35 *
W_2	294	159.13	319	177.92	1.39 *
Q_1	294	21,238.70	319	23,267.70	1.12 *
Q_2	294	35,649.40	319	38,851.60	1.06 *

* $p > 0.05$

f_i : degrees of freedom

SS_i : Residual sums of squares

Model I $(W_i ; Q_i) = \mu+\alpha_j+\beta_k+\phi_l+\tau_m+\eta_n+ \sum_{1 \& 2 \text{ order}} \text{interactions} + e$

Model II $(W_i ; Q_i) = \mu+e$

μ General mean

α_j Effect of SD

β_k Effect of LE

ϕ_l Effect of AD

τ_m Effect of D1

η_n Effect of D2

e Experimental error

Source: Tables 7.10, 7.11, 7.12 and 7.13

Table 7.15

ANOVA in SD_{W_1} by SD, AD, LE, D1 and D2

*** Analysis of Variance ***

Variate: SD_{W_1}

Source of Variation	DF	SS	MS	VR
SD	1	1.9515	1.9515	17.530
AD	1	3.4432	3.4432	30.930
LE	1	1.5625	1.5625	14.035
D1	1	0.3671	0.3671	3.298
D2	1	0.0088	0.0088	0.079
SD.AD	1	0.0632	0.0632	0.568
SD.LE	1	0.4041	0.4041	3.630
AD.LE	1	0.0011	0.0011	0.010
SD.D1	1	0.0096	0.0096	0.087
AD.D1	1	0.0200	0.0200	0.180
LE.D1	1	0.0038	0.0038	0.034
SD.D2	1	0.0005	0.0005	0.004
AD.D2	1	0.5145	0.5145	4.622
LE.D2	1	0.2050	0.2050	1.842
D1.D2	1	0.1538	0.1538	1.381
SD.AD.LE	1	0.0136	0.0136	0.122
SD.AD.D1	1	0.0194	0.0194	0.175
SD.LE.D1	1	0.0323	0.0323	0.290
AD.LE.D1	1	0.3655	0.3655	3.283
SD.AD.D2	1	0.4534	0.4534	4.073
SD.LE.D2	1	0.0249	0.0249	0.224
AD.LE.D2	1	0.0583	0.0583	0.524
SD.D1.D2	1	0.1031	0.1031	0.926
AD.D1.D2	1	0.3377	0.3377	3.034
LE.D1.D2	1	0.0004	0.0004	0.003
RESIDUAL	38	4.2304	0.1113	
GRAND TOTAL	63	14.3479		

GRAND MEAN 0.698
TOTAL NUMBER OF OBSERVATIONS 64

Table 7.16

ANOVA in SD_{W_2} by SD, AD, LE, D1 and D2

*** Analysis of Variance ***

Variate: SD_{W_2}

Source of Variation	DF	SS	MS	VR
SD	1	2.0046	2.0046	18.765
AD	1	3.3807	3.3807	31.647
LE	1	1.5080	1.5080	14.117
D1	1	0.3285	0.3285	3.075
D2	1	0.0080	0.0080	0.074
SD.AD	1	0.0717	0.0717	0.672
SD.LE	1	0.4938	0.4938	4.622
AD.LE	1	0.0100	0.0100	0.093
SD.D1	1	0.0202	0.0202	0.189
AD.D1	1	0.0341	0.0341	0.319
LE.D1	1	0.0018	0.0018	0.017
SD.D2	1	0.0145	0.0145	0.135
AD.D2	1	0.6228	0.6228	5.830
LE.D2	1	0.1715	0.1715	1.605
D1.D2	1	0.1270	0.1270	1.189
SD.AD.LE	1	0.0214	0.0214	0.200
SD.AD.D1	1	0.0168	0.0168	0.158
SD.LE.D1	1	0.0176	0.0176	0.165
AD.LE.D1	1	0.4072	0.4072	3.812
SD.AD.D2	1	0.4171	0.4171	3.904
SD.LE.D2	1	0.0307	0.0307	0.287
AD.LE.D2	1	0.0619	0.0619	0.579
SD.D1.D2	1	0.1099	0.1099	1.029
AD.D1.D2	1	0.2706	0.2706	2.533
LE.D1.D2	1	0.0000	0.0000	0.000
RESIDUAL	38	4.0594	0.1068	
GRAND TOTAL	63	14.2097		

GRAND MEAN 0.690
TOTAL NUMBER OF OBSERVATIONS 64

Table 7.17

ANOVA in SD_{Q_1} by SD, AD, LE, D1 and D2

*** Analysis of Variance ***

Variate: SD_{Q_1}

Source of Variation	DF	SS	MS	VR
SD	1	265.52	265.52	22.659
AD	1	315.91	315.91	26.958
LE	1	95.83	95.83	8.178
D1	1	21.15	21.15	1.805
D2	1	0.19	0.19	0.017
SD.AD	1	7.19	7.19	0.614
SD.LE	1	36.55	36.55	3.119
AD.LE	1	0.14	0.14	0.012
SD.D1	1	0.12	0.12	0.010
AD.D1	1	1.57	1.57	0.134
LE.D1	1	1.32	1.32	0.113
SD.D2	1	0.25	0.25	0.022
AD.D2	1	68.96	68.96	5.885
LE.D2	1	30.75	30.75	2.624
D1.D2	1	21.30	21.30	1.818
SD.AD.LE	1	1.25	1.25	0.107
SD.AD.D1	1	0.08	0.08	0.007
SD.LE.D1	1	1.21	1.21	0.104
AD.LE.D1	1	57.29	57.29	4.889
SD.AD.D2	1	63.79	63.79	5.444
SD.LE.D2	1	1.60	1.60	0.137
AD.LE.D2	1	3.95	3.95	0.337
SD.D1.D2	1	5.94	5.94	0.507
AD.D1.D2	1	41.11	41.11	3.508
LE.D1.D2	1	0.42	0.42	0.036
RESIDUAL	38	445.29	11.72	
GRAND TOTAL	63	1488.71		

GRAND MEAN 8.23
TOTAL NUMBER OF OBSERVATIONS 64

Table 7.18

ANOVA in SD$_{Q_2}$ by SD, AD, LE, D1 and D2

*** Analysis of Variance ***

Variate: SD$_{Q_2}$

Source of Variation	DF	SS	MS	VR
SD	1	582.37	582.37	28.979
AD	1	321.86	321.86	16.016
LE	1	78.23	78.23	3.893
D1	1	17.17	17.17	0.855
D2	1	1.85	1.85	0.092
SD.AD	1	39.91	39.91	1.986
SD.LE	1	23.39	23.39	1.164
AD.LE	1	0.00	0.00	0.000
SD.D1	1	5.35	5.35	0.266
AD.D1	1	2.43	2.43	0.121
LE.D1	1	22.44	22.44	1.117
SD.D2	1	3.84	3.84	0.191
AD.D2	1	45.08	45.08	2.243
LE.D2	1	1.76	1.76	0.088
D1.D2	1	6.40	6.40	0.318
SD.AD.LE	1	0.00	0.00	0.000
SD.AD.D1	1	2.13	2.13	0.106
SD.LE.D1	1	38.47	38.47	1.914
AD.LE.D1	1	56.45	56.45	2.809
SD.AD.D2	1	27.05	27.05	1.346
SD.LE.D2	1	27.56	27.56	1.371
AD.LE.D2	1	20.93	20.93	1.041
SD.D1.D2	1	17.62	17.62	0.877
AD.D1.D2	1	17.57	17.57	0.874
LE.D1.D2	1	0.36	0.36	0.018
RESIDUAL	38	763.65	20.10	
GRAND TOTAL	63	2123.89		
GRAND MEAN		10.92		
TOTAL NUMBER OF OBSERVATIONS		64		

