IMPERIAL COLLEGE OF SCIENCE AND TECHNOLOGY

(University of London)

Department of Management Science

COMPUTER-BASED FACILITY SCHEDULING

by C. D. J. Waters

A thesis submitted in fulfilment of the requirements for the M.Phil

January 1976

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

Many individuals have offered advice and assistance in the preparation of this report. I am extremely grateful to all these people.

In particular I would like to thank my supervisors Prof. S. Eilon and Dr. J. O. Jenkins for their continued support and encouragement.

The BBC has also been extremely helpful in the provision of, and offering advice towards the solution of, a suitable practical problem. I would, therefore like to thank the BBC in general, and in particular M. Graham, C. Lashmar, G. Pexton and T. Smith, for their assistance.

In addition this project could not have been completed without the continued support, both moral and financial, given by Mrs. E. A. and Mrs. E. N. Waters.

Finally my thanks are extended to Mrs. C. Robb who typed and set-out the report in its final form.

SYNOPSIS

This report considers the formulation of a scheduling sub-system, which may be attached to an existing computerised Management Information System, in order to enhance the overall operation.

The general scheduling problem to be tackled is described, and an examination is made of the important features of computerised M.I.S.s. From these the overall form which the sub-system should have, in order to provide a satisfactory solution to the problem, is derived. In particular attention has been paid to the actual scheduling element of the sub-system, with several alternative forms considered.

In order to show that the sub-system developed is capable of practical as well as theoretical application a real world problem is examined. The modifications necessary to the sub-system, in order to move from a general to a specific problem, are then described, with the results observed.

Comparisons are made between the practical performance of the sub-system as compared with both theoretical results and the alternative manual scheduling approach.

SECTION A

THE FORMULATION AND SOLUTION OF A GENERAL PROBLEM

1. INTRODUCTION

1.1 Background to the Study

The growth in power of computers during past years has given an increased opportunity for organisations to benefit from the development of a completely integrated Management Information System.

This concept of an integrated system does not necessarily imply that such systems should either be designed or installed in a single step, but it does indicate that any new components of a developing system should be designed to interlock with those parts of the system already in operation.

In most real situations, for instance, the requirements of the system either grow, or only become known, after some period of operation, while the facilities and characteristics required of the system will almost invariably alter with time. Furthermore, on a practical level, the necessary resources (capital, labour and equipment) will not often be available at any one time, but will accumulate slowly over a period.

It may be implied, therefore, that while it is possible to consider a Management Information System (M.I.S.)

as a single discrete entity, incapable of either flexibility or growth, it would be foolish to do so. In fact, one of the major advantages of a well designed M.I.S. lies in its ability to make allowances for changes as they arise.

Because of the desirability of integrated growth there is a continuing demand for modules, or subsystems, which can be attached to an existing Management Information System to enhance its total operation.

This report describes a complete machine scheduling sub-system which can be used in certain cases for such a system expansion, and also describes a practical situation where the sub-system has been used.

1.2 A Summary of the General Problem Tackled

The problem considered is essentially one in which schedulers gather requests for machine usage from a number of sources, produce schedules as efficiently as possible within the constraints imposed, and distribute the results to interested recipients.

The situation can be represented diagrammatically as shown in Fig. 1.

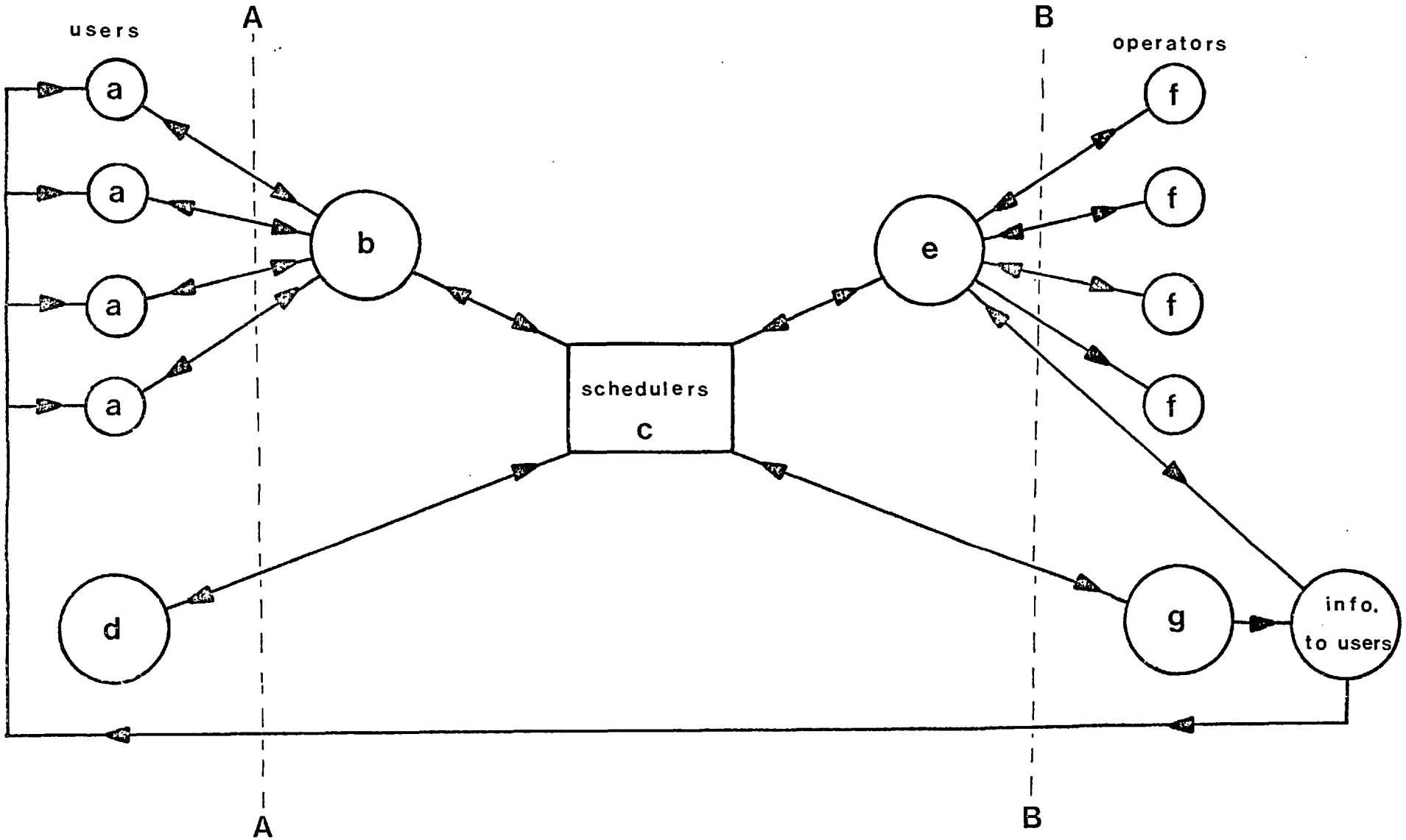


Fig 1 - a schematic representation of the problem tackled

In Fig. 1 many individual machine users (a) calculate their requirements for machine usage and send them to a central temporary store for information (b). At some stage the schedulers (c) pick-up the total information relevant to a particular period, together with the constraints imposed for that period (d) and produce schedules (e). These are sent to machine operators (who may in some cases also be the machine users) and associated departments (f). Any remaining unscheduled or unschedulable requests for machine usage (g) are dealt with as necessary.

Copies of the schedules prepared (e) and left-over requests (g) are sent to the machine users who act on this information as necessary.

In all cases two-way information flows are essential between the people involved in the scheduling process, in order that they may be kept fully informed about the state of the schedules and the alterations made.

The two interfaces between the scheduling sub-system and the larger Management Information System occur at 'A' and 'B'. Within these limits alterations can be made to the scheduling sub-system without any consequential effects on the M.I.S.. The contents of the inputs and outputs to the module are, however, fixed,

unless comparatively major changes are made to other parts of the system. Despite the fixed nature of these contents, variations are possible in the formats in which they are presented.

Within this simple framework several complicating factors are present in the problem considered which must be allowed for. The most important of these are:

1. Each request is different and has several variable characteristics, including type of processing required (i.e. function), duration, importance, number of machines used and amount of notice given before machine time is needed.

2. When scheduled the jobs are processed once only on one or more machines working simultaneously, although occasionally two requests are linked so that one must be finished before the other is started.

3. The machines are of different types, each of which is capable of handling some subset of the request functions, although no machine can handle all functions.

4. There may be a fixed or preferred start-time and/or machine associated with a request. In this context 'fixed' means that if the start time or machine

specified by the user is not available no other start time or machine is suitable and the request cannot be scheduled. 'Preferred' indicates that some other suitable start time or machine may be used if that specified by the user is not available, but the result will not be as satisfactory.

5. Before each job a variable period of preparation must be allowed before processing can begin. This time varies with function and machine.

6. Demand for machine time is heavy, usually being greater than supply.

7. The schedules for each period are issued to different people at different times, with the early schedules being distributed before the last requests arrive. Updating of the early schedules is required as changes are made to give the later results.

8. Discussions are necessary between the schedulers and the machine users to determine exact requirements for difficult or incompletely given requests. Similarly discussions are necessary between the schedulers and the machine operators in cases where machine usage is unusual or difficult to assess. The schedulers have to examine the requirements of the many would-be machine

users and balance them with the shortage of available machine time. Where agreement cannot be reached between diverse opinions, the schedulers have to make subjective decisions, based on experience, to determine priorities and produce the most satisfactory results.

9. A high proportion of requests are altered by the users after they have been received by the schedulers.

10. The computing capacity available is limited, and it is desirable that a minimum of these resources be used in order to obtain results.

With these characteristics in mind, a more detailed representation of the problem can be evolved, as shown in Fig. 2. This describes schematically the activities of the three main participants in the scheduling process (machine users, schedulers and operators) during the time of a complete scheduling cycle.

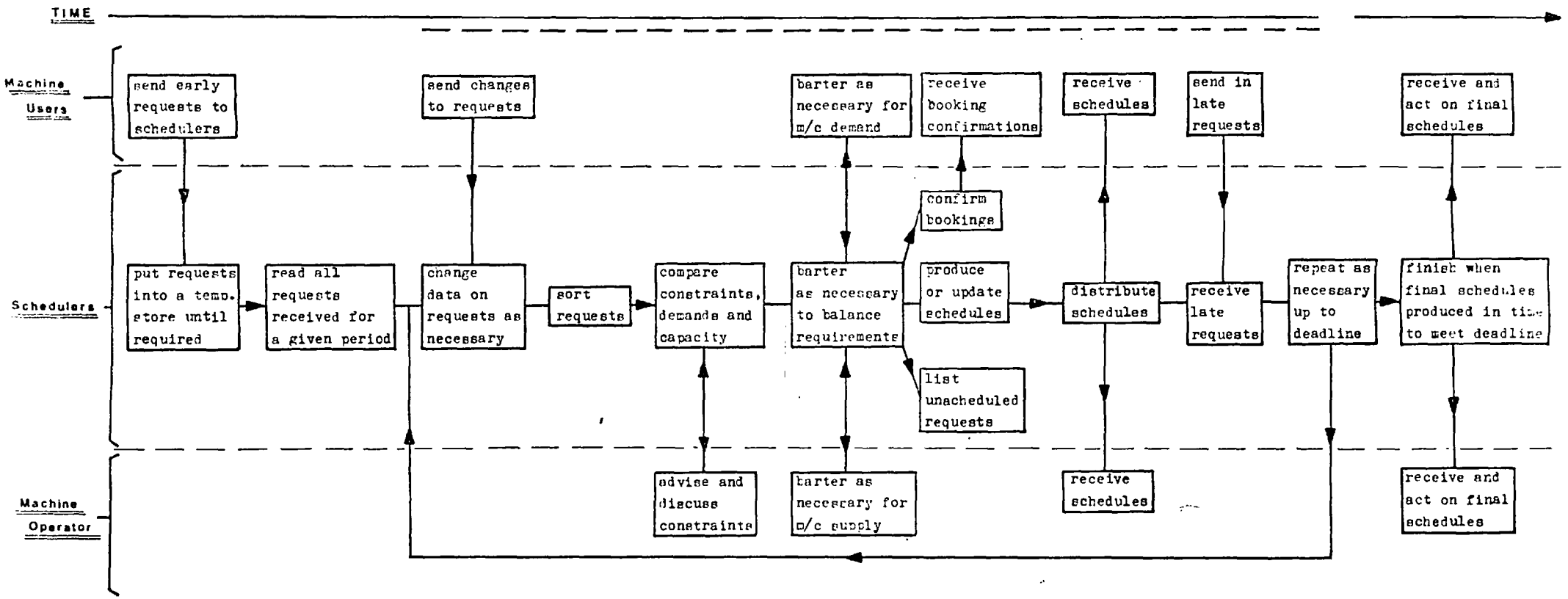


Fig. 2 - a representation of the activities of the participants in the scheduling process

The problem is to provide a computerised scheduling sub-system which will effectively replace an existing manual sub-system, with a minimum of alteration to the larger Management Information System, and in such a way as to improve the overall performance without reduction in flexibility.

The first step in the design of this sub-system is to examine in some detail the structure of a Management Information System (M.I.S.). From this examination the best approach to the formulation of a suitable sub-system will be more clearly discernable.

2. MANAGEMENT INFORMATION SYSTEMS

2.1 Definition and Basic Concepts

If a number of the different definitions proposed for 'Management Information System' are considered it is found that although they vary in detail there is general agreement on the principles involved.

For example Radley[14] wrote, "A Management Information System may be defined as a system which collects pertinent facts relating to the external and internal operations of an organisation, and which converts the collected data into information formats which are both relevant and meaningful to the needs of the organisation. The information stored should be readily available to management to aid them in their decision making and control processes."

Chandor[3] comes to the conclusion that a M.I.S. is, "A system which may perform routine commercial processing functions, but which is designed so that much processing will also produce information that will be presented to management ..... to assist in decision making. The implication is that the results will be produced speedily, perhaps requiring real-time processing, to enable management to ascertain the progress of the organisation in terms of satisfying its major objectives."

These two definitions indicate that the main concepts involved in a Management Information System are:

(a) the collection and storage of raw data

(b) the processing of this data, where necessary, to produce basic information

(c) the sorting of this information so that it is readily available for management to aid their decision making.

It is in the first, and more particularly the second, of these that the correspondence between a M.I.S. and a data handling system can be observed. A M.I.S. may, for example, be regarded as a practical extension of a basic data handling system, where refinements are added as necessary to allow for the type and function of the data processed.

Although the concepts mentioned above are of general applicability, it is this element of data handling which has led to the introduction of computer assistance in most of the larger Management Information Systems. The ready availability of computers and the unsuitability of manual methods for processing large amounts of data have meant that computerised M.I.S's are the only realistic alternative in medium-sized or larger organisations.

2.2 Computerised Management Information Systems

A computer is the only means of efficiently handling data in a large-scale or complex information system. For some purposes manual procedures cannot cope with the amount of processing to be done in the time available, while at other times the complexity is such as to make manual processing impossible. In either case computer processing would provide the most practical solution.

The overall advantages that computer operations may bring to a system will depend to a large extent on the function and organisation of the system. Firnberg[8], for instance, summarises the benefits which can be gained from using a computer in a M.I.S. as follows:

"Firstly, a virtually unlimited volume of data can be received and stored .....

"Next, this data can be processed and innumerable calculations undertaken at fantastic speeds .....

"Finally, data and information can be made available, transmitted and presented in a variety of ways and over great distances ....."

In the same manner Dearden[4] says, "A computer is able to store vast amounts of data, to retrieve this data quickly, and to do arithmetic and logic operations

at a speed measured in millionths of a second. And it can perform these operations practically without machine error."

On a more practical level Lock[12] has stated the primary advantage of computerised scheduling to be, "the ability to process large volumes of data with low risk of error and in a short space of time ..... (which) enable a planner to produce his schedules with a speed and accuracy which would otherwise be impossible."

Obviously many more such quotations can be given, but the main benefits to be derived from the use of a computer in a Management Information System are clear from these indications. Computer use will make available:

(a) a very large, compact storage capacity
(b) the facility for the rapid retrieval of stored data
(c) the ability to do a large number of calculations very rapidly
(d) negligible risk of machine errors during processing
(e) the ability to transmit results 'instantaneously' to remote sites
(f) the formats of outputs can easily be varied to suit individual needs or situations.

In addition to these considerable benefits the use of computers has several drawbacks which must also be examined. These disadvantages arise from both the

inherent characteristics of the machines, and from the way in which the systems are designed and operated. It is thus possible to classify them according to whether the disadvantage is primarily associated with hardware or software.

The main drawbacks associated with computer hardware are that it is expensive to acquire and maintain (although not necessarily more expensive than any possible alternative), comparatively difficult to communicate with by untrained people, inflexible in operation (for instance computers cannot make subjective decisions, correct 'obvious' mistakes, easily accept verbal or handwritten inputs or produce 'original' responses) and is prone to breakdown.

The other type of complaint brought against computers is mainly due to the design of the software used. It is common, for example, to find computer systems;

(a) evolved from more than one system and not combined into one integrated unit
(b) do little more than reflect manual processes which are deficient in themselves, and unsuited to computer applications.
(c) emphasise short term control over long term planning
(d) inflexible and slow to respond to environmental changes
(e) produce too much information

(f) designed by computer personnel with more regard to producing a 'good' system than satisfying management needs.

On evaluation of the relative advantages and disadvantages of computer usage, it is generally felt that computers provide the best solutions for fairly large or complex problems where subjective decisions do not have a major importance, and particularly where routine processing is involved. Manual methods are better for simpler problems and situations where subjective decisions are more relevant.

It may be noted, however, that even in cases where a computerised system would obviously provide the best solution to a problem, its introduction is not always welcomed by those concerned. As Lock[12] said, "The introduction of a computer ..... is likely to give rise to some apprehension, if not actual hostility."

It is an obvious comment that an otherwise perfect system is useless if nobody wants to use it.

The reasons for this 'apprehension' are varied, but the following factors may be important:

(a) resistance to change from old familiar practices
(b) the aura of mystery surrounding computers
(c) the tales of disasters associated with the implementation of computer systems

(d) the idea that a new computer will replace existing workers and possibly bring unemployment.

The overall conclusion which can be drawn from this brief discussion is that computers provide the best, if not the only, means of tackling some problems. However, the introduction of a new computer system must be done with care so as to take into account both management requirements and the general feelings of employees.

2.3 On-Line Management Information Systems

In his definition of a Management Information System Chandor[3] indicated that computerised systems often call on real-time (or possibly on-line) processing to speed-up the information flow. This technique was first made available in about 1959, when Gearing[9] reported, "We have received the first announcement of the Ferranti ORION data processing system ..... It would appear that on-line working will now be economical in this fast business machine, because ..... of the facilities for time-sharing and parallel processing."

In more recent years, however, it would be safe to say that most medium-sized computerised management information systems use on-line processing in one form or another as a matter of course.

The general characteristics of an on-line system (and in particular a real-time system) have been described by Dearden[5] as,

> "1) Data will be maintained 'on-line'(as opposed to magnetic tapes which have to be searched) - directly available in computer memory or in random access files.
> 2) Data will be updated as events occur (cf periodic updating of batch processes.)
> 3) Computer can be interrogated from remote terminals."

When batch processing alone is employed delays will inevitably occur at several points during the journey of instructions and/or data from the manager to the computer, and during the return of output information to the manager. On-line processing has been seen as the only way of avoiding these delays and transmitting information to and from the computer with sufficient speed.

Burck[2] summarises the situation for real-time processing by defining, "A real-time Management Information System (is) ..... one that delivers information in time to do something about it."

The use of on-line computing is not without its advantages and disadvantages, as is the use of any

computing technique. The major of these are listed in the following two sections.

(a) Advantages of On-Line Computing

i. 'dynamic' situations can be dealt with easily

ii. service to users is instantaneous

iii. many users may utilise the facilities simultaneously

iv. the central computer is a large general-purpose machine

v. it can be used by organisations with limited resources, as small amounts of computer time may be bought conveniently from a bureau

vi. compact and easily understandable video displays can be produced.

vii. it is useful in situations where one question has to be answered before the next can be asked

viii. it can be used for direct communication between many points

ix. uneven loading of facilities is effectively handled

x. the users' control over the system is more flexible as it allows the operating procedures to vary continuously throughout the processing.

(b) Disadvantages of On-Line Computing

i. large amounts of input and output cannot be handled easily at a terminal

ii. difficulties arise with jobs needing a very long C.P.U. time

iii. large amounts of storage are needed if many users are connected to the system

iv. there are problems associated with unauthorised access to private information

v. at times of high demand response time for even simple instructions may be slow

vi. higher capital costs are involved because of the need for terminals and additional software

vii. reliability is more important than for purely batch systems

The desirability or otherwise of using on-line computing will depend to a large extent on the use to which it is put. In general terms the rapidly increasing use of this technique indicates that there is a very large number of applications where the benefits far outweigh any drawbacks.

The type of terminal used for on-line computing again will depend on the use to which the system is put. Perhaps the most flexible types of terminal, however, are video-display units (V.D.U.'s). These allow very high speeds of data transmission, silent operation, ease of use (by, for instance, well designed 'forms' to be completed on the VDU screen), easy verifying of input data and so on.

Gladwin[10] has described one situation where VDUs have proved very successful. He says, "One of the most expensive and time consuming part of any computer organisation is the daily preparation of basic data and subsequently the reading of this data into the computer system. Key punch and key verifier operators are becoming more costly to employ and more difficult to engage." He concludes that in his opinion video terminals are the only satisfactory means of information transmission where large amounts of data have to be dealt with by inexperienced clerks.

It seems likely that in the future the benefits to be gained from the use of V.D.U.s will become increasingly apparent. As they become more sophisticated the cost for a given level of sophistication may be expected to fall and and they will become less expensive in comparison to the main alternative, the teletypewriter.

2.4 The Position of the Scheduling Sub-System within a MIS

It has already been said that the purpose of this report is to describe a scheduling sub-system which can be attached to an existing Management Information System without requiring major changes in this larger system.

Now, having examined the general nature of Management Information Systems, it is possible to determine

the position of a scheduling sub-system within a MIS, and to state the facilities which would be required by a user when considering the problem posed in section 1.2.

It is possible to summarise the main components necessary in a computerised Management Information System as;

(a) an information store or data base containing stored data relevant to the management function in an easily accessible form

(b) a means of maintaining this data base by inputing new or altered data

(c) the ability to make certain calculations and processing runs using this data

(d) a method of speedily obtaining the results of such calculations and transmitting them to the relevant management

In this list 'b' and 'd' may often be handled advantageously as on-line operations, unless there is some reason why this is undesirable (such as the production of too much output to be conveniently transmitted via a terminal). If on-line processing is prohibited then punched cards, line printers or some alternative input or output device must be used.

Again the observation may be made that a Management Information System contains a major element

of 'data processing'. For this reason it has been said previously that a MIS may be considered as a practical extension of a data processing system. It follows that any general comments made about a MIS may also be relevant for a data processing system.

It is possible to represent the elements of a Management Information System (and in particular a computerised MIS) as shown in Fig. 3. Any scheduling sub-system designed to expand an existing MIS must fit into this general structure. The problem is the method of connecting these various elements, and the determination of important points which need particular attention.

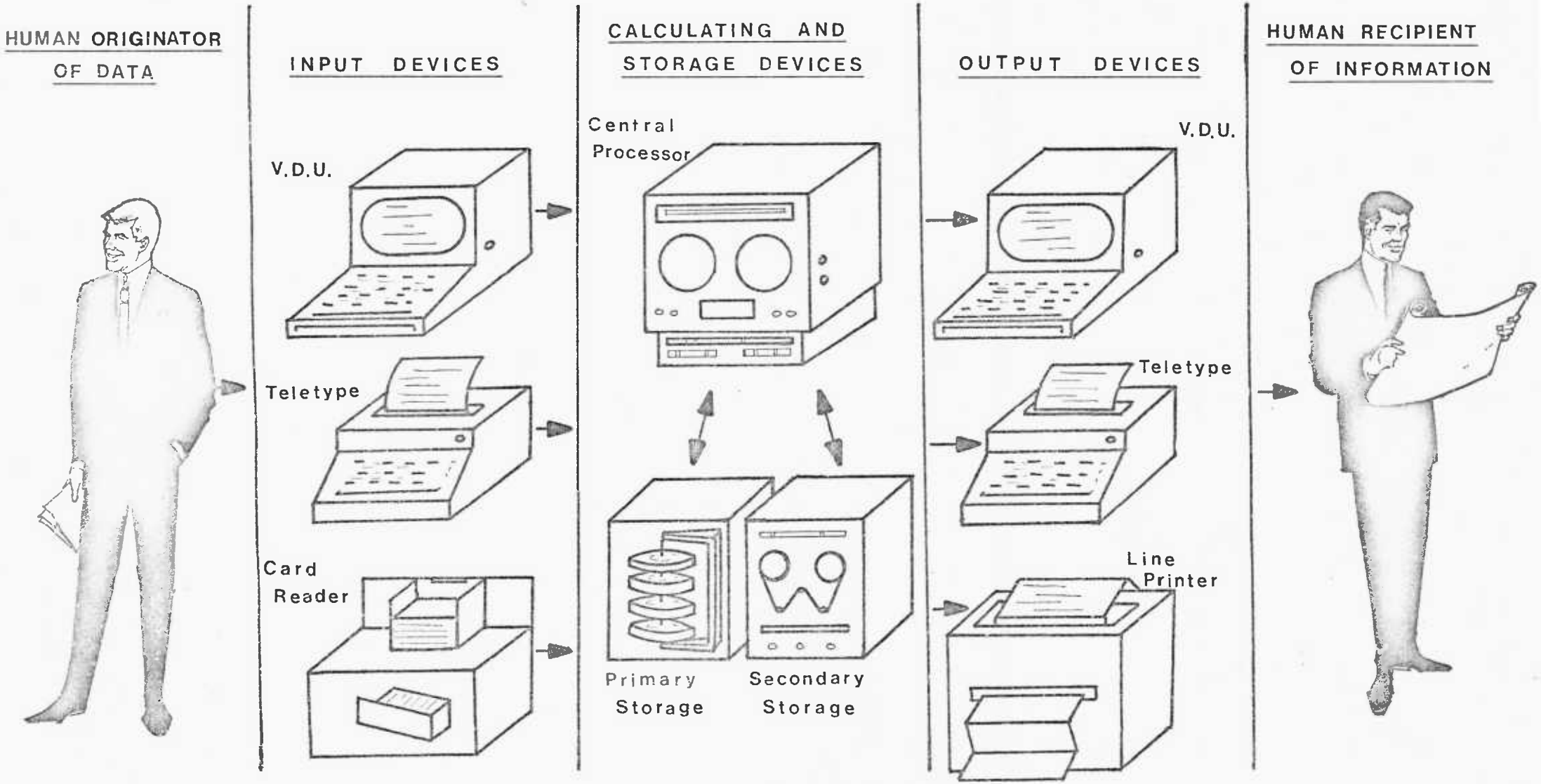


Fig 3 - showing the general structure of a Management Information System

3. A DESCRIPTION OF THE SCHEDULING SUB-SYSTEM PROPOSED

3.1 Initial Considerations

If the scheduling problem described in section 1.2 is again examined it is apparent that the total sub-system must be able to perform the following functions:

(a) input booking requests for machine time into a computer file

(b) alter requests as necessary after they have been entered

(c) delete (and possibly later replace) requests from the file

(d) search for and print details of a particular specified request, or group of requests which share some specified characteristic

(e) produce initial schedules for the machines

(f) allow manual alteration of these initial schedules in cases where subjective decisions are necessary, or where late requests are received which must be scheduled but do not warrant a total rescheduling

(g) make available at any time statistics of machine loading, time available etc.

(h) alter the constraints and conditions of the scheduling process as needs arise

In the design of the scheduling sub-system it is necessary to produce an information flow which will both satisfy these requirements and fit into the general structure of Management Information Systems as illustrated in Fig. 3.

An illustration of the sub-system designed for this purpose is shown in the form of an information flow diagram in Fig. 4.

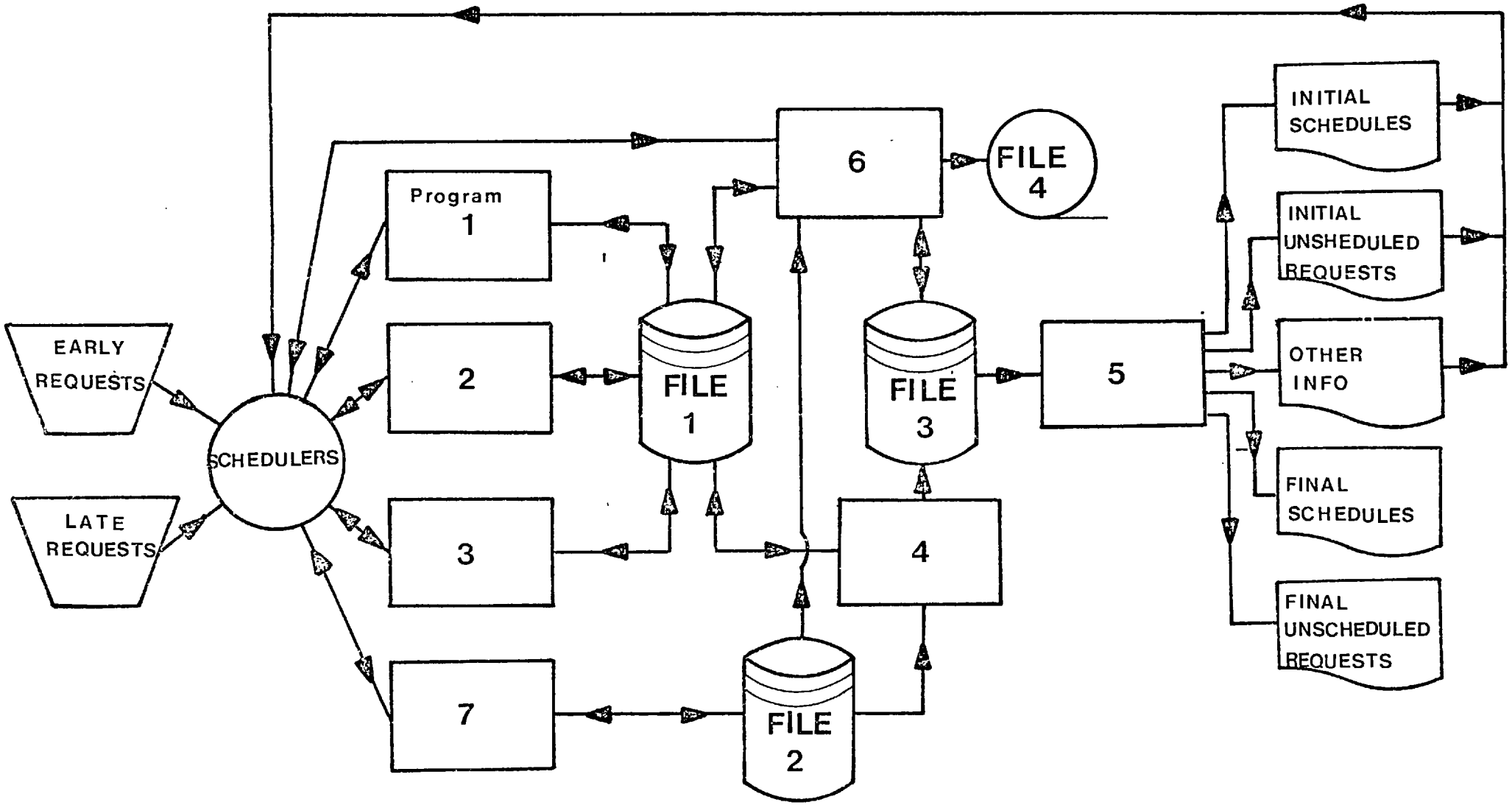


Fig. 4 - the information flow diagram for the scheduling sub-system developed

In practice all elements of this sub-system will be required on-line at all times. There will be no straightforward pathway taking each element in turn and then passing on to the next in a predetermined sequence, but the facilities will be used as requirements dictate and in a more or less random order.

In order to describe this diagram more easily, however, the course through a simplified hypothetical scheduling cycle is examined.

Early requests for machine time are sent by would-be machine users to the schedulers. These use Program 1 to input all the available information to the Master File (File 1).

Any later alterations or additions to the requests already on the Master File are made using Program 2.

Program 3 is needed to feed-back to the schedulers details of requests, loads, utilisations and any other information they may need.

At some point Program 4 is directed to read from the Master File all relevant information concerning requests for a particular scheduling period, and from File 2 the constraints imposed for this period. The initial schedules are then automatically produced and stored internally in File 3, to be referred to as necessary.

The Report Writer, Program 5, lists the initial schedules, together with requests which cannot for any reason be scheduled automatically. Further details of machine utilisation etc. are also given at this point, and the results are returned to the schedulers.

Almost inevitably late requests will be sent to the schedulers. After examining these late requests, together with the unscheduled requests from the automatic scheduling run, the schedulers will decide either to do a complete automatic reschedule (if there are a large number of important requests to be considered), or to update the initial schedules by hand (for comparatively few extra requests).

For manual updating of the initial schedules Program 6 is called, which firstly reads the constraints from File 2, together with the initial schedules from File 3. The schedulers then direct the program to make changes where desired, and with reference to the Master File, to produce the Final Schedules. Any proposed move which would be outside the imposed constraints, or contains some other error is automatically rejected.

File 1 and File 3 are updated with the alterations made while Program 5 is again called to print the final schedules and results.

A permanent record of the results obtained is kept on the tape File 4, which can be kept and referred to as required.

Inevitably the constraints imposed on the scheduling will change from time to time, and in order to keep the sub-system working efficiently File 2 must be updated as necessary, using Program 7.

All updatings and alterations of files and schedules may be repeated whenever necessary.

3.2 An expanded Description of the Basic Sub-system

One of the first decisions which has to be made with any such system expansion is the balance between the level of automation and human discretion. A balance has to be drawn, in this case, between doing too much by machine (and therefore not allowing sufficient flexibility in the system), and leaving too much to the schedulers (and thereby not reducing their workload sufficiently to justify the cost of the sub-system).

In the solution illustrated it can be seen that the schedulers have complete control over the system, but by doing all the processing necessary to prepare the initial schedules automatically a large proportion of their routine workload is removed. At the same time leeway is available to alter the schedules manually in

the light of unusual circumstances or requirements.

The schedulers communicate with the computer via terminals, in an on-line mode, and where possible the outputs are returned through these terminals. However, as there is a large amount of information produced with the schedules this particular output would be best directed to a line printer. In order to reduce delays this should be done at some point where the delivery time is not critical. For example a command to initiate the automatic scheduling run at the end of a working day could make the print-out available before the schedulers arrive at work on the following morning.

Following the comments in section 2.3 to the effect that video display units provide a useful means of communication in on-line operations, it is thought that they will be particulalry useful in this context. Here a fast response, silent operation in an office, often having small amounts of output and often requiring no hard copies of output make a VDU both feasible and desirable in this application.

There are basically three types of processing to be considered in the proposed sub-system:

(a) completely manual

(b) on-line computer opeations using video display units

(c) computer processing runs, where the input is read from data files and the output is sent to the line-printer.

The first of these is external to the computer system and involves form handling, the posting of schedules and other similar functions. This will, to a large extent, be fixed by the rest of the information handling system.

The second, which can be referred to as the 'on-line' or more specifically the 'interactive part' of the sub-system, updates and interrogates the data files whenever necessary.

Finally, a 'batch part' contains the scheduling algorithm and report writers. As all contact between the schedulers and the computer is via a terminal, this should be referred to more fully as a 'terminal-initiated batch processing section'.

The original information flow diagram shown in Fig. 4 can now be redrawn to illustrate the mode of operation present, and allow for the addition of two extra programs needed to overcome the difficulty of

printing large amounts of information on a terminal printer.

In Fig. 5 Program 8 calculates and prints an analysis of the schedules stored in File 3, while Program 9 lists the input data stored in the Master File, in any desired format. All other files and programs are the same as those described in Fig. 4.

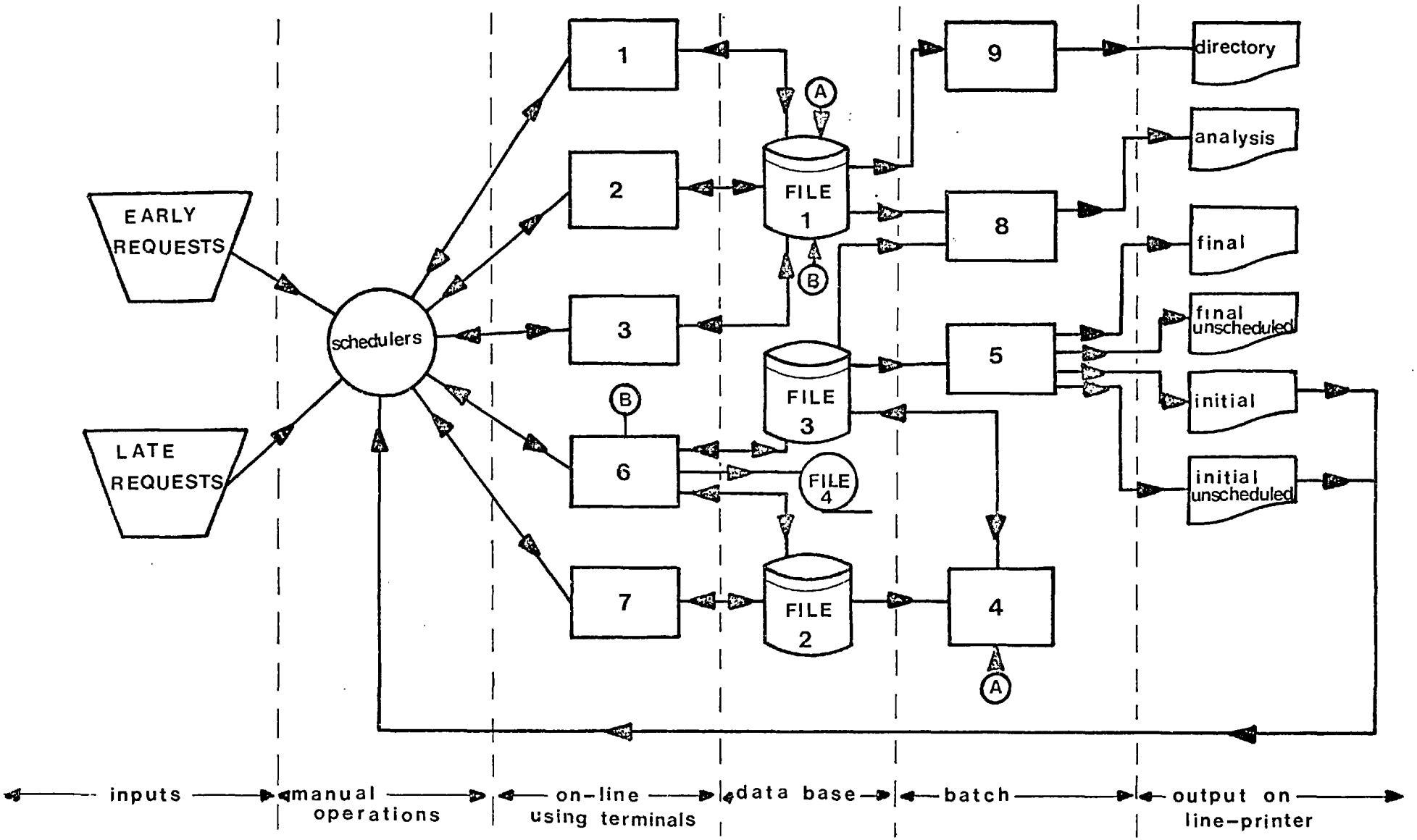


Fig. 5 - the modified information flow diagram showing the mode of operation

4. THE ELEMENTS OF THE PROPOSED SCHEDULING SUB-SYSTEM

4.1 Manual Operations

These are the parts of the sub-system which do not involve the use of computers. In particular they include sections where the schedulers receive and transmit documents (either written or verbal), and where they work on these documents by hand. Such operations include the posting and receipt of input data by the schedulers, the manual sorting of this data and the return of the initial and final schedules from the line-printer.