Table 7.19

Analysis of residual sums of squares when control variables were used to explain $\overline{\overline{SD}}_{W_i}$ & $\overline{\overline{SD}}_{Q_i}$

Characteristics in the simulation model:

Q0 = 200 ; MU = 20 ; PU = 0.50 ; IS = 10

Units	Model I		Model II		$\frac{(SS_2-SS_1)*f_1}{SS_1*(f_2-f_1)}$
	f_1	SS_1	f_2	SS_2	
SD_{W1}	38	4.23	60	7.39 (1)	1.29 *
SD_{W2}	38	4.06	60	7.32 (1)	1.38 *
SD_{Q1}	38	445.30	61	907.30 (2)	1.71 *
SD_{Q2}	38	763.70	61	1219.70 (2)	< 1 *

* p > 0.05

f_i : degrees of freedom

SS_i : Residual sums of squares

Model I $(SD_{Wi} ; SD_{Qi}) = \mu+\alpha_j+\beta_k+\phi_l+\gamma_m+\eta_n+$ Σ interactions 1 & 2 order $+ \epsilon$

Model II (1) $SD_{Wi} = \mu+\alpha_j+\beta_k+\phi_l+\epsilon$

(2) $SD_{Qi} = \mu+\alpha_j+\beta_k+\epsilon$

μ General mean

α_j Effect of SD

β_k Effect of LE

ϕ_l Effect of AD

γ_m Effect of D1

η_n Effect of D2

ϵ Experimental error

Source: Tables 7.15, 7.16, 7.17 and 7.18

Table 7.20

$\overline{\overline{SD}}_{Wi}$ in SD, LE and AD

Characteristics in the simulation model:

Q0 = 200 ; MU = 20 ; PU = 0.50 ; IS = 10

Factors	Levels	$\overline{\overline{SD}}_{W1}$	$\overline{\overline{SD}}_{W2}$
SD	3	0.52	0.51
	6	0.86	0.87
LE	10	0.46	0.46
	20	0.92	0.92
AD	1	0.85	0.84
	2	0.53	0.54
General Mean		0.69	0.69
$\hat{\sigma}$ (df = 38)		0.33	0.33

- D1 & D2 did not show any significant contribution to the residual sums of squares.

Table 7.21

$\overline{\overline{SD}}_{Wi}$ by SD, LE and AD

(Linear Model)

Characteristics in the simulation model:

Q0 = 200 ; MU = 20 ; PU = 0.50 ; IS = 10

Conditions			$\overline{\overline{SD}}_{W1}$	$\overline{\overline{SD}}_{W2}$
SD	LE	AD		
3	10	1	0.45	0.43
		2	0.13	0.13
	20	1	0.91	0.89
		2	0.59	0.59
6	10	1	0.78	0.79
		2	0.47	0.49
	20	1	1.25	1.28
		2	0.93	0.95

- $\overline{\overline{SD}}_{Wi} = \hat{\mu} + \hat{\alpha}_j + \hat{\beta}_k + \hat{\phi}_l$

μ: General Mean

α_j: Effect of SD

β_k: Effect of LE

ϕ_l: Effect of AD

Source: Table 7.20

Table 7.22

Glim Outputs I

Regression Model:

$$E(\overline{SD}_W) = \alpha+\beta_1(SD-4.5)+\beta_2(LE-15)+\beta_3(AD-1.5)$$

Cycle	Deviance	DF
1	0.50000193E-04	4

ESTIMATE	S.E.	PARAMETER
0.6887	0.1250E-02	α
0.1125	0.8333E-03	β_1
0.4625E-01	0.2500E-03	β_2
-0.3175	0.2500E-02	β_3

(CO)VARIANCE MATRIX

1	1.5625E-06			
2	1.1511E-13	6.9445E-07		
3	1.0076E-14	9.8711E-15	6.2500E-08	
4	-1.7739E-15	1.6557E-13	3.0725E-14	6.2500E-06

SCALE PARAMETER TAKEN AS 0.1250E-04

$$-\overline{SD}_W = (\overline{SD}_{W_1} + \overline{SD}_{W_2})/2$$

Source: Table 7.21

Table 7.23

$\overline{\overline{SD}}_{Qi}$ * in SD and LE

Characteristics in the simulation model:

Q0 = 200 ; MU = 20 ; PU = 0.50 ; IS = 10

Factors	Levels	$\overline{\overline{SD}}_{Q1}$	$\overline{\overline{SD}}_{Q2}$
SD	3	6.20	7.90
	6	10.30	13.90
LE	10	6.00	8.70
	20	10.50	13.20
General Mean		8.20	10.90
$\hat{\sigma}$ (df = 38)		3.40	4.50

* n = 32

- No significant differences were found in AD, D1 and D2 (p > 0.50). See table 7.19

Table 7.24

$\overline{\overline{SD}}_{Qi}$ by SD and LE

(Linear Model)

Characteristics in the simulation model:

Q0 = 200 ; MU = 20 ; PU = 0.50 ; IS = 10

Conditions SD	LE	$\overline{\overline{SD}}_{Q1}$	$\overline{\overline{SD}}_{Q2}$
3	10	4.0	5.7
	20	8.5	19.2
6	10	8.0	11.7
	20	12.5	16.2

- $\overline{\overline{SD}}_{Qi} = \hat{\mu} + \hat{\alpha}_j + \hat{\beta}_k$

μ: General Mean

α_j: Effect of SD

β_k: Effect of LE

Source: Table 7.23

Table 7.25

Glim Outputs II

Regression Model:

$$E(\overline{SD}_{Qi}) = \alpha_i + \beta_{1i}(SD-4.5) + \beta_{2i}(LE-15)$$

1) YVAR: $\overline{SD}_{Q_1}$

Cycle	Deviance	DF
1	0.91707353E-13	1

ESTIMATE	S.E.	PARAMETER
8.250	0.1514E-06	α_1
1.333	0.1009E-06	β_{11}
0.4500	0.3028E-07	β_{12}

SCALE PARAMETER TAKEN AS 0.9171E-13

2) YVAR: $\overline{SD}_{Q_2}$

Cycle	Deviance	DF
1	0.21077532E-11	1

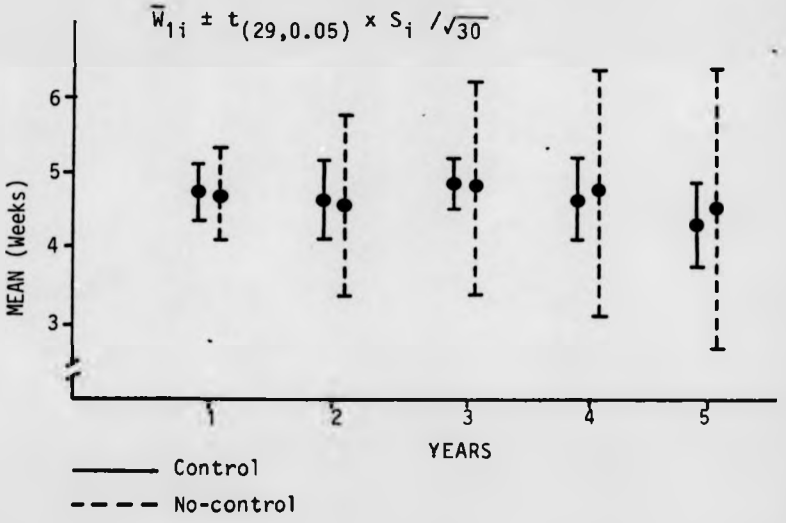
ESTIMATE	S.E.	PARAMETER
10.95	0.7259E-06	α_2
2.000	0.4839E-06	β_{21}
0.4500	0.1452E-06	β_{22}

SCALE PARAMETER TAKEN AS 0.2108E-11

Source: Table 7.24

Graph 7.1

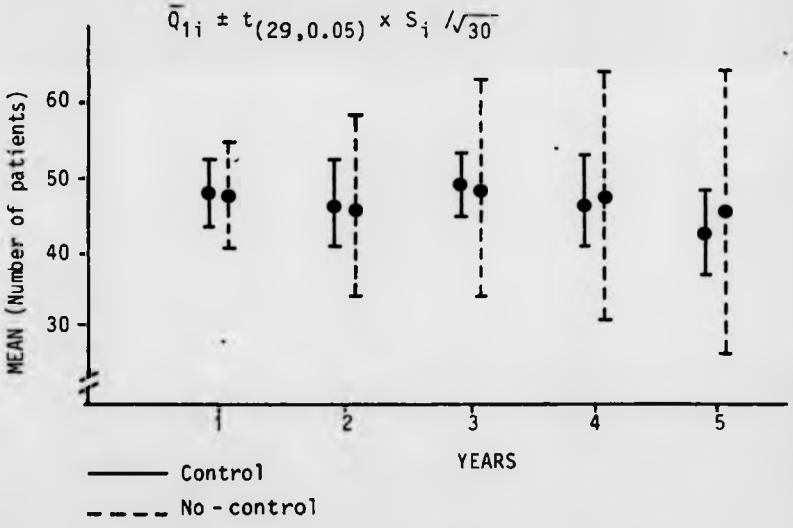
Mean Waiting Time for Urgent Cases per Year



Source : Table 7.1

Graph 7.2

Mean Number of Urgent Waiting per Year



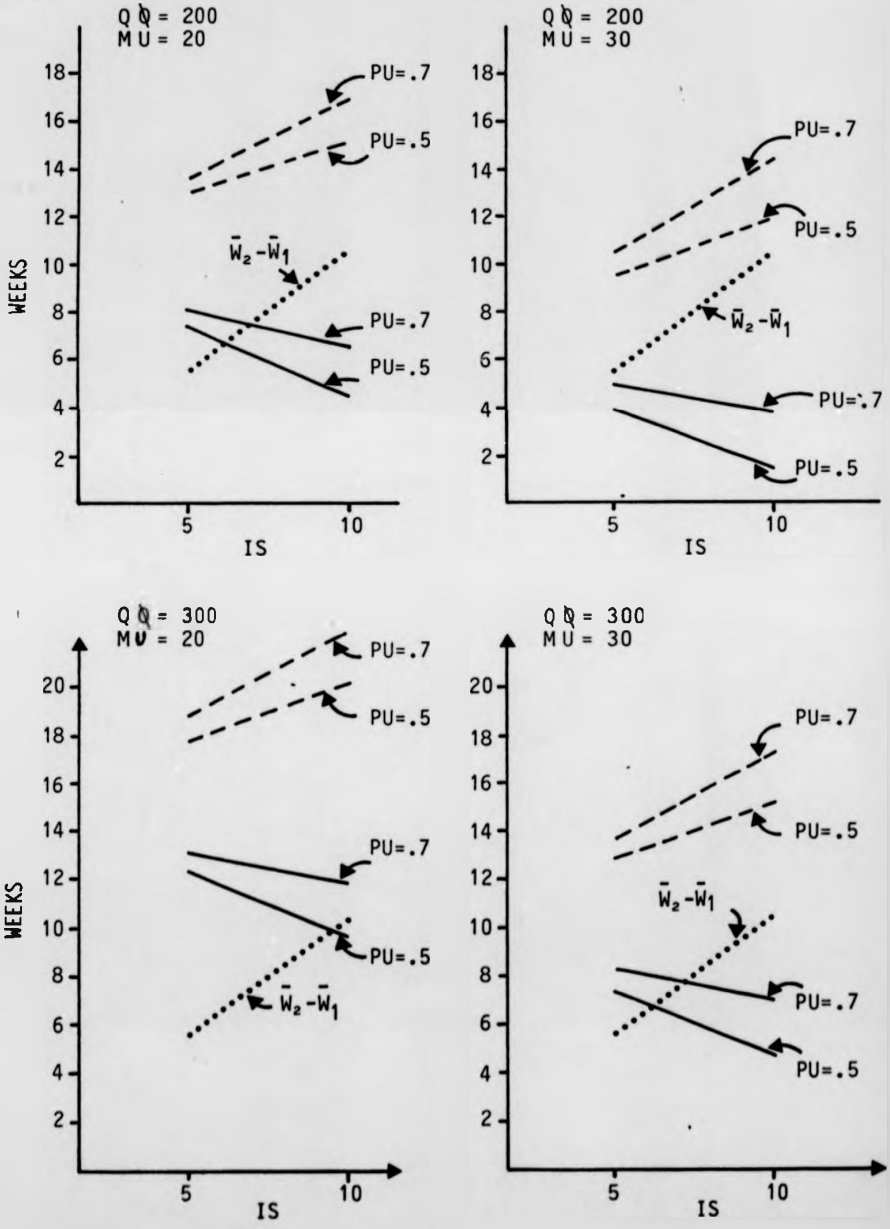
Source : Table 7.1

Graph 7.3

$\bar{W}_1$ by QQ, MU, PU & IS

——— Urgent Cases ($\bar{W}_1$)

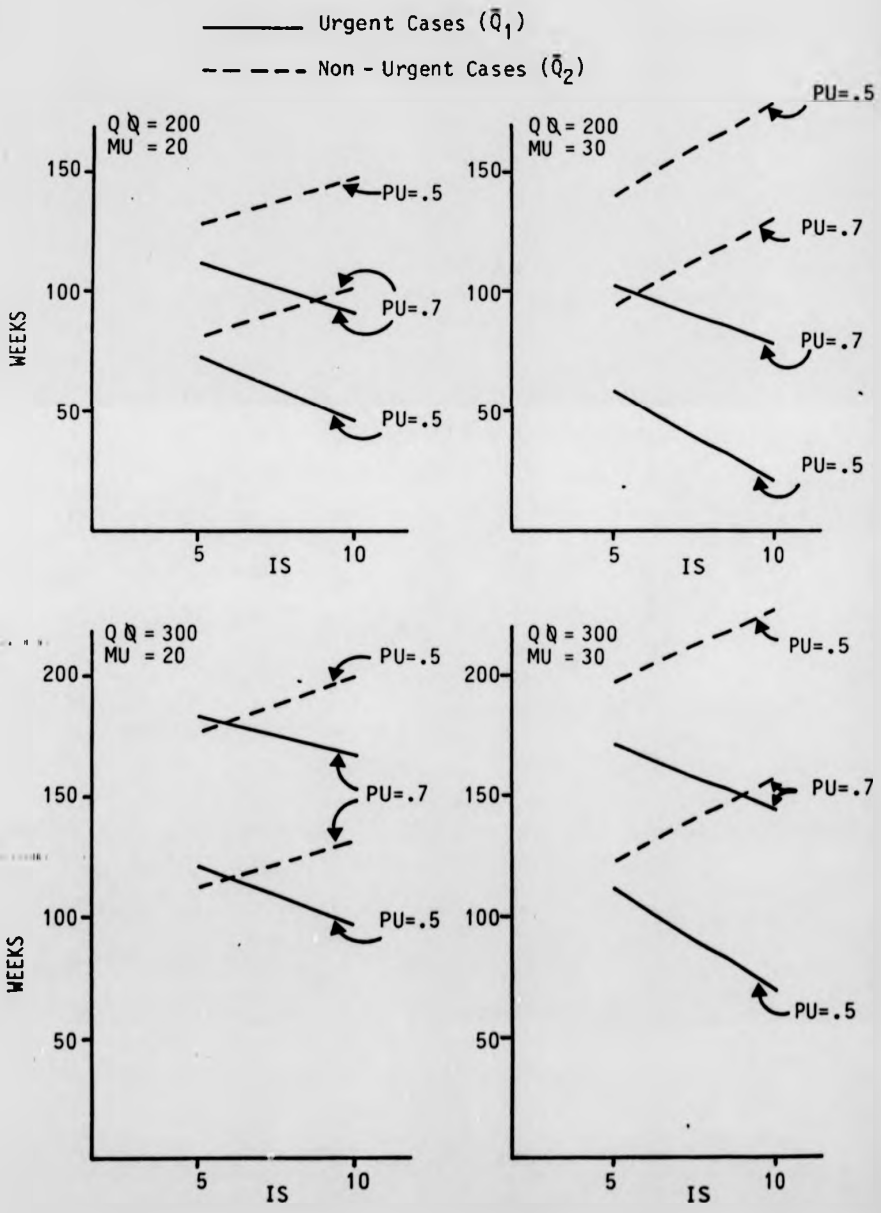
– – – – Non - Urgent cases ($\bar{W}_2$)