The information involved in these parts of the sub-system must, to a large extent, be considered as fixed, as no alterations can be made here without affecting the overall information flow in the MIS. However, a distinction may be made between the input information sent to the schedulers by would-be machine users, and the output describing the results of the scheduling runs as returned by the schedulers. In the former case the information handled must be considered as fixed (unless extra work is undertaken to produce additional inputs), while in the latter case the present level of information flow can be considered as the minimum acceptable level.

Additional output information may be made available, but it should be noted that with the use of computers in the

sub-system it is very easy to produce large amounts of data. The temptation to distribute the output more widely, and consequently perhaps produce more wastage of time and resources, should be avoided, with relevant results being distributed only to those whose jobs would be made easier by their receipt.

It must be assumed here that no major reorganisation is possible which will alter either the sources of the input data (i.e. the would-be machine users and the machine operators) or the destinations of the finished schedules.

With the introduction of a computerised scheduling sub-system the actual handling of forms by the schedulers should be reduced considerably. Instead of producing the complete schedules by hand they will basically be required to do manually only the initial sorting of data, with any translations necessary to give inputs to the computer, and later examine the printed schedules. All other manipulations can be done automatically, or with computer assistance. Once the data has been input to the computer the original forms are not needed and may be kept as records for as long as necessary.

4.2 The On-line Computer Operations

As video display units are to be used for all on-line operations there are several factors to be considered:

(a) a VDU works on a 'page at a time' basis (when used as a pure VDU and not as a teletype compatible VDU). Hence a large amount of information should be put onto each screen before it is transmitted. This minimises the delays caused by slow responses, thinking time etc.

(b) overwriting of information on a VDU screen is possible. 'Forms' may therefore be displayed and completed by filling the relevant sections, corrections may be made by overwriting and so on.

(c) there is a limited amount of space on a VDU screen (usually 80 characters by 25 lines, or 56 characters by 18 lines). Coding of information is therefore desirable, both to reduce typing time and also to allow more information to be put onto each screen.

In order to simplify the further examination of this part of the sub-system it can be divided into two parts:

(a) the part of the sub-system which is concerned with the upkeep and interrogation of data files (Programs 1, 2, 3, and 7 in Fig. 5)

(b) that part which makes alterations to the automatically produced initial schedules (Program 6).

The purpose of the individual programs has been described in section 2, but further consideration of the form they should take and the facilities which should be offered are given in the following sections.

4.2.1 On-line File Updating and Interrogation Programs

Program 1

This is the program which is used to input information about requests for machine time onto the Master File (File 1).

The design of the input form will, of course, depend upon the application. However, there are some general guidelines which may be considered for all circumstances. For example:

(a) the use of 'forms' which can be filled with required information often gives the most satisfactory results.

(b) 'free formats' will save both time and trouble if they can be used

(c) if possible it is beneficial for all information about a particular request to be fitted onto a single VDU page.

(d) the data should be compact to speed-up the time needed to type the input, and also to reduce the chance of miscounting spaces. Coding is desirable wherever possible.

(e) as much information as possible should be derived automatically or assumed by the computer

(f) error checks and validations should be performed at the input stage so that mistakes can be rectified at once (while the scheduler can still remember the request, and before errors are introduced into the system.)

A simple example of a possible screen format which incorporates most of these features is given in Fig. 6.

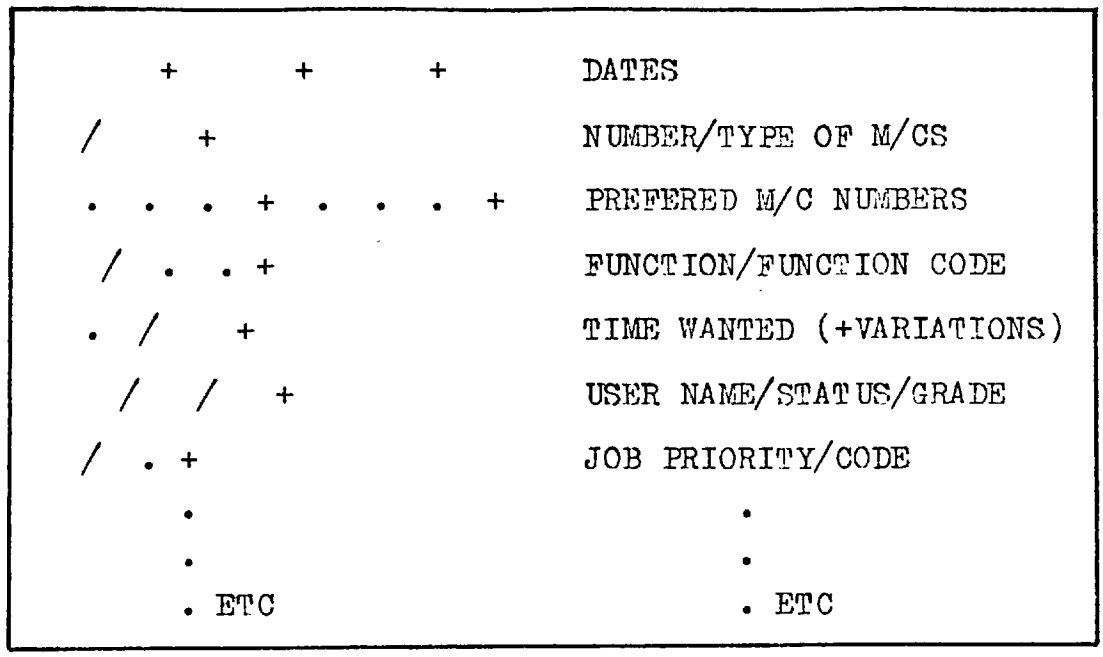


Fig. 6(a) - showing a 'form' which could be used for inputing data.

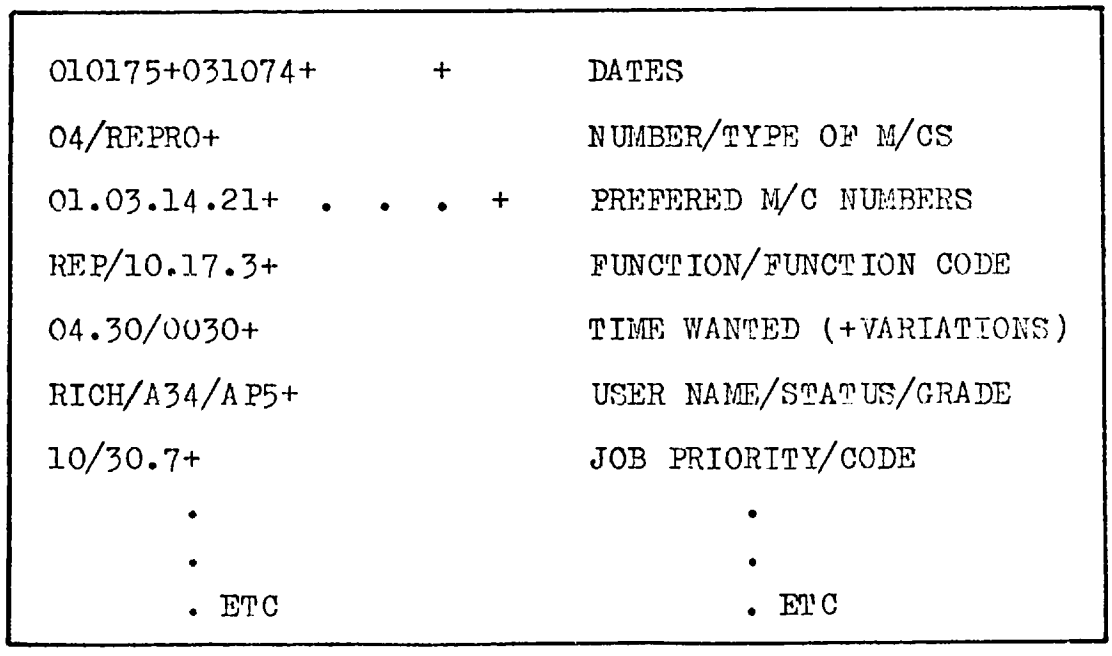


Fig. 6(b) - showing the above 'form' after being completed by the schedule

Program 2

If, at any time, users change their requirements for machine usage, or more information becomes available about a request which has already been stored in the Master File, a program is required which will speedily alter or add to the stored information. A video display unit is ideal for this purpose, as information can be displayed on the screen and altered by simply overwriting the incorrect version.

In order to standardise the VDU displays, it is probably most desirable to use the same format in this program as was used in program 1. Thus a screen similar to Fig. 6 (b) can be produced, with any new data added where necessary and transmitted to the data files.

If a request has to be cancelled, rather than overwriting all the information stored with blanks a simple code can be used to mark the request in the files. A similar code can later be used, if necessary, to remove the marker and reinstate the request in its original form.

Program 3

It may often be necessary to have rapid access to a single request, or a group of requests sharing some common characteristic(s). For instance, if a one hour gap is introduced into the initial schedule by some cancellation,

it may be advantageous to find all those requests which are unscheduled, have a duration of say 55 mins $\pm$ 5 mins and have a function which can be performed on the machine in question. The alternatives can then be examined by the schedulers and the most important or urgent suitable job can be fitted into the gap.

In other cases would-be machine users may simply want to know some information about a request they made some time ago, and about which they have forgotten all the details. All the requests for machine usage which this user has made can then be listed, and the appropriate one examined in a short time on the V.D.U. screen.

Unfortunately a computer terminal is not suitable for handling large amounts of print-out. Thus if several requests are found in an initial search there are four alternatives open which help in the gathering of the desired information.

a. more details can be given to reduce the number of requests found, thus enabling the results to appear on a V.D.U. screen

b. all the results found can be displayed on a V.D.U. screen a few at a time, each being overwritten by the next when required

c. a compact format can be adopted, using coding or omitting information of secondary importance

d. the requests can be identified on the V.D.U. for quick reference, while a complete listing of all the required information is sent to a line printer.

Program 7.

The machines' characteristics may not remain constant over long periods, but may be changed from time to time. In addition maintenance periods, breakdowns etc. will inevitably break into the smooth-running of the machines. A program is, therefore, needed which will update the information describing the machines' characteristics (stored in File 2) without requiring major rewriting of the programs themselves.

As with the updating of the Master File, a video display unit again provides a very convenient method of doing this file amendment. The required information may be brought onto the screen and overwritten as necessary. Alternatively an instruction may be given for the alteration to be done automatically. For example an instruction may be given to leave a named machine unscheduled during a period of maintenance; the machine can then be reintroduced when ready with a similar command.

When new machines are introduced into operation, or old ones removed, the procedures necessary can be treated in a similar manner.

Again the form taken by this program will depend on the nature of the machines used, and the number of variations possible in their function and mode of operation.

4.2.2 Program for Altering the Initial Schedules

Program 6

When the initial schedules have been automatically produced and returned to the schedulers, there will probably be some alterations necessary. These arise either because relatively important requests have arrived too late to be included in the initial scheduling run, or because subjective decisions are necessary which cannot be made by the computer.

This program is designed to alter the initial schedules as necessary in order to produce the final schedules.

Almost invariably the initial schedules will be produced on a line-printer in the form of bar charts or chronological listings, usually with a large amount of information associated with each job. Difficulties

will arise when the schedules, or relevant parts of them, have to be displayed on a V.D.U. screen in such a way as to enable the schedulers to see the effects of the changes they make.

Probably the best way of doing this is to show miniature bar charts of the schedules on the V.D.U., omitting most of the information available but containing some identification for each job. An example of this technique is illustrated in Fig. 7, where a unique identifying number is associated with each job scheduled. Reference tables can be prepared automatically, and consulted to give the information missing on the V.D.U. bar charts.

By specifying the requests for machine time which have to be added to or deleted from the schedules (either by overwriting or by instructions) the effects of manual alterations can be easily displayed on the screen.

If the bar charts are too detailed to be easily readable, or if more detailed information is required, the relevant section of the schedules may be identified and enlarged on the V.D.U. screen. Examples of this type of manipulation are given in Fig. 7.

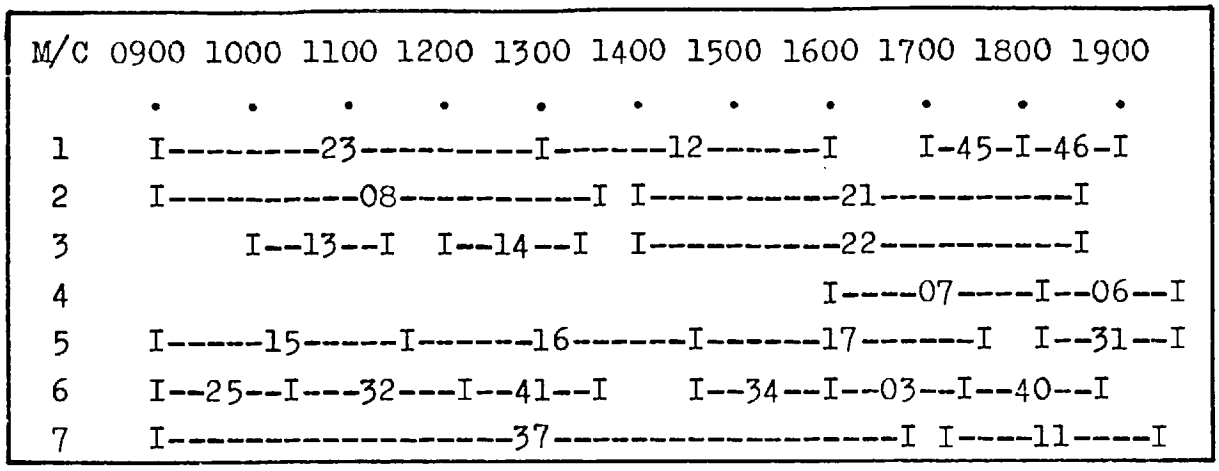


Fig. 7(a) - showing the basic schedule formal on a VDU screen

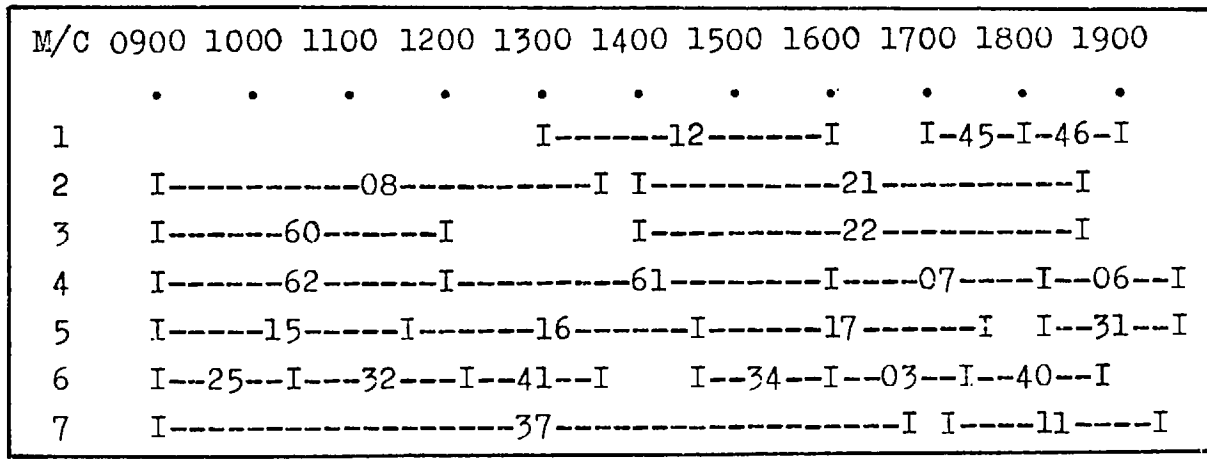


Fig. 7(b) - the same schedule after 'DELETE' 23, 13, and 14 and 'ADD' 60, 61 and 62

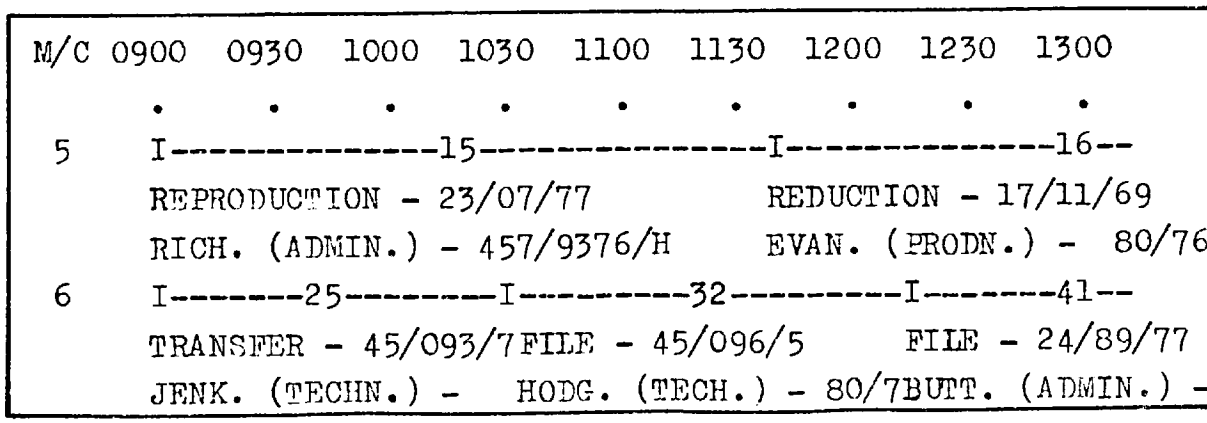


Fig. 7(c) - the schedule after selective expansion

4.3 The Batch Computer Operations

This section refers to those parts of the sub-system where the schedulers initiate 'batch' programs, which read data from the files and send results on the line-printer. As the programs are themselves stored on files, and recalled as necessary, the term 'batch' is somewhat inaccurate, but is used to distinguish this type of operation from the on-line processing.

This part of the sub-system can also be further divided into two parts. One of these automatically produces the initial schedules, while the other, the 'report writer', sends various results to the line printer.

4.3.1 Report Writing Programs

In addition to the schedules themselves, there are several other results which could be produced best on a line printer. These include listings, in various formats, of requests stored in the Master File (to produce a 'directory' of requests), requests left-over after the automatic scheduling run, figures for machine utilisation and tables of spare capacity.

Program 5.

After they have been produced during a computer run the schedules will probably be stored in data files in the form of numerical lists. The print-outs, on the other hand, are likely to be required in the form of bar charts or chronological listings.

This program is needed to translate the schedules from the form they are stored in the data files into the desired print-out format. In cases when there are requests left unscheduled in the initial computer run, these can also be listed conveniently at this point.

One of the major advantages of computer usage is that an output can be printed in a variety of formats with very little effort. Thus individual user requirements can be determined, and the most suitable format for the print-outs selected from a variety available. The exact formats available will depend on the overall user requirements and personal preferences.

Program 8

Accurate analyses of the results of the scheduling run will be needed, and these can be calculated and printed using this program.

It is often the case that accurate measures of machine utilisation etc. are not available in existing information systems. This may be either because it is impossible to derive them manually, or else the effort involved in getting them is too great to justify the rewards.

With a computer, however, it is a simple matter to produce breakdowns of results automatically and in whatever format is required. Typically such analyses will include:

a) analyses of input data

b) analyses of jobs scheduled

c) analyses of requests not scheduled

d) machine utilisations

e) machine availability for the period

Program 9

It has already been stated that one of the major disadvantages of using computer terminals of the keyboard type is that large amounts of information cannot be printed conveniently. This program is needed, therefore, in order to produce a 'directory' of the requests for machine use stored in the data files. Several different forms will be required, giving listings,

for example, according to job number, function, type and so on.

This directory can then be used in conjunction with the on-line operations, to identify a job displayed on a V.D.U. and specify all its characteristics without using too much space on the V.D.U. screen.

4.3.2 The Automatic Scheduling Program

Program 4

This program, when initiated by the schedulers, reads all the relevant details about requests for machine usage and constraints imposed for the scheduling run from the data files, and automatically prepares the initial schedules. As such it is the most complicated program in the sub-system and can, perhaps, be considered as the most important. For this reason the factors involved in the design of this program are discussed more fully in Chapter 5.

5. THE AUTOMATIC SCHEDULING ELEMENT

5.1 A Synopsis of the Problem

The main characteristics of the scheduling problem tackled have been described in section 1.2. It is convenient, however, to list at this point those characteristics most relevant to the scheduling element. This will help to illustrate the thinking behind some of the decisions made later-on.

The main characteristics are:

a) each job is processed once, usually on one machine

b) sometimes a job uses two or three machines working in parallel

c) two jobs are sometimes linked so that one must finish before the other can start

d) there is a static situation in that all the jobs to be included in the automatic scheduling run are known at the start of the production of initial schedules.

e) each job has a particular function specified

f) there is a basic order of job importance which depends on function, together with additional orders based on other considerations

g) there is a variable period of preparation before any job can begin to be processed, which depends among other things on job function

h) each machine can handle certain prescribed job functions, but no machine can cope with all functions

i) there is an order of machine preferences associated with each job function

j) each request can have a different duration

k) there may be a fixed or preferred start time associated with each request for machine time

l) there may be a fixed or preferred machine associated with each request

m) demand for machine time is usually greater than supply

n) there are sufficient staff to operate the machines at all timesthroughout the working day

o) the initial schedules will be updated later by using on-line updating runs, in the light of new information available

p) the computer capacity and CP time available is limited, and a minimum of these resources should be used.

Based on these characteristics it is necessary to determine the best means of solution for the automatic scheduling problem. One of the most significant questions which must be answered concerns the choice between an optimal solution and a sub-optimal 'satisficing' one. At this point the terms 'optimal' and 'satisficing' need

not be defined more fully, as the details of what constitutes a 'good' or 'bad' schedule will not appreciably affect the general principles involved in the decision as to the most satisfactory approach to a solution. It is only later, when determining the quality of the schedules produced, that the criteria for 'goodness' need be examined.

5.2 The Choice between Optimising and Satisficing Solutions

When confronted by the vast amount of literature available on scheduling (Eilon and King[6] for example list 462 abstracts collected between 1950 and 1966) it would be easy to assume that all the major problems have been overcome. This, however, would be erroneous. It is still impossible to solve the majority of realistic scheduling problems by any of the tested optimising techniques.

The most obvious of the optimising techniques is complete enumeration, where all possible schedules are examined and the 'best' is determined. However, when the number of possible schedules is examined it can be seen that this approach is not feasible for even a comparatively trivial problem. As an illustration of this the simple case of scheduling 'n' jobs on 'm' machines is examined. If n jobs are to be scheduled on one machine there are n! possible sequences.

If 'm' machines can be used the number of possible schedules rises considerably. In general terms the number of possible schedules for 'n' jobs on 'm' machines is $(n!)^m$. Thus for even a comparatively small problem with four jobs on four machines the number of possible schedules is:

$$(4!)^4 \simeq 3.3 + 10^5$$

This figure will rise rapidly as the size of the problem is increased.

Although a large number of these will be duplicates, allowance is not made for any complicating factors such as priorities and deadlines. The magnitude of the problem can therefore be judged from this figure.

For real problems this sort of enumeration is beyond the scope of even the largest computer to evaluate in a usefully short time.

In order to reduce the number of solutions which need to be enumerated, algorithms have been constructed which attempt to quickly 'home-in' on an optimal solution. These algorithms themselves, however, often run into computing difficulties which make them unsuitable for use in solving real problems.

Gupta[11] summarises the situation by saying, "Scheduling n-jobs on M-machines has been an area of constant research for the past two decades. However, in spite of several research efforts, practical scheduling problems cannot be optimised effectively and efficiently. The reported progress in the field of scheduling is almost insignificant as compared to the practical requirements of the problem".

This view is supported by Balas[1] who wrote, "Machine sequencing is one of the most frequently occurring real-world problems, which while theoretically 'solved', cannot be handled for a realistic problem size by any of the existing optimising techniques".

These two comments illustrate the split which exists between 'theoretical' optimising solutions and 'satisficing' solutions to practical problems. At present these two types of solution must be regarded as separate, and there seems very little hope of providing optimal solutions to real problems in the near future.

There are, in general terms, only two types of problem for which optimal solutions can be obtained: small-scale problems involving a few machines and jobs, and larger problems in which constraints and restrictions

are relaxed to make solution easier. Iterative approaches to the solutions of problems of this type are illustrated schematically in Fig. 8.

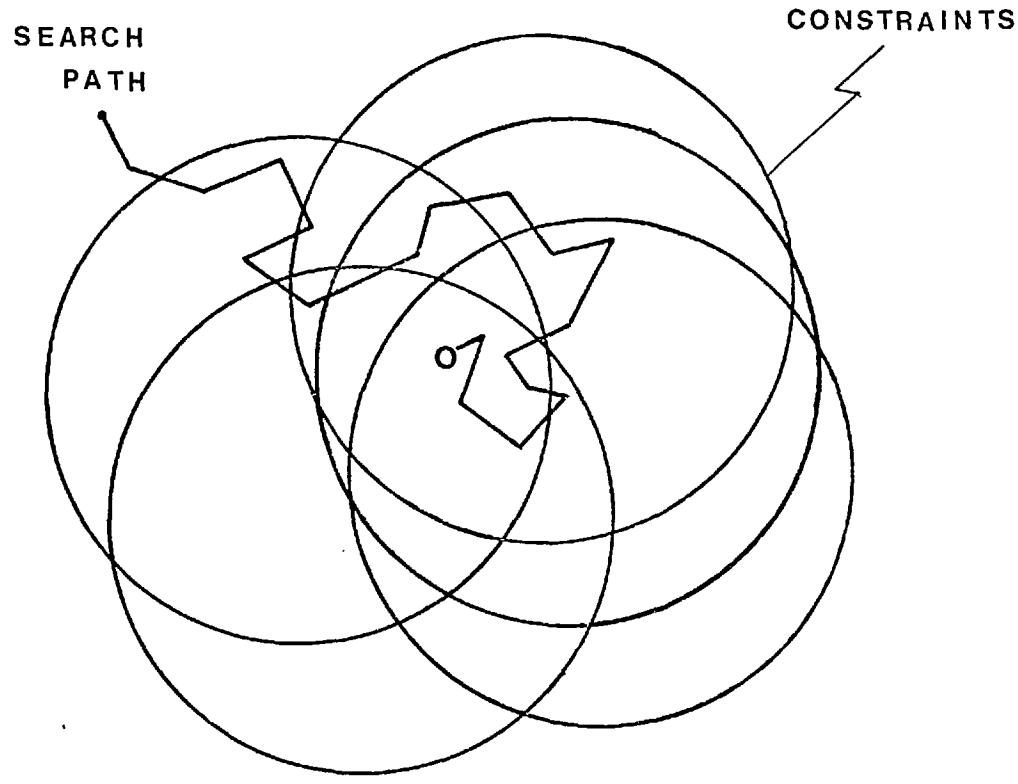


Fig 8 (a) - illustrating one possible iterative route to the optimal solution 'O'.

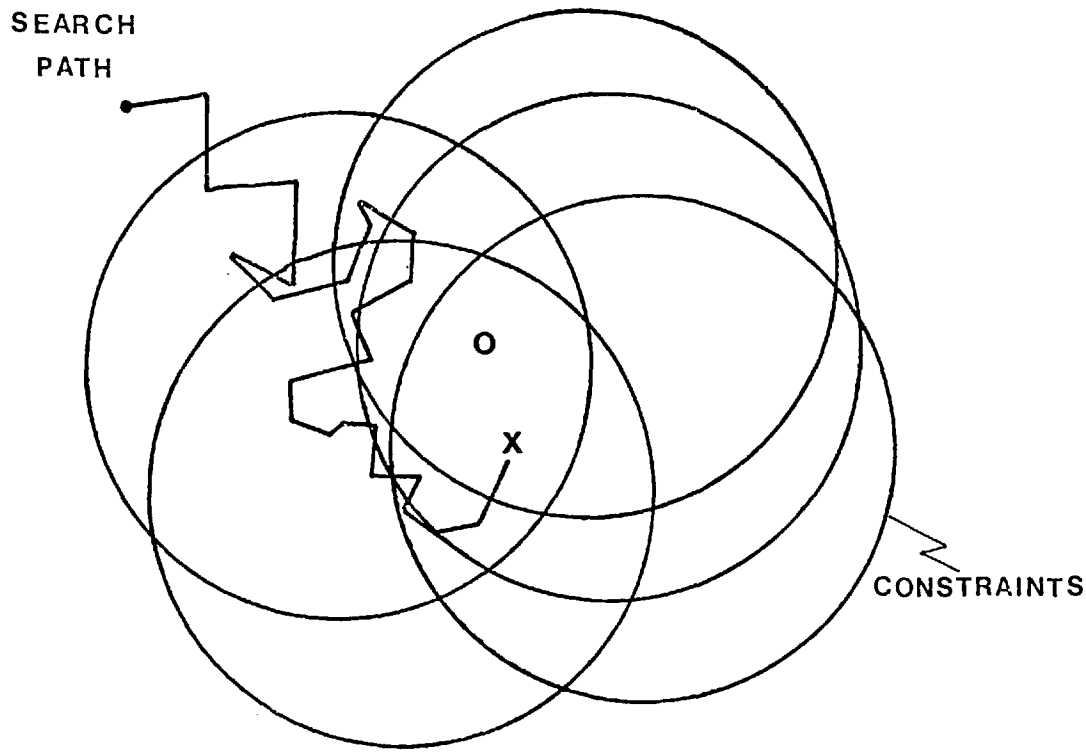


Fig 8 (b) - illustrating one possible iterative route to a sub-optimal solution 'X' which is within all the constraints

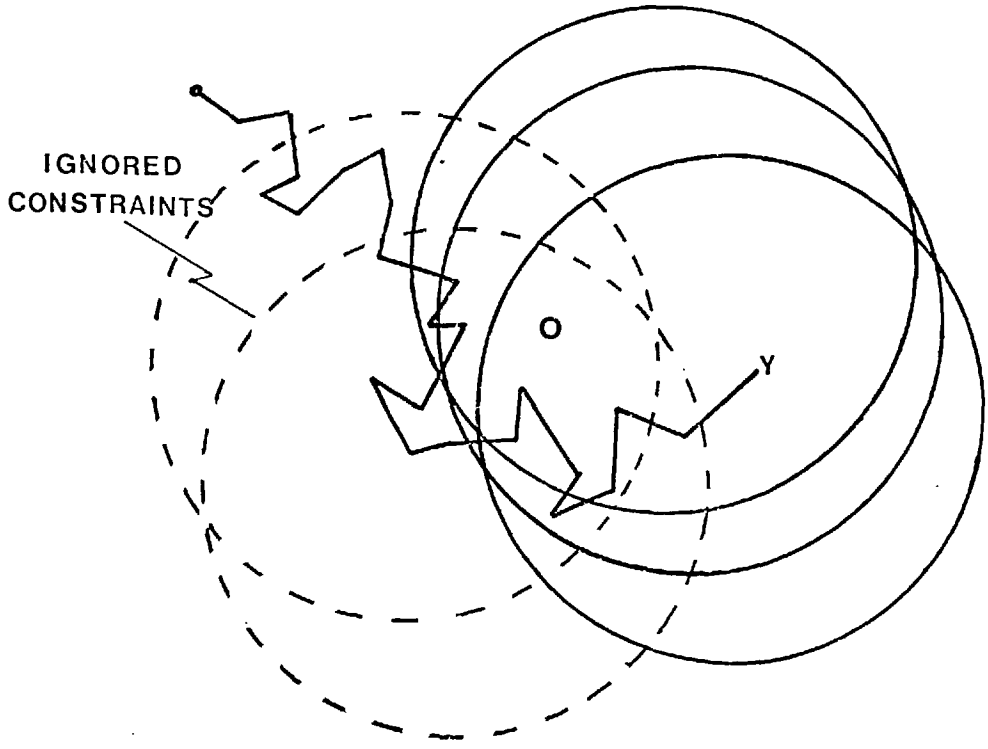


Fig 8 (c) - illustrating one possible iterative route to an optimal solution 'Y' where some of the constraints have been ignored

A real scheduling problem will usually be much more complex than a theoretical formulation. In particular constraints are more rigid, the number of interacting factors is generally greater, the interactions are more complex and subjective decisions are likely to be more relevant.

It may clearly be concluded that if a realistic problem is to be studied an optimal solution is unlikely to be found. The criterion for success in a real scheduling situation is not usually optimality, but the production of a good or 'satisficing' schedule. This means that the schedule produced should achieve a certain acceptable standard when measured against some predetermined criteria.

Even if it were possible to produce optimal schedules for the particular problem considered in this report, the benefit to be gained from them may not be worth a large expenditure of effort. Any optimal or near optimal schedules produced will probably have its quality reduced as the initial schedules are altered manually to take into account late requests and other factors. Thus optimal initial schedules could be changed later to give non-optimal final schedules.

It may be concluded, therefore, that in the particular problem considered the most satisfactory

approach to a solution is by some technique which will yield 'good' though not necessarily optimal schedules, using a minimum of available resources.

5.3 Scheduling Techniques Available for Obtaining Sub-optimal Results

There are two major approaches open for obtaining such sub-optimal, satisfycing solutions:

(a) Simulation - which involves the building of a mathematical model of the system to be investigated. Experiments are then performed on the model and the factors which produce good solutions can be identified. These may then be transferred to the real system.

(b) Heuristics - which are methods of obtaining results by calling on past experience of similar situations. They depend on techniques such as loading rules and priorities.

These approaches often have no backing by formal mathematics, but have been found to give very useful results in many cases. Perhaps their major advantage is that they are fairly easy and cheap to apply.

Simulation has been widely used for some time as a powerful technique for obtaining solutions to problems which would prove difficult by other means. Its ease of

use and the high quality of results (although usually sub-optimal) have helped to make it a 'respectable' technique.

Also, largely because it is readily suited to computer use, the number of applications where it is used is growing rapidly. In particular computer simulation languages have been developed which simplify even more the procedure necessary to obtain results.

Heuristics on the other hand, perhaps because it is essentially a 'practical' method of solution often lacking in rigorous mathematical justification, has become known as the technique to be used as a last resort. Although it is widely used as the 'best' means of obtaining a solution in cases where limited resources are available, or where the problem is too complex to be attempted by other methods, its use is often considered an admission that a 'second best' approach has been used.

This somewhat short-sighted attitude may, however, be changing. Certainly heuristics are becoming used increasingly as it is realised that the results obtained, although usually sub-optimal, are at least as good as those given by most other non-optimising techniques. Moreover the effort and expenditure necessary

to obtain these results is usually sufficiently lower to justify the use of this approach.

With optimal solutions to real problems obviously some way away, good heuristics are increasingly seen as a means of obtaining satisfactory results. As Bakshi and Arora[13] say, "The search for optimal solutions in sequencing problems still remains in a far from satisfactory position. There is an increasing trend to accept sub-optimal solutions. Scheduling based on heuristic methods which will give reasonably good sub-optimal solutions is getting more and more popular with management in real life problems".

The preceding comments have been based on the assumption that it is possible to differentiate completely between heuristic and simulation approaches. In practice this assumption is not entirely valid, due to the overlap in meaning of the two terms. This may be caused, in part, by the vague definition attached to the term 'model' with simulation techniques. There is little difference, for example, between some simple simulation models and the corresponding heuristic approach. At the same time methods of obtaining solutions to sequencing problems which involve the use of loading rules are commonly referred to as simulation techniques. Thus there is, in practice, no clear cut

distinction between the two terms.

In the particular scheduling problem tackled in this report there are several factors which must be considered in determining the better approach to a solution. These include:

(a) the number of jobs to be scheduled
(b) the number of machines to be used
(c) the number of variables associated with each job
(d) the number of variables associated with each machine
(e) the complexity of the rules for determining which jobs can be scheduled on which machines
(f) the computer resources available.

Because each job is unique, and as there are a large number of possible variations in job characteristics, it is thought that a system of loading rules would be ideally suited to this application. Simulation approaches would prove less satisfactory, due to the complexity and structure of the model which would have to be built.

As the jobs to be scheduled are sufficiently different from each other, a system of loading rules can be devised which will differentiate the requests and put them in some order of priority in which they can be

scheduled. This would prove more satisfactory, and less costly in resources, than a more formal simulation approach.

It is therefore concluded that the approach to a solution which should be examined further is a heuristic approach, relying on a system of loading rules.

The determination of the most suitable form for these loading rules is considered in section 5.4.

5.4 Determination of the Most Suitable Loading Rules for Solving the Problem

The loading rules devised must depend on those variable characteristics of the requests for machine time which effect the way in which they are scheduled. Notably a lot of the previous work done comparing loading rules for various scheduling situations concentrates on the job duration. For instance Eilon et al[7] have shown that in their study of job-shop scheduling the most satisfactory results were obtained by scheduling the jobs in order of increasing duration, i.e. the shortest job is considered first.

From the previous descriptions of the particular problem considered in this report it can be

seen that the variable characteristics of the requests for machine time include:

(a) job duration

(b) function - which is equivalent to importance, and also determines the machines which may be used

(c) whether or not there is a preferred/fixed machine specified for the job

(d) whether or not there is a preferred/fixed start time for the job

(e) the arrival time of the requests for machine usage at the schedulers.

These factors all add further complications to the problem considered. As an example, if requests are considered for scheduling in order of function (i.e. most important first) then an early request may be scheduled in a gap which is needed for a later request with 'fixed' characteristics. Conversely if requests are considered in the order of 'most fixed first' (that is considering those requests on fixed machines and at fixed times first) all the machine time available may be taken up with comparatively unimportant fixed jobs, while important ones with more flexibility will remain unscheduled.

If either of these straightforward approaches are used the most satisfactory results are unlikely to

be obtained. This is true even though the criteria chosen for determining the quality of a schedule (i.e. for deciding what constitutes a 'good' or 'bad' schedule) will inevitably have a considerable effect on the loading rules adopted in any practical situation.

In the problem considered a compromise must be drawn in order to produce efficient schedules which take into account fixed and preferred conditions, the relative importance of requests and so on.

In order to determine the exact effect of varying the order in which requests are considered for scheduling, a series of loading rules had to be evolved and tested using realistic data. These would evaluate the various alternatives open, by placing different emphasis of each of the most relevant factors, and comparing the results obtained.

The three most important factors which would effect the scheduling in the problem considered are duration, function and fixed/preferred conditions. Each of these is discussed briefly in the following paragraphs.

(a) Fixed/preferred conditions

It is possible to identify nine varying amounts by which a request may be fixed on a specified machine

or at a specified time, as illustrated in Fig. 9.

'Fix Code'	Machine	Start Time
1	any	any
2	preferred	any
3	any	preferred
4	preferred	preferred
5	fixed	any
6	any	fixed
7	fixed	preferred
8	preferred	fixed
9	fixed	fixed

Fig. 9 - showing the varying amounts by which a request can be fixed

In this context 'fixed' means that no other time or machine can be used to schedule a request if that specified by the schedulers is not available. 'Preferred' indicates that some other time and/or suitable machine may be used if the specified time and/or machine is not available, but the result will be less satisfactory. 'Any' means that any available time on a suitable machine can be used with equally good results. A 'suitable' machine is one which is capable of handling the request function.

This 'fix code' can be used to identify the amount of flexibility of movement of the request. In general terms the higher the fix code, the less flexibility a request has and the more fixed it is.

The emphasis placed on the degree to which a request is fixed at a given time or on a given machine will obviously effect the result of the scheduling. For instance, if requests are scheduled in an order which is not dependent on the fixed conditions, it is possible that early requests which have a greater degree of freedom will be scheduled so that they occupy spaces which are needed for later more fixed requests.

It is likely that more satisfactory schedules will be produced if some account is taken of the fix code. In particular it is probably advisable to consider the requests in the order of decreasing fix code for the scheduling run.

(b) Function

As the function also gives an indication of the importance of a request similar considerations apply. Notably if requests are taken for scheduling in an order independent of function, important requests may be left unscheduled due to a lack of free time on the machines.

It is a straightforward matter to encode functions so that an increasing or decreasing function number is associated with increasing or decreasing importance. Then the best results may be expected when the most important requests are scheduled first. If this is not done (and bearing in mind that demand for machine time is greater than supply) all the available machine time may be taken with comparatively unimportant jobs, leaving insufficient space for the more important ones.

(c) Duration

There are basically three possible orders into which the requests can be put with regard to duration:

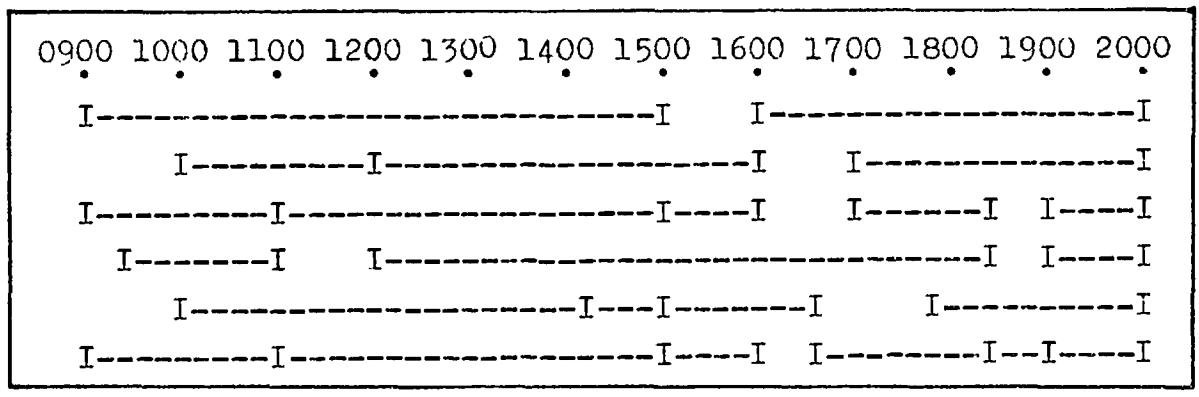
i. shortest first

ii. longest first

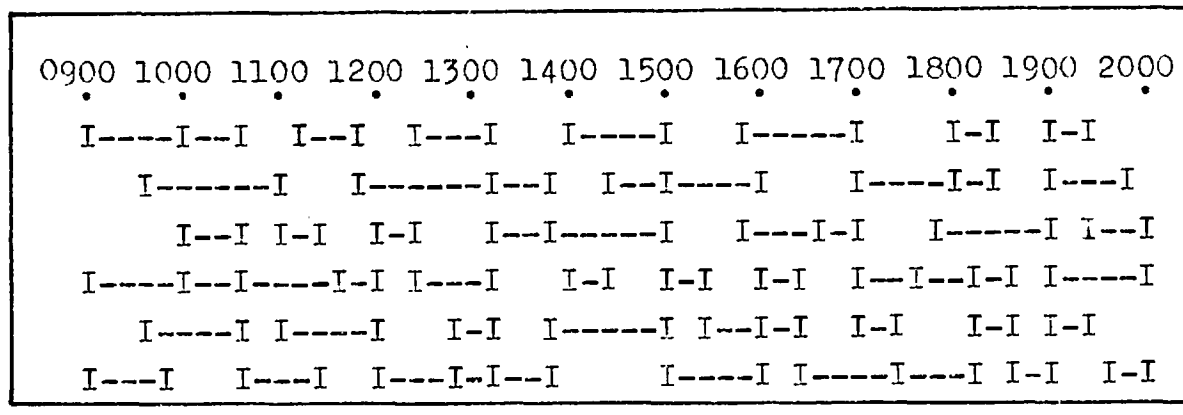
iii. independent of duration

In addition there is a further alternative which may be advantageous in cases where there are more requests for machine time than can be scheduled. If the longest jobs are scheduled first, few requests may be scheduled, but efficient schedules may be given. Conversely scheduling the shortest jobs first may produce inefficient schedules, but more jobs could be

scheduled. This principle is crudely illustrated in Fig. 10, while a more rigorous approach is given later in section 5.7.



(a) possible results of scheduling the largest jobs first.



(b) possible results of scheduling the shortest jobs first.

Fig. 10 - illustrating possible effects of scheduling jobs according to duration

In an attempt to compromise between these two extremes an alternative was devised which ignored a proportion of the longest requests, and considered the remainder in the order longest first. The proportion of requests ignored rises with the demand for machine time or the number of requests submitted.

This principle, which is illustrated schematically in Fig. 11 and Fig. 12, was proposed in order to attempt a solution which incorporated the best features of solution 'a' and 'b' in Fig. 10, and yet gave a better result than could be obtained by taking the requests in an order independent of duration.

One elementary drawback with this approach is that the longest requests will always be ignored, unless there is an exceptionally low demand for machine time. Even if they are resubmitted for a second scheduling run they will be ignored. Unfortunately these long requests are also likely to be the most difficult to add manually to the initial schedules.

A set of loading rules was devised and tested to determine which of the three factors considered was most important in the production of a schedule, and which combination of factors produced the schedules which were judged 'best' when measured against certain criteria.

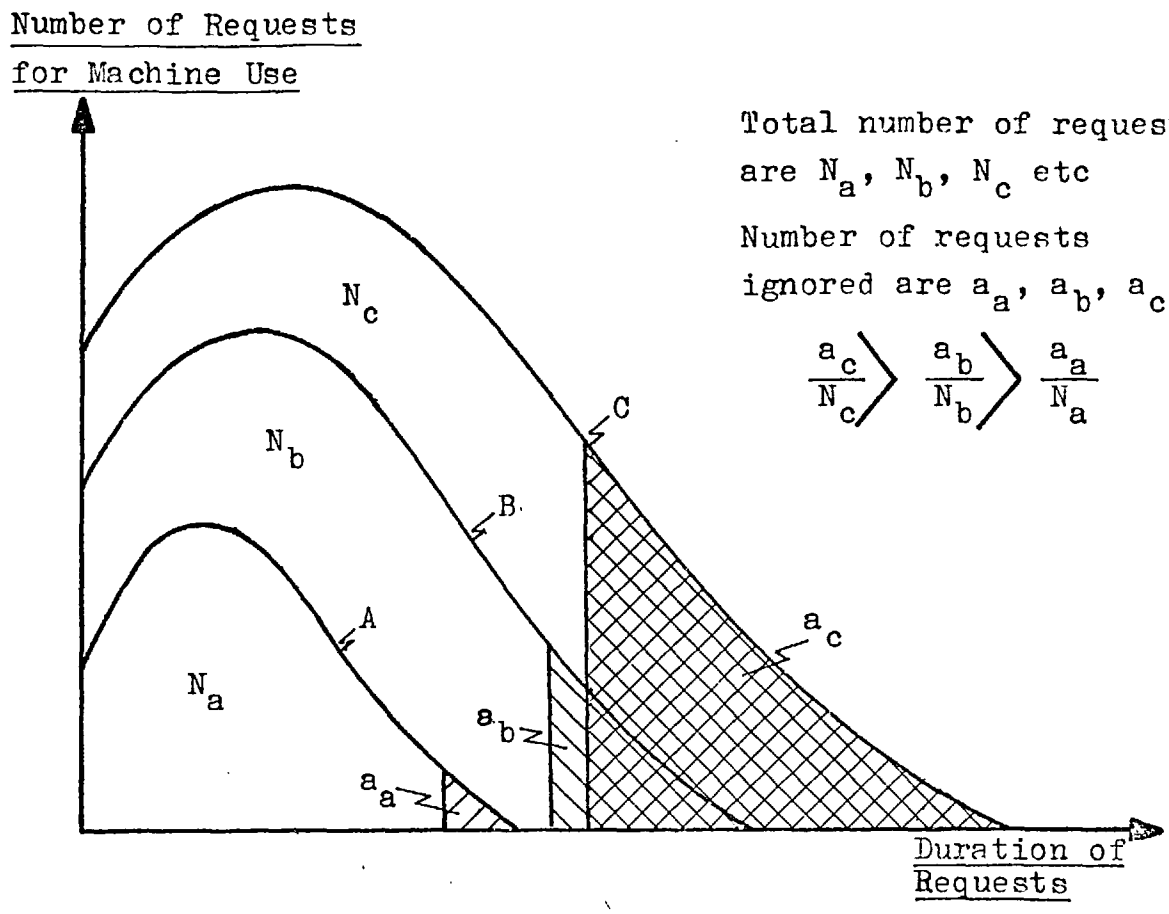


Fig 11 - showing the higher proportion of requests ignored with increasing numbers of requests for machine use

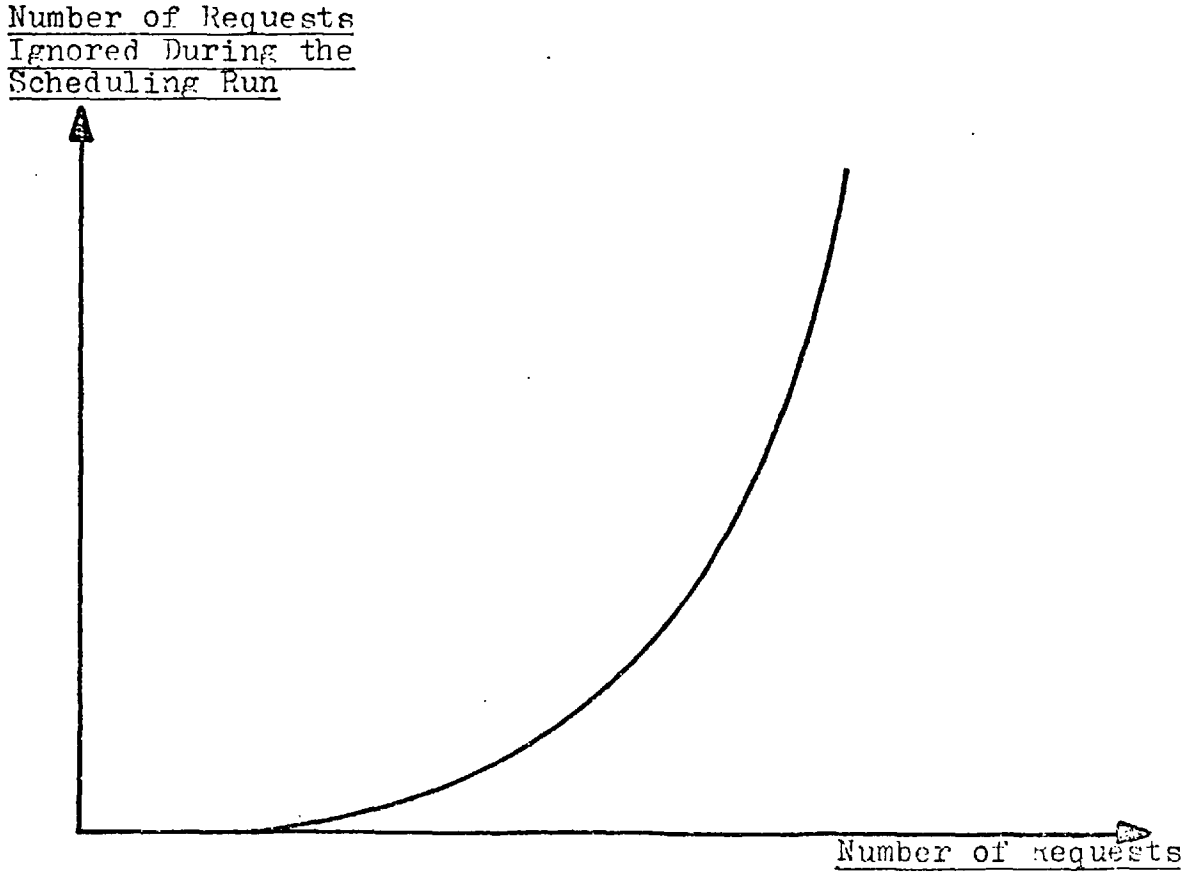


Fig 12 - showing the relationship between the number of requests for machine use and the number of these ignored

Every combination of the factors duration, fix code and function were considered in producing the following twenty loading rules.

	arrival[6]	shortest first	longest first	reduced then[7] longest first
Independent[1]	1	6	11	16
Function[2]	2	7	12	17
Fix code[3]	3	8	13	18
Fix code Function[4]	4	9	14	19
Function Fix code[5]	5	10	15	20

Fig. 13 - illustrating the twenty loading rules considered

Notes:

(1) Independent of function or fix code

(2) i.e. in order of the most important function first

(3) in order of decreasing fix code - i.e. most fixed first

(4) in order of most fixed, then most important first

(5) in order most important then most fixed first

(6) i.e. arrival order at the schedulers, and therefore independent of request duration

(7) ignoring a proportion of the longest requests, and then taking the remainder in the order of longest first, as described in the previous paragraphs.

Within each group the requests are considered in the order in which they arrive at the schedulers i.e. earliest first.

Thus loading rule number 15 takes the requests in the order:

1. most important first - then within each function-
2. most fixed first - then within each fix code-
3. longest first - then for requests with the same durations-
4. earliest first

and loading rule number 7 takes the requests in the order:

1. most important first - then within each function-
2. shortest first - then for requests with the same durations
3. earliest first

Loading rule number 1 simply takes the requests in the order they arrive at the schedulers. This can, therefore, be considered the 'base line' performance with which the other loading rules can be compared.

It is plain that the point at which the requests are sorted according to duration is immaterial to the final order.

5.5 The Test Data Used for the Scheduling Runs

The data used can be considered in two parts, being:

(a) data concerning the requests for machine use

(b) data concerning the machines being used

5.5.1 Data used for the Requests

The job data used was derived to give a range of durations from half an hour to nine hours, which did not conform to a regular pattern but which did show certain trends.

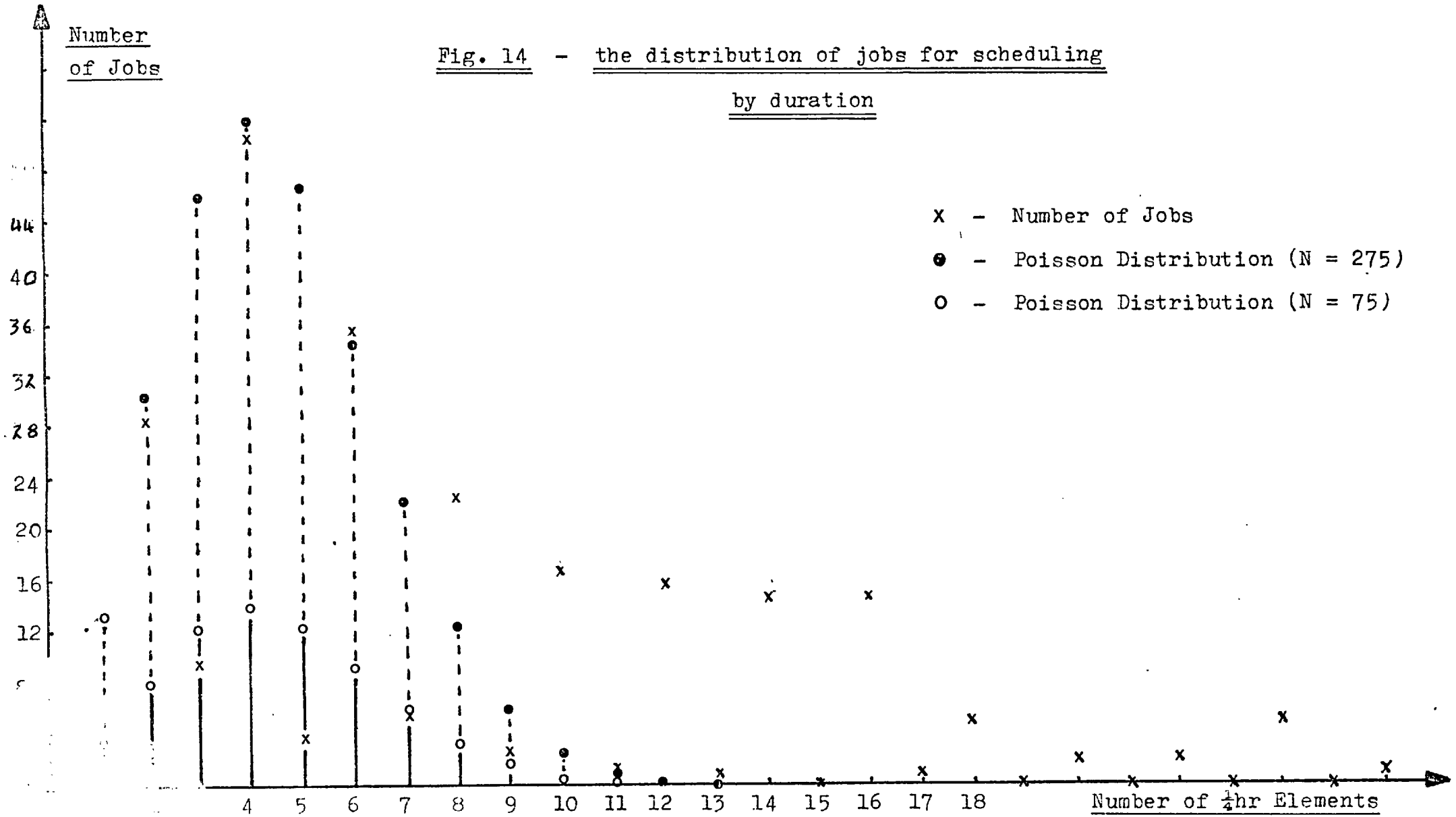
The method of producing this data was evolved during the formulation of a solution to the problem, and was based on those approaches which it was thought gave close approximations to the real data which may be associated with such problems.

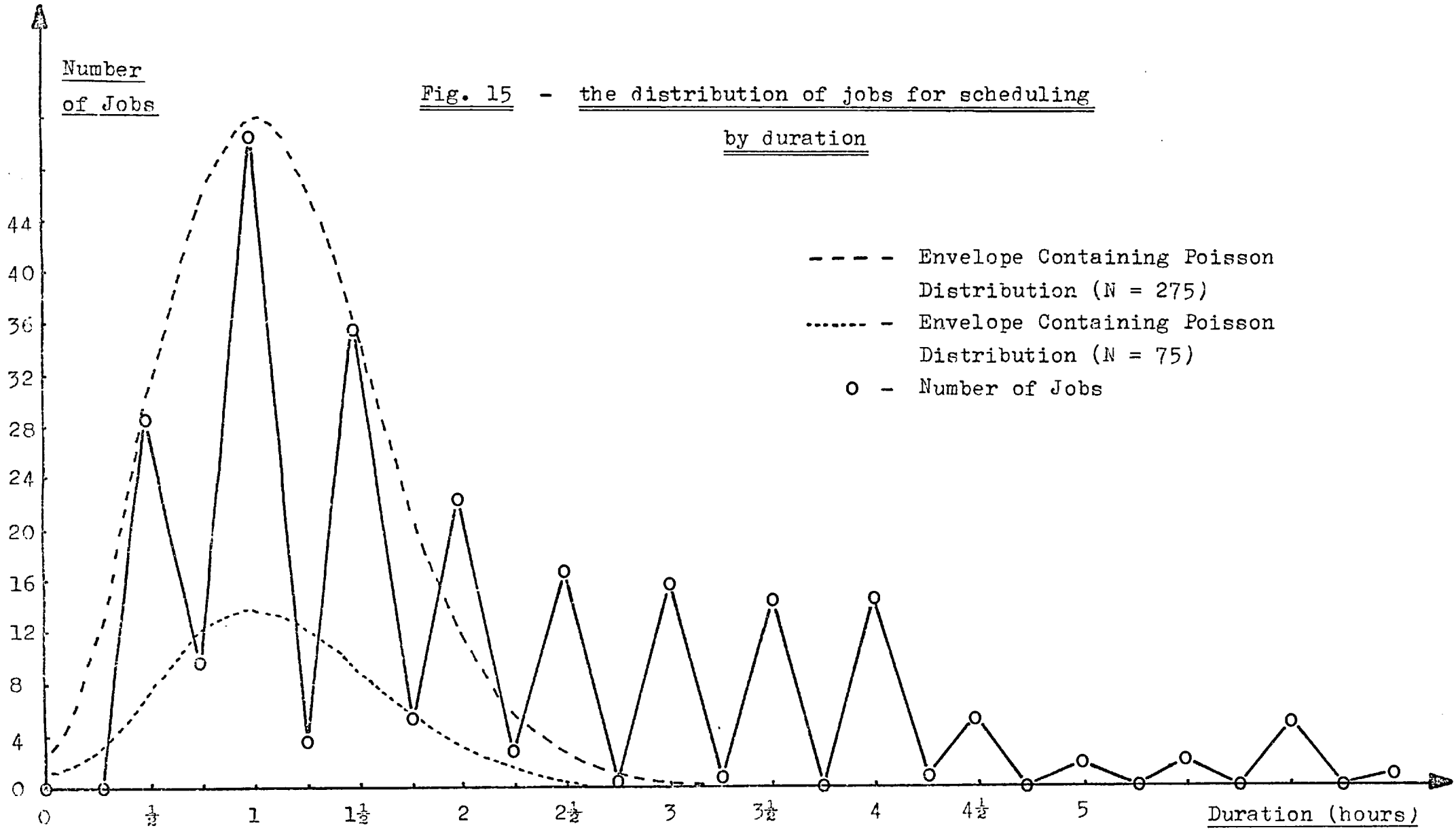
Request durations were taken in discrete elements of a quarter of an hour each, and the number of requests for each duration element were considered to be distributed basically along the lines of a Poisson Distribution. In practice two Poisson Distributions were taken; both with a mean of 4.5 units, but one with 275 requests and one with 75 requests. The number of requests with each duration was taken, for the test data,

to be equal to the number of requests shown when the two distributions were considered alternately. Further requests were then added with durations generally longer than the Poisson Distributions, while others were subtracted, on a random basis, until a total of 250 requests remained.

The numbers involved in this process are illustrated in Fig. 14, while the final distribution of request durations is shown more clearly in Fig. 15.

Fig. 14 - the distribution of jobs for scheduling by duration




Number of Jobs
Fig. 15 - the distribution of jobs for scheduling by duration
Envelope Containing Poisson Distribution (N = 275)
Envelope Containing Poisson Distribution (N = 75)
o - Number of Jobs
0
4
8
12
16
20
24
28
32
36
40
44
0
1/3
1
1 1/2
2
2 1/2
3
3 1/2
4
4 1/2
5
Duration (hours)

The number of requests used was 250. This, although a comparatively small sample, was consistent with the scale of the problem considered. It was both small enough to be applicable to many practical situations (where up to 250 jobs may be handled during a particular scheduling period), and yet large enough to give representative results.

Each job was allocated an identifying number from 1 to 250 in such a way that the durations were in a random order with respect to these numbers.

The nine fix codes were allocated, and the jobs were considered to be for one of nine functions (referred to as numbered 1 to 9 for convenience), both of which were distributed in a random order.

Periods of preparation, necessary before processing jobs can begin, were added randomly with durations varying from five minutes to one hour.

Preferred and fixed start times and machine numbers (see section 5.5.2) were allocated, again in a random manner, wherever necessary.

A more complete analysis of the data concerning requests used for the test runs is given in Appendix A.

5.5.2 Data used for the Machines

Twenty five machines were considered for the scheduling, each of which was different and had distinct characteristics. An identifying number, from 1 to 25, was given to each machine. The working day, when all machines were taken to be free for scheduling, was the sixteen hour period from 0900 to 0100.

For these tests it was assumed that staffing availability would not affect the schedules, but if there was enough free time on the machines to schedule a job sufficient operating staff would also be availalbe.

Each machine was considered suitable for processing certain functions, and (arising from the difference in machine capabilities) each function had associated with it an order of preference for machines. In practice each machine was considered capable of handling up to seven functions, while each function could be processed by up to fifteen machines.

The order of machine preference associated with each function was chosen so that each machine would be able to handle a similar number of functions (from five to seven). However, recognising that some would have generally better facilities than others, the

order in which the machines would be considered for scheduling jobs varied with function.

For the test runs conducted the order of machine preference associated with each function is illustrated in Fig. 16.

Function number	Machine Numbers - decreasing order of preference														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4	5	6	7	8	14	15	16	17	18	9	10	11	12	13
2	1	2	3	4	10	11	12	14	15	16	17	18	19	21	22
3	1	2	3	4	5	6	7	8	9	10	17	18	19	24	25
4	1	2	3	4	5	6	7	8	9	14	15	16	21	22	23
5	5	6	7	8	14	15	16	17	18	19	20	25	21	22	23
6	10	9	8	7	6	5	11	12	13	20	21	22	23	24	25
7	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
8	1	2	3	10	11	12	13	19	20	21	22	23	24	25	9
9	25	24	23	22	21	20	13	12	11	10	9	1	2	3	4

Fig. 16 - showing the order of machine preference for each function.

Provided that no fixed or preferred machine conditions were imposed on a request, each machine associated with the request function would be examined,

in the order shown in Fig. 16, to find a suitable gap for the job. If no suitable gap was found on any of the machines associated with the request function, the request is left as unschedulable.

If a preferred machine was specified for a request this machine would be examined first, but if no suitable gap was found the other machines are scanned in the preferred order shown in Fig. 16 as before.

5.5.3 Machine Loadings for the Test Runs

For the test scheduling runs varying numbers of requests were taken from 25 to 250, with steps of 25, for each of the loading rules considered. It can be seen from the details of the requests given in Appendix A that the time requested in many cases was greater than the machine time available. This is consistent with one of the original conditions of the problem that the machines should be overloaded. The loading of the machines for each size of test run is given in Fig. 17

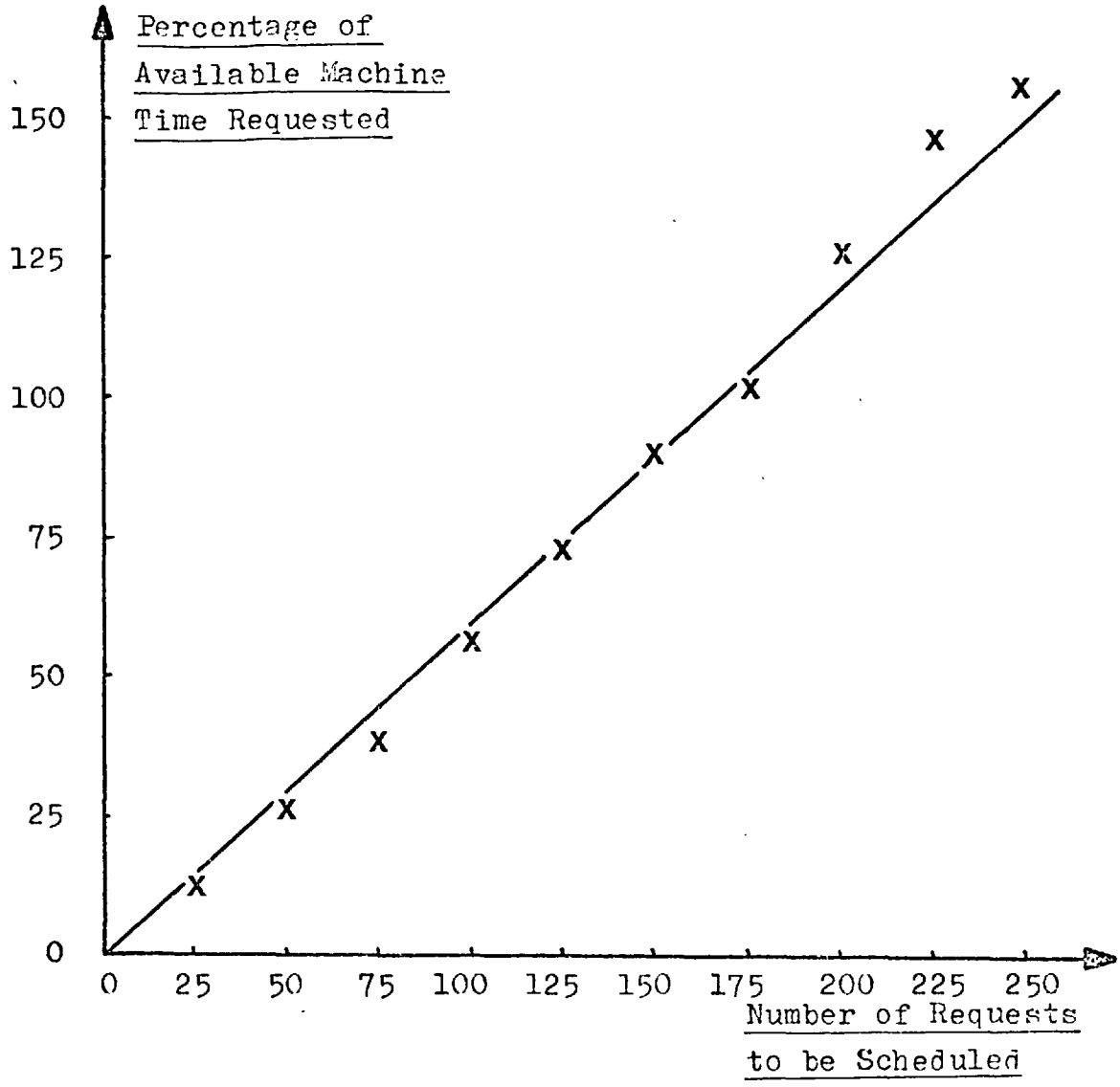


Fig. 17 - showing the percentage of available machine time which is requested with varying numbers of jobs

5.6 A Description of the Scheduling Program

The scheduling program was written in Fortran, as this is one of the most widely used, and therefore generally understood, computer languages. In addition

the translation of the scheduling program into Fortran was thought to present few problems.

The program was stored on a disc file, and called-up as required from a terminal. The necessary input data was read from one set of data files, while any output was sent to another set. This output could then be read, structured and sent to a line printer as required.

Initially it was decided to store the basic information about requests in a random access file. Then for the scheduling run the characteristics of each request were read in turn, and an array was built-up which contained only the requests' identifying numbers in the order in which they were to be scheduled. The program then attempted to schedule each request in turn, in the order they appeared in this array, while referring to the relevant data stored for each request in the random access file.

The random access file was then updated with information about the position of the job in the schedule, or a mark was made to indicate that the request could not be scheduled.

This approach is illustrated schematically in Fig. 18.

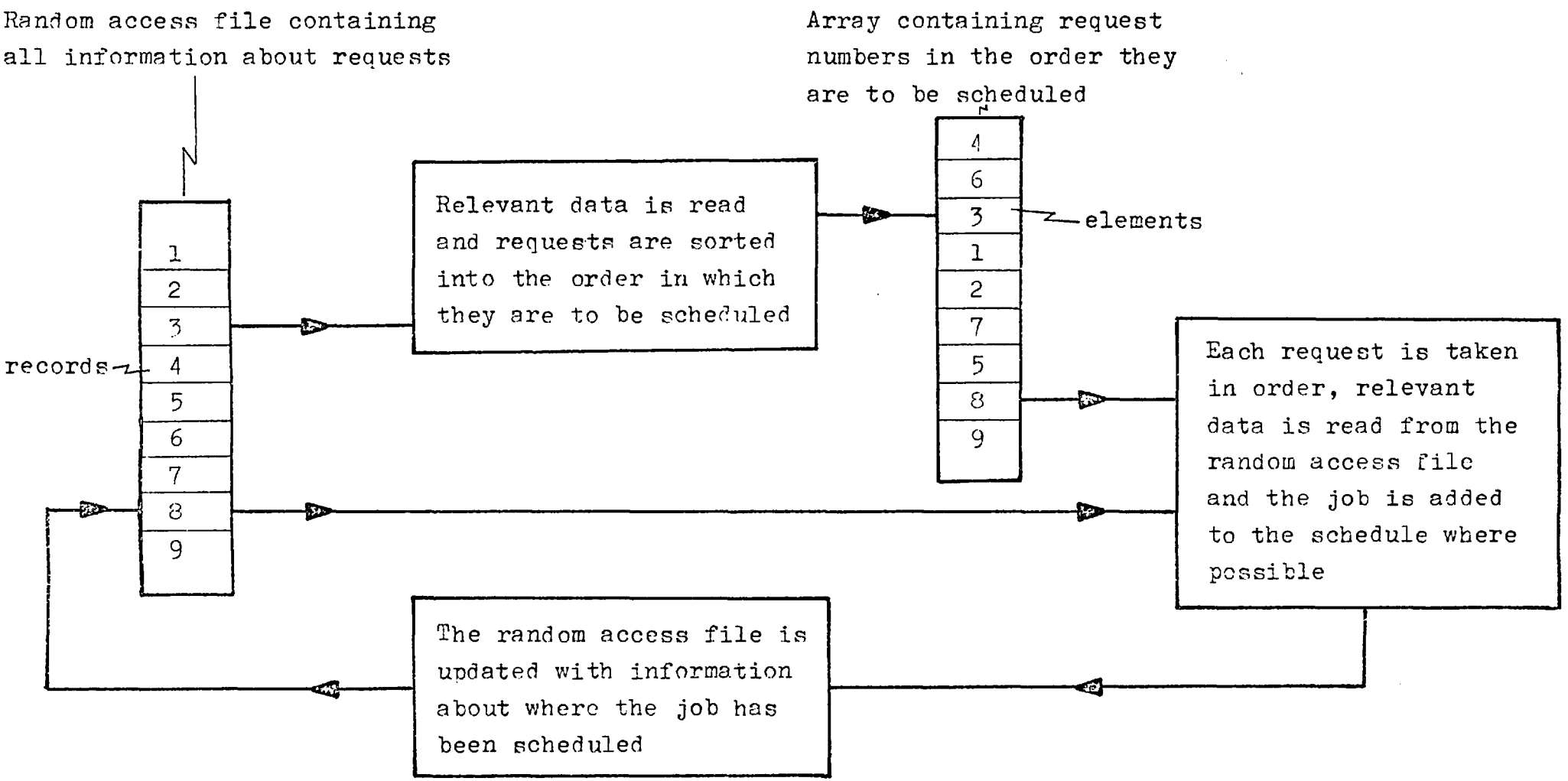


Fig. 18 - illustrating the original approach adopted for file handling

This approach was tested using loading rule 15 (taking the requests in the order most important first, most fixed, longest and then earliest) with varying numbers of requests. An ICL 1905E Computer was used for the tests, and in particular the multi-access system 'Maximop', developed by Queen Mary College, London. This system has the advantages of both very good random access file handling facilities, and the facility for communications via video display units.

The general approach of using random access files to store the basic data has the advantage that a comparatively small amount of core space is needed. At the same time, however, the difficult file manipulations and large amount of arithmetic processing necessary use a large amount of CP time. It can be seen from Fig. 19 that the 'Mill Time' used, as recorded by the computer and adjusted to allow for different numbers of users, is too great to justify this approach to the problem.