Source : Table 7.7

Graph 7.4

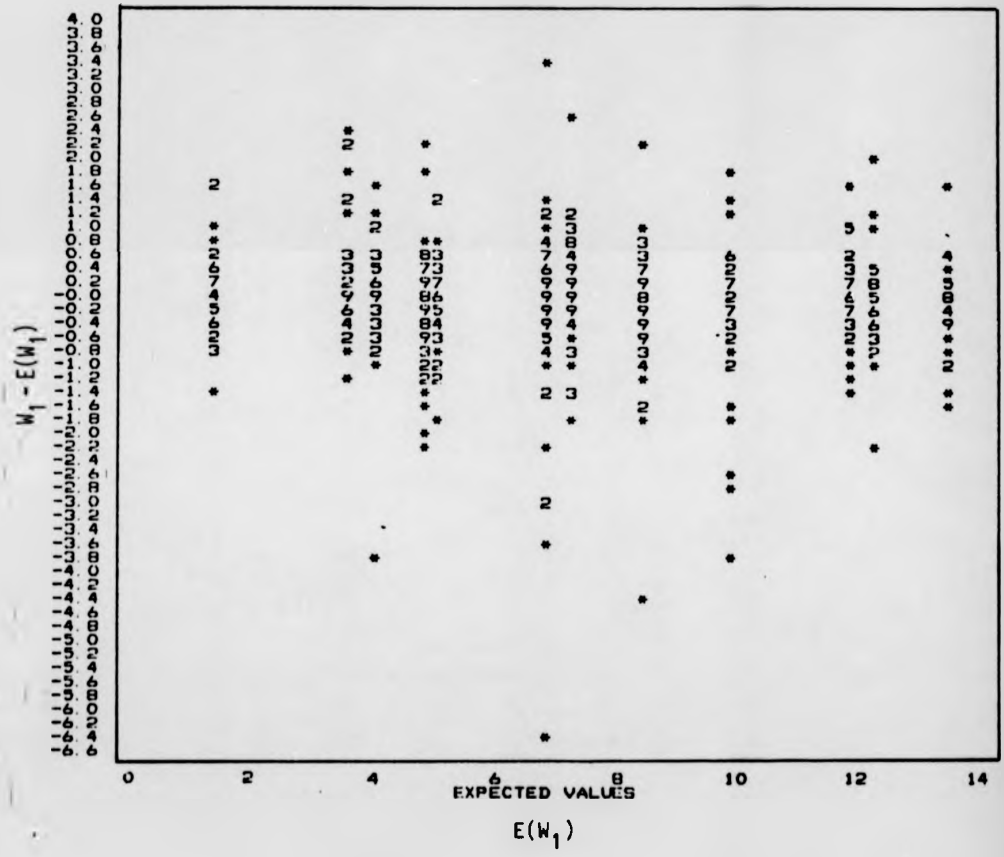
$\bar{Q}_i$ by QQ, MU, PU & IS



Source : Table 7.7

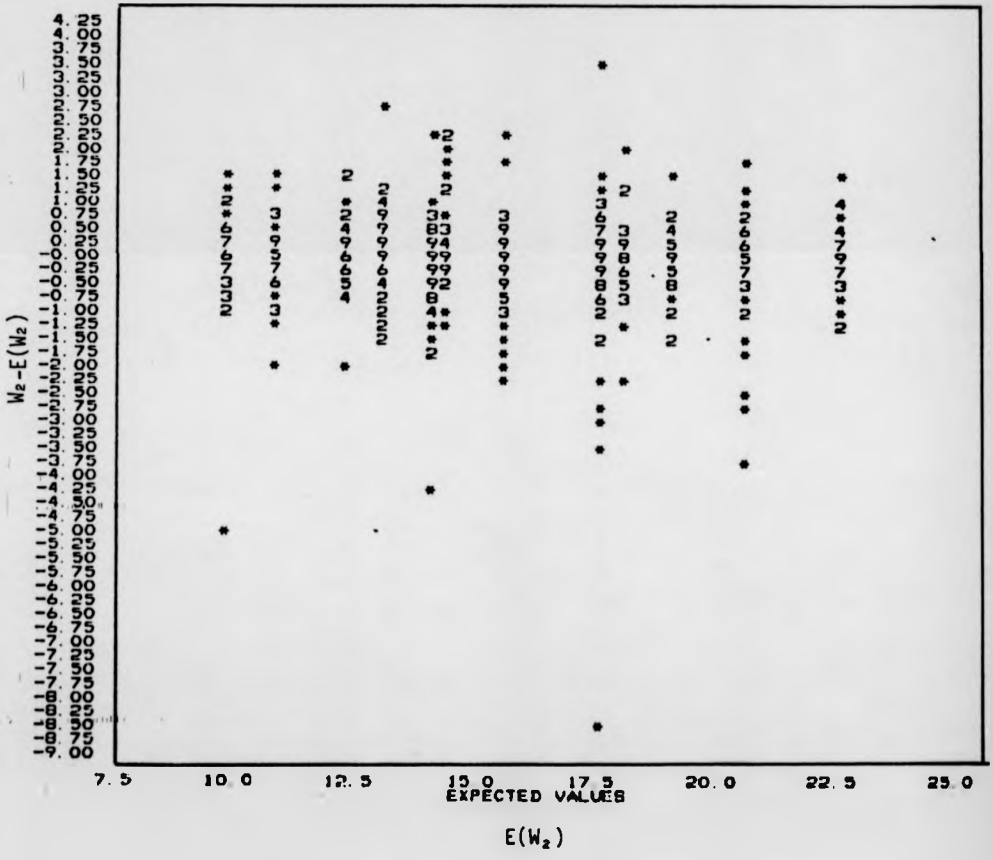
Graph 7.5

Mean Waiting Time for Urgent Cases [$(W_1 - E(W_1))$ vs $E(W_1)$]



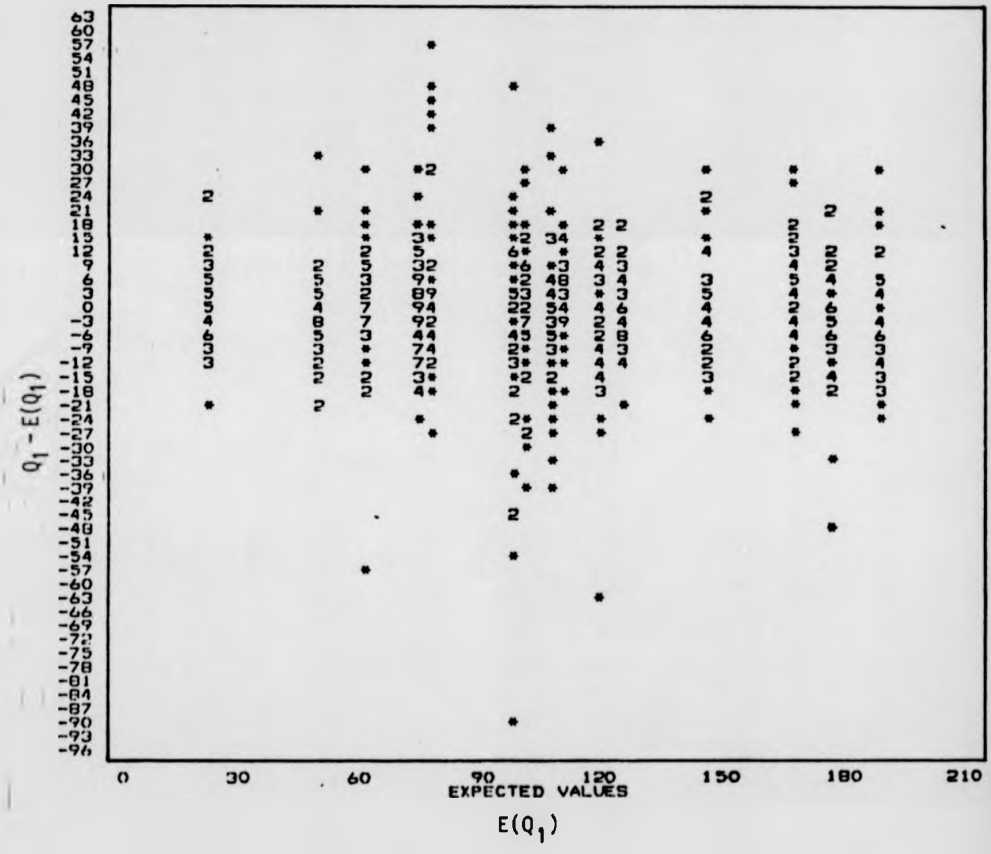
Graph 7.6

Mean Waiting Time for Non-Urgent Cases [$(W_2 - E(W_2)$ vs $E(W_2)$]



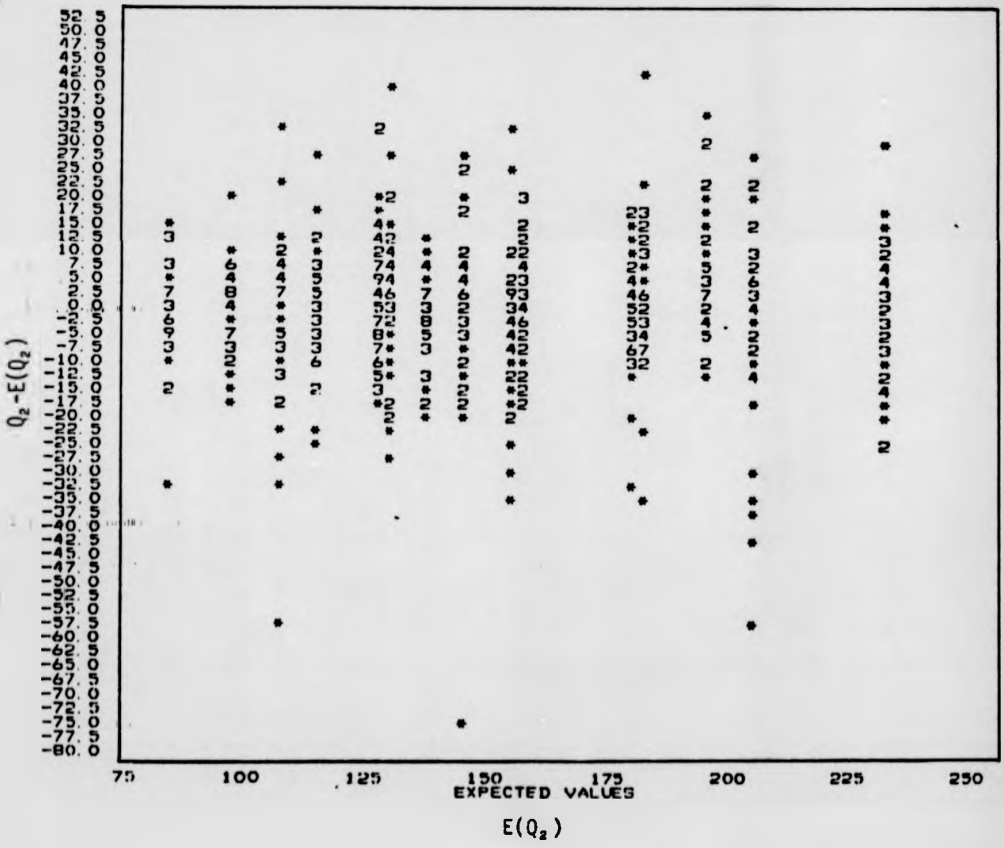
Graph 7.7

Mean Number of Urgent Cases on Waiting List [$(Q_1 - E(Q_1))$ vs $E(Q_1)$]