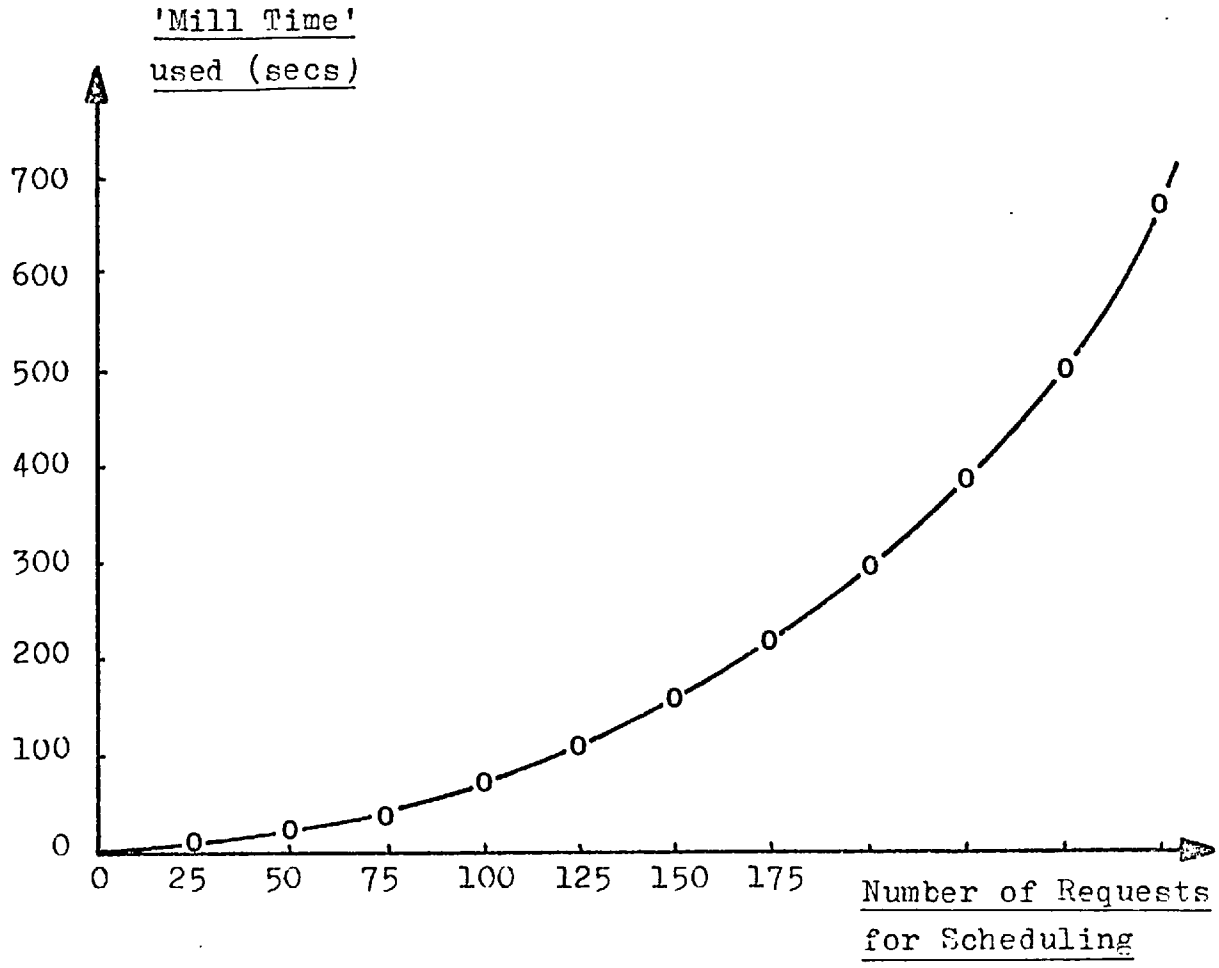


Fig. 19 - core times used for the tests on an ICL 1905E computer (using 6.5K words).

The second approach, which was finally used for all test runs, was to hold all the relevant data concerning the requests sequentially on a disc file. This was transferred to a large array in the computer core when needed for the scheduling run. This array

could then be sorted, reordered and referred to as necessary, and updated with any data which was altered during the scheduling run. After processing the array is copied back onto the data file.

For this approach a larger amount of core space is necessary, but this is more than compensated for by the large reduction in the CP time requirement.

The CDC 6400 Computer at Imperial College, London, operating under the 'Kronos 2.0' system (and later the 'Kronos 2.1' system) was used for these tests.

This second approach, using the faster CDC 6400 computer and keeping the necessary information about the requests in the core during scheduling runs, was far more economical in terms of computer resources than the first approach, using random access files on the smaller, slower ICL machine.

Comparable figures for the core times needed for obtaining results for loading rule 15 are given in Fig. 20.

Number of Requests	Core (Mill) time used (secs)	
	CDC 6400	ICl 1905E
25	1.9	30
50	2.2	44
75	2.3	73
100	2.5	114
125	3.0	164
150	3.6	223
175	4.5	299
200	6.1	386
225	7.4	502
250	8.6	671
Core required	16.8K words (60 bit words)	6.5K words (24 bit words)

Fig. 20 - Comparison of the resources needed to obtain results for scheduling rule number 15.

The drawbacks of using the CDC 6400 computer at Imperial College are that it is incapable of supporting video display units, and its random access file handling capabilities are not as good as those available with 'Maximop'.

For those parts of the sub-system which have been described as on-line operations both random access files and V.D.U.s are considered extremely useful. Thus with the resources available any practical developments towards a scheduling sub-system along the lines preferred have to be considered in two parts:

(a) an on-line part using V.D.U.s and random access files, tested on an ICL 1905E computer with 'Maximop' and
(b) a terminal initiated batch part on a CDC 6400 computer operating under 'Kronos 2.0' (and later 'Kronos 2.1').

This does not, of course, mean that in practical situations the sub-system should be divided in this way. The constraints imposed by the resources available made this artificial split necessary.

The scheduling program itself works by storing the schedules in the elements of a matrix. The working day is divided into five minute periods for each machine, with each period represented by the corresponding element of a matrix. Each matrix element is initially set to zero. Then as a job is added to the schedule the relevant matrix elements are checked and switched to the value '1' to indicate that this time is no longer free. Details of start times and machine used for jobs scheduled are returned

to the data files.

When the schedules are printed the necessary information is gained directly from the data files, without reference to the schedule recording matrix.

This general approach is illustrated in Fig. 21, while a more detailed flow diagram of the steps involved in such a program is given in Appendix B. The details of the program examined in Appendix B have been somewhat amended to take into account the further requirements of a practical example of this program's use.

Initially the schedule recording matrix, which contains the values 'O' and '1' as necessary, was stored using bits rather than words for each element. However, this approach was considered to reduce the generality of the solution given, as some computers may lack the necessary facilities for this type of operation. The extra storage needed for storing each element in a word rather than a bit is comparatively small compared with the rest of the core requirements, especially if use is made of the 'COMMON' storage facilities.

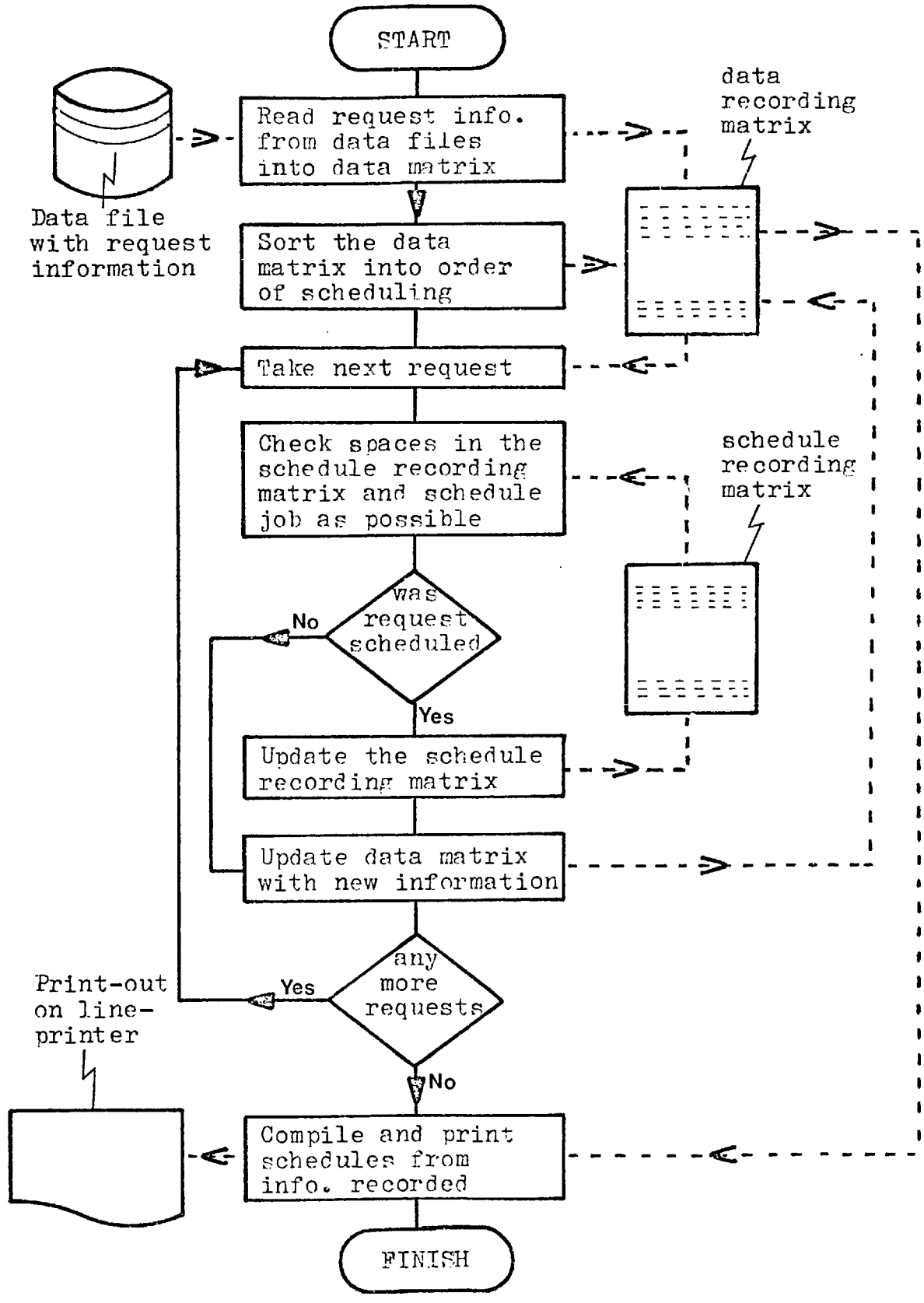


Fig. 21 - illustrating the general approach adopted for the scheduling program

5.7 Theoretical Results of the Scheduling Runs

To schedule a job satisfactorily an acceptable gap is required on a suitable machine (assuming there are no additional constraints imposed by staffing considerations). This process of adding jobs to suitable gaps will continue until no more requests can be scheduled, at which point the scheduling run is terminated. This will happen when:

(a) all the time available is occupied
(b) the shortest remaining request is longer than the longest gap, or
(c) there remain only requests with fixed conditions which cannot be met.

With a fairly large number of requests it is expected that conditions (a) and (b) will be the critical ones, even when a high proportion of the requests have fixed conditions. Validity checks on the input data will ensure that no two requests would be fixed in the same spot, and that condition (c) would, therefore, be rarely critical.

In particular condition (b) is likely to be most important, due to the small probability of scheduling jobs to exactly fill the machine time available.

Because of the fixed and preferred conditions which are put on the requests for machine usage, jobs will

not be scheduled in a continuous manner, with one job starting when the last job has finished. They will be scattered around almost randomly as these fixed and preferred conditions are satisfied. Moreover, scheduling the most fixed requests first, as suggested earlier, will increase this tendency to scatter the jobs around.

A rudimentary schedule taken at some stage during the automatic scheduling run will consist of jobs spread more or less randomly throughout the time bands and separated by gaps of varying lengths. This situation is illustrated in Fig. 22.

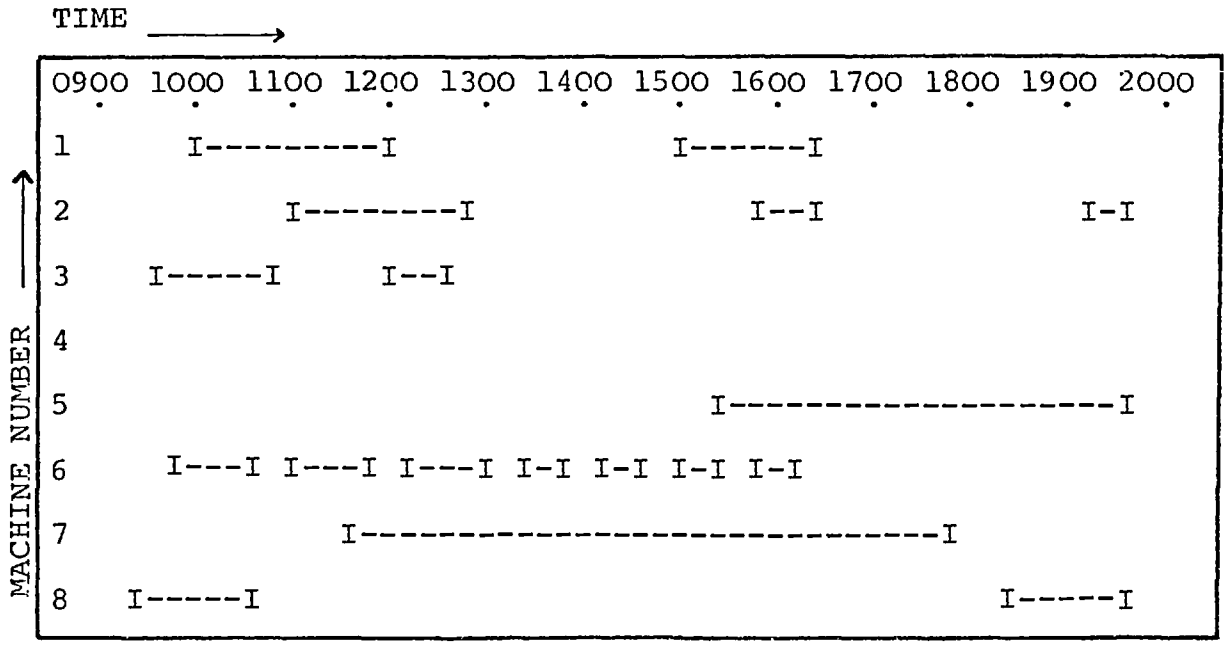


Fig. 22 - Illustrating the state of the schedules at some point during the automatic scheduling run.

If it were possible to ignore the fixed and preferred conditions placed on requests the scheduling would be a great deal easier, for continuous sequencing, with one job starting at the finish of the last one, would be feasible. In the scheduling problem considered this is only possible for jobs with no fixed or preferred conditions which can be fitted into any gaps remaining between the more fixed jobs.

When the scheduling has reached the stage illustrated in Fig. 22 the next request to be considered is taken. If possible it is added to the schedule in an appropriate gap, either to satisfy preferred/fixed start time and/or machine conditions or at any place where it will fit. If there is not a suitable gap for the request it is left unscheduled and the next request is considered.

Obviously as more jobs are added to the schedule the average length of the gaps remaining between scheduled jobs will decrease, while the number of gaps will tend to increase. This trend is illustrated in Fig. 23.

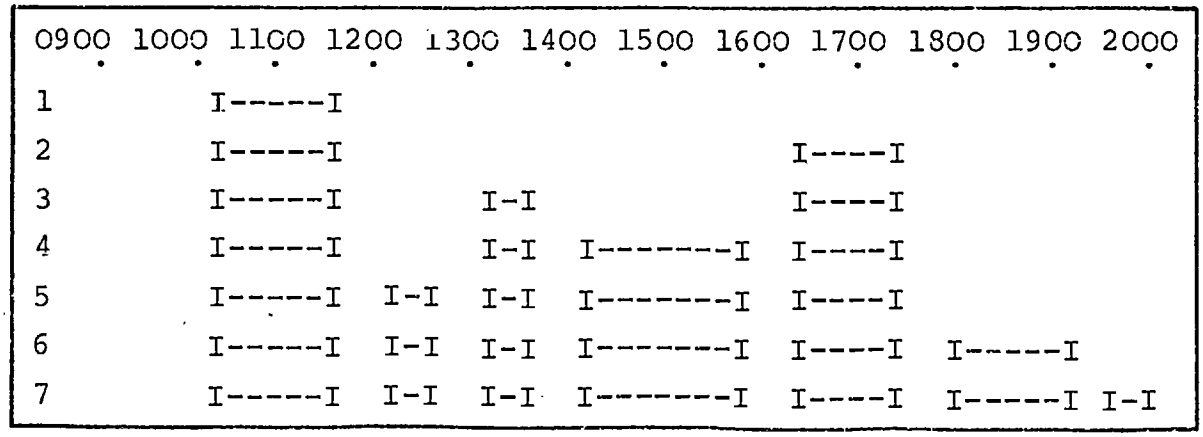


Fig. 23 - illustrating the increasing number and decreasing average length of gaps between jobs as more jobs are scheduled.

If the jobs are scheduled separately, in a random manner and with gaps before and after each one, the average gap length is:

$$\bar{G} = \frac{\text{total duration of gaps}}{\text{total number of gaps}} = \frac{MT - \sum_{i=1}^{n} d_i}{n + M}$$

where: $\bar{G}$ = average gap length

M = number of machines to be scheduled

T = time to be scheduled per machine per day

n = number of jobs scheduled

d_i = the duration of the ith job scheduled

In practice jobs may be scheduled so that there are no gaps between them. This may be allowed for by adding a factor, 'n_a' to the denominator of the previous equation, which then becomes:

$$\bar{G} = \frac{MT - \sum_{i=1}^{n} d_i}{(n - n_a) + M} \qquad \text{with } n_a \ll n$$

For the four orders of scheduling considered earlier, based on duration (longest, reduced longest, shortest and independent or random first) the following graphs illustrate the trends to be found for the average duration of unscheduled requests, total time scheduled and average gap length, with increasing numbers of jobs scheduled.

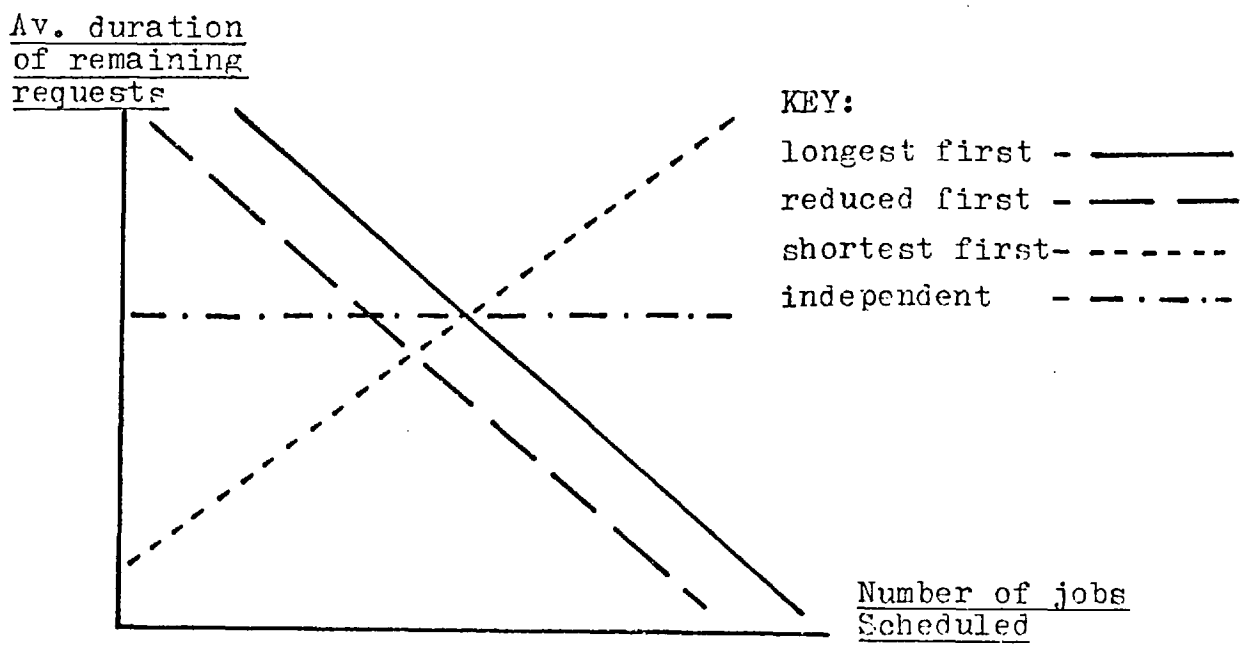


Fig. 24 (a) - trends for the average duration of unscheduled requests

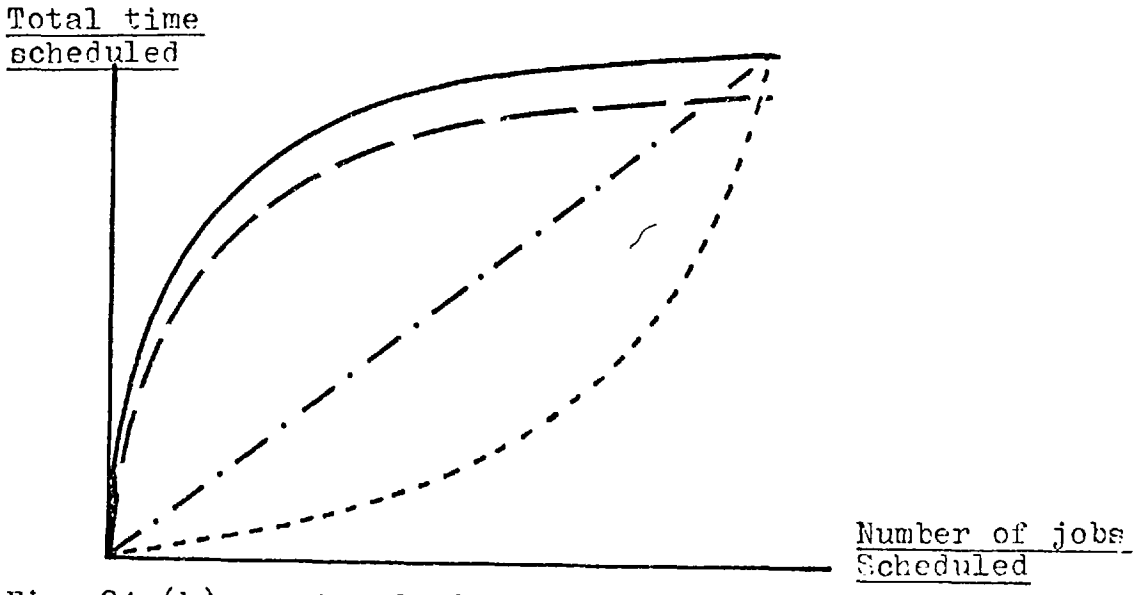


Fig. 24 (b) - trends for the total time scheduled

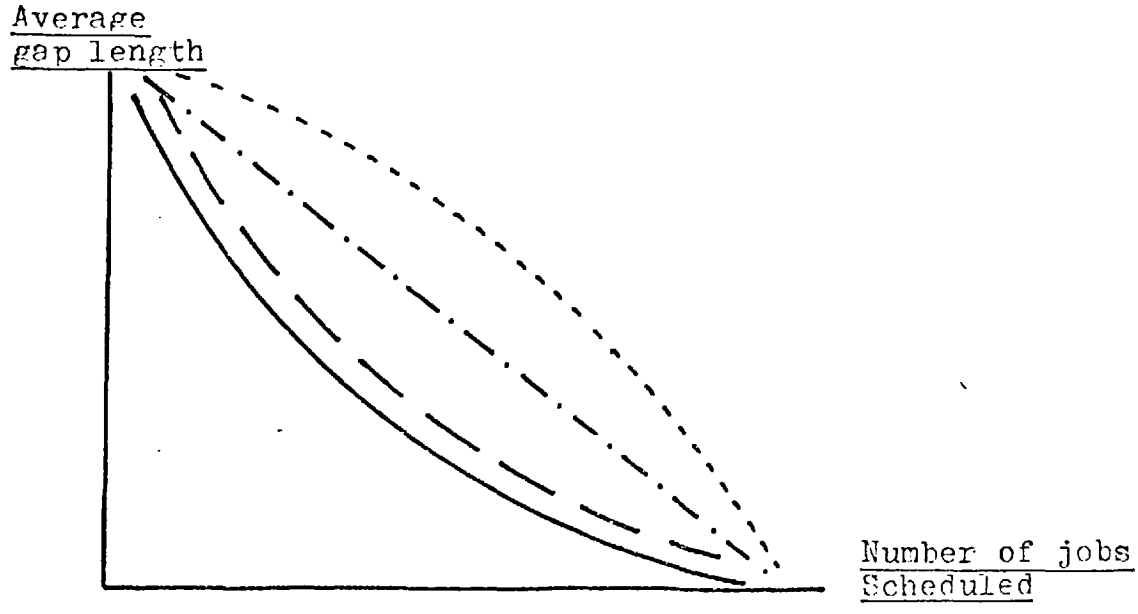


Fig. 24 (c) - trends for the average gap length

With a large number of gaps introduced into the schedules, the distribution of gap lengths with number of jobs scheduled will tend to Normal. This can be illustrated as shown in Fig. 25.

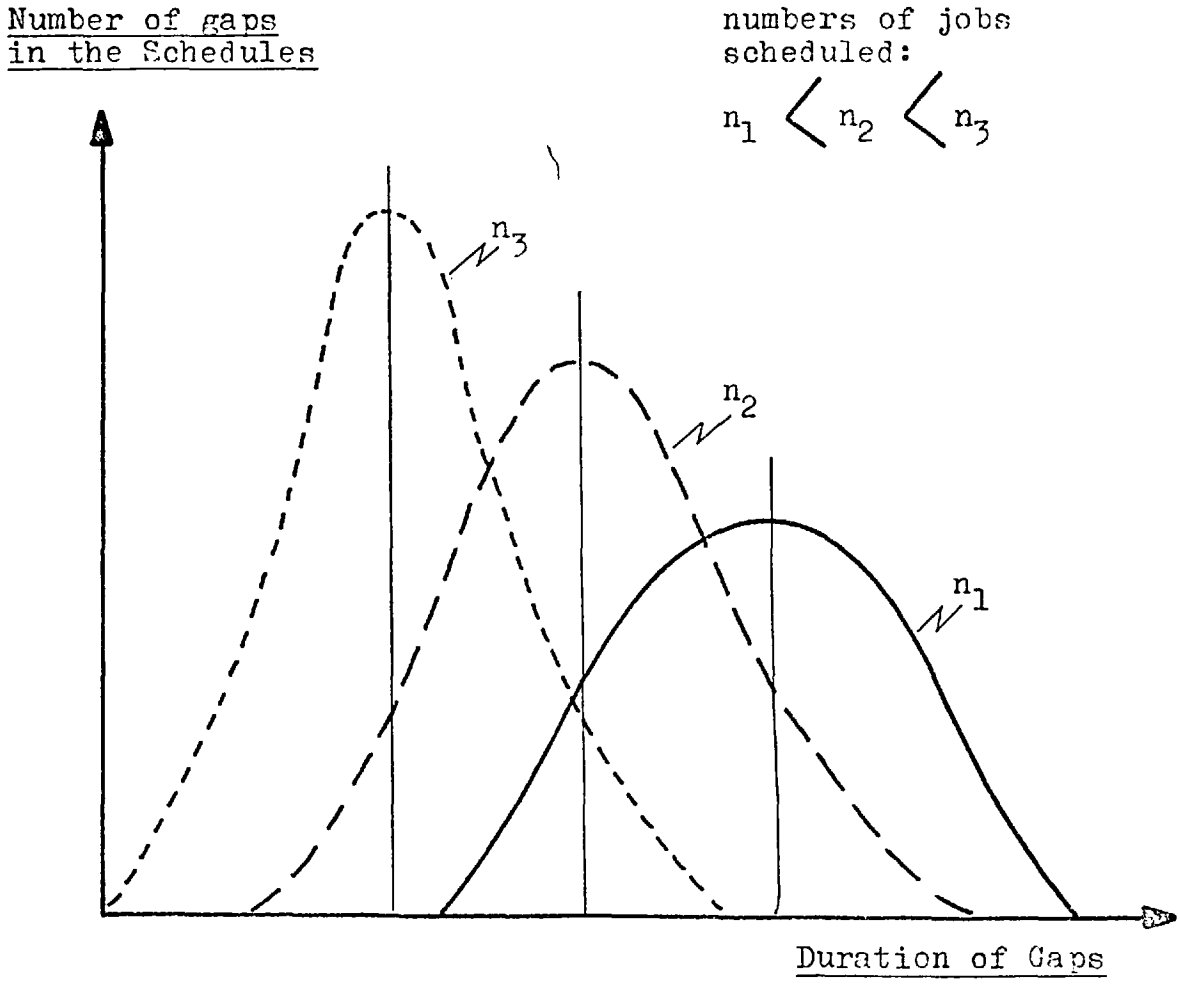


Fig. 25 - illustrating the distribution of gap lengths with an increasing number of jobs scheduled (n)

In this distribution it has been shown that the mean value for gap duration may be taken as:

$$\bar{G} = \frac{MT - \sum_{i=1}^{n} d_i}{n - n_a + M}$$

The Normal distribution for gap lengths is therefore:

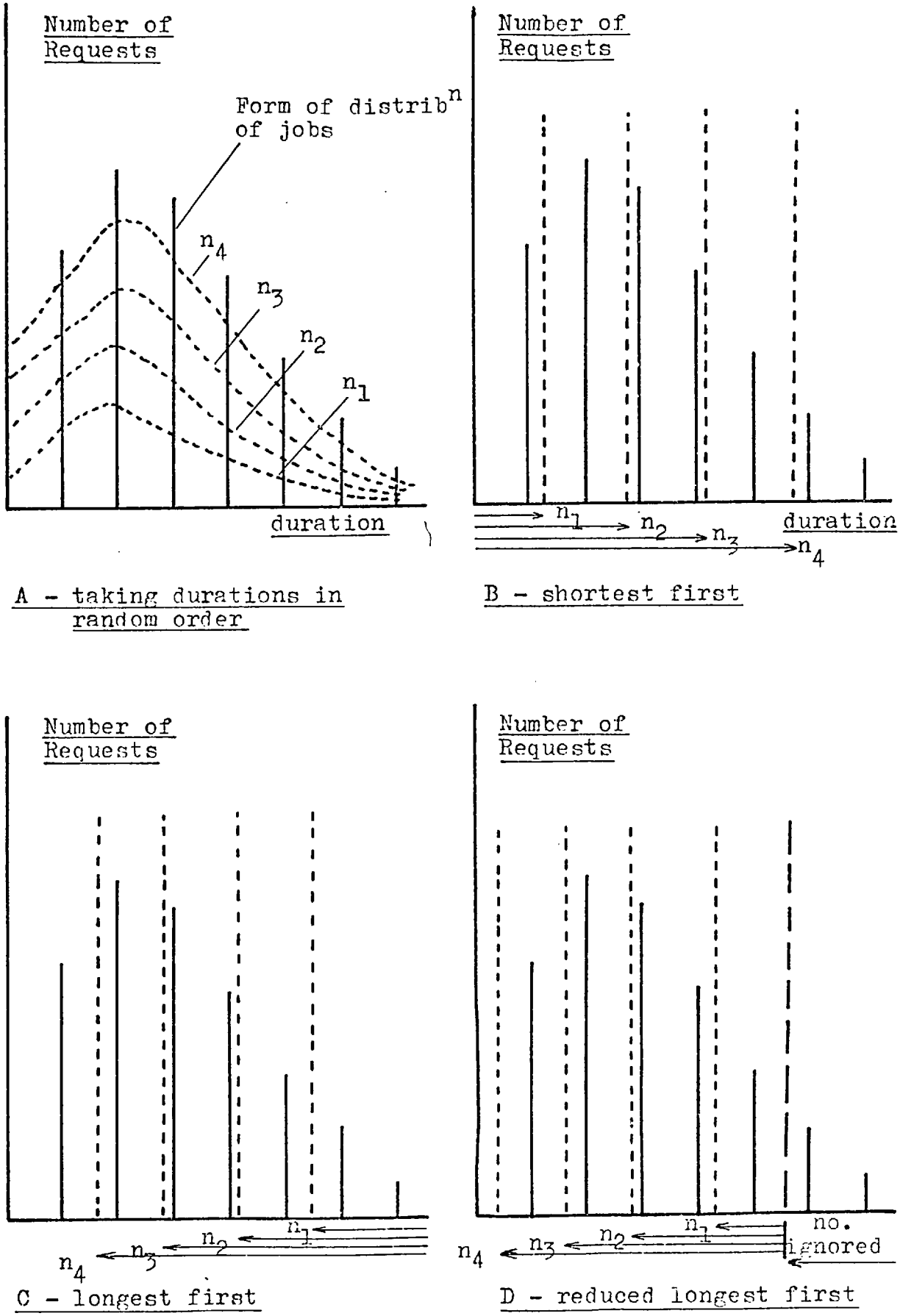
$$P(G) = \frac{1}{\sigma\sqrt{2\pi}} \quad e^{\frac{-(G-\bar{G})^2}{2\sigma^2}}$$

Here σ (the standard deviation of the sample) would be difficult to evaluate, and it is probable that an estimated value would have to be taken.

In the same way the distribution of requests still to be examined for scheduling can be considered, and the results shown in Fig. 26 obtained. It is assumed that the initial job durations have a Poisson distribution of the form:

$$P(d) = \frac{e^{-\lambda} (\lambda)^d}{d!}$$

Where : d = request duration and λ = mean duration of all requests



A - taking durations in random order

B - shortest first

C - longest first

D - reduced longest first

Fig. 26 - illustrating the distribution of requests scheduled with duration ($n_1 < n_2 < n_3 < n_4$)

Looking at these graphs it may be recognised intuitively that the most efficient shcedules appear to be produced when both the gap length and the unscheduled request durations are decreasing. This means that scheduling the longest jobs first will provide the best machine utilisation.

If, however, there are a large number of long request the most satisfactory schedules may be produced by ignoring the longest of them, i.e. taking the requests in the order 'reduced longest first'.

This hypothesis can be examined further if a combination of Fig. 25 and Fig. 26 is considered, as illustrated in Fig. 27. Here:

A - shows the results of scheduling the requests in an order independent of duration

B - requests are scheduled shortest first

C - longest first

D - reduced longest first.

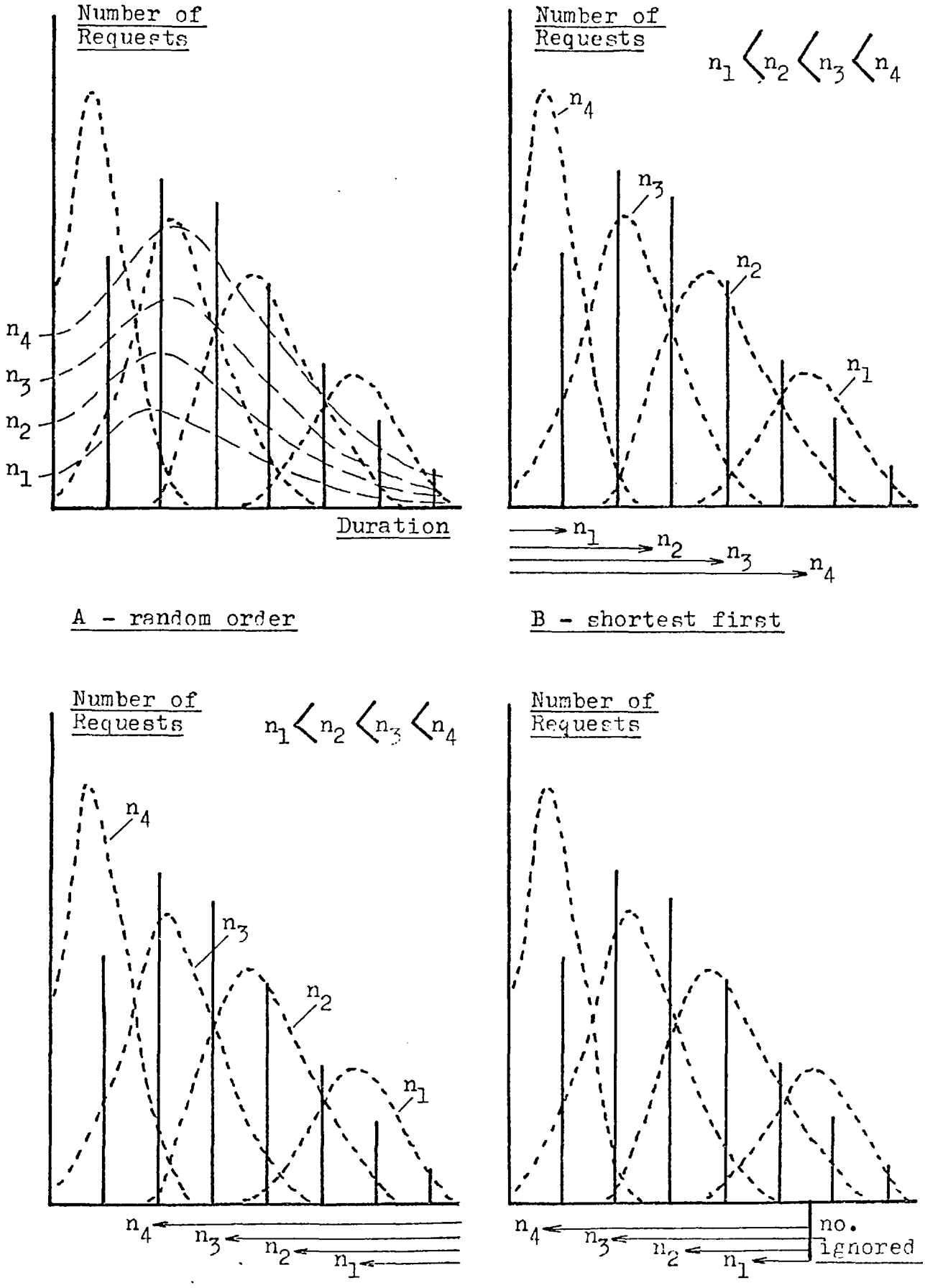


Fig. 27 - relationship between gap length and the duration of jobs at points in the scheduling

From these graphs it can be seen that in case 'B', when the requests are considered in the order shortest first, less than n_4 requests can be scheduled. Before this point the longest gap becomes shorter than the shortest job remaining.

In cases 'C' and 'D', however, this problem does not arise, and requests can be scheduled until there is no more machine time available.

Scheduling the requests in an order independent of duration gives a result between these two, where some requests, though not all, can be scheduled at any point. Here the critical factor may be the machine time available, or it may be the gap length remaining.

With any number of requests the point represented by n_4 will be reached before all available machine time is used, provided that there is not a very large number of short requests or there are very few requests to be scheduled. In the first of these exceptions the duration of scheduled jobs will remain small, while in the second case the gap length will remain large.

It can be concluded, therefore, that in most cases more machine time will be scheduled when the requests are considered in some order which involves the longest ones being considered first.

If there is a greater demand for machine time than supply available, then the number of requests scheduled will be limited. In particular, if the longest requests are scheduled first, then it seems likely that fewer of them will be scheduled.

Interesting results can be obtained to confirm this by considering the points at which the average duration of the remaining requests equals the average gap duration. This is the point where, on average, no further requests can be scheduled.

At any point during the scheduling the average duration of requests still to be considered for scheduling is:

$$\lambda = \frac{T_o - \sum_{i=1}^{n} d_i}{N - n} = \frac{\text{total time requested, but still unshceduled}}{\text{number of requests still unscheduled}}$$

where:

λ = average duration of requests still to be considered for scheduling

T_o = total time requested

N = total number of requests

n = number of requests scheduled

d_i = the duration of the ith request scheduled

Thus at the point where the average duration of the remaining requests equals the average gap duration:

$$\frac{MT - \sum_{i=1}^{n} d_i}{(n-n_a) + M} = \frac{T_o - \sum_{i=1}^{n} d_i}{N - n}$$

Scheduling the longest requests first will fill all the available machine time when:

$$\sum_{i=1}^{n_1} d_i \simeq MT$$

Conversely scheduling the shortest first will produce a critical point when:

$$\sum_{i=1}^{n_2} d'_i \ll MT$$

where 'n_2' is the number of jobs scheduled and d'_i is the duration of the i_{th} job when taken in the order shortest first

Now:
$$\sum_{i=1}^{n_1} d_i = \frac{T_o\,(n_1 - n_a + M) - MT(N - n_1)}{2n_1 - n_a + M - N} \qquad \text{----------(1)}$$

and
$$\sum_{i=1}^{n_2} d'_i = \frac{T_o\,(n_2 - n_b + M) - MT\,(N - n_2)}{2n_2 - n_b + M - N} \qquad \text{----------(2)}$$

As (1) $\geqslant$ (2):

$$(T_o n_1 - T_o n_a + T_o M - MTN + MTn_1)(2n_2 - n_b + M - N) \geqslant$$

$$(T_o n_2 - T_o n_b + T_o M - MTN + MTn_2)(2n_1 - n_a + M - N)$$

On simplification this becomes:

$$An_2 + Bn_1 \geqslant An_1 + Bn_2$$

where: $A = 2(T_o M - T_o n_a - MTN)$

$$B = T_o M - T_o n_a - NT_o - MTN + M^2 T - MTn_a$$

and $n_a = n_b$ which can be accepted for large values of n_1 and n_2

From this it can be seen that $\underline{n_2 \geqslant n_1}$ when $A > B$

i.e. when:

$$T_o M - T_o n_a - MTN > M^2 T - MTn_b - NT_o$$

i.e. $\underline{T_o > MT}$ (With $M + N\ n_a$)

Thus whenever the demand for machine time is greater than the supply, scheduling the longest requests first will schedule fewer jobs than scheduling the shortest first, provided the initial condition:

$$\sum_{i=1}^{n_1} d_i \geqslant \sum_{i=1}^{n_2} d_i$$

is met.

Conversely it can be shown that to achieve the best machine utilisation the requests should be considered in order of decreasing duration.

Using the same notation as before, scheduling the longest job first will mean that n_1 jobs are scheduled before the average gap length equals the average duration of the remaining requests, while taking the shortest jobs first will mean n_2 jobs are scheduled before this point is reached. Here n_1 and n_2 have the following values:

$$n_1 = \frac{NMT - N\sum_{i=1}^{n_1} d_i - MT_o + M\sum_{i=1}^{n_1} d_i - n_a\sum_{i=1}^{n_1} d_i + T_o n_a}{MT + T_o - \sum_{i=1}^{n_1} d_i} \quad \text{----(1)}$$

$$n_2 = \frac{NMT - N\sum_{i=1}^{n_2} \sqrt{\sigma_i} - MT_o + M\sum_{i=1}^{n_2} \sqrt{\sigma_i} - n_b\sum_{i=1}^{n_2} \sqrt{\sigma_i} + T_o n_b}{MT + T_o - \sum_{i=1}^{n_2} \sqrt{\sigma_i}} \quad \text{----(2)}$$

Now as $n_2 \geqslant n_1$ it can be seen that;

$$(NMT - N\sum_{i=1}^{n_2} \sqrt{\sigma_i} - MT_o + M\sum_{i=1}^{n_2} \sqrt{\sigma_i} - n_b\sum_{i=1}^{n_2} \sqrt{\sigma_i} + T_o n_b)(T_o + MT - 2\sum_{i=1}^{n_1} d_i$$

$$\geqslant (NMT - N\sum_{i=1}^{n_1} d_i - MT_o + M\sum_{i=1}^{n_1} d_i - n_a\sum_{i=1}^{n_1} d_i + T_o n_a)(T_o + MT - 2\sum_{i=1}^{n_2} \sqrt{\sigma_i})$$

For large values of n_1 and n_2 it can be taken that $n_a = n_b$, from which:

$$\sum_{i=1}^{n_1} d_i \; (2MT_o - 2NMT - 2T_o n_b + NT_o + NMT - T_o M - M^2 T + T_o n_a + MTn_a)$$

$$\gg \sum_{i=1}^{n_2} \sigma_i \; (2MT_o - 2NMT - 2T_o n_a + NT_o + NMT - T_o M - M^2 T + T_o n_b + MTn_b)$$

or:

$$\sum_{i=1}^{n_1} d_i \gg \sum_{i=1}^{n_2} \sigma_i$$

Thus scheduling the longest request first produces the better machine utilisation.

Apart from spreading the jobs throughout the time bands during the scheduling runs, the presence of fixed and preferred conditions has further significance. Notably, as was suggested earlier, the most satisfactory schedules will be produced if the requests are considered in decreasing order of fixedness.

However, if either or both of these features are taken into account during the scheduling run the effects of taking the requests in strict order according to duration are reduced.

Thus if weight is given to scheduling the most important jobs, or satisfying fixed and preferred conditions, the best machine utilisation and highest possible number of jobs scheduled will probably not be achieved. There is also likely to be some direct relationship between the proportion of requests with fixed conditions and the quality of the resulting schedule.

If there are insufficient requests to fill the available machine time the differences given by the various loading rules proposed will obviously be reduced. This is especially true for very small numbers of requests, where most of them will be scheduled using any of the rules, provided that sufficient data checks have been performed on the input data. This was illustrated on page 102, where it was shown that scheduling the shortest requests first meant that the highest number of requests were scheduled provided that the requested time was greater than the available machine time.

A summary of the results expected from the loading rules proposed is as follows:

(a) considering the shortest requests first will schedule the greatest number of jobs, provided there are too many requests for all of them to be scheduled

(b) Considering the longest requests first will produce the best machine utilisation, provided there are too many requests for all of them to be scheduled.

(c) scheduling in the order of reducing importance will probably give the most satisfactory results

(d) scheduling in the order of reducing fixedness will probably give the most satisfactory results

(e) factors (c) and (d) above will decrease the effects of factors (a) and (b), so that there will be less differences in the schedules when these are taken into account

(f) comparatively poorer machine utilisation will result when there are a large number of fixed conditions on the requests

(g) there is likely to be less difference between the various loading rules when the total time requested is small compared with the total machine time available.

5.8 Actual Results of the Test Scheduling Runs

Having considered in theory the results to be expected from the loading rules, test runs were made using the data described in section 5.5 Runs using the twenty loading rules evolved were made, scheduling between 25 and 250 requests.

A summary of the results obtained is shown in Fig. 28. Here the number of jobs scheduled, total time, machine utilisations and fixed and preferred conditions satisfied are listed and compared.

A more complete description of the results obtained is given in Appendix C.

Approach Number	No. of jobs Scheduled		Time sched. (hrs.mins)		Best av. Machine Utilis.	% Fixed Condit. Satisfied			% Pref. Condit. Satisfied			% Function 1 Scheduled			% Function 2 Scheduled		
	at 250	max	at 250	max		250	max	min	250	max	min	250	max	min	250	max	min
1	164	164*	371.30	378.45	94.7	41.8	94.1	41.8*	46.6	100	46.6	60.0	88.9	60.0*	51.9	100	51.9*
2	168	168	372.15	377.10	94.3	56.0	100	56.0	45.8	93.8	45.8	96.0	100	95.7	92.6	100	90.5
3	187	187	365.00	366.10	91.6	86.6	100	86.6	48.3	100	48.3	84.0	100	84.0	77.8	100	77.8
4	182	182	357.10	365.25	91.4	87.3	100	87.3	50.8	100	50.8	92.0	100	91.3	88.9	100	88.9
5	172	172	376.05	376.05	94.0	61.2	100	61.2	42.4	100	42.4	100	100	100	92.6	100	90.5
6	179	179	312.35	312.35*	78.1*	57.5	100	57.5	53.4	100	53.4	72.0	100	68.4	63.0	100	63.0
7	171	171	361.50	361.50	90.5	58.2	100	58.2	47.5	100	47.5	100	100	100	92.6	100	90.5
8	185	185	349.50	351.40	87.9	85.8	100	85.8	52.5	100	52.3	80.0	100	78.9	81.5	100	81.5
9	184	184	359.40	359.40	90.4	87.3	100	87.3	48.3	100	48.3	96.0	100	94.7	88.9	100	88.9
10	174	174	375.20	375.20	93.8	61.2	100	61.2	45.8	100	45.8	100	100	100	92.6	100	90.5
11	120	139*	395.30	397.15	99.3	21.6	85.3	21.6*	19.5	100	19.5*	36.0	77.8	26.1*	33.3	100	33.3*
12	164	164*	382.10	382.10	95.5	54.5	100	54.5	39.0	100	39.0*	96.0	100	95.7	96.3	100	95.2
13	185	185	374.20	374.20	93.7	87.3	100	87.3	44.1	100	44.1	84.0	100	84.0	85.2	100	85.2
14	182	182	362.05	368.40	92.2	86.6	100	86.6	48.3	100	48.3	88.0	100	88.0	85.2	100	85.2
15	170	170	379.20	379.20	94.8	61.2	100	61.2	45.8	100	45.8	100	100	100	92.6	100	90.5
16	170	170	378.40	378.40	94.7	41.0	85.3	41.0*	39.0	100	39.0*	40.0	77.8	33.3*	48.1	100	41.2*
17	183	183	365.35	367.40	91.9	59.7	100	59.7	45.8	100	45.8	88.0	100	87.0	96.3	100	95.2
18	189	189	366.50	370.10	92.6	84.3	100	84.3	48.3	100	48.3	84.0	100	84.0	88.9	100	85.7
19	189	189	360.35	365.30	91.4	85.2	100	85.2	41.5	100	41.5	92.0	100	91.3	92.6	100	92.6
20	186	186	364.40	367.55	92.0	66.4	100	66.4	53.4	100	53.4	92.0	100	91.3	96.3	100	90.5

* indicates a very poor result

Fig. 28 - showing a summary of the results obtained from the twenty loading rules

The overall trends predicted in chapter 5.7 are verified in the test scheduling runs. For example, if graphs are drawn for the number of requests scheduled and the total time scheduled for loading rules 1 (independent of duration), 6 (shortest first), 11 (longest first) and 16 (reduced longest first) the results shown in Figs. 29 and 30 are obtained.

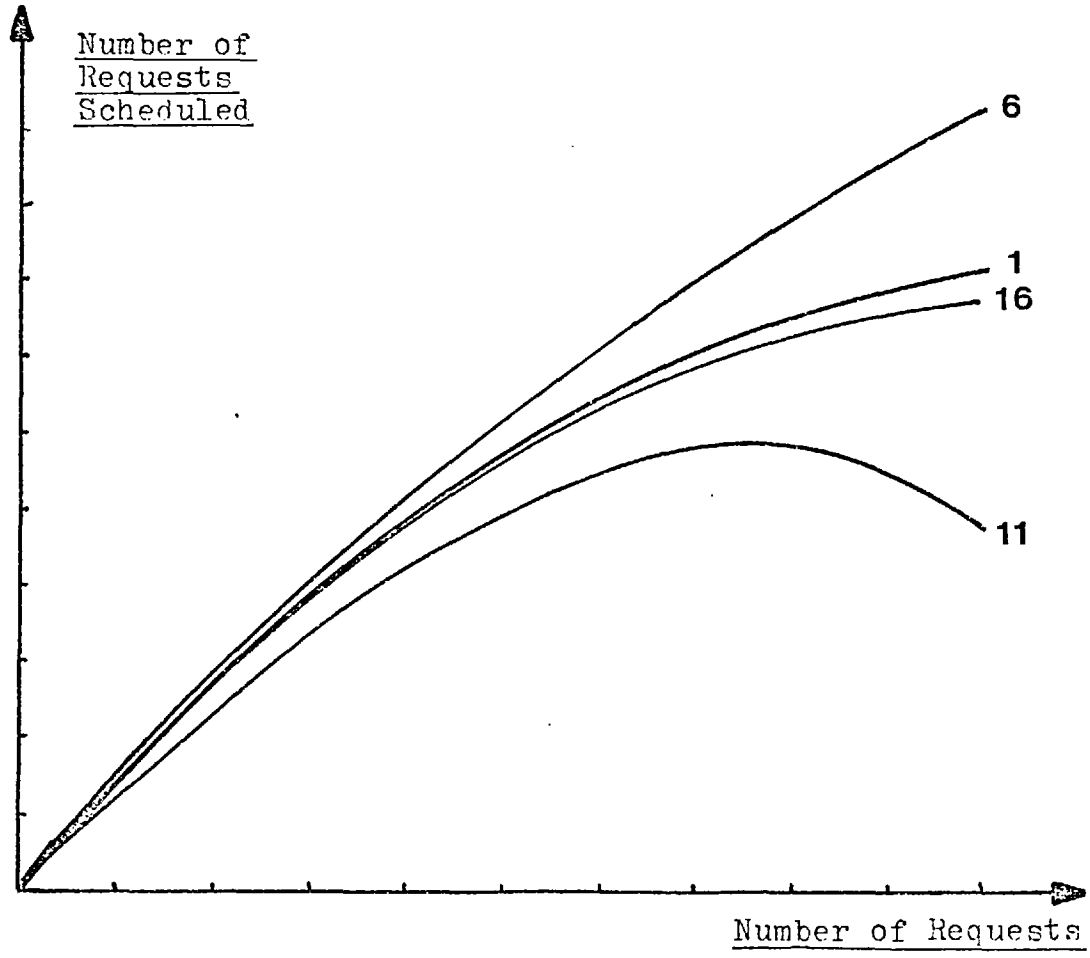


Fig. 29 - illustrating the trends in numbers of requests scheduled with various numbers of requests received (loading rules 1,6,11,16)

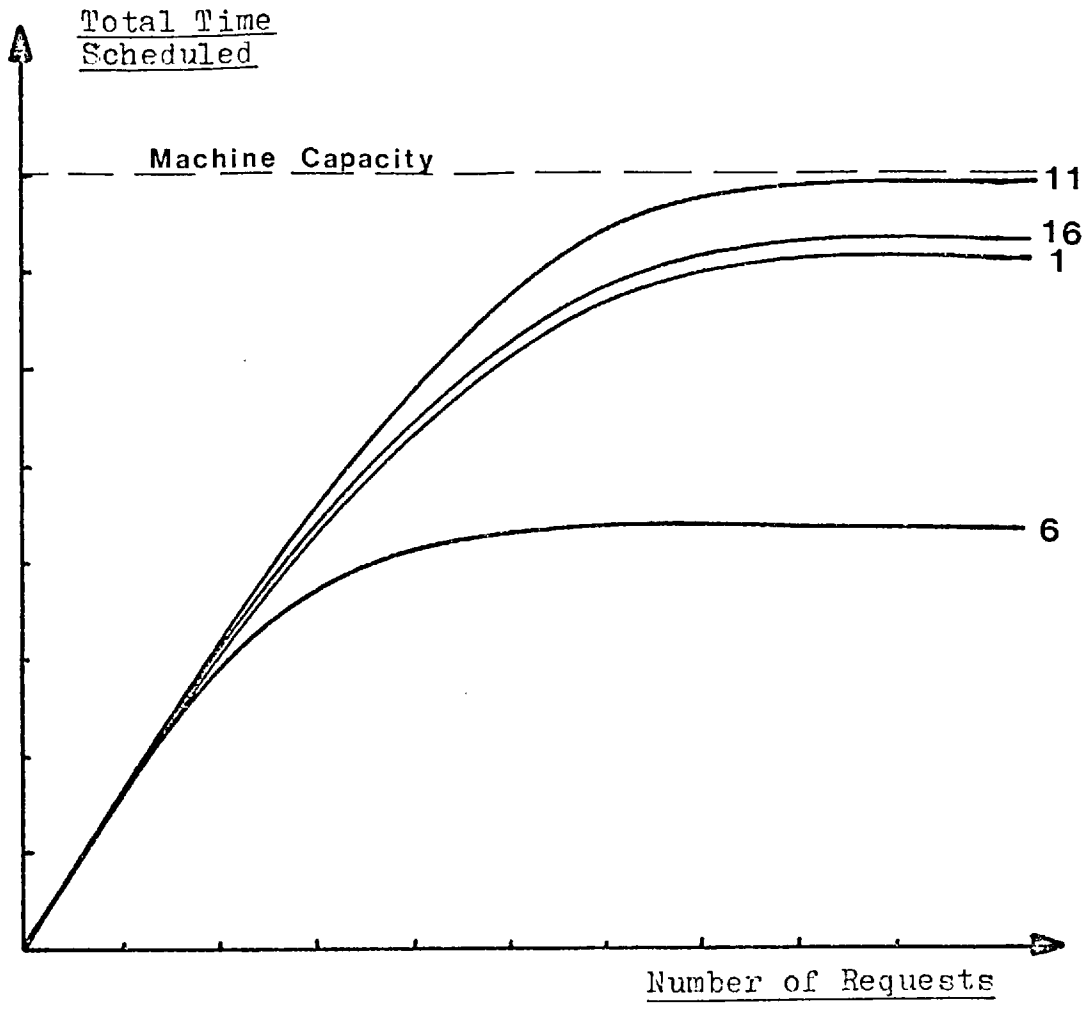


Fig. 30 - illustrating the trends in times scheduled with various numbers of requests received (for loading rules 1, 6, 11 and 16)

Clearly scheduling the shortest requests first results in more jobs being scheduled, while scheduling the longest requests first gives better machine utilisation. In addition it can be observed that these variations become more pronounced with increasing time requested.

For different numbers of requests queued for scheduling the two most significant predictable trends are:

(a) when taking the requests in the order shortest first the number of jobs scheduled steadily increases with increasing time requested, but the time scheduled reaches a plateau before the machine capacity is reached

(b) when taking the requests in the order longest first the machine utilisation reaches a plateau near to maximum capacity, while the number of jobs scheduled reaches a peak and then declines. In an attempt to avert the effects of this decline in the number of requests scheduled when the longest requests are considered first, the loading rules based on 'reduced longest first' were introduced.

When examining the results obtained from the test runs it is possible to put the loading rules into an order of preference based on their performance compared with certain predetermined criteria. Fig 31, for example, lists the best loading rules when judged by the most important six criteria.

Criteria Considered	Best Loading Rules in Order of Performance
(a) maximum number of jobs scheduled	18 and 19, 3, 20, 8 and 13, 9, 17
(b) maximum time scheduled (best machine utilisation)	11, 12, 15, 1 and 16, 2, 5, 10
(c) maximum minimum percentage of fixed conditions satisfied	4 and 9 and 13, 3 and 14, 8, 19, 18
(d) maximum minimum percentage of preferred conditions satisfied	6, 20, 8, 4, 3 and 9 and 14 and 18
(e) maximum minimum percentage of function 1 requests scheduled	5 and 7 and 10 and 15, 2 and 12, 9, 4 and 19 and 20
(f) maximum minimum percentage of function 2 requests scheduled	12 and 17, 19, 2 and 5 and 7 and 10 and 15, 20

Fig. 31 - showing the best loading rules when judged by six different criteria (equally good results are indicated by 'and')

The main point to be observed is that the decision as to which is the best schedule depends to a large extent upon the criteria which are considered most important. Several of the approaches perform poorly whichever criteria are used, while others perform better or worse depending on how they are judged.

It can be said, for instance, that approaches number 9 and 19 give results which are generally fairly good when judged by any of the criteria mentioned in Fig. 31. Conversely loading rule number 11 usually gives fairly poor results, and yet gives the best machine utilisation.

In general terms the following conclusions can be drawn from the results obtained:

(a) loading rules 1, 6, 11, 12 and 16 gives results which perform so badly when judged against at least one of the criterion that they are unlikely to be acceptable (asterisked in Fig. 28)

(b) loading rules 6 and 10 give comparatively poor machine utilisation

(c) numbers 1, 2, 6, 7, 11, 12, 16 and 17 give poorer satisfaction of fixed conditions that the rest.

Bearing in mind the commercial objectives which are usually present in the performance of such a scheduling sub-system, it is likely that approach 15 will frequently be an attractive alternative. However, in cases where there is an emphasis on some specific performance, say scheduling more high priority jobs, or giving better machine utilisation, there may be a preference for approaches 10 or 11.

Again the overriding comment which must be made is that there is no single 'best' loading rule, but only one which best achieves some criterion which is taken as the measure of performance for a particular situation.

SECTION B

THE SOLUTION OF A PRACTICAL SCHEDULING PROBLEM

6. A DESCRIPTION OF THE PRACTICAL PROBLEM CONSIDERED

6.1 Introduction

In Section A a situation was considered where a computerised scheduling sub-system was required for the solution of a specified type of problem. A solution was proposed for this type of problem which involved on-line computer operations using video display units, batch type operations initiated via these terminals and the comparison of several loading rules evolved during the study.

It was said on several occasions during the description of the proposed solution that although guidelines may be given the details of the sub-system would depend on the particular situation in which it is used. Moreover emphasis was placed on the fact that this was essentially a search for a solution to a problem which could be met in practical situations. The next step, therefore, is to move from the general solution to a specific one capable of meeting a particular application.

Section B describes the application considered, together with the modifications necessary to the sub-system in order to provide a solution to the practical problem. In particular decisions concerning the best loading rules, VDU layouts and so on will be explained.

One practical example of the type of situation described in Section A arises in the scheduling of video-tape machines in large television broadcasting organisations. Such a problem, arising in the British Broadcasting Corporation, has already been described in some detail[15], and it is beneficial to take this example to illustrate a practical use of the scheduling sub-system developed.

In the BBC requests for video-tape machine usage are sent by Production Departments to the Programme Planning Department, who produce and distribute complete schedules to machine operators, control rooms and several other destinations.

A description of the general situation is given in the following sections.

6.2 Background to the Problem Faced by the BBC

BBC television has as its objective the production and transmission of television programmes, on two channels, for about 20 hours a day. This involves it in the making of some 6,000 complete programmes a year at a total cost of around £100m.

In the mid 1960s the BBC was faced with the problem of increasing costs, caused by increased hours of transmission, the growth of BBC 2, the introduction of colour services and the general effects of inflation,

while its income, controlled by Government policy, was to a large extent stationary.

It was obvious that some major rethinking was necessary within the organisation to overcome this trend for an increasing budget deficit.

Towards this end the initial stages of a Television Management Information System (TMIS) were formulated with the explicit objectives of determining:

a) that the service was making the best use of programme monies
b) whether productivity of resources could be increased
c) whether complexity could be reduced.

It was concluded at this time that these objectives could only be achieved with the aid of computers. It followed that in October 1969 the first computer operations were introduced into the Corporation.

Not surprisingly this System had undergone continuous developments with the purpose of extending its scope and making its overall performance more sophisticated.

Bearing in mind objective (b) of TMIS mentioned above, an obvious extension to the existing system is that of scheduling certain scarce technical resources, among

them video-tape machines. This is the problem to be tackled here.

In particular two valid reasons can be put forward for the development of a new system for scheduling these machines:

a) In the first place the machines are comparatively expensive both to buy and to operate. For example, a major manufacturer of video-tape machines quotes a purchase price of up to £75,000 and up to £50 an hour running costs. At these prices the BBC could have a capital outlay in London of about £2m and annual running costs of between £2m and £4m. A small percentage saving in these costs would obviously produce a sizeable cash benefit;

b) secondly the old manual system for scheduling the machines is reaching the end of its ability to cope with the increasing demands put on it. Figs. 32 and 33 illustrate the rise not only in the number of machines to be scheduled, but also in the utilisation of each machine. While the system copes adequately during the summer, at times of higher demand the system begins to break down, with schedules getting poorer and outside charges to hire extra machines rising (if these extra machines are available for use).

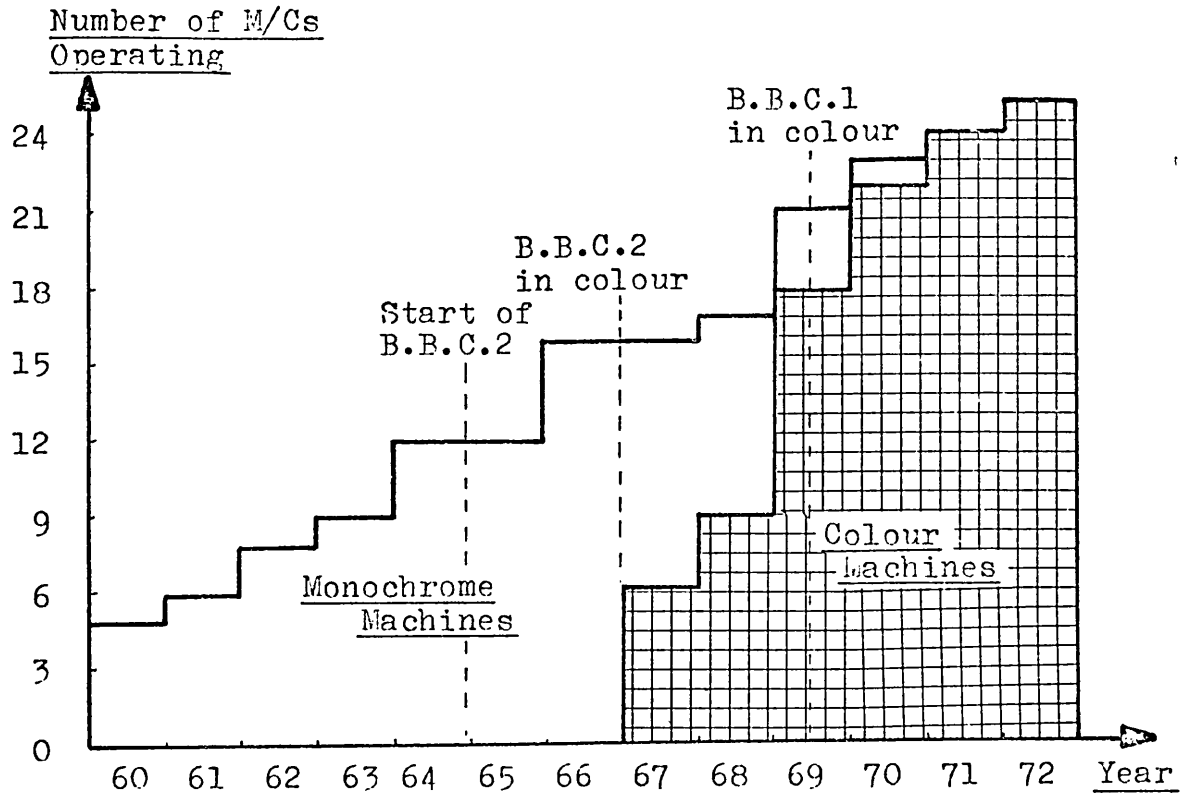


Fig. 32 - showing the number of video-tape m/cs in operation

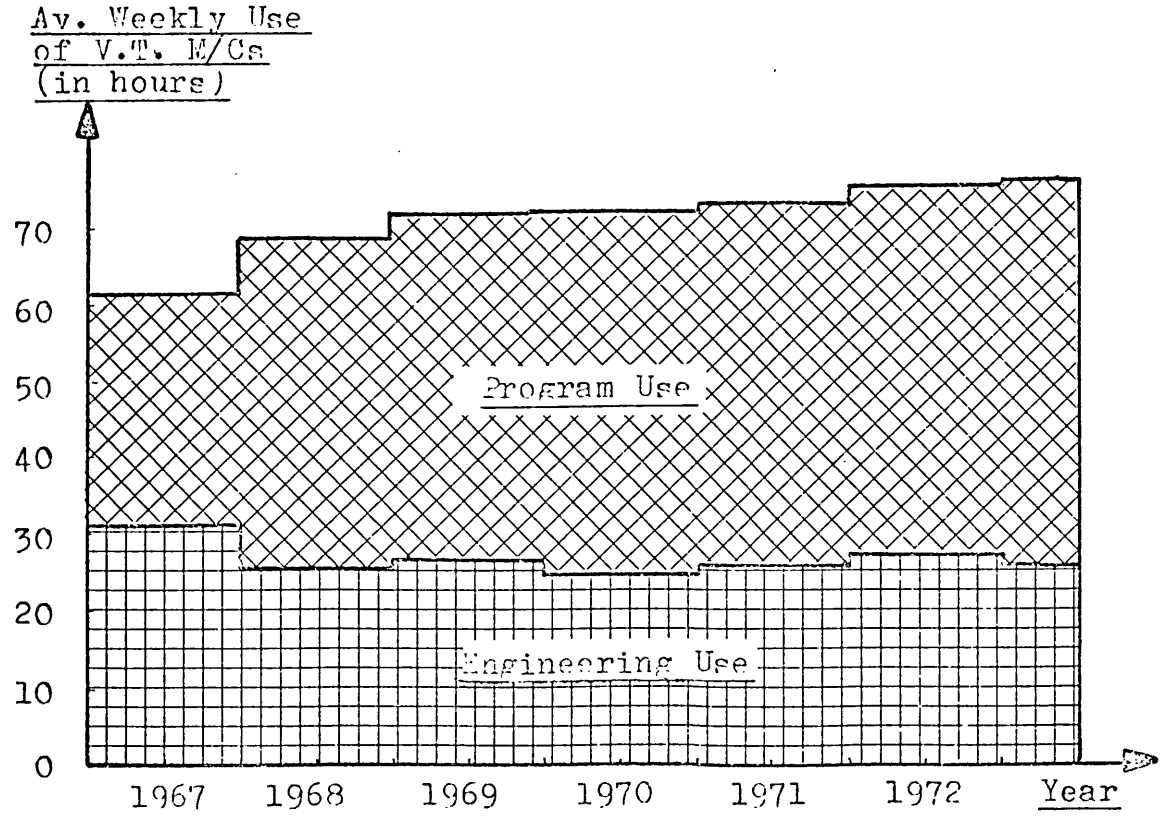


Fig. 33 - showing the average weekly use of V.T. machines

6.3 A Description of the Present Scheduling System

As has already been said a full description of the present system for scheduling video-tape machines at the BBC has been given in reference number 15. However, a brief summary describing the general characteristics of the situation is given at this point.

The scheduling procedure starts some six to eight weeks before the video-tape machines (VTs) are needed. An 'allocation clerk' enters dates, page and machine numbers on a standard form used to produce the schedules. This form is illustrated in Fig. 34 as it appears when completed at the end of the manual scheduling.

Details of regular or 'block' bookings are then added in pencil, these being read from a 'block bookings file' which is kept up to date by frequent checks. Also at this time machine maintenance times, regular playback sessions, and advance notice of special facilities needed for major events are added. Finally 'transmission plans' and 'current studio arrangements' are used to fill in any further machine requirements known at this stage.

Five weeks before the machines are to be used, a 'recording assistant' takes a week's draft schedules from

the allocation clerk and completes the scheduling during the following five weeks.

Request forms for machine time from the production departments are examined, and any discrepancies or problems are cleared up over the telephone. At this point new recordings are given a unique identifying number.

From transmission and studio usage already scheduled, the staffing pattern for machine operators is worked out. In practice this is fairly straightforward, but occasionally problems do arise. At the start of the enquiry it was ascertained that if machine time is available an operator or operators can invariably be found.

Spare time for machines and men is then filled as efficiently as possible with the remaining requests for machine usage. In order to do this, and still keep within the constraints of the operation, it is necessary to repeatedly change and rearrange the bookings. This is particularly so as late requests are continually arriving, and each new arrival may necessitate many changes in the schedules. In practice these alterations are made by continually rubbing out and rewriting, in pencil, the schedules already produced.

Rather than build-up a backlog of requests, excess work may be put out to external hire. Rearrangements are then necessary to the schedules to determine the cheapest means of processing the extra work, bearing in mind that hire charges are more expensive than equivalent internal operations.

Although there is a general order of priority for machine use, with some functions (notably transmissions) being considered more important than others, this may be overriden at any time by unusual circumstances. One of the main characteristics of this particular scheduling problem is that the situation is extremely flexible, with hard and fast rules being difficult to determine. The sub-system must be able to cope with this flexibility, and the schedulers must maintain complete control over the operations in order to allow for the subjective decisions necessary.

The scheduling process is complicated by the fact that more information and requests are arriving by telephone and post all the time. This includes the updating of transmission plans five times during the scheduling (at 8 - 9 weeks before transmission date, 5 - 6 weeks, 3 - 4 weeks, 2 - 3 weeks and ten days).

The importance of the new requests must be weighed against existing booking scheduled, and where necessary changes made. This results in further negotiations with production departments, and possible cancellation or postponement of existing bookings.

Five days before the machines are due to be used copies of the 'provisional' schedules are sent out to a number of people connected with the use of VT machines. Any subsequent changes must be relayed to these people.

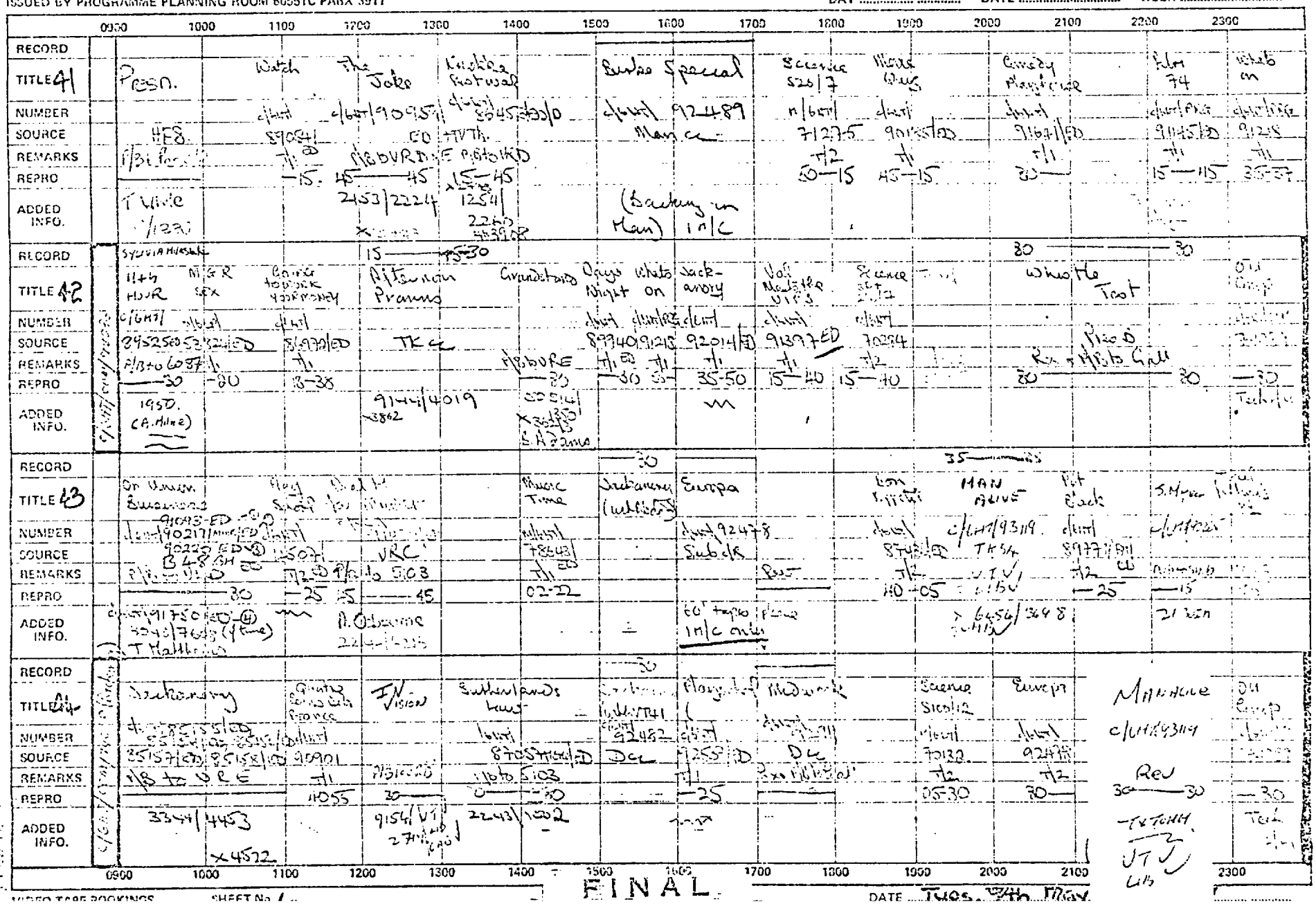
Three days later the schedules become 'revised' and more copies are sent out. After this only essential changes are made. Last checks are made of recording times etc. so that the 'final' schedules are distributed the day before the machines are to be used.

The distribution of schedules cannot in reality be this regular, because the recording assistants do not normally work at week-ends, and unavoidable occurences (such as machine breakdowns, major national events causing transmission reorganisations and so on) make the whole scheduling process somewhat less regular than indicated. The basic system as described, however, gives a fair indication of the general procedures followed during a complete scheduling cycle.

Fig. 34 shows an example of the completed scheduling form as it appears at the end of the scheduling cycle. When finalised this form is sent away to be typed, copied and distributed as necessary.

The complete scheduling process can be summarised as shown in Fig. 35.