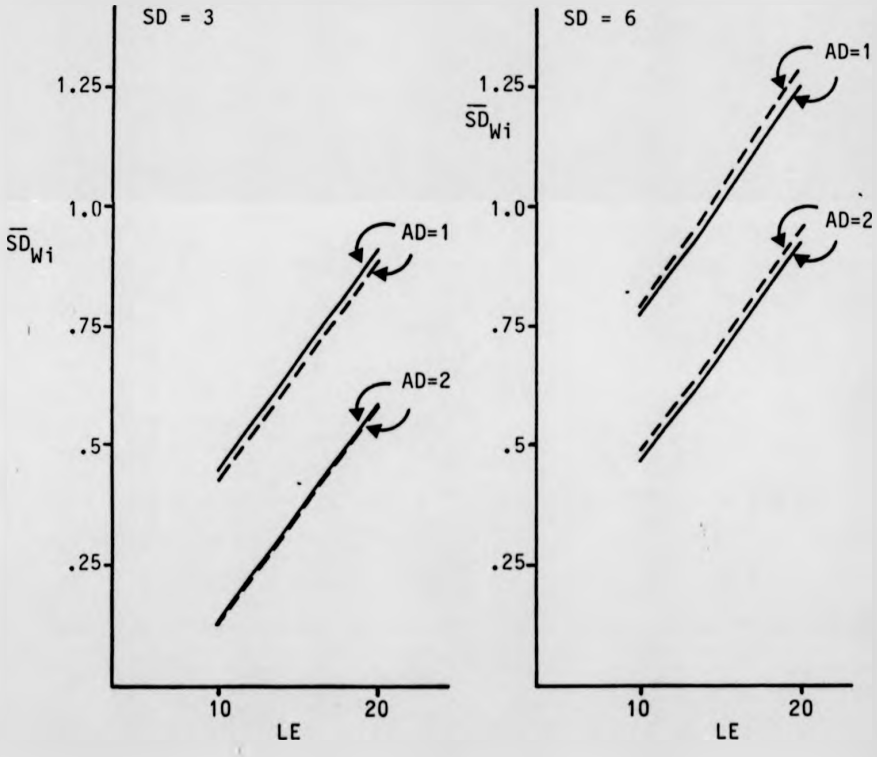
Graph 7.8

Mean Number of Non-Urgent Cases on Waiting List [$(Q_2-E(Q_2))$vs $E(Q_2)$]



Graph 7.9

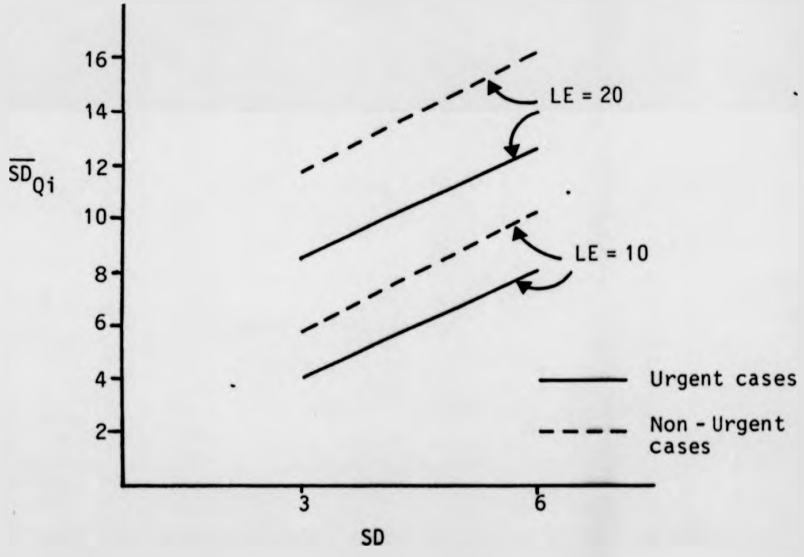
$\overline{SD}_{Wi}$ by SD, LE and AD



Source : Table 7.21

Graph 7.10

$\overline{SD}_{Qi}$ by SD & LE



Source : Table 7.24

REFERENCES

1 Alvey N., et al, An Introduction to GENSTAT, Academic Press, London, 1982.

2 Anonymous, "Any Way out of the Waiting-Lists?", The Lancet, Feb. 18, 1978, 397-398.

3 Anonymous, "The Mystery of Waiting Lists and Waiting Times", The Lancet, Nov. 14, 1981, 1122.

4 Anonymous, "How Secure are Figures for Waiting N.H.S. Patients?", The Lancet, Nov. 21, 1981, 1179-1180.

5 Armitage P., Statistical Methods in Medical Research, Blackwell Scientific Publication, G.B., 1971.

6 Bailey N., "A Study of Queues and Appointment System in Hospital Out-Patient Department, with special reference to Waiting Times", Journal of the Royal Statistical Society, Series B (Methodological): 14, 1952, 185-199.

7 Bailey N. and Thompson M., Systems Aspects of Health Planning, IIASA, Austria, 1975.

8 Barber R., "A Unified Model for Scheduling Elective Admissions", Health Services Research, Winter 1977; 12(4): 407-415.

9 Bhattacharyya G. and Johnson R., Statistical Concepts and Methods, Wiley, USA, 1977.

10 Blanco M. and Pike M., "Appointment System in Out-patients Clinics and the Effect of Patients Unpunctuality", Medical Care, Aug-Sept. 1964; II(3): 133-141.

11 Barron W., "FAILED APPOINTMENTS, Who Misses Them, Why They are Missed, and What can be Done", Primary Care, Dec. 1980; 7(4): 563-573.

12 Bigby J., et al., "Appointment Remainders to Reduce no-show rates: A Stratified Analysis of their Cost-Effectiveness", Journal of the American Medical Association, Chicago ILL, Oct. 7, 1983; 250(13): 1742-1745.

13 Brent R., "A Gaussian Pseudo-Random Number Generator [G5]", Communications of the ACM, Dec. 1974; 17(12): 704-706.

14 Burgoyne R.W., Acosta F. and Yamamoto J., "Telephone Prompting to Increase Attendance at Psychiatric Outpatient Clinic", American Journal of Psychiatry, Washington D.C., March 1983; 140(3): 345-347.

15 Buttery R. and Snaith A., "Surgical Provision, Waiting Times and Waiting Lists", Health Trends, 1980; 12: 57-61.

16 Chatfield C., The Analysis of Time Series: Theory and Practice, Chapman and Hall, London, 1975.

17 Coleman R., Stochastic Process, George Allen & Unwin Ltd., England, 1974.

18 Cullis J. and Jones P., "Inpatient Waiting: A Discussion and Policy Proposal", British Medical Journal, London, Nov. 12, 1983; 287(6403): 1483-1486.

19 Culyer A. and Cullis J., "Hospital Waiting List and the Supply and Demand of Inpatient Care", Social and Economic Administration, 1975; 9(1): 13-24.

20 Davis P., "Why dont Patients turn up?", Health and Social Service Journal, Basingstoke, July 26, 1984; 94(4907): 886-887.

21 Department of Health and Social Security, "Orthopaedic Services: Waiting Time for Out-patients Appointment and In-Patients Treatment", Report of the Working Party to the Secretary of State for Social Service, London, 1981.

22 Devlin H., "Programmed Elective Surgery", Harrogate Seminar Reports, Waiting for Hospital Treatment, DHSS, London, Jan. 1980; 18-25.

23 Deyo R. and Inui T., "Droputs and Broken Appointments", Medical Care, Nov. 1980; XVIII(11): 1146-1155.

24 Dove H. and Schneider K., "The Usefulness of Patients Individual Characteristics in Predicting No-Shows in Out-patients Clinics", Medical Care, July, 1981; XIX(7): 734-740..

25 Draper N. and Smith H., Applied Regression Analysis, Wiley, USA, 1968.

26 Esogbue A., "Experiments on Scheduling Disciplines in Surgery: A Simulated Queueing Approach", OpSearch, 1971; 8: 264-280.

27 Fisher R. and Yates F., Statistical Tables for Biological, Agricultural and Medical Research, Oliver and Boyd, Tweeddale Court, Edinburgh, 1970.

28 Fordyce A. and Phillips R., "Waiting List Management by Computer", The Hospital, 1970; 6(9): 303-305.

29 Fox D., George J. and Cahuin R., "A Hospital throughput Model in the Context of Long Waiting List", Journal of the Operational Research Society, Oxford, Jan. 1983; 34(1): 27-37.

30 Frost C., "How Permanent are NHS Waiting List?", Social Science and Medicine, 1980; 14c: 1-11.

31 Frost C., "Surgical Waiting List: An Economics View", Harrogate Seminar Reports, Waiting for Hospital Treatment, DHSS, London, Jan. 1980, 44-53.

32 George J., et al, "The Long Long Trial.....", Health and Social Service Journal, Basingstoke, March 1982; 92(4788): 332-335.

33 Gillham G. and Arch D., "Review of Patient Time Spent ...in Out-Natal Clinics", Hospital and Health Services Review, London, June 1982; 78(6): 171-174.

34 Grower S., et al, "Improving Appointment Keeping by Patients new to a Hospital Medical Clinic with Telephone or Mailed Reminders", Canadian Medical Association Journal, Ont., Nov. 15, 1983; 129(10): 1101-1103.

35 Healy M. J. R., The Statistics of Linear Models, Preliminary Draft, May, 1980.

36 Healy M. J. R., Personal Communication.

37 John & Quenouille, Experiments Design and Analysis, Charles Griffin, England, 1977.

38 Kanon D., "Simulation of Waiting Line Problems in a Hospital Setting", Medinfo 74, North Holland Publishing Company, 1974, 503-507.

39 Kendall M., et al, The Advanced Theory of Statistics. 4th ed., Vol. 3, Charles Griffins, England, 1983.

40 Létourneau C., "Classification of Surgical Procedures", Hospital Management. July 1965; 99(7): 37-39.

41 Luck G., Luckman J., Smith B. and Stringer J., Patients, Hospitals, and Operational Research. Tavistock Publications, G.B., 1971.

42 Luckman J. and Murray F., "Organising Inpatient Admissions", Selected Papers on Operational Research in the Health Services. Edited by Barry Barber, Printed by the Operational Research Society, Birmingham, U.K., 1976.

43 Luckman M., et al, "Management Policies for Large Wards Units", Appendix 5, Control of Waiting List, Health Report No. 1. Institute for Operational Research, 1968.

44 Management Advisory Service, "Study of the Management of Inpatient Waiting List for Surgery, ENT, Plastic Surgery and Gynaecology in the Oxford and South Western Health Regions", Management Advisory Service (Oxford and South Western Health Regions), Cheltanham: Management Advisory Service, 1983.

45 Meissner L. and Organick E., FORTRAN 77, Featuring Structured Programming. Addison-Wesley, USA, 1980.

46 Montgomery D., Design and Analysis of Experiment. Wiley, USA, 1976.

47 Mood A., Graybill F. and Boes D., Introduction to the Theory of Statistics. 3rd ed., McGraw-Hill, USA, 1974.

48 NAG, Fortran Mini Manual, Mark 10, Published and Printed by Numerical Algorithms Group Ltd., U.K.

49 NAG, GLIM Manual, Release 3, (GM3), 1978.

50 Nash A. and Sewell W., Waiting List Management, Chapter 3, Computer in Health Care, Simposium Proceedings, heald at the North Staffordshire Medical Institute for Postgraduate Education and Research, Hartshill Road, Stoke-on-Kent, 16th-17th April, 1975.

51 Ottestad P., Statistical Models and their Experimental Application, Griffins Statistical Monograph & Course, London, 1970.

52 Phoenix C., "Waiting List Management and Admission Scheduling", Spectrum 71, A Conference in Medical Computing, Edited by M. E. Abrams, Butterworths, 1972, 75-85.

53 Rothamstead Experimental Station, GENSTAT Manual, A General Statistical Program, England, Oct. 1977.

54 Sanderson H., "What's in a Waiting List?", British Medical Journal, Nov. 6, 1982; 285: 1368-1369.

55 Searle S., Linear Models, Wiley, USA, 1971.

56 Silvey S., Statistical Inference, Chapman and Hall, G.B., 1975.

57 Snaith H., "Apply and Demand in the NHS", British Medical Journal, 28 April, 1979, 1159-1160.

58 Weaver P., "Waiting List: the Negleted Statistic", Dimension Health Care, May 1981; 58(5).

59 Welch J. and Bailey N., "Appointment System in Hospital Out-patient Department", The Lancet, May 31, 1952, 1105-1108.

60 West R. and Jenkins R., "Problems of Patients Waiting for Orthopaedic Out-patients Appointment", Hospital and Health Services Review, Harlow, May 1984; 80(3): 126-130.

61 West R. and McKibbin B., "Shortening Waiting List in Orthopaedic Surgery Out-Patient Clinic", British Medical Journal, March 6, 1982; 284: 728-730.

62 Wightman J., "The Management Problem of a rising Waiting List", Harrogate Seminar Reports, Waiting for Hospital Treatment, DHSS, London, Jan. 1980, 40-43.

63 Yates J., "In-Patient Waiting List Statistics: Inacurate or Misleading?", Harrogate Seminar Reports, Waiting for Hospital Treatment, DHSS, London, Jan. 1980, 5-13.

64 Yeates W., "Waiting List and Urological Cases", Harrogate Seminar Reports, Waiting for Hospital Treatment, DHSS, London, Jan. 1980, 26-28.