Fig. 34 - illustrating the present scheduling form ($\frac{5}{8}$ actual size)



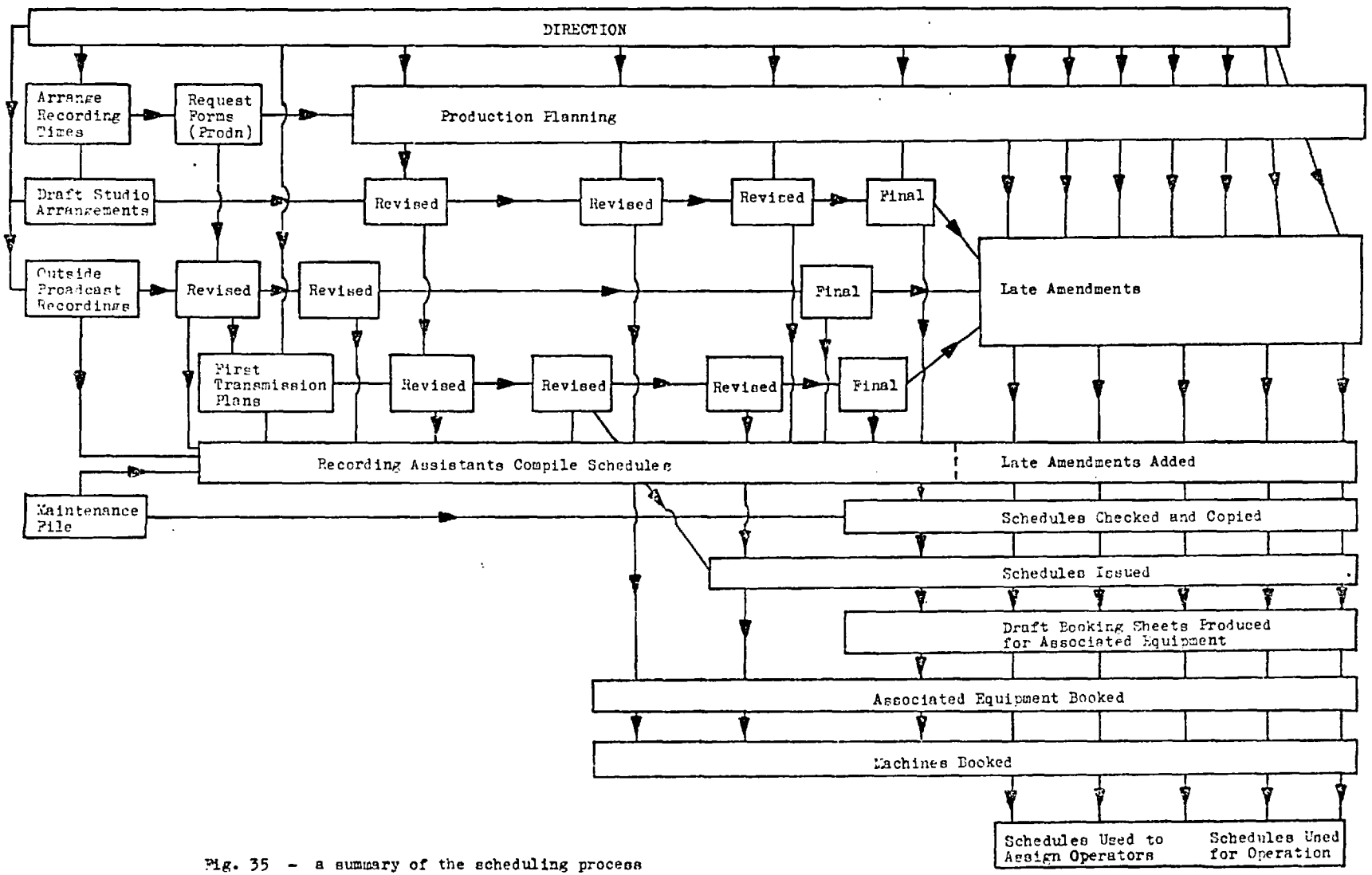


Fig. 35 - a summary of the scheduling process

6.4 A Description of the Machines and Requests for Machine Time

6.4.1 The Requests for Machine Time

In general terms the requests for machine time have features similar to those listed in section 1.2, but they have in addition several other factors to be considered. In particular each request is different and may have some or all of the following characteristics:

(a) A title (up to 32 alphanumeric characters).

Each job is associated with a television programme whose title it takes

(b) Project number (up to 12 alphanumeric characters)

For costing purposes the production departments are responsible for their machine usage. Project numbers are used to identify the department and project concerned.

(c) Recording number (24 alphanumeric characters).

Each video tape is given a unique number to identify it, and also give some indication of its nature.

(d) Source code (2 alphanumeric characters) and

(e) Destination code (2 alphanumeric characters). In many

cases VT machines are used to record from cameras at remote sites, or playback to screens at remote sites. In these cases it is easier to use a short code to identify the source or destination of recordings or playbacks in order to ensure that the necessary wiring circuits are available, while minimising computer storage requirements.

(f) Function code (2 numbers)

As in the theoretical study described in Section A there are ten functions to be considered. These are coded, to reduce storage requirements, in the following manner:

1. Machine maintenance
2. Transmission on BBC 1
3. Transmission on BBC 2
4. Record
5. Edit/review/dub using a pair of machines
6. Edit/review/dub on three machines
7. transfer (525 or 405 lines to 625 lines, or film to video tape etc.)
8. Edit/review/dub on two machines (also used as a continuation of function 5)
9. Technical line-ups etc.
10. Playbacks

These are listed above in an approximate order of importance.

(g) Start time (4 numbers)

Preferred or fixed start times may be given (in hours and minutes on a 24 hour clock).

(h) Duration (4 numbers). (hours and minutes)

The duration, excluding machine preparation time, will vary from about 2 minutes to 15 hours in a single day.

(i) Line-up time (2 numbers)

This is the machine preparation time needed before each job. It varies between about 15 minutes and 60 minutes, depending primarilly on the function of the job.

(j) Duration flexibility (2 numbers)

In some instances, especially with long jobs, the duration is not critical to within a few minutes. If a request cannot be scheduled at its full duration an amount of flexibility may be specified (in minutes) by which the request may be shortened in order to fit it into the schedule.

(k) Machine number (three sets of 2 numbers).

The machines are referred to by a two digit

number. When a fixed or prefered machine is given then only 2 digits are used, but on jobs which need three machines there must be sufficient space to record all the numbers.

(l) Fix code (one number).

As explained in section 5.4. there are nine possible variations in the amount by which a job is 'fixed' in one place in the schedule.

(m) Number of machines required (1 number).

This can vary between one and three for each job. Requests for more machines, working in parallel, will have to be treated as more than one job.

(n) Comments (48 alphanumeric characters).

For nearly all jobs there will be a certain amount of information which does not fit into the above categories. This can be added to a request in the form of comments, which do not alter the scheduling, but add to the information available.

In addition to the preceeding information which may be associated with each request, there are several other trends which may be noted. These include:

(a) there tends to be a correlation between function and the degree to which a request is fixed, in particular the lower the function code, the higher the fix code. Thus maintenance sessions and transmissions are normally more fixed than are playbacks etc.

(b) there is a correlation between duration and the degree to which a request is fixed (i.e. the shorter the request, the higher its fix code).

(c) the requests are of three basic types: regular or block bookings, normal one-off requests or low priority request which can be held over from day to day in order to fill any gaps which arise in the schedules.

(d) it is possible to carry-over some one-off requests from one day to another if this is advantageous.

(e) the number of requests for machine time which have to be scheduled each day is usually between 125 and 175, although this is subject to seasonal, weekly and other less predictable variations.

(f) certain times of the day are more popular with machine users, particularly afternoons.

(g) meal breaks for the machine operators have to be allowed for in long editing sessions. These amount to one hour in a

five hour session, two hours in a ten hour session and so on.

6.4.2. The Machines to be Scheduled.

The B.B.C. has, at the time of this study, twenty five video tape machines, each of which has an identifying number (in order to simplify matters they are referred to as 1 to 25 for the purpose of this description). Each of these machines is different.

Although most of the machines can handle most of the functions if necessary, there is an approximate order of preference of machines for functions. This is based on the fact that some machines are 'better' at handling certain functions than others.

In addition there are specialised machines which are usually reserved for particular functions. For example four 'machine pairs' are set aside as editing machines to be used for function '5'. If, however, these machines have spare capacity each pair may be used as a single machine for other functions. Conversely if more time is required for function '5' jobs, two separate machines may be utilised as for a function '8' job.

Other specialised machines include two

'transmission suites' which are largely reserved for transmissions, although they may be used for other functions during idle periods. These particularly occur early in the mornings. Again other machines may be used for transmissions under certain circumstances, such as during maintenance periods or where special facilities are needed.

Initially up to fifteen machines were considered, in order of preference, for each function. There were certain functions, however, which could not be processed by this number of machines, notably functions '5' and '6'.

The machines work a fifteen hour day, from 0900 to 2400, although this is somewhat flexible in that carry-over beyond midnight is possible in certain circumstances. This sort of carry-over, or early starts, may, however, cause staffing problems and such decisions are best left for the schedulers. The automatic scheduling was limited to the fifteen hour period indicated.

The order of preference of machines for each function is illustrated below. These were derived primarily from the machine capabilities, but an attempt was also made to spread the machine loadings evenly.

Function	Order of machine preference
1 - maintenance	any - but must be named
2 - transm. BBC1	22,23,24,25,13,10,12,19,20,21,2,4,14, 15,16.
3 - transm. BBC2	24,25,22,23,13,10,12,19,20,21,2,4,14, 15,16.
4 - record	1,9,10,11,12,13,17,18,19,2,3,4,14,15, 16.
5 - edit pair	5,6,7,8. *1
6 - edit with 3 m/cs	14,2 *2
7 - transfer	19,20,21,17,18,1,2,3,4,9,10,11,12,13, 14.
8 - edit with 2 m/cs	1,3,9,11,15,17,13 *3
9 - tech. line up etc	1,2,3,9,10,11,12,13,14,15,16,19,20,21 23
10 - playbacks	1,2,3,4,21,20,19,18,17,16,15,13,12,11, 25

NOTES

*1 - double machines

*2 - for three machine edits the first machine number is given while the consecutive two machines are assumed. Thus the machine used are 14,15 and 16, then 2, 3 and 4.

*3 - similarly for two machine edits the order of machine preference is 1 and 2, 3 and 4, 9 and 10 then 11 and 12 etc.

7. APPROACH TO THE SOLUTION OF THE B.B.C.'S VIDEO-TAPE MACHINE SCHEDULING PROBLEM.

7.1. Initial Consideration

7.1.1. Computing Resources Available

The scheduling sub-system described in Section A has to be altered in minor ways in order to suit the practical situation. Even then it has to be recognised that the system described here will have to undergo further alteration before it can be put into actual use.

A primary reason for this is the type of computers used. It has already been stated that the sub-system was developed in two separate parts; one containing on-line operations using video display units on an ICL 1905E computer and the other containing terminal-initiated batch scheduling programs on a CDC 6400. The BBC, however, has an ICL 1904S computer which supports on-line operations with "DRIVER". This system is unobtainable on any computer which was available for the development of the sub-system.

The system description given here illustrates how the sub-system could be used to solve the B.B.C.'s problem, but does not constitute a complete solution as all the necessary development work must now be transfered to

the B.B.C.'s computer.

7.1.2. Source of the Data used for the Tests

One of the major problems associated with testing the sub-system under real conditions with the B.B.C.'s problem is that no historic input data was available. Old schedules were stored for some period, but these showed the necessary information after it had been altered by negotiations and scheduling. No record was kept of the original inputs.

As the scheduling data arrived over an approximately eight week period, much of it from telephone conversations, it was a tedious and difficult task to collect much original data. The solution adopted to overcome this was to produce some simulated data based on past schedules for the primary, rigorous tests, and use real input data for two further tests as it became available.

It should be clearly stated here that the simulated data used was far more testing and difficult to accommodate than the real data supplied.

7.1.3. The Period Taken for the Scheduling

The scheduling of video-tape machines at the B.B.C. is obviously a continuous process. However,

consideration was initially confined to a single complete day's schedules, taken in isolation.

The initial tests of the sub-system were carried out on a simulated set of requests for a single day. These included requests containing all combinations of the various characteristics which could be met in practice.

Results were obtained from this initial data which illustrated the practical operation of the sub-system. When all operations had been checked and tested, with modifications made where necessary and conclusions drawn, the sub-system was tested with a further two days of real data.

7.1.4. Storage of the Basic Data

Each of the three sets of data considered was taken in complete isolation. It seemed that very little benefit could be derived from further refining the sub-system to allow for continuous scheduling, when its operation was proved satisfactory, and its direct transfer to B.B.C.'s computer was impossible. Moreover storage limitations imposed on the computers used made it practical for only a relatively small amount of data to be stored

at any time.

These two reasons can also be invoked to defend the decision to store all three types of requests (block or regular bookings, normal and low priority requests) in the same file. Normally they would be stored separately, and read as necessary. However, it would have been a pointless exercise to artificially separate them and recombine them for the scheduling run.

7.2. A Description of the Sub-System as Modified for the Practical Application

The sub-system, as modified for the practical problem faced, will obviously be very similar to the description given in Chapter 3. However, the modification necessary for this particular application, and some of the further details necessary, can now be described.

The overall information flow diagram given in Fig. 5 is now modified to take the form shown in Fig. 36. For convenience the programs have been given names appropriate to their function.

This diagram can easily be followed by reference to the description in Chapter 3 and the following notes. Appendix D contains the flow diagrams for all the programs shown in Fig 36.

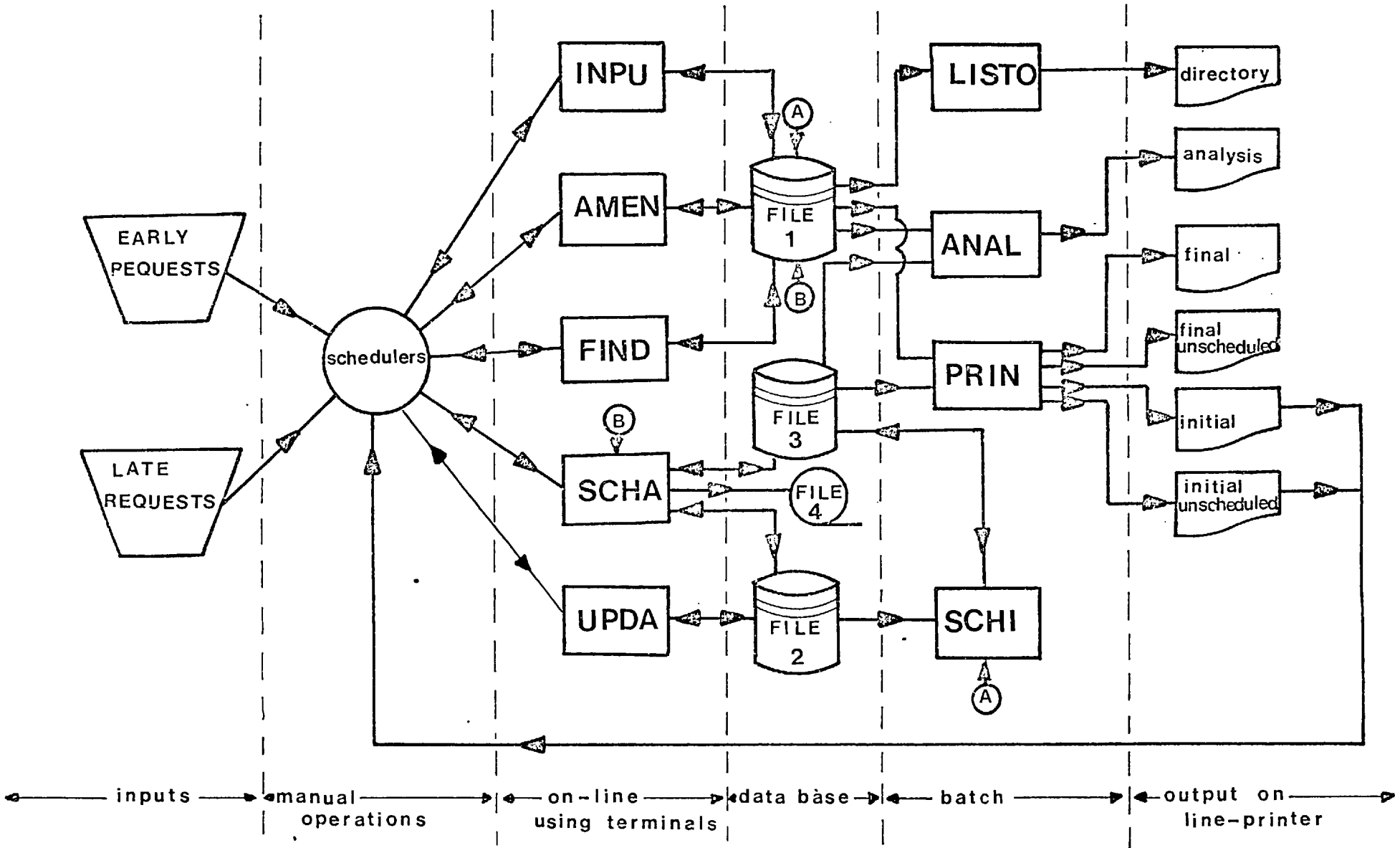


Fig. 36 - the final information flow diagram for the scheduling sub-system

(a) Program 'INPU' (program 1 in section 2) - inputs data concerning requests to the Master File.

(b) Program 'AMEN' (program 2) - amends as necessary information already stored on the Master File

(c) Program 'FIND' (program 3) - finds any request or group of requests stored on the Master File and relays the relevant details to the schedulers

(d) Program 'SCHI' (program 4) - automatically produces the initial schedules using the data supplied

(e) Program 'PRIN' (program 5) - which prints the initial schedules in the form of bar charts, together with other relevant information

(f) Program 'SCHA' (program 6) - manually updates the initial schedules as directed by the schedulers

(g) Program 'UPDA' (program 7) - alters the machine characteristics and constraints as stored in File 2, whenever updating is necessary.

(h) Program 'ANAL' (program 8) - analyses and prints the results of the scheduling runs

(i) Program 'LISTO' (program 9) - lists the requests stored on the Master File as directed

The files also correspond to those described earlier, and thus contain:

File 1. - The Master File - all the information input for

requests. In practice this file was subdivided into three further parts both for convenience and to reduce file handling times. The parts contain:

a) the fixed part of the request data which does not alter during the course of the scheduling run (eg title, project number) excluding comments and some numerical data. This data is not referred to during the scheduling run.

b) the variable part of the data which may be altered during the course of the scheduling run (eg start time, machine number) together with some fixed numerical data (eg function code) which must be referred to during scheduling.

c) comments

File 2 - contains a list of the machine numbers associated with each function in decreasing order of preference.

File 3 - contains a record of the time scheduled at any point of the scheduling. The day is divided into five minute elements, with free time indicated by a blank in the relevant position in this file, and scheduled time indicated by a '1'.

Thus if the first few records of each of these files are listed the results shown in Figs. 37, 38 and 39 are obtained.

	Title	Project Number	Recording Number
1	NATIONWIDE		VTC/6HT/92910
2	HOME INTERNATIONAL TITLES	7344/2 50	VTC/6HS/92914
3	PLAYSCHOOL	3354/3419	VTC/6HT/92931
4	PLAYSCHOOL	3354/3419	VTC/6HT/92932
5	DOCTOR WHO	2344/7037	VTC/6HT/91962/ED
6	TOP OF THE FORM	7043/1209	VTC/6HT/90953/ED
7	OUT OF USE		
8	THE NINE TAILORS	2340/7238	VTC/6HT/89930/ED
9	WHISTLE TEST		
10	PLAY FOR TODAY - THE FAMILY		VTC/6HT/89213/ED/ED
11	SUTHERLAND'S LAW	2243/1001	VTC/6HT/87056/MELO/ED
12	WINNERS AT THE WHEEL	5354/9001	VTC/6HT/89509
13	TRAIL HORIZON		
14	OUT OF SCHOOL	0814/1340	VTC/6HT/91446/ED/ED
15	PRESENTATION PROMOTIONS	9144/9919	
16	O.U.COMP		

Fig. 37 (a) - the 'fixed' part of the data stored in File 1

```
 1   5 930 3303015 5400 0 0 0
 2   5 0 0 53030 0 0100 0 0 0
 3   11445 04530 0 0600 0 0 0
 4   42015 04530 0 0600 0 0 0
 5   511 0 3 030 011402 0 0 0
 6   21855 025 5 0 0600 0 0 0
 7   122 0 2 0 0 019900 0 0 0
 8   22135 05530 0 0600 0 0 0
 9  10 0 0 1 030 0 0100 0 0 0
10   4 0 0 245452019500 0 0 0
11  10 9 0 1 030 0 0300 0 0 0
12  101345 03030 018400 0 0 0
13   21940 0 520 025800 0 0 0
14  1011 0 11530 0 1800 0 0 0
15  10 930 1 0 0 0 9800 0 0 0
16   92230 03030 522300 0 0 0
```

Record Number
Function
Start Time
Duration
Line-up Time
Durat. Flexibility
1st M/C Number
Fix-Code
Indicator
No. of Machines
2nd & 3rd M/C Nos.
Marker

Fig. 37 (b) - the variable part of the data stored in File 1

Comments

```
 1 RX ELEC EDIT AS REQD
 2 MARTIN. 2M/CS + DUB
 3 CP/FELGATE
 4 CP/FELGATE. 1M/C ONLY. P/B TO GALLERY
 5 FURTHER ELEC ED. TX 11.5.74.  2M/CS
 6 OE/WRIGHT 10/15
 7
 8 DL/BEYNON
 9 P/B ON SITE
10 TO BE TRANSMITTED ON 10.08.74
11 P/B TO J103. PRODUCER BRINGING TAPES TO V.T.
12 P/B ON SITE. X7882
13 T/NC2
14 JONES. PLAYBAC  TO 614 VILLIERS HOUSE
15 PLAYBACK ON SITE
16 TECH L/U
```

Fig. 37 (c) - showing the comments stored in File 1 for the first sixteen requests

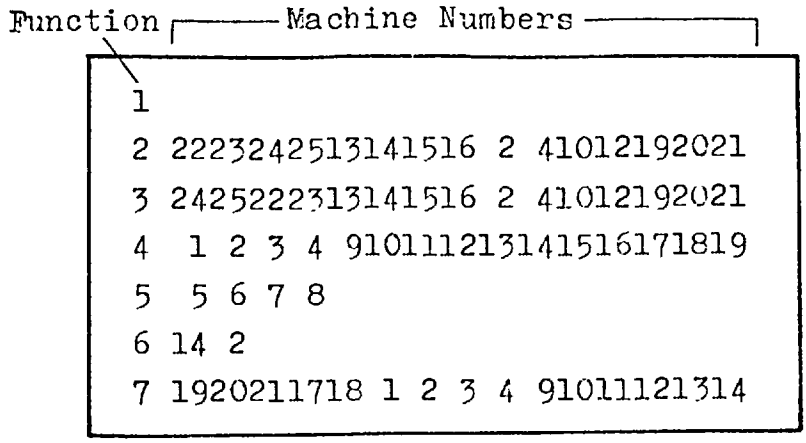


Fig. 38 - showing the first seven records stored in File 2 - the machine numbers associated with functions

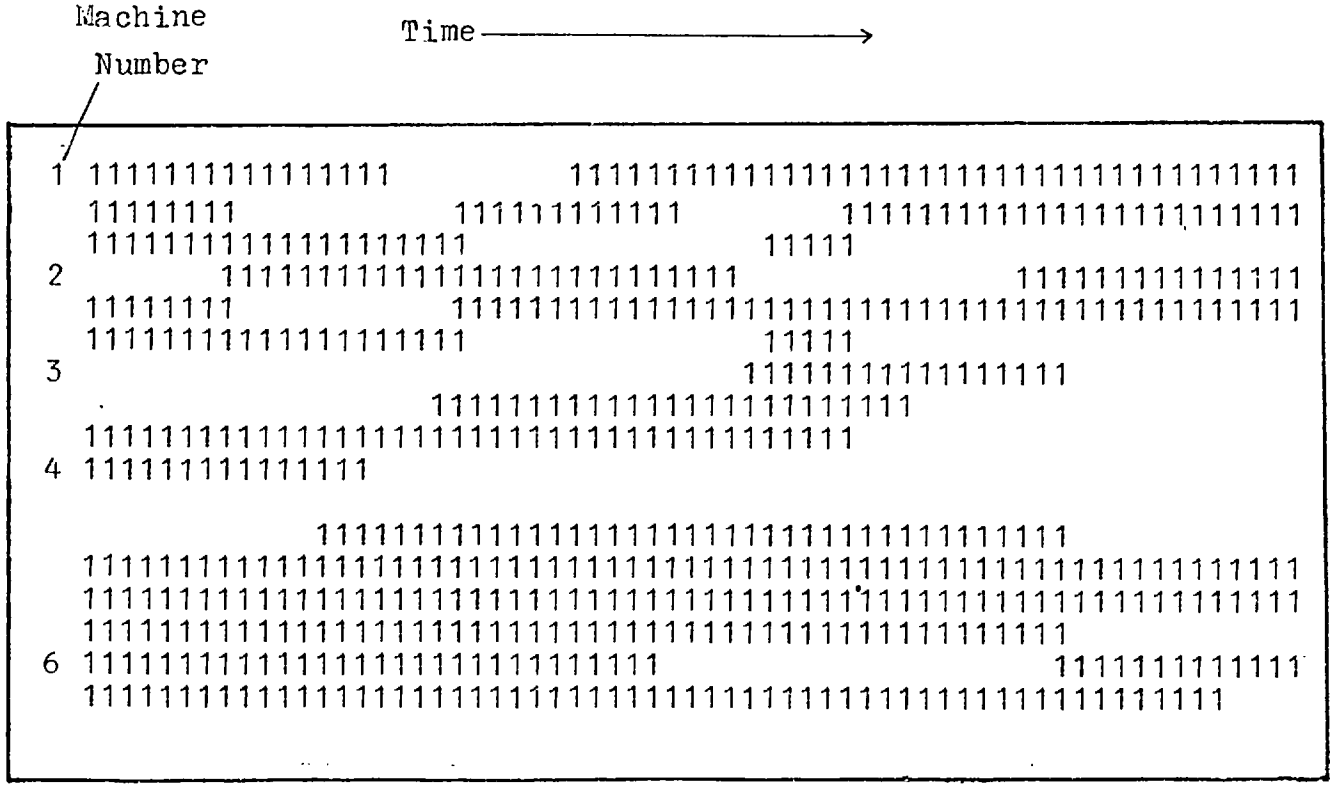


Fig. 39 - showing the first six records of File 3, indicating which times are filled in the schedules

8. A DESCRIPTION OF THE INDIVIDUAL PROGRAMS EVOLVED

The basic principles of the programs developed for the sub-system were described in Chapters 3 and 4. This Chapter is an extension of these initial comments, and the programs described are based on the ideas illustrated earlier.

Examples of the operational methods of the programs developed for the particular application are described in the following paragraphs, while the broad principles have been covered in the preceeding chapters.

For each program there is a brief indication of its operation and some results obtained from test runs. A complete explanation of each program in the information flow diagram is given in Appendix D.

The on-line and terminal-initiated batch operations are considered separately.

8.1 On-Line Operations

8.1.1 Program 'INPU'.

This program reads the data concerning requests for machine time entered on a VDU screen, checks the data for errors and validity, and writes it in the correct position on the data files.

The error checking involves determining that numeric data is in the correct position and in the right format, that all essential information is included, and that the information given is consistent (for example that machine numbers and/or start times are given with certain fix codes).

One immediate problem which was presented by the use of the multi-access system 'MAXIMOP' was the limitation on the video-display units which could be used. The system could only support ICL mk 1 V.D.U.s. This meant that the screen size was limited to 20 lines of 54 characters. All input and output formats had to be designed within thisconstraint, with an attempt made to maintain clarity of presentations, especially where fairly large amounts of information were involved.

The detailed facilities offered by this program can be derived from the flow diagram illustrated in Appendix D(1). Examples of the general type of operation are given in Fig. 41.

```
_SRUN
     HELLO  - I AM READY WHEN YOU ARE
 THE NEXT REQUEST TO BE ENTERED IS NUMBER  44
                                      *  TITLE
  .  .  *                           C   DATE
           *                              PROJECT NUMBER
                              *           RECORDING NUMBER
  *                               C   FUNCTION
     .  .  *                      C   DURATIONS
 .     .  . .    *                C   FIX,START,M/C ETC
  /  *                                   SOURCE/DESTINATION_COPY
 OK - NUMBER  44 IS REPEATED AS NUMBER  43
                                      *  TITLE
  .  .  *                           C   DATE
           *                              PROJECT NUMBER
                              *           RECORDING NUMBER
  *                               C   FUNCTION
     .  .  *                      C   DURATIONS
 .     .  . .    *                C   FIX,START,M/C ETC
  .  *                                 SOURCE/DESTINATION_ NOMO
   OK    -   GOODBYE
    YOU HAVE ENTERED  1 REQUESTS.
```

Fig. 41 (a) - illustrating the 'COPY' facility

```
10.23.19_ SRUN
     HELLO  - I AM READY WHEN YOU ARE
 THE NEXT REQUEST TO BE ENTERED IS NUMBER  45
                                      *  TITLE
  .  .  *                           C   DATE
           *                              PROJECT NUMBER
                              *           RECORDING NUMBER
  *                               C   FUNCTION
     .  .  *                      C   DURATIONS
 .     .  . .    *                C   FIX,START,M/C ETC
  /  *                                   SOURCE/DESTINATION
_REPE
PARENTS AND CHILDREN
01.01.75
3053/4883
VTC/6HT/91842/ED
  *                               C   FUNCTION
     .  .  *                      C   DURATIONS
 .     .  . .    *                C   FIX,START,M/C ETC
  /  *                                   SOURCE/DESTINATION
```

Fig. 41 (b) - illustrating the 'REPEat' facility

Fig. 41 - showing some of the operations of program 'INPU' with the 'form' layouts.

8.1.2 Program 'AMEN'

It may be necessary to alter the information concerning requests for machine use after it has been stored in the data files. This can be done by using program 'AMEN'. All information about the request to be altered is displayed on a V.D.U. screen, and changes are made by overwriting as desired.

Additional facilities may be necessary to remove a request from the files to allow for cancellations. Program 'AMEN' marks the relevant request so that it is ignored from further consideration. However, the option of replacing a request in the original form has been introduced.

An example of the working of this program is given in Fig. 42, while a detailed flow diagram is illustrated in Appendix D(2).

```
10.32.04_ SRUN
PLEASE ENTER THE FUNCTION WANTED (ALTE,DELE,REPL)_DELE 36
REQUEST NUMBER  36 HAS BEEN DELETED
DO YOU WANT TO DELETE ANY MORE REQUESTS?_NO
 FINISHED?_NO
PLEASE ENTER THE FUNCTION WANTED (ALTE,DELE,REPL)_REPL 36
 REQUEST NUMBER  36 HAS BEEN RESTORED AS FOLLOWS:-
TITLE                MUSIC TIME
DATE                 01.01.75
PROJECT NUMBER
RECORDING NUMBER
FUNCTION              2
START TIME           1122
DURATION              020
LINE-UP TIME         30
TIME FLEXIBILITY      0
MACHINE NUMBER        0   0   0
FIX-CODE             6
NUMBER OF MA/CS       0
CHEDULED STATE     0
SOURCE/DESTINATION    /

DO YOU WANT TO RESTORE ANY MORE REQUESTS?_YES
PLEASE ENTER THE REQUEST NUMBER._21
REQUEST NUMBER  21 HAS NOT BEEN DELETED     -    TRY AGAIN
PLEASE ENTER THE FUNCTION WANTED (ALTE,DELE,REPL)_ALTE 49
THERE ARE ONLY  45 REQUESTS IN THE FILE
PLEASE ENTER THE FUNCTION WANTED (ALTE,DELE,REPL)_ALTE 45
REQUEST NUMBER  45IS AS FOLLOWS:- OVERWRITE AS NECESSARY
TITLE
DATE                 01.01.75
PROJECT NUMBER
RECORDING NUMBER
FUNCTION              0
START TIME            0 0
DURATION              0 0
LINE-UP TIME          0
TIME FLEXIBILITY      0
MACHINE NUMBER        0   0   0
FIX-CODE             0
NUMBER OF MA/CS       0
CHEDULED STATE     0
SOURCE/DESTINATION    /
```

Fig. 42 - illustrating some of the operations possible with program 'AMEN'. Information may be altered by overwriting.

8.1.3 Program 'FIND'.

The purpose of this program is to search the data files in order to find all those requests which have certain named characteristics in common. For example, all requests with durations of half an hour may be listed if a gap of this duration is to be filled in the schedules.

The facility for multiple searches has also been included. Thus it would be possible to list all those requests with, for example, durations of half an hour, function 2, fix code 3 and project number 1644/2820. In addition a certain flexibility has been introduced to certain of the variables so that requests with durations of half an hour plus or minus fifteen minutes, say, may be listed.

An example of the working of this program is given in Fig. 43, while a detailed flow diagram is illustrated in Appendix D(3).

```
16.18.42_ SRUN
        HELLO   -   READY TO START
PLEASE ENTER THE NAME AND VALUE WANTED_RECO VTC/6HT/92932
_NOMO
  OK  -  THERE WERE  1 SEARCHES GIVING THE FOLLOWING RESULTS
   4  PLAYSCHOOL                              3354/3419
VTC/6HT/92932                        /
 4  20.15   0.45  30   0   0  6  0  0   0   0
CP/FELGATE. 1M/C ONLY. P/B TO GALLERY

THE TOTAL NUMBER OF REQUESTS FOUND IS  1
DO YOU WANT TO FIND ANY MORE?_YES
PLEASE ENTER THE NAME AND VALUE WANTED_NUMB 35
  35  OPEN UNIVERSITY                         0525/7592
VTC/6HT/AM.74093                     /
 9   0. 0   1.15  15   0   0  1  0  0   0   0
REV

DO YOU WANT TO FIND ANY MORE?_YES
PLEASE ENTER THE NAME AND VALUE WANTED_DURA 0100
WITH WHAT FLEXIBILITY?_15
_LINE 15
_CODE 01
_FUNC 09
_NOMO
  OK  -  THERE WERE  4 SEARCHES GIVING THE FOLLOWING RESULTS
  35  OPEN UNIVERSITY                         0525/7592
VTC/6HT/AM.74093                     /
 9   0. 0   1.15  15   0   0  1  0  0   0   0
REV

THE TOTAL NUMBER OF REQUESTS FOUND IS  1
DO YOU WANT TO FIND ANY MORE?_YES
PLEASE ENTER THE NAME AND VALUE WANTED_FUNC 09
_DURA 0100
WITH WHAT FLEXIBILITY?_00
_STAR 2230
WITH WHAT FLEXIBILITY?_00
_PROJ 1644/2819
_NOMO
  OK  -  THERE WERE  4 SEARCHES GIVING THE FOLLOWING RESULTS
  26  CANNON - 'THE DEAD SAMARITON'     1644/2819
VTC/6HT/92949                        /
 9  22.30   1. 0  30   0   0  1  0  0   0   0
DUBBED YESTERDAY

THE TOTAL NUMBER OF REQUESTS FOUND IS  1
DO YOU WANT TO FIND ANY MORE?_YES
PLEASE ENTER THE NAME AND VALUE WANTED_
BLANK INPUT VALUE  -  TRY AGAIN_TITL
BLANK INPUT VALUE  -  TRY AGAIN_TITL OPEN UNIVERSITY
_NOMO
  OK  -  THERE WERE  1 SEARCHES GIVING THE FOLLOWING RESULTS
  35  OPEN UNIVERSITY                         0525/7592
VTC/6HT/AM.74093                     /
 9   0. 0   1.15  15   0   0  1  0  0   0   0
REV

THE TOTAL NUMBER OF REQUESTS FOUND IS  1
DO YOU WANT TO FIND ANY MORE?_NO
      OK   -   GOODBYE
```

Fig. 43 - illustrating some of the operations of program 'FIND'

8.1.4 Program 'STAT'.

This program is similar to program 'FIND' in that information about requests is relayed to the schedulers. In program 'STAT' the information given concerns the number and durations of requests with each function or fix code. The proportions of these scheduled are also given.

Further information can also be gained about the machine utilisations with this program.

An example of the operation of this program is given in Fig. 44, while a detailed flow diagram is illustrated in Appendix D(4).

```
12.41.52_ SRUN
    HELLO   -   READY TO START WHEN YOU ARE
PLEASE ENTER DATA WANTED_FUNC

 FUNCTION    NUMBER    TIME    NUMBER    TIME
  /CODE      RQSTD     RQSTD   SCHED     SCHED
 ********    ******    *****   ******    *****

     1          6      21.15      6      21.15
     2         13       9.35     13       9.35
     3          3       2.30      3       2.30
     4          2       3. 0      2       3. 0
     5          5      22.30      4      20.15
     6          0       0. 0      0       0. 0
     7          1       1. 0      1       1. 0
     8          0       0. 0      0       0. 0
     9          4       6. 0      3       5.15
    10          8      10.45      7      10.10


TOTAL NO. OF REQUESTS =  44    -  41 SCHED
TOTAL TIME = 76 HRS 35 MINS  -  73 HRS  0 MINS SCHED

PLEASE ENTER DATA WANTED_TOTA

NO. OF REQUESTS = 44 - 41 SCHEDULED (WITH MULTIPLES)
TIME REQUESTED  = 76.35 - 73.00 SCHEDULED (WITH MULTIPLES)

PLEASE ENTER DATA WANTED_UTIL

 MACHINE     NUMBER      TIME      PERCENT UTIL
 *******     ******      ****      ************

     1          3       12.30         83.33
     2          1        6. 0         40.00
     3          0        0. 0          0.00
     4          0        0. 0          0.00
     5          0        0. 0          0.00
     6          4        6.15         41.67
     7          0        0. 0          0.00
     8          0        0. 0          0.00
     9          0        0. 0          0.00
    10          0        0. 0          0.00
    11          6        8.30         56.67
    12          3        6.15         41.67
    13          2        5.40         37.78
    14          0        0. 0          0,00
    15          4        7.10         47.78
    16          2        4.20         28.89
    17          0        0. 0          0.00
    18          4        5. 0         33.33
    19          0        0. 0          0.00
    20          0        0. 0          0.00
    21          0        0. 0          0.00
    22          5        8.30         56.67
    23          1        1. 0          6.67
    24          4        6.15         41.67
    25          2        2.30         16.67

TOTAL NUMBER = 41 (93.2 PER CENT OF REQUESTS)
TOTAL TIME   = 73.00 (INCLUDING LINE-UP)
```

Fig. 44 (a) - illustrating some of the operations of Program 'STAT'

```
PLEASE ENTER DATA WANTED_CODE

FUNCTION    NUMBER     TIME      NUMBER     TIME
 /CODE      RQSTD      RQSTD     SCHED      SCHED
********   ******     *****     ******     *****

    1          7       14.30       6       12.15
    2          2        4.15       2        4.15
    3          2        2.30       2        2.30
    4          3       13.15       3       13.15
    5          0        0. 0       0        0. 0
    6         17       14.10      15       12.50
    7          0        0. 0       0        0. 0
    8          6        7.55       6        7.55
    9          7       20. 0       7       20. 0
   10          0        0. 0       0        0. 0

TOTAL NO. OF REQUESTS =  44    -   41 SCHED
TOTAL TIME = 76 HRS 35 MINS  -   73 HRS  0 MINS SCHED

PLEASE ENTER DATA WANTED_NOMO

     - OK -      GOODBYE
```

Fig. 44 (b) - illustrating further possible operations with program 'STAT'

8.1.5 Program 'UPDA'.

Almost invariably the facilities offered by various machines will alter with time, while new ones are brought into operation and old ones are withdrawn from service. This program alters, as necessary, the orders of preference of machines for each function, as they are stored in the data files.