VIII) General Conclusions

Throughout this reseach we identified three main areas where simulation analysts have concentrated their efforts. These areas are:

i) Statistical Research.- Different statistical models are used to solve mathematical problems, and to evaluate the efficiency of generation of random variables (chapter II), pseudo-random number generators (chapter III) and variance reduction techniques (chapter IV)

ii) Applied.- Simulation techniques are used to solve complex management problems or to develop new management methods (chapter V).

iii) Software.- New computer programs with simulation purposes have been designed, not just for mainframes, but also for microcomputers. For an updated presentation of these languages see the journal SIMULATION, October 1985, where 65 languages for mainframes and 48 for microcomputers, including versions of GPSS, DYNAMO and SIMSCRIP for micros are described.

The above simulation areas are not mutually exclusive. Sometimes we could find the development of all of them in a single simulation work. In our waiting list problem, we concentrated our efforts not only in the application of simulation to a health problem, but also in

the design of a computer program to select patients from a waiting list using a scoring system, and in finding the expected waiting time per priority when the traffic intensity in a queueing model was equal to one ($\rho=1$).

The common point between the three areas of simulation (statistical research, applied, and software) is the simulation model. The construction of this model is a difficult task to develop because of the two elements of conflict embodied in the model - realism and simplicity. We wish to construct a model of a real system that neither oversimplifies the system to the point where the model becomes trivial nor carries so much detail that it becomes clumsy and expensive to simulate. The tendency is nearly always to simulate too much detail rather than too little. Thus one should always design the model around the questions to be answered rather than imitiate the real system exactly.

Sometimes the questions to be answered in the study of a system can be found with the use of a mathematical theory, such as: queueing theory, linear programming, inferential statistics, probability theory, and differential equations theory. The common objective of these

theories and simulation is the pursuit of scientific knowledge about the behaviour of a given system. Computer simulation becomes a relevant tool for system analysis when (see figure 8.1):

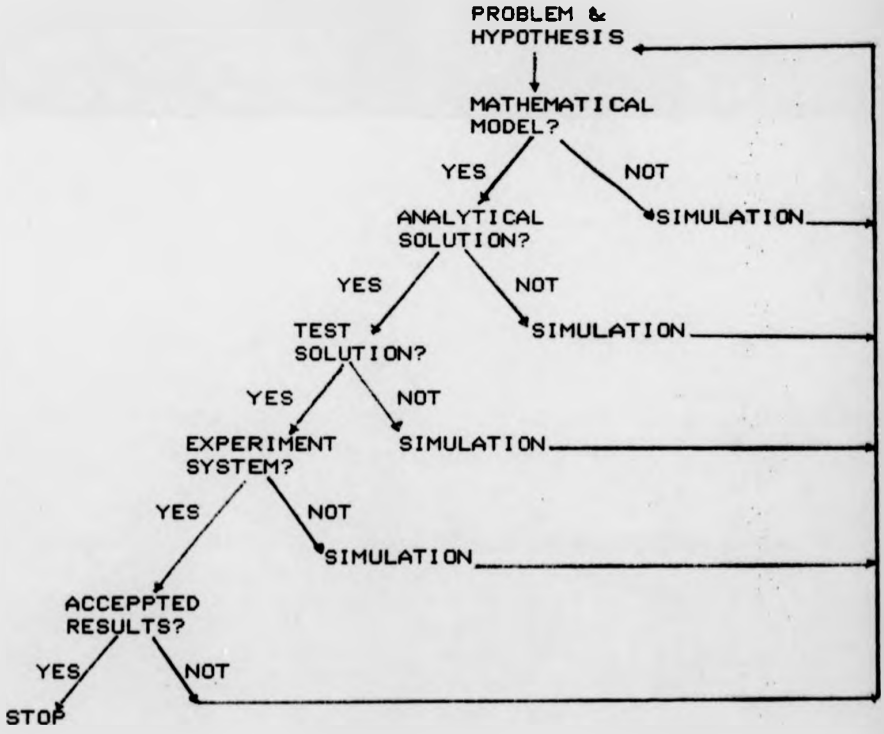
1) It is difficult to find an explicit mathematical expression which can represent the behaviour of a system. For example, an expression to explain the psychological behaviour to receive medical care in a community with private and government health services and with their own customs to treat themselves for some diseases (i.e., by using herbs, or by visiting a healer).

2) Once a mathematical model has been defined, it is difficult or impossible to find an analytical solution. For example, a system may be represented as a queueing model, but if the traffic intensity is one, the current concepts of queueing theory can not be applied (chapter VI).

3) It is very difficult or expensive to test the analytical solution in the real world.

4) It is simply impossible, impracticable, or uneconomic to conduct a controlled experiment in a real system. For example, to experiment with a new management procedure in an emergency department can produce serious risks in the health of the patients.

figure 8.1 System Experimenting Process and Simulation



Therefore, following figure 8.1, instead of calling simulation as a tool of 'last resort' , we consider that it should be called as an 'alternative resort' in the system experimentation process.

Although the principal reasons for choosing simulation are presented in figure 8, there are several other reasons, which are described in the following lines:

i) Simulation can be used as a pedagogical device for teaching both students and practitioners basic skills in theoretical analysis, statistical analysis, and decision making.

ii) The experience of designing a computer simulation model may be more valuable than the actual simulation itself. The knowledge obtained in designing a simulation study frequently suggests changes in the system being simulated. The effects of these changes can then be tested via simulation before implementing them on the actual system.

iii) Simulation can be used to experiment with new situations about which we have little or no information, so as to anticipate what may happen.

iv) Simulation can serve as a 'preservice test' to try new policies and decision rules for operating a system, before the risk is run of experimenting on the real system.

v) Simulation enables one to study dynamic systems in either real time, compressed time, or expanded time.

vi) When new elements are introduced into a system, simulation can be used to anticipate bottlenecks and other problems that may arise in the behaviour of the system.

Simulation is indeed a very versatile tool. However, it is by no means a panacea. Simulation is inherently an imprecise technique. It provide only statistical estimates rather than exact results, and it only compares alternatives rather than generating the optimal one. Furthermore, simulation is a slow and costly way to study a problem. It usually requires a large amount of time and expense for analysis and programming, in addition to considerable computer running time. Finally, there is no "cook book" type of approach to computer simulation. The approach taken by the researcher to statistical design and analysis of simulation studies must result from the unique character of the problem at hand.

To conclude, it is important to say that simulation studies are not a substitute for competent and efficient management; they only provide information that can be used in the decision-making process, but they do not eliminate or reduce the need for decision-making, nor substitute the skill or judgement of decision-makers.

ACKNOWLEDGEMENTS

My biggest debt of gratitute is to Professor M.J.R. Healy, head of the medical statistics department, for his comments and suggestions throughout my studies.

I would like to thank the computer services unit staff, specially to Sam Salem, for the advice to carry out the computing work. Thank to Allison Douglas for her efforts to improve my literary style.

I must thank in particular my wife Lizette who ineffable kindness and patience helped me typing and reading this thesis, and who gave me a lot moral support.

This research would not have been possible without the financial support of Consejo Nacional de Ciencia y Tecnología (CONACYT) of México, to which I am indebted.