There are two methods of operation involved. In the first commands are given to initiate changes. For example, machine number 10 can be deleted from the lists (which may be necessary when it is withdrawn from service) or machine number 20 can be replaced by machine number 10 (when a new machine is introduced to replace the old machine 20).

The second method of operation involves the overwriting of information displayed on a V.D.U. screen. Thus complete lists of machine preferences for each function are displayed and alterations made by overwriting as required.

An example of the operations of this program is given in Fig. 45, while a detailed flow diagram is illustrated in Appendix D(5).

```
09.58.42_ SRUN
   HELLO   -   READY TO START
PLEASE INPUT FUNCTION WANTED (REPL,DELE,SUMM)_DELE
PLEASE ENTER MACHINE NUMBER_12
MACHINE NUMBER 12 HAS BEEN DELETED
THE MACHINE ORDER IS NOW:-
FUNCTION  1 (MAINTNCE) -     1  2  3  4  5
           6  7  8  9 10 11 13 14 15  0
FUNCTION  2 (TRANSM-1) -    25 24 23 22 21
          20 19 18 17 16 15 14 13 11  0
FUNCTION  3 (TRANSM-2) -     1  2  3  4  5
           6  7  8  9 10 16 17 18 19 20
FUNCTION  4 (RECORD  ) -    21 22 23 24 25
           1  2  3  4  5  6  7  8  9 10
FUNCTION  5 (EDIT-1PR) -    11 13 14 15 16
          17 18 19 20 21 22 23 24 25  0
FUNCTION  6 (EDIT-3MC) -    13 14 15 16 17
          18 19 20  1  2  3  4  5  6  0
FUNCTION  7 (TRANSFER) -     1  2  3  4  5
           0  0  0  0  0  0  0  0  0  0
FUNCTION  8 (EDIT-2MC) -    25 24 23 22 21
          20  0  0  0  0  0  0  0  0  0
FUNCTION  9 (TECH L-U) -     1  2  3  4  5
           6  7  8  9 10 11 13 14 15  0
FUNCTION 10 (PLAYBACK) -    25 24 23 22 21
          20 19 18 17 16 15 14 13 11  0
DO YOU WANT TO CHANGE ANY MORE? YES
PLEASE INPUT FUNCTION WANTED (REPL,DELE,SUMM)_REPL 10 30
MACHINE NUMBER 10 HAS BEEN REPLACED BY NUMBER 30
THE MACHINE ORDER IS NOW:-
FUNCTION  1 (MAINTNCE) -     1  2  3  4  5
           6  7  8  9 30 11 13 14 15  0
FUNCTION  2 (TRANSM-1) -    25 24 23 22 21
          20 19 18 17 16 15 14 13 11  0
FUNCTION  3 (TRANSM-2) -     1  2  3  4  5
           6  7  8  9 30 16 17 18 19 20
                .                 .
                .                 .
                .                 .
FUNCTION  9 (TECH L-U) -     1  2  3  4  5
           6  7  8  9 30 11 13 14 15  0
FUNCTION 10 (PLAYBACK) -    25 24 23 22 21
          20 19 18 17 16 15 14 13 11  0
DO YOU WANT TO CHANGE ANY MORE?_YES
PLEASE INPUT FUNCTION WANTED (REPL,DELE,SUMM)_SUMM
THE MACHINE ORDER IS NOW:-
FUNCTION  1 (MAINTNCE) -     1  2  3  4  5
           6  7  8  9 30 11 13 14 15  0
FUNCTION  2 (TRANSM-1) -    25 24 23 22 21
          20 19 18 17 16 15 14 13 11  0
FUNCTION  3 (TRANSM-2) -     1  2  3  4  5
           6  7  8  9 30 16 17 18 19 20
                .                 .
                .                 .
                .                 .
FUNCTION  9 (TECH L-U) -     1  2  3  4  5
           6  7  8  9 30 11 13 14 15  0
FUNCTION 10 (PLAYBACK) -    25 24 23 22 21
          20 19 18 17 16 15 14 13 11  0
THIS MAY BE ALTERED BY OVERWRITING
```

Fig. 45 - illustrating some of the operations of program 'UPDA'

8.1.6 Program 'SCHA'

After the initial shedules have been produced automatically a facility is needed which enables the schedulers to alter them as they feel necessary. This is the program which provides this facility.

The schedulers direct the program to display, on a V.D.U. screen, the schedules of the machines under consideration. Obvious space limitations prevent all the schedules appearing on the screen at the same time.

Requests are added to and subtracted from the schedules as directed by the schedulers, with the visual display clearly indicating the effect of the various changes. By referring to a general index of requests, or by using program 'FIND', the schedulers can determine which requests are best suited to any gaps they may introduce into the initial schedules.

A system of overwriting the schedules displayed on the V.D.U. was considered, but in practice this option proves too cumbersome to be of any real value.

An example of the procedure involved with this program is given in Fig. 46, while a detailed flow diagram is illustrated in Appendix D(6).

```
12.26.12_ SRUN
   HELLO   -   READY TO START WHEN YOU ARE
WHICH MACHINES DO YOU WANT DISPLAYED?_41 42 43 44
M/C 0900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 2100
    .    .    .    .    .    .    .    .    .    .    .    .    .
  41  0012T     I0014RI   0013P     II    II    II    X       I-I
  42       X    X    II    0024R     II  II   0020RI          0028
  43  I--------00290--------I        I--0030R--II-I  II      0033T
  44  0034R     II    X         II   I--0039R--I040DII    II
READY  -  PLEASE CONTINUE
_SUB 0014
  41  0012T                 0013P     II    II    II    X       I-I
READY  -  PLEASE CONTINUE_ADD 0014
  41  0012T     I0014RI   0013P     II    II    II    X       I-I
READY  -  PLEASE CONTINUE_NOMC
  -  OK  -  GOODBYE
```

```
14.23.11_ SRUN
   HELLO   -   READY TO START WHEN YOU ARE
WHICH MACHINES DO YOU WANT DISPLAYED? 10 11 12 13 14
M/C 0900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 2100
    .    .    .    .    .    .    .    .    .    .    .    .    .
  10 I-------0055R-------I    I-------0061T-------I
  11 I-------0056R--------I--0051D--II-I      I--0069D--I    I-I
  12 I-------0057R--------I--------0070D-------I-I       I-I  0071T
  13          0052D     0053D
  14 I-----------------0066R----------------I-------0073D-------I
READY  -  PLEASE CONTINUE
_SUB 0070
  12 I-------0057R-------I                  I-I       I-I  0071T
READY  -  PLEASE CONTINUE
 SUMM
INCORRECT INSTRUCTION - TRY AGAIN
 SHOW
WHICH MACHINES DO YOU WANT DISPLAYED? 17
M/C 0900 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 2100
    .    .    .    .    .    .    .    .    .    .    .    .    .
  17 0097D     I----0006R-----I                  0098D     I-I II
READY  -  PLEASE CONTINUE
 ADD 0006
THIS BOOKING HAS ALREADY BEEN SCHEDULED
READY  -  PLEASE CONTINUE
 ADD 0070
WHICH MACHINE DO YOU WANT TO PUT THIS ON?_17
WHAT TIME IS THE BOOKING TO START?_1300
  17 0097D     I----0006R-----I--------0070D-------I0098D    I-I II
READY  -  PLEASE CONTINUE
_NOMO
  -  OK  -  GOODBYE
```

Fig. 46 - illustrating some of the operations of program 'SCHA'

In some instances, for example when there are a large number of short jobs scheduled on a machine, the bar chart format illustrated in Fig. 46 may not provide very clear displays. In these circumstances lists of the jobs scheduled on each machine may be preferable to bar charts. Two alternative forms of listing are illustrated in Fig. 47.

MACHINE	21	22	23	25
JOB	145	65	109	3
FROM	0900	0900	0930	1000
TO	0930	1100	0945	1400
JOB	47	110	8	34
FROM	0930	1100	1000	1600
TO	1245	1130	1330	2400

Fig 47a - illustrating one form of a 'listed' format for displaying schedules

```
MACHINE NUMBER 1
NUMBER 101    START 1030    FINISH 1230
NUMBER  47    START 1230    FINISH 1400
NUMBER 115    START 1430    FINISH 2000
NUMBER   3    START 2000    FINISH 2400
MACHINE NUMBER 2
NUMBER 123    START 0900    FINISH 1000
NUMBER  74    START 1230    FINISH 2330
```

Fig 47b - illustrating an alternative form of 'listed' format

8.2 Terminal - Initiated Batch Programs

There are four programs to be considered in this section, of which three are fairly straightforward (programs 'ANAL', 'PRIN' and 'LISTO'), while the last (program 'SCHI') will need more explanation. Hence the first three are considered in this section while the automatic scheduling program is discussed in more detail in Chapter 9.

As the output of these programs is sent to a line-printer there is not so severe a constraint on either the volume of output of its format. One area of difficulty arose, however, in reaching agreement on the format adopted for the initial schedules. The results selected are illustrated in Chapter 9.

Again the operations of the programs are described briefly with examples of the print-outs obtained. Details of the programs can be obtained from the flow diagrams described in Appendix D.

8.2.1 Program 'ANAL'.

When the scheduling program has been run this program analyses the results obtained and relays associated information to the schedulers. Particularly of interest are the numbers and times requested and scheduled for each function and fix code. In addition the utilisation of each machine is of interest.

The data for scheduled jobs can be compared with either the original input, or with the slightly modified data present after the scheduling run (for example a request for a machine pair may be altered to two separate requests, each on a separate machine).

An example of the output from this program is shown in Fig. 48, while a flow diagram for its operation is given in Appendix D(7).

ANALYSIS OF RESULTS

	REQUESTS NUMBER	REQUESTS TIME	REQUESTS SCHEDULED NUMBER	REQUESTS SCHEDULED TIME	PERCENT SCHEDULED NUMBER	PERCENT SCHEDULED TIME
FUNCTION 1	8	37. 2	8	37. 2	100.00	100.00
FUNCTION 2	24	18.47	24	18.47	100.00	100.00
FUNCTION 3	10	8.15	10	8.15	100.00	100.00
FUNCTION 4	23	62.45	23	62.45	100.00	100.00
FUNCTION 5	15	51.15	13	51.45	86.67	84.49
FUNCTION 6	9	82.15	3	27. 2	33.33	43.37
FUNCTION 7	1	1. 2	1	1. 2	100.00	100.00
FUNCTION 8	0	0. 0	4	29. 2	0.	0.
FUNCTION 9	9	16. 2	9	16. 2	100.00	100.00
FUNCTION 10	13	21. 2	13	21. 2	100.00	100.00

MACHINE	REQUESTS ON IT	TIME SCHEDULED	PERCENT UTILISATION
1	7	13.15	88.33
2	6	12.45	85.00
3	2	14.30	96.67
4	2	14.30	96.67
5	4	13.45	91.67
6	3	12. 0	80.00

Fig. 48 - sample print-out from program 'anal' (an expanded version of the print-out from program 'stat') ($\frac{5}{8}$ actual size)

8.2.2 Program 'PRIN'.

The schedules produced by program 'SCHI' are stored in the computer in the form of numerical lists associating jobs with machines. The schedulers, and later the machine users and operators, require the schedules to be presented as bar charts. Program 'PRIN' analyses the information stored in the data files concerning jobs scheduled and prints the results in the required format.

The facility of producing a variety of formats may be advantageous in certain circumstances. For example the schedulers may require the schedules in the form of bar charts, while machine operators might prefer chronological listings of jobs on machines. In these cases this program may be easily modified to allow such flexibility. In the solution described, however, a single format was presented which gave the most satisfactory result. An example of this print-out is shown in Fig. 49.

A detailed flow diagram for the program is illustrated in Appendix D(8).

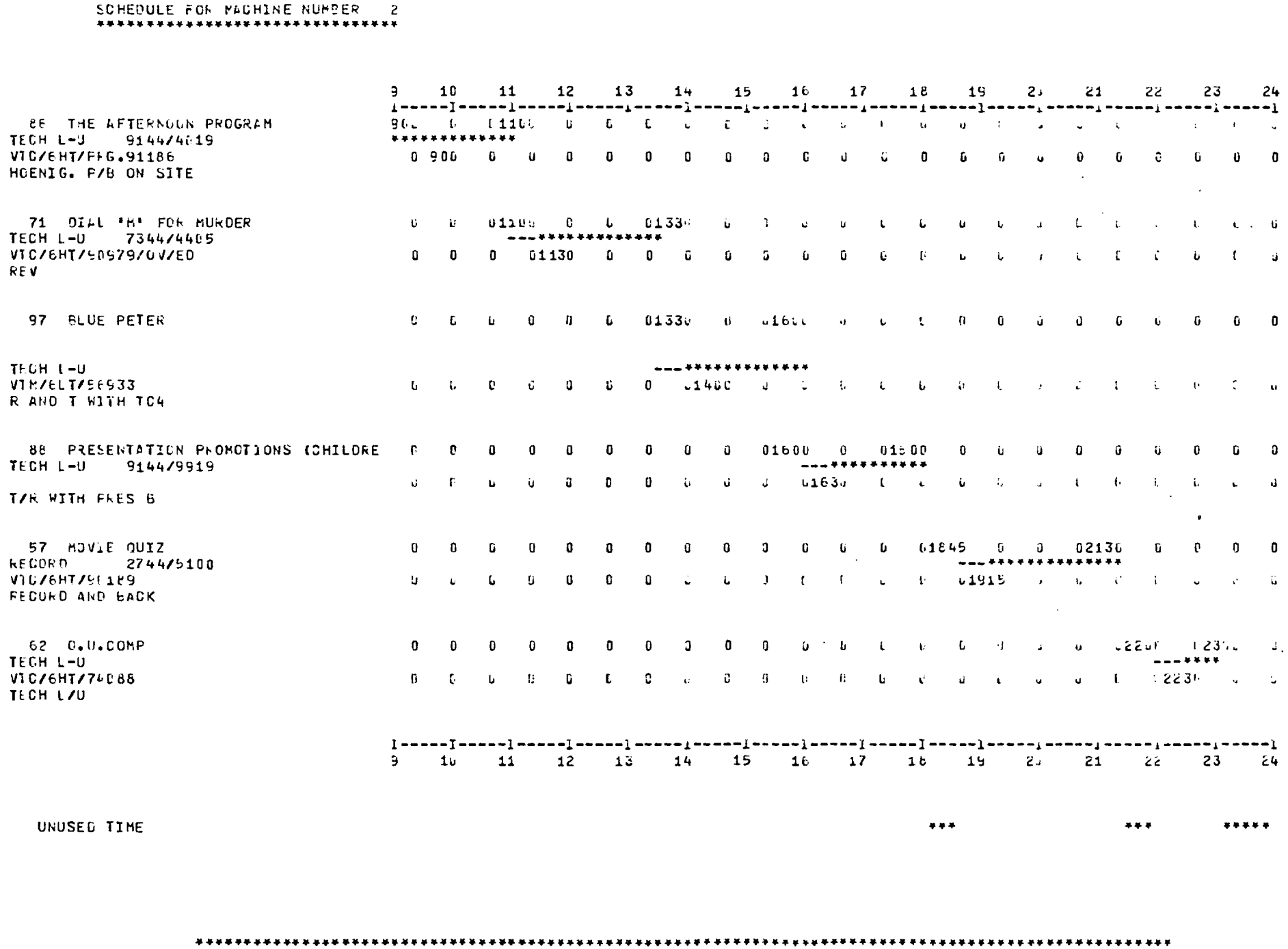
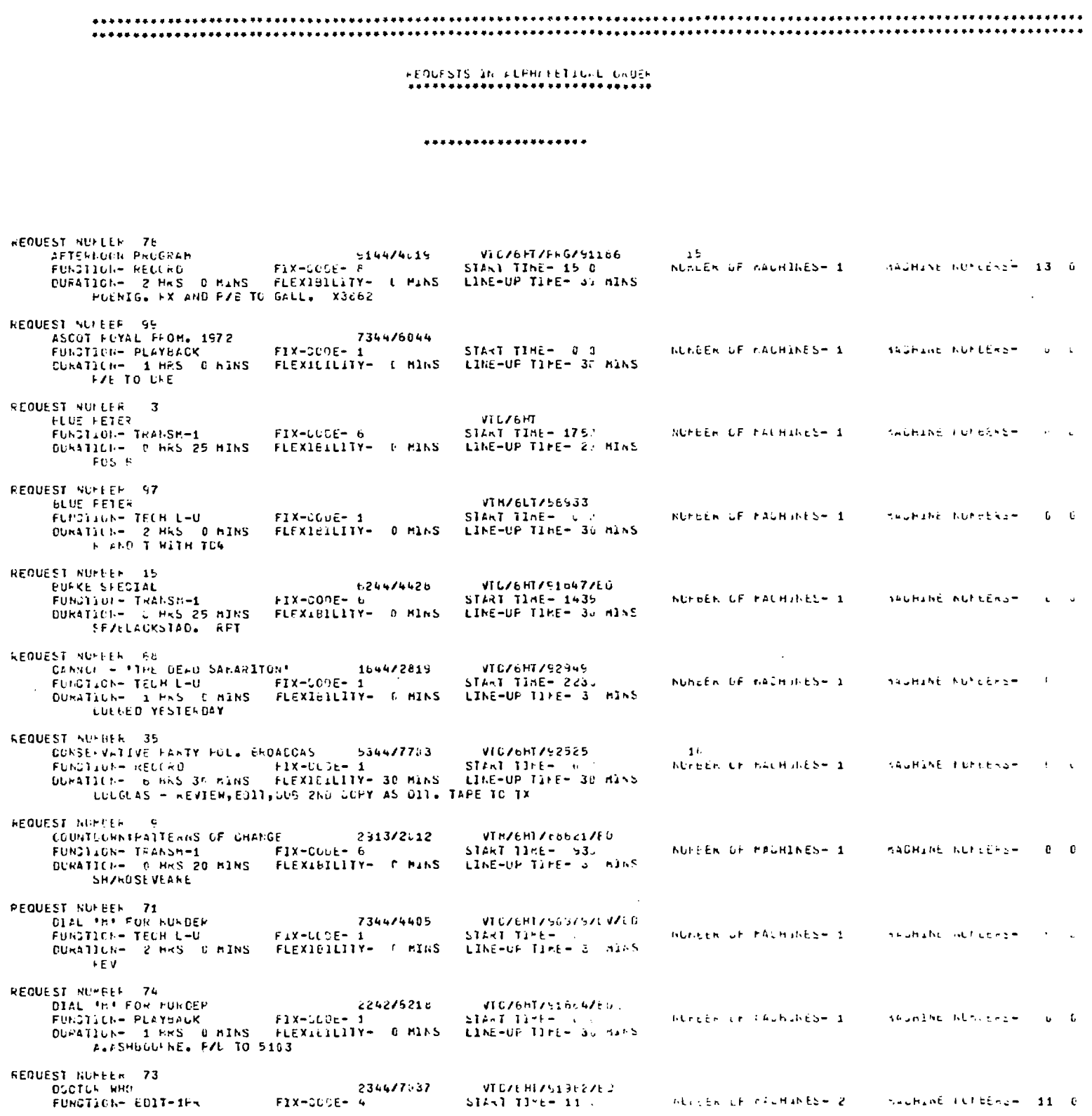


Fig. 49 - sample print-out from program 'prin' (reduced to $\frac{5}{8}$ actual size)



REQUESTS IN ALPHABETICAL ORDER

REQUEST NUMBER 76
AFTERNOON PROGRAM 5144/4019 VTC/6MT/PRG/91166 15
FUNCTION- RECORD FIX-CODE- F START TIME- 15 0 NUMBER OF MACHINES- 1 MACHINE NUMBERS- 13 0 0
DURATION- 2 HRS 0 MINS FLEXIBILITY- 0 MINS LINE-UP TIME- 30 MINS
PUENIG. FX AND F/E TO GALL. X3662

REQUEST NUMBER 95
ASCOT ROYAL FROM. 1972 7344/6044
FUNCTION- PLAYBACK FIX-CODE- 1 START TIME- 0 0 NUMBER OF MACHINES- 1 MACHINE NUMBERS- 0 0 0
DURATION- 1 HRS 0 MINS FLEXIBILITY- 0 MINS LINE-UP TIME- 30 MINS
F/E TO DAE

REQUEST NUMBER 3
BLUE PETER VTC/6MT
FUNCTION- TRANSM-1 FIX-CODE- 6 START TIME- 1750 NUMBER OF MACHINES- 1 MACHINE NUMBERS- 0 0 0
DURATION- 0 HRS 25 MINS FLEXIBILITY- 0 MINS LINE-UP TIME- 2[illegible] MINS
POS 8

REQUEST NUMBER 97
BLUE PETER VTM/6LT/56933
FUNCTION- TECH L-U FIX-CODE- 1 START TIME- [illegible] NUMBER OF MACHINES- 1 MACHINE NUMBERS- 0 0 0
DURATION- 2 HRS 0 MINS FLEXIBILITY- 0 MINS LINE-UP TIME- 30 MINS
F AND T WITH TD4

REQUEST NUMBER 15
BURKE SPECIAL 6244/4426 VTC/6MT/91047/E0
FUNCTION- TRANSM-1 FIX-CODE- 6 START TIME- 1435 NUMBER OF MACHINES- 1 MACHINE NUMBERS- 0 0 0
DURATION- 0 HRS 25 MINS FLEXIBILITY- 0 MINS LINE-UP TIME- 30 MINS
SF/BLACKSTAD. RFT

REQUEST NUMBER 88
CANNON - 'THE DEAD SAMARITAN' 1644/2819 VTC/6MT/92949
FUNCTION- TECH L-U FIX-CODE- 1 START TIME- 2230 NUMBER OF MACHINES- 1 MACHINE NUMBERS- 1 0
DURATION- 1 HRS 0 MINS FLEXIBILITY- 0 MINS LINE-UP TIME- 3 MINS
LOGGED YESTERDAY

REQUEST NUMBER 35
CONSERVATIVE PARTY POL. BROADCAS 5344/7703 VTC/6MT/92525 10
FUNCTION- RECORD FIX-CODE- 1 START TIME- [illegible] NUMBER OF MACHINES- 1 MACHINE NUMBERS- 1 0 0
DURATION- 6 HRS 30 MINS FLEXIBILITY- 30 MINS LINE-UP TIME- 30 MINS
LOLOLAS - REVIEW,EDIT,DOS 2ND COPY AS DIT. TAPE TO TX

REQUEST NUMBER 9
COUNTDOWN:PATTERNS OF CHANGE 2313/2612 VTM/6MT/c86c1/E0
FUNCTION- TRANSM-1 FIX-CODE- 6 START TIME- 930 NUMBER OF MACHINES- 1 MACHINE NUMBERS- 0 0 0
DURATION- 0 HRS 20 MINS FLEXIBILITY- 0 MINS LINE-UP TIME- 0 MINS
SH/ROSEVEARE

REQUEST NUMBER 71
DIAL 'M' FOR MURDER 7344/4405 VTC/6MT/9[illegible]/E0
FUNCTION- TECH L-U FIX-CODE- 1 START TIME- [illegible] NUMBER OF MACHINES- 1 MACHINE NUMBERS- 1 0 0
DURATION- 2 HRS 0 MINS FLEXIBILITY- 0 MINS LINE-UP TIME- 3 MINS
REV

REQUEST NUMBER 74
DIAL 'M' FOR MURDER 2242/5218 VTC/6MT/91[illegible]/E0
FUNCTION- PLAYBACK FIX-CODE- 1 START TIME- [illegible] NUMBER OF MACHINES- 1 MACHINE NUMBERS- 0 0 0
DURATION- 1 HRS 0 MINS FLEXIBILITY- 0 MINS LINE-UP TIME- 30 MINS
A.ASHBOURNE. F/E TO 5103

REQUEST NUMBER 73
DOCTOR WHO 2344/7037 VTC/6MT/9[illegible]/E0
FUNCTION- EDIT-1PR FIX-CODE- 4 START TIME- 11 [illegible] NUMBER OF MACHINES- 2 MACHINE NUMBERS- 11 0 0

Fig. 50 - a sample print-out from program 'lists' (reduced to ⅝ actual size)

9. THE AUTOMATIC SCHEDULING PROGRAM

9.1 Requirements of the Program

From the results obtained in the series of trial scheduling runs described in Chapter 5 it was concluded that the most satisfactory results could generally be obtained from loading rule number 15. This considered the requests for machine time in the order most important, most fixed, longest and earliest arrival first.

This conclusion was based on the assumption that an approximately equal importance would be given to each of the possible criteria by which the schedules could be judged.

In the problem faced by the BBC the requirements of the scheduling program have been clearly stated. These are:

(a) always schedule top priority requests
(b) give a high utilisation of the machines scheduled
(c) schedule all fixed time requests
(d) schedule a high percentage of 'preferred machine' requests
(e) make a minimum use of computer store and CP time

While probable that the approach which proved most satisfactory with the simulated data in the initial tests would satisfy most of these conditions, a second series of tests was undertaken to ensure that this was indeed the case.

To ensure that the results of the first tests were of general applicability several more of the loading rules were taken for further tests. From these results the best loading rule for this particular application was determined.

Of the loading rules considered earlier numbers 16 to 20 were irrelevant with the low levels of machine loading encountered. These rules only become applicable when there are far too many requests for all of them to be scheduled, and some mechanism is needed to eliminate the very long requests.

Loading rules numbers 1 to 5 all gave similar results, and in each case these were below the standards reached by other loading rules.

Numbers 6 to 8 and 11 to 13 did not give enough emphasis to one or other of the factors to be considered in judging the performance of the schedules.

Thus loading rules 9, 10, 14 and 15 were tested further, with rules 1 and 11 included for comparison purposes.

9.2 Data Used for the Test Runs

It has already been explained that historic data for requests is not kept, past schedules were, therefore, examined to extract data for this set of tests. A heavily

booked day was selected and the requests scheduled on this day were taken, with the relevant details examined. Further requests were added at random until a set of requests was obtained which had the desired characteristics.

In total 99 requests were considered, which were equivalent to 112 individual requests if those requiring more than one machine were separated. The total time requested amounted to 289 hours 17 minutes (equivalent to about 82 percent of available machine capacity).

The general characteristics of the input data are illustrated in Fig. 51, but particular mention may be made of the following observations:

(a) there were half the number of requests to be scheduled that there were in the first series of tests. Thus it is likely that there will be far less significant differences in the schedules produced by the different loading rules.

(b) there was a relationship between the 'fixed time' requests and durations, mainly caused by transmissions

(c) more requests required simultaneous processing on three machines than there was machine time available to service them, using the constraints imposed. This means that it was impossible to schedule all the requests using any approach.

The third observation has particular significance in that it puts an upper limit on the time which can be scheduled, and this limit is at a level considerably lower than would normally be expected. Although possible to schedule some 290 hours of requests, the data is so constructed that it would not usually be possible to schedule one of the longer requests for three machines. The maximumtime which may normally be scheduled is, therefore, around 265 hours, although this may be surpassed in certain circumstances.

9.3. Results Obtained from the Loading Rules Tested

For these tests the loading rules considered were chosen either because they performed generally well in the previous tests, or else they gave some sort of guide to the performance of those rules expected to give better results.

In this test loading rules 1, 9, 10, 11, 14 and 15 were tested, and the 'best' result from the scheduling runs was found. The loading rule which gave this result was then further tested with two sets of real input data to check the validity of the results obtained.

A summary of the results obtained from the

first of these tests is given in Fig 52.

It should be noted that these results, given as percentages of requested numbers, are not strictly comparable. During the course of the scheduling some compound requests are split into two or three separate requests, while their function is changed from number '5' to number '8'. It is thus possible for more than 100% of requests to be scheduled in some cases. Similarly a decreased figure will appear in other circumstances.

The total figures show an absolute comparison, while the other figures can generally be relied upon to give very fair comparisons.

Function / Fix-code	1	2	3	4	5	6	7	8	9	10	Total
1	0	0	0	4	2	1	0	0	4	7	18
2	0	0	0	0	5	1	0	0	0	0	6
3	0	0	0	0	1	0	0	0	1	4	6
4	0	0	0	3	4	1	0	0	1	1	10
5	0	0	0	0	0	0	0	0	0	0	0
6	0	19	8	10	0	0	0	0	0	1	38
7	0	0	0	0	0	0	0	0	0	0	0
8	0	3	0	4	0	0	1	0	1	2	11
9	8	1	0	0	1	0	0	0	0	0	10
Totals	8	23	8	21	13	3	1	0	7	15	99

(a) - distribution of functions and fix-codes

Duration Range (hrs.mins)	Number of Requests
0.00 - 0.15	10
0.16 - 0.30	26
0.31 - 0.45	4
0.46 - 1.00	14
1.01 - 1.30	10
1.31 - 2.00	11
2.01 - 3.00	7
3.01 - 5.00	7
5.01 - 15.00	10

(b) - distribution of durations

Fig 51 - showing some of the characteristics of the data used for the second series of tests

Loading rule / Function	1	9	10	11	14	15
1	87.50	100.00	100.00	62.50	100.00	100.00
2	100.00	100.00	100.00	91.67	100.00	100.00
3	100.00	100.00	100.00	100.00	100.00	100.00
4	100.00	100.00	100.00	95.65	100.00	100.00
5	86.67	80.00	86.67	86.67	86.67	86.67
6	66.67	33.33	33.33	100.00	33.33	33.33
7	100.00	100.00	100.00	100.00	100.00	100.00
8	100.00	100.00	100.00	100.00	100.00	100.00
9	100.00	100.00	100.00	100.00	100.00	100.00
10	100.00	100.00	100.00	100.00	100.00	100.00

Fig 52a - showing the percentage of all requests scheduled (by number) by function

Loading rule / fix-code	1	9	10	11	14	15	
1	100.00	80.00	90.00	100.00	80.00	90.00	
2	62.50	75.00	75.00	125.00	75.00	75.00	
3	100.00	116.67	100.00	100.00	100.00	100.00	
4	90.91	100.00	100.00	100.00	100.00	100.00	
5	100.00	100.00	100.00	100.00	100.00	100.00	*
6	100.00	100.00	100.00	95.24	100.00	100.00	
7	100.00	100.00	100.00	100.00	100.00	100.00	*
8	100.00	100.00	100.00	91.67	100.00	100.00	
9	84.62	100.00	100.00	76.92	100.00	100.00	

Fig 52B - showing the percentage of all requests scheduled (by number) by fix-code.

(NOTE: * - no requests were made with this fix-code)

loading rule	1	9	10	11	14	15
total number of requests scheduled	106	107	108	108	106	108
total time scheduled (hrs. mins)	246.47	247.47	253.17	280.47	241.17	253.17

Fig 52c - showing the total number and duration of scheduled requests.

From this table of results it can be seen that loading rules 10 and 15 prove to be the most generally acceptable. Because of its superior performance in the first set of trials it is most likely that approach number 15 will give the best results in the application described for the BBC. It was, therefore, decided to test this rule with two further sets of data collected specifically for this purpose. From these it was shown that entirely satisfactory results could be obtained.

The general characteristics of this final testing data were similar to those of the previous tests, and there is little point in supplying details of them. Suffice to say that in all the trials undertaken loading rule number 15, which considered requests in the order

most important function, most fixed, longest and earliest arrival first, provided satisfactory results when judged by the criteria set by the BBC.

9.4. The Program Used

The program used to obtain these results can be examined in the flow diagram shown in Appendix B. The final results of the program are the bar charts provided by program 'PRIN', but during the course of the scheduling run various messages are printed to draw attention to various points of interest to the schedulers. An example of these messages produced during one run is illustrated in Fig 53.

```
NUMBER OF RE UESTS TO BE SCHEDULED IS  99
****************************************

 REQUEST  61 NOW INCLUDES A 1HR MEAL BREAK
-RE UEST  90 NOW INCLUDES A 1HR MEAL BREAK
 REQUEST 90 CANNOT BE FITTED ON THE PREFERED M/C        ( 8)
 RE UEST 73 CANNOT BE FITTED ON THE PREFERED M/C        ( 11)
 REQUEST 73 CANNOT BE FITTED ON THE PREFERED M/C        ( 11)
 REQUEST   0 NOW INCLUDES A 1HR MEAL BREAK
 RE UEST  5  NOW INCLUDES A 1HR MEAL  REAK
 RE UEST 93 CANNOT  E FITTED ON THE PREFERED M/C        ( 8)
 REQUEST 94 CANNOT BE FITTED ON THE PREFERED M/C        ( 5)
 RE UEST  48 NOW INCLUDES A 1HR MEAL BREAK
 RE UEST 48CANNOT BE SCHEDULED (ON M/CS  5  6  7  8  0  0  0  0  0
  0  0  0  0  0)
 REQUEST 67 CANNOT BE FITTED ON THE PREFERED M/C        ( 15)
 REQUEST 67 CANNOT BE FITTED ON THE PREFERED M/C        ( 15)
 RE UEST  58 NOW INCLUDES A 1HR MEAL  REAK
 RE UEST 58 CANNOT  E FITTED ON THE PREFERED M/C        ( 2)
 RE UEST 58 CANNOT  E FITTED ON THE PREFERED M/C        ( 2)
 REQUEST 58CANNOT BE SCHEDULED (ON M/CS  14  2  0  0  0  0  0  0  0
  0  0  0  0  0)
 RE UEST  53 NOW INCLUDES A 1HR MEAL BREAK
 REQUEST 53CANNOT  E SCHEDULED (ON M/CS  14  2  0  0  0  0  0  0  0
  0  0  0  0  0)
```

Fig. 53 - illustrating some of the messages printed during the course of a scheduling run

9.5 The Design of the Program Print-Out

The format of the present hand-written scheduling sheets was illustrated in Fig. 34. There the schedules for each day are written on 8 separate sheets, with up to 300 lines of information of any width up to several hundred characters.

The BBC expressed the view that bar charts would provide the only generally acceptable form for the presentation of results.

Several alternative approaches were examined to determine how well-presented results could be obtained using bar charts, while still showing the amount of information on the present schedules.

After several initial suggestions had been made and discounted the format which gained the most favourable reception was one which gave horizontal bars as illustrated in Fig. 54.

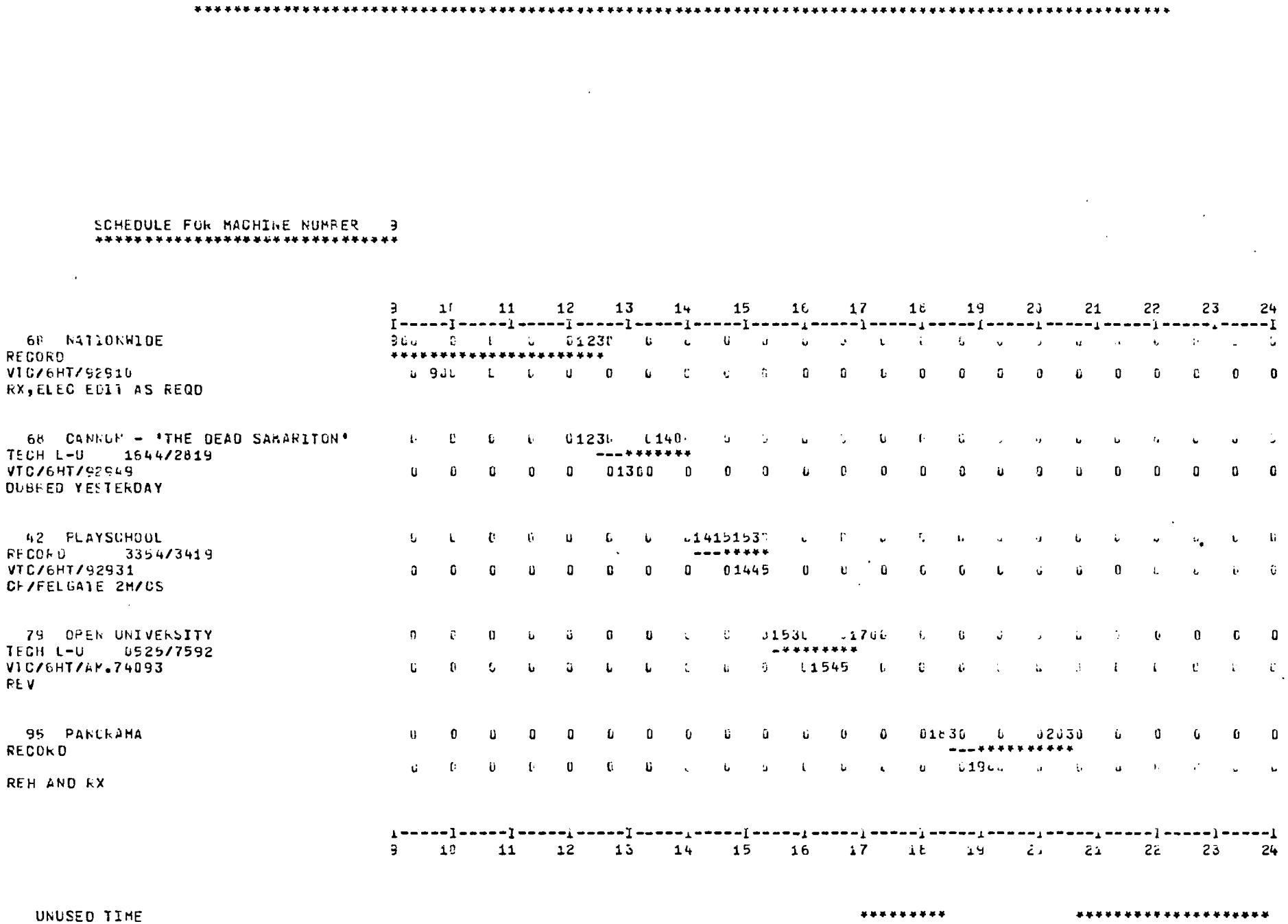


Fig. 54 - most popular bar chart format for the schedules ($\frac{5}{8}$ actual size)

10. DISCUSSION OF THE SCHEDULING SUB-SYSTEM DEVELOPED TO SOLVE THE PROBLEM FACED BY THE BBC

10.1 The Achievement of Objectives

The programs developed to solve a general scheduling problem, and then modified as necessary to handle the specific problem faced by the BBC, appear to have satisfied all the conditions imposed on them. In order to determine whether this is indeed the case a detailed examination of the objectives of the sub-system, and its actual performance, is necessary.

The first questions to be raised in evaluating the performance of the sub-system concern the general facilities available to deal with routine data handling. Are the facilities for inputing, adding to, altering or deleting data adequate? Are there sufficient error checks? Is the sub-system suited to the application, or is it a poor modification of an 'off the peg' solution?

It can be seen from the general discussion of the programs that all these requirements are met fully. The original sub-system was devised in such a way that all necessary data handling facilities would be present. Moreover the sub-system was modified to handle the BBC's problem in such a way as to ensure the complete

compatibility between the requirements and the solution offered.

It is, therefore, thought that these general criteria of success have been adequately met, as exemplified in the results of all the test runs, whether using simulated or real data.

More specific objectives concerning various parts of the sub-system have been described by the BBC. In the main these deal with the performance of the loading rule used, and can be stated as follows, with comments on their satisfaction included.

(a) Always schedule top priority bookings.

As can be seen from Fig 52a loading rule 15 schedules 100% of the top four most important functions. During the four sets of test run on these loading rules, including the initial tests using hypothetical data, loading rule 15 has always scheduled 100% of function 1 requests, and never less than 90.5% of function 2, 93.8% of function 3, 88.9% of function 4 and 91.7% of function 5.

(b) Offer a high percentage utilisation of scheduled machines

It is obvious that a high machine utilisation can only be achieved if there are sufficient requests to occupy

a high proportion of the available machine time. In the final three test series the total time requested was not great enough to provide high machine utilisations. For example in the results quoted in Fig. 52 289 hours and 17 minutes of machine time was requested of which 253 hours and 17 minutes was scheduled. This meant that although almost 88% of request time was scheduled the average machine utilisation was about 62.5%.

During the initial trials when more time was requested the machine utilisation reached an average of 94.8% of available time, and would have continued to rise if more requests for time had been added.

(c) Schedule all fixed time bookings

Fix code 6, 8 and 9 involve fixed start times. It can be seen from the results shown in Fig. 52b that this condition is satisfied in that particular test. This was also true in the final test runs. However in the initial tests the percentage of fixed start requests which were scheduled began to fall below 100% when demand for machine time rose past 70% of available time.

This would tend to indicate that although this condition will be met in most practical situations there is a limit beyond which it will be increasingly deficient. In the initial tests the level of satisfaction fell to about 70% with a demand of almost 160% of available time.

(d) Schedule a high percentage of 'preferred machine' bookings.

This involves fix codes 2, 4 and 8. Again the second set of results indicates that the degree to which this objective was met only once fell below 100%.

In the primary trials the figure declined with increasing requests, but in all cases a satisfactory performance was recorded.

(e) Permit rapid rescheduling if required

The files were so designed, on the CDC 6400 computer, that requests were read from one file and the results of the scheduling run were transmitted to a second file. Thus the actual scheduling was done using a copy of the stored data. The original data was always stored intact, so that for a complete rescheduling it is only necessary to take another copy of this data and overwrite the initial results.

(f) Be able to schedule or reschedule a limited number of machines, or groups of machines if required.

This facility was examined, but on consideration it was thought unnecessary.

The purpose of rescheduling a group of machines would be to either omit a previously scheduled machine from consideration, or else to rearrange the jobs already

scheduled on the machines. In the former case this could result in high priority requests being transferred to other machines which may not have sufficient space for them: i.e. low priority requests would be taking space needed for high priority jobs. In the latter case a simple rearrangement would be pointless, as the same requests would be considered for scheduling as are at present fitted into the schedules.

Individual requests can be moved manually by using Program 'SCHA'. Alternatively a complete rescheduling can easily be performed, but the disruption caused by rescheduling groups of machines would not be worth the results obtained.

(g) Handle regular 'block' bookings automatically

As has been explained earlier it was not possible to incorporate this feature with the limited file space available. It would, however, be a very straightforward procedure to keep these regular requests in one file which is copied as necessary to the 'master data file before the automatic scheduling run begins.

(h) Be capable of easy modification to accommodate new machines and facilities.

Program 'UPDA' is used to change any machines which may change their function or order of suitability for

particular functions. Using this program the order in which machines are considered for each function can be altered using a VDU, and machines can be deleted and replaced by new machines with simple messages sent to the computer.

Altering the order in which the functions are scheduled, or introducing completely new functions would pose more problems. The programs would have to be slightly modified by altering 'DIMENSION' statements and using different codes for each new function. Although this would be comparatively simple, it would need the involvement of a programmer to make the changes as necessary. In a practical system it would be fairly straightforward to add an additional facility so that these changes could be made by the schedulers using their video terminals.

(i) Make minimum use of computer store and mill

The times needed to run the programs 'SCHI', 'ANAL' and 'PRIN' for the comparative tests with 112 requests are give in Fig 55. This table shows the CP time used to obtain the results in CDC 6400 CP seconds. As the results are sent to a file before being printed, an indication of the time necessary to

read the results from this file and print them on the line printer is useful. These figures are also included (in the same units).

Loading Rule	Run Time Prog SCHI	Run Time Prog ANAL	Print Time	Run Time Prog PRIN	Print Time
1	4.2	2.2	0.2	6.3	1.8
9	5.5	2.4	0.2	6.2	1.7
10	5.4	2.2	0.2	6.1	1.7
11	4.6	2.3	0.2	6.4	1.8
14	5.1	2.2	0.2	6.4	1.8
15	5.4	2.4	0.2	6.1	1.7

Fig 55 - showing the CP times for six loading rules with 112 jobs.

(all figures are rounded to one place of decimals)

Having decided that loading rule 15 would give the best results for the BBC, the programs were amended slightly for the final test runs using real data. Thus the final characteristics of the programs run in 'batch' mode are as follows (again all times are in CDC6400 CP seconds rounded to one place of decimals):

Program	Run Time	Core (CDC words)	Time to Print Results
SCHI	6.4	11.7K	0.0
PRIN	6.9	9.3K	2.1
ANAL	2.6	7.7K	0.3
LISTO	6.4	9.0K	1.7

Fig 56 - showing the CP time for the final versions of the 'batch' programs

In Fig. 56 the time to print results quoted for program 'SCHI' refers to the time needed to read and print the messages which are produced during the scheduling run.

It is obvious that the times quoted will vary considerably with the number of requests to be scheduled. However, the figures given above will give a fairly accurate estimate of the times used in an average days scheduling.

Fig. 57 shows how the time taken to run program 'SCHI' increased with the number of requests considered during the first set of tests (using results for loading rule number 15).

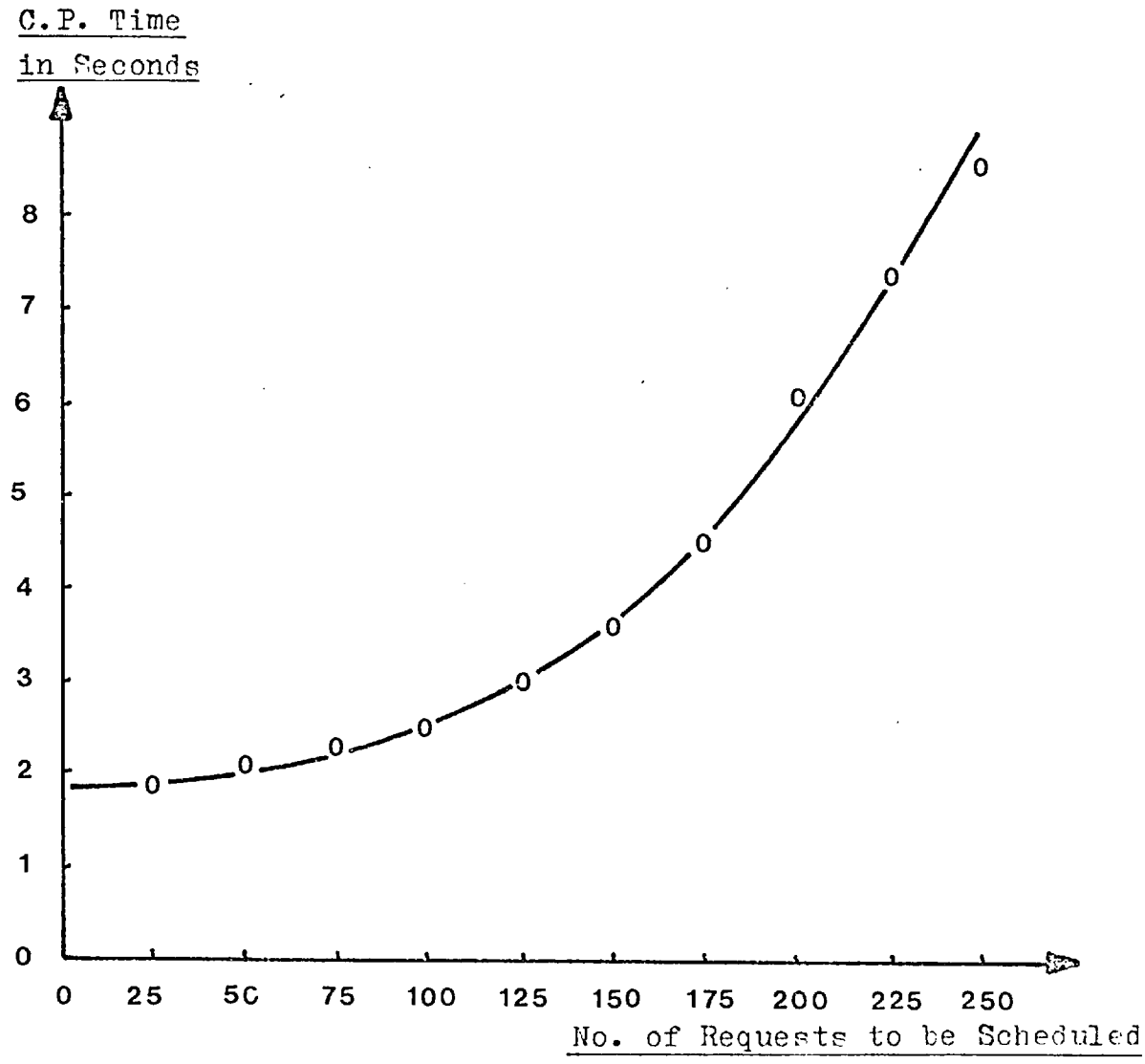


Fig. 57 - showing the C.P. time necessary to run program 'SCHI' with varying nos of requests

The CP time for the on-line programs will obviously vary enormously, depending on the number of operations done in a single run. Values for completing basic runs for each program, where only one operation is completed, may be useful. Fig. 58 gives the times for such runs, together with the core needed. In this table the run-times are in ICL 1905E mill seconds and core occupancy is given in 24 bit words. The CP times are corrected to allow for a standard loading of the system.

Program	Run Time (approx total)	Core (ICL words)	Approx Time for a single element (eg one search)
INPU	15.2	8.1K	0.9
AMEN	11.1	7.0K	1.0
STAT	8.6	6.5K	-
UPDA	12.6	6.0K	2.5
FIND	14.0	6.8K	1.8
SCHA	9.9	12.8K	1.5

Fig. 58 - showing the approximate times for running the on-line programs

10.2 Benefits and Drawbacks of the Proposed Sub-system Over the Existing Manual system

As was stated earlier, the sub-system described in this report must only be considered as a working basis of the final scheduling system that might be installed in the BBC. The following comments, therefore, are those most likely to apply to the real problem.

10.2.1 The Time Taken for the Scheduling

The most obvious effect of the proposed

sub-system is that it replaces a tedious manual operation spread over more than five weeks by a computer operation lasting a few seconds.

This has several consequences on the data used. For example, the significance of alterations to the input data will be reduced (as the data is stored in computer files, prior to being scheduled, where amendments have little effect). In the present system the requests are scheduled at an early stage with later alterations often necessitating rescheduling.

The facility also exists for a complete rescheduling of requests in a very short space of time, measured in seconds rather than days as at present. If a complete rescheduling is not necessary minor adjustments can be made to the initial schedules with a minimum of effort, by directing a program to make the changes via a video display unit. With the present system even minor changes would prove difficult. The proposed sub-system also has the advantage that changes made in the schedules may be examined, and if they are found to be unsatisfactory the schedules can be returned to their original state with ease. Again this facility is lacking in the existing system.

10.2.2 Effects on the Schedulers

The people most directly concerned with the proposed sub-system will be the schedulers themselves, who at present produce the schedules manually.

The proposed sub-system removes a tedious job from the schedulers and leaves them free to deal with the more interesting problems requiring subjective decisions. The amount of satisfaction they obtain from their job may, therefore, be expected to rise.

Less staff will be needed to operate the new system. This may mean that either a reduction in staff could be made, or else the schedulers will have more time available for discussions with machine operators and users, in order to sort-out any difficulties which may arise. A choice may be made, therefore, between reducing the cost of the scheduling or increasing its quality.

In practice the latter course may be taken, as the loss of contact between the schedulers and the machine users during the early part of the scheduling may mean that extra discussions are necessary at a later stage in the scheduling operation. The implication here is that the schedulers will not be as conversant with the schedules as at present, because they will only come into contact with them at a relatively late stage. A case may be made, in this context, that the

proposed sub-system lacks complete flexibility in that any subjective decisions can only be introduced into the scheduling at a comparatively late stage.

The schedulers will inevitably lose some of their present expertise as the necessity to keep a complete knowledge of the machines' facilities and capabilities is reduced. This could mean a shorter training period for the schedulers, but may also be a disadvantage as the schedulers will be less knowledgeable about their work. This may be countered, however, by the fact that the results obtained will depend less on the individual expertise of the schedulers than is the case at present.

10.2.3 Comparison of Information Available

On a superficial examination it may be said that certain information will be less easily accessible with the new sub-system, as it is stored in a computer file rather than being readily available on the scheduler's desk. This is not generally the case. Input documents, initial schedules with later amendments and directories of all requests may be held by the scheduler as with the existing system. The only differences with the proposed sub-system is that some of the documents will be printed instead of handwritten, directories will be available, which will reduce the time required to find information about a particular request, and all information will be readily accessible on the computer if required. Rather than make certain information less

accessible, the overall effect will be to make all information which can be accessed at the present more readily available.

In addition to the information which the schedulers have at present details of machine utilisation, patterns of requests, fluctuations in demand, lists of requests still to be scheduled and several other types of information are available with the proposed sub-system. With this increased knowledge the schedulers may smooth-out fluctuations in demand and possibly arrange from day to day the requests to be scheduled, so that efficient machine utilisation may result, and the number of requests delayed (or possibly processed on hired machines) will be reduced.

10.2.4 Consequences Arising from the Use of Computers

Arising from the use of computers are several associated factors which may be examined. For example the hand-written schedules which are produced at present will be replaced by printed ones. These may, in addition, be produced in a variety of formats to suit individual requirements. The outputs given will, therefore, be clearer and easier to interpret, more suited to individual needs and easier to produce.

If required the facility for automatically printing receipts for requests sent in, and transmitting, to production departments, relevant information about the space

reserved for particular requests may easily be included.

Routine requests, which are at present kept in a register and added to the schedules as required, may be stored permanently in a computer file and added to the schedules automatically. Thus the schedulers will be saved additional time which they may spend on the more subjective parts of the scheduling.

Errors and validation checks will be made automatically, both as a check for simple errors in the requests submitted, and also to determine if requested facilities are available. With the existing manual system such checks are not possible, as the schedules are built-up by the schedulers relying on their expertise and knowledge. If they make a mistake in the compilation of the schedules this will go unnoticed until a stage where correction is more difficult.

It is probable that if the proposed scheduling sub-system is installed, additional computerised developments will be made in associated areas. For these the transfer of information from one part of the overall computer system to another may be made possible with little further amendment to the scheduling sub-system. In particular it would be an

obvious advantage if the data prepared in one sub-system and needed in another were transferred automatically without human intervention.

Because the scheduling sub-system relies completely on the availability of a computer it may be vulnerable to machine breakdowns. Although little damage can be done to the stored data if a program is interrupted by a computer failure (particularly as the programs developed work largely on copies of data stored rather than the actual data), the delays caused by such occurrances may be, at the least, irritating. At critical times in the scheduling long delays could have fairly serious results. The only reassurance possible is that, on the whole, computers are very reliable and are not prone to extended failures.

10.2.5 Economic Factors

Inevitably with the proposed sub-system there will be an initial capital cost incurred for the purchase of hardware (VDUs and printers) and for the final development of the software. In addition there will be a continuing cost of computer facilities.

Against these costs must be weighed the expected savings. These include the possible reduction

in staff mentioned previously, and the higher machine utilisation which may reduce charges for hiring outside machine time, or delay the purchase of additional machines needed for increasing demands.

Assuming that the staffing levels are maintained at their present levels, with a corresponding increase in the quality of the schedules produced, the quantifiable benefits to be gained from the new sub-system must be judged primarily on an increase in machine utilisation. With 25 machines available for 15 hours a day there is a total machine availability of 375 hours per day. For each one per cent increase in overall machine utilisation, and assuming the running costs quoted by manufacturers of £50 per hour, a saving of more than £68,000 a year could result.

Video terminals cost, at the present, about £3,000 and so it may be seen that if a one percent increase in machine utilisation were achieved, the savings made during the first year would more than pay for the hardware and necessary software development. Exact costs and savings are impossible to calculate, but the introduction of the sub-system described would be an extremely attractive proposition if measured on a purely economic value.

SECTION C

CONCLUSIONS

CONCLUSIONS

The object of this report was to describe a complete scheduling sub-system which could be attached to a larger computerised Management Information System. This sub-system was to be capable of solving a general scheduling problem with the characteristics described in Chapter 1.

Before the sub-system was formulated an investigation into the general properties of M.I.S.s was undertaken. From this the best approach to the solution of the problem was found, and a sub-system was designed which satisfied all the conditions imposed. A set of ten programs was written, of which six were purely on-line, operated via V.D.U.s, and four were terminal initiated batch programs, stored on discs and recalled as necessary.

The sub-system incorporated an automatic scheduling program. Available literature was searched in order to determine the best approach to this program, and it was concluded that a series of loading rules should be tested and the results compared. Twenty loading rules were devised and the results were compared using a series of 200 test runs. The practical results obtained corresponded closely to theoretically predicted results.

Although the 'best' results were impossible to determine without reference to a particular set of criteria, it was thought that to schedule requests in the order: most important function, most fixed, longest and earliest arrival first, produced generally satisfactory results.

In order to remove the generality of the sub-system developed, elaborate the details and prove its usefulness, a practical situation was examined where the sub-system could be used.

The problem examined was the scheduling of video tape machines at the BBC. The various alterations and modifications necessary to the sub-system for this application were described, again with some emphasis on the automatic scheduling program. A further series of runs was made to test the schedules produced by six of the original loading rules. From these it was concluded that the loading rule which performed 'best' in the original test series adequately met the requirements of the BBC. This was confirmed with a further two runs made using real (as opposed to simulated) data.

Flow diagrams were drawn for all the ten programs written, with samples of print-outs to illustrate their methods of operation.

Finally, mention was made of the benefits and drawbacks thought to be associated with the new scheduling sub-system.

REFERENCES

1. Balas E. (1970)

"Machine Sequencing: Disjunctive Graphs and Degree-Constrained Sub-graphs." in Naval Research Logistics Quarterly vol 17 no. 1 pp 1 - 10

2. Burck G. and the Editors of 'Fortune' (1965)

"The Computer Age" Published in New York by Harper and Row

3. Chandor A., Graham J., and Williamson R. (1970)

"A Dictionary of Computers" Pub. in England (Harmondsworth, Middlesex) by Penguin Books Ltd.

4. Dearden J. (1964)

"Can Management Information be Automated?" in 'Reprints from Harvard Business Review - Computer Management Series' part 1 pp 62 - 69

5. Dearden J. (1966)

"Myth of Real-Time Management Information" in 'Reprints from Harvard Business Review - Computer Management Series' part 1 pp 103 - 112

6. Eilon S. and King J. (1967)

"Industrial Scheduling Abstracts (1950 - 1966)" Pub. in London by Oliver and Boyd.

7. Eilon S. and Pace A. J. (1970)

"Job Shop Scheduling with Regular Batch Arrivals" in 'Proceedings of the Institute of Mechanical Engineers' vol 184 part 1 pp 301 - 310

8. Firnberg D. (1973)

"Computers, Management and Information" Pub. in London by George Allen and Unwin Ltd.

9. Gearing H.W. (1959)

"On-Line, Off-Line or Shared Time?" in The Computer Journal vol 2 no 3 p 150-editorial comment

10. Gladwin B. J. (1969)

"The Utilisation of Graphic Display Units as the Main Form of Computer Input" in the Computer Journal vol 12 no. 2 p. 114.

11. Gupta J. N. D. (1971)

"The Generalised n-Job, M-Machine Scheduling Problem" in Opsearch vol. 8 no. 3 pp 173 - 185

12. Lock D. (1971)

"Industrial Scheduling Techniques" Pub. in London by Gower Press Ltd.

13. Mahendra S. Bakshi and Sant Ram Arora (1969)

"The Sequencing Problem" in Management Science Dec. 1969 p B262.

14. Radley G. W. (1973)

"Management Information Systems" Pub. in England (Aylesbury, Bucks) by International Textbook Company Ltd.

15. Waters C. D. J. (1972)

"The Scheduling of Technical Resources" M.Sc. Report at Imperial College of Science and Technology, University of London.

APPENDICES

Appendix A - A Description of the Job Data Used for the Test Runs

Appendix B - The Flow Diagram for the Automatic Scheduling Program 'SCHI'

Appendix C - The Results Obtained from the Initial Series of Test Scheduling Runs

Appendix D - Flow Diagrams for the Programs Developed

APPENDIX A

A DESCRIPTION OF THE JOB DATA USED FOR THE TEST RUNS

A description of the job data used for the primary set of test runs has been described in section 5.5.1. The details given in this Appendix show the breakdown of these requests by function and fix code.

No. of requests	25	50	75	100	125	150	175	200	225	250
Duration										
0.30	3	7	13	15	18	19	24	26	27	29
0.45	0	0	0	0	1	1	1	1	5	10
1.00	6	13	20	28	31	38	41	43	48	51
1.15	0	0	0	0	0	0	0	1	2	2
1.20	0	0	0	0	0	0	1	1	1	1
1.25	0	0	0	0	0	0	0	1	1	1
1.30	5	9	12	16	19	23	29	30	32	35
1.40	0	0	0	0	1	1	1	1	1	1
1.45	0	0	0	0	0	1	1	2	2	6
2.00	5	8	10	10	13	16	19	21	22	23
2.15	0	0	0	0	0	0	1	1	1	3
2.30	2	2	3	6	8	8	10	13	15	17
2.45	0	0	0	0	0	0	0	0	1	1
3.00	3	5	8	8	9	11	12	13	15	16
3.15	0	0	0	0	0	0	0	1	1	1
3.30	1	3	4	7	11	12	13	13	14	15
4.00	0	3	5	6	7	11	11	14	15	15
4.15	0	0	0	0	0	0	0	1	1	1
4.30	0	0	0	0	1	2	2	3	4	5
5.00	0	0	0	1	1	1	1	2	2	2
5.30	0	0	0	1	1	1	1	2	2	2
6.00	0	0	0	0	1	2	4	4	5	5
6.30	0	0	0	1	1	1	1	1	2	2
7.00	0	0	0	1	1	1	1	1	1	1
7.30	0	0	0	0	0	0	0	1	1	1
8.00	0	0	0	0	1	1	1	1	1	1
8.30	0	0	0	0	0	0	0	0	1	1
9.00	0	0	0	0	0	0	0	2	2	2

Distribution of jobs with durations for the various numbers of jobs scheduled (25 to 250)

Fix Code	1	2	3	5	6	7	8	9	Total
Function									
1	3	3	3	2	2	3	3	6	25
2	2	3	2	3	3	7	4	3	27
3	4	1	5	2	9	3	2	2	28
4	7	7	3	5	3	3	0	1	29
5	8	6	5	4	2	1	3	0	29
6	9	6	3	5	2	2	2	2	31
7	9	4	8	4	1	2	1	2	31
8	8	6	3	4	3	2	2	0	28
9	7	5	3	2	2	0	2	1	22
Total	57	41	35	31	27	23	19	17	250

Distribution of functions with fix codes for 250 Requests

Fix Code	1	2	3	5	6	7	8	9	Total
Function									
1	6.15	9.20	4.25	4.00	2.00	5.30	9.00	9.30	50.00
2	7.00	5.15	1.30	7.25	4.15	17.35	10.50	6.30	60.20
3	18.00	3.30	9.00	8.20	32.10	8.00	5.45	2.55	87.50
4	9.45	21.30	6.40	9.10	10.45	11.00	0.00	1.00	69.40
5	30.20	18.35	5.25	7.10	6.00	5.00	6.45	0.00	79.15
6	24.15	8.00	11.15	10.00	3.45	2.50	2.30	6.15	68.50
7	9.20	11.30	13.00	7.20	2.00	4.30	4.00	6.40	58.20
8	17.15	14.45	6.00	17.00	4.30	8.40	5.15	0.00	73.25
9	24.10	25.15	7.30	9.00	5.15	0.00	7.00	3.00	81.10
TOTAL	145.20	117.40	64.45	79.25	70.40	63.05	51.05	35.50	628.50

Distribution of the times requested for functions and fix codes for 250 Requests (in hours.minutes)

No. of requests	25	50	75	100	125	150	175	200	225	250
Line-up Time (mins)										
0	2	9	13	19	20	22	28	34	40	50
10	0	0	0	1	1	1	1	2	2	6
15	9	16	25	32	37	43	50	55	62	66
20	0	0	0	2	5	7	8	9	9	11
25	0	0	0	0	0	0	0	1	1	2
30	14	25	37	45	59	71	78	85	93	95
35	0	0	0	0	0	0	0	0	0	1
40	0	0	0	1	1	2	2	2	2	2
45	0	0	0	0	2	4	8	11	15	16
60	0	0	0	0	0	0	0	1	1	1

Distribution of line-up Times for up to 250 Requests

APPENDIX B

THE FLOW DIAGRAM FOR THE AUTOMATIC SCHEDULING PROGRAM 'SCHI'

This program reads data concerning requests from the data files, together with the constraints imposed, and produces schedules as possible.

The information concerning requests is stored in an array during the scheduling, and is updated with machine numbers and times whenever a request is scheduled. When the scheduling run is completed this updated information is written back onto the data files. Information about the time which is free is stored in a second array, in which the elements represent five minute time elements on corresponding machines. A '1' in an array time element indicates that this particular five minute time element is already filled, while an '0' indicates that the time is still free to be scheduled.

The general procedure followed in the program is that a request is examined; if no fixed or preferred conditions are specified, the order of machine preference is determined from its function. A routine is then followed which searches the relevant elements of the schedule recording matrix until a suitable gap is found. The values of these elements are then switched from '0' to '1' and the information stored concerning the request is updated with machine number and time.

Options in the program allow for complicating procedures, such as preferred or fixed conditions, the addition of meal breaks, duration flexibility etc.

The particular situation taken in the flow diagram illustrated schedules the requests in the order:

1. most important function (numbers 1 - 10)
2. most fixed (fix codes 9 - 1)
3. longest
4. earliest

In the flow diagram the elements are as follows:

START

1. Sets up the initial conditions for the program.
2. Reads all the information available for the requests of a particular day, from the data files.
3. Writes a message to the output file giving the number of requests to be scheduled in the day
4. Sets the relevant variables, including the schedule recording matrix, to zero.
5. Arranges the requests in the order longest first.
6. Sets the next function to be considered
7. Sets the next fix-code to be considered
8. Is this a valid fix-code? Yes - 9, No - 64
9. Takes the next (or first) request to be considered
10. Is this request blank or deleted? Yes - 12, No - 11
11. Does this request have the right function and fix code (as set in elements 6 and 7) Yes - 13, No - 12
12. Are there any more requests to be considered? Yes - 9, No - 7
13. Does this request have to be scheduled after another earlier request? Yes - 14, No - 17
14. Has this earlier request been scheduled? Yes - 16, No - 15
15. Writes a message to the output file that the earlier request has not been scheduled.

16. Sets the earliest possible start time for the request (i.e. the finish time of the earlier request)
17. Calculates the start time and duration of the request in terms of five minute elements from 0900 (where a start time is specified, otherwise a blank is left)
18. Is the request for editing lasting longer than five hours? Yes - 19, No - 20
19. Adds a one hour meal break to the duration
20. Sets the finish time in terms of five minute elements from 0900 (again where a finishing time has been specified)
21. Directs the program to the relevant part as determined by the fix code. The destinations are:

Fix code	Machine	Start Time	element sent to
1	any	any	39
2	preferred	any	36
3	any	preferred	28
4	preferred	preferred	25
5	fixed	any	22
6	any	fixed	28
7	fixed	preferred	44
8	preferred	fixed	25
9	fixed	fixed	50

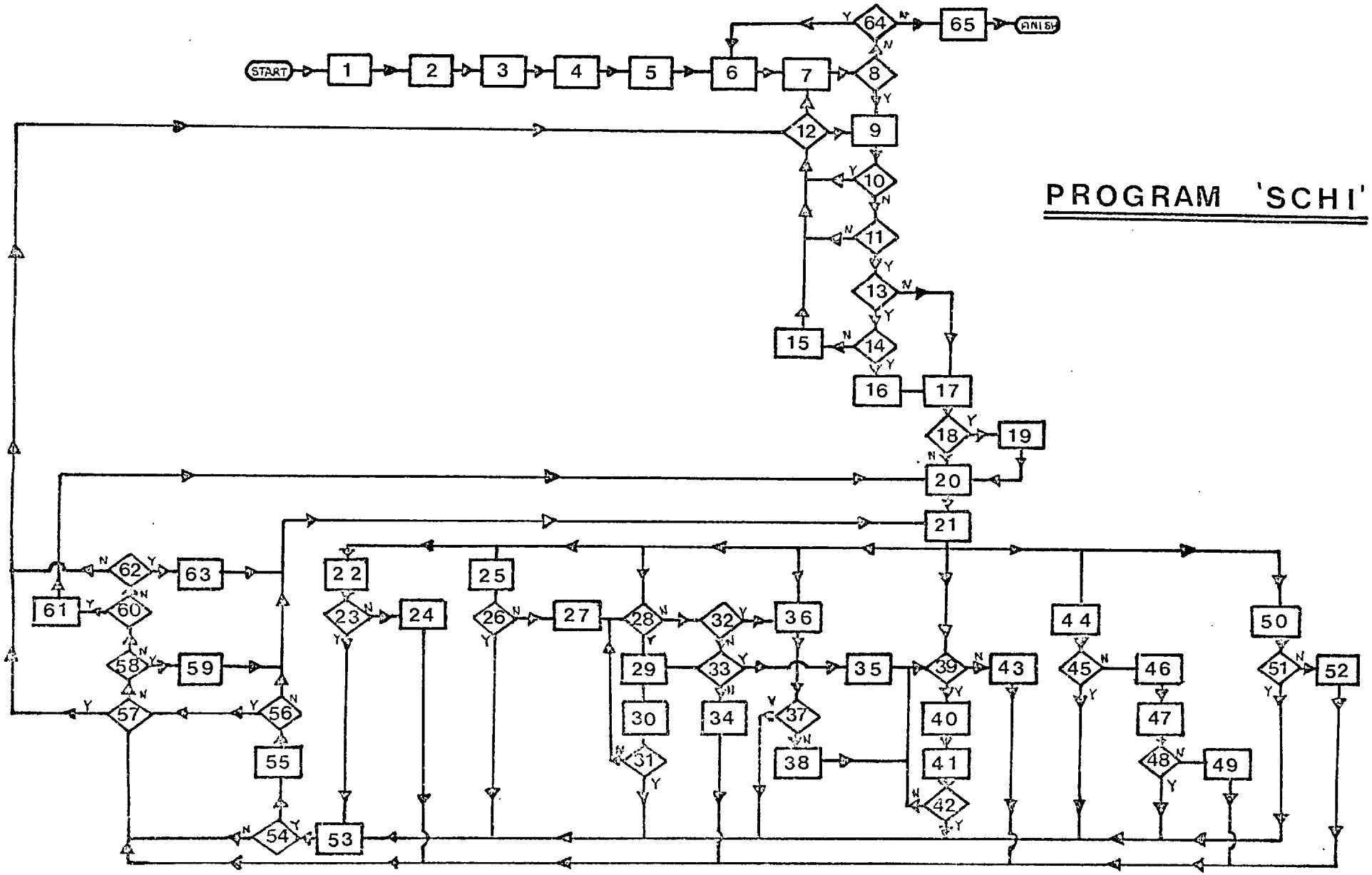
22. Calls the Subroutine 'SPA1', which searches the schedules of a named machine to determine if a suitable gap exists, and if there is one it schedules the request in it.
23. Was the request scheduled on this machine? Yes - 53, No - 24
24. Writes a relevant message to the output file

25. Calls the Subroutine 'SPA2', which searches the schedules of a named machine to determine if a specified time is free, and if it is the request is scheduled there.
26. Has the request been scheduled on this machine? Yes - 53, No - 27
27. Writes a relevant message to the output file
28. Are there any more machines associated with the function of this request? Yes - 29, No - 32
29. Takes the next machine number.
30. Calls the Subroutine 'SPA2' to see if the request can be fitted at the specified time on this machine.
31. Was the request scheduled on this machine? Yes - 53 No - 28
32. Is the fix-code equal to four (i.e. preferred time on a preferred machine)? Yes - 36, No - 33
33. Is the fix-code equal to 3 (i.e. preferred time on any machine)? Yes - 35, No - 34
34. The fix-code for the request must be either six or eight and the request cannot be scheduled at the fixed start time. A message is written to the output file.
35. Writes a message to the output file.
36 Calls the Subroutine 'SPA1' to see if the request can be fitted at any time on the named machine.
37. Was the request scheduled on this machine? Yes - 53, No - 38
38. Writes a message to the output file.
39. Are there any more machines associated with the function of this request? Yes - 40, No - 43
40. Takes the next machine number.
41. Calls the Subroutine 'SPA1' to see if the request can be scheduled at any time on the named machine.
42. Was the request scheduled on this machine? Yes - 53, No - 39

43. Writes a message to the output file
44. Calls the Subroutine 'SPA2' to see if the request can be scheduled at the specified time on the named machine.
45. Was the request scheduled on this machine? Yes - 53, No - 46.
46. Writes a message to the output file.
47. Calls the Subroutine 'SPA1' to see if the request can be scheduled at any time on the named machine.
48. Was the request scheduled on this machine? Yes - 53, No - 49
49. Writes a message to the output file
50. Calls the Subroutine 'SPA2' to see if the request can be scheduled at the given time on the specified machine.
51. Was the request scheduled on this machine? Yes - 53, No - 52
52. Writes a message to the output file
53. Confirms the start time etc.
54. Is the request to be scheduled on more than one machine? Yes - 55, No - 57
55. Call the Subroutine 'MULTI'. This subroutine checks the machine number and time where the request is scheduled, together with the number of machines which are required by the request. The schedules of the machines associated with the first machine are searched to determine if a corresponding gap exists. If the correct times are free on all machines the request is scheduled.
56. Was the request successfully scheduled on these machines? Yes - 57, No - 21
57. Has the request been successfully scheduled at some point during the course of the program? Yes - 12, No - 58
58. Does the request need more than one machine? Yes - 59, No - 60

59. Resets the value of the schedule recording matrix so that the same gaps need not be tried again.
60. Is there any flexibility in the duration of the request? Yes - 61, No - 62
61. Reduces the duration by the amount of the flexibility.
62. Is the request to be on a pair of machines? Yes - 63, No - 12
63. Resets the request as using two separate machines.
64. Are there any more functions to be considered? Yes - 6, No - 65
65. Writes a terminating message, and copies the updated information concerning the requests, together with the schedule recording matrix, onto the data files

FINISH


PROGRAM 'SCHI'
START
FINISH
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

APPENDIX C

THE RESULTS OBTAINED FROM THE INITIAL SERIES OF TEST SCHEDULING RUNS

For the initial series of test scheduling runs, the results obtained from each of the twenty loading rules devised were compared. A summary of these results, with varying numbers of requests from 25 to 250, is given in the following tables. Emphasis is placed on the percentage of requests scheduled (in terms of both times and numbers), percentages of fixed and preferred conditions satisfied and the distribution of jobs scheduled by function.

Summary of the results obtained from loading rule Number 1. (arrival order).

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	96.0	97.1	12.6	90.0	87.5	100	100
50	98.0	98.6	25.9	93.3	93.3	90.9	100
75	98.7	99.0	38.3	94.4	93.8	88.2	100
100	96.0	95.8	53.4	82.6	95.2	85.7	91.7
125	92.0	91.4	67.6	71.0	85.2	81.5	88.5
150	84.0	84.0	76.3	59.5	69.4	75.0	71.9
175	81.7	81.2	85.1	56.5	61.9	67.5	58.5
200	75.5	74.0	94.7	50.0	55.1	55.1	58.7
225	70.2	62.8	91.6	45.3	49.1	50.0	52.8
250	65.6	59.1	92.9	40.8	42.9	45.0	48.3

Summary of Results

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	66.7	100	100	100	100	100	100	100	100
50	83.3	100	100	100	100	100	100	100	100
75	88.9	100	100	100	100	100	100	100	100
100	88.9	90.0	100	90.0	100	92.9	100	100	100
125	80.0	83.3	91.7	85.7	100	83.3	100	100	100
150	76.9	62.5	68.8	77.8	89.5	84.2	94.4	100	100
175	70.6	64.7	66.7	80.0	90.0	82.6	91.3	95.2	87.5
200	68.4	52.4	59.1	70.8	83.3	83.3	87.5	91.3	78.9
225	65.2	56.5	54.2	66.7	74.1	74.1	85.2	80.8	71.4
250	60.0	51.9	46.4	62.1	69.0	67.7	87.1	75.0	68.2

Distribution of Requests Scheduled by Function

Summary of the results obtained from loading rule Number 2. (function, arrival order)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	66.7	100
50	100	100	26.3	100	100	81.8	100
75	98.7	97.7	37.8	94.4	100	88.2	100
100	96.0	94.2	52.5	82.6	95.2	90.5	91.7
125	97.6	96.8	71.6	90.3	96.3	92.6	84.6
150	92.7	90.4	82.1	78.6	94.4	71.9	75.0
175	88.6	83.3	87.4	73.9	85.7	62.5	70.7
200	84.0	72.1	92.3	61.1	75.5	51.0	56.5
225	72.9	64.7	94.3	54.7	63.6	44.4	54.7
250	67.2	59.2	93.1	52.1	60.3	41.7	50.0

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	91.7	100
100	100	100	100	90.9	100	92.9	100	92.9	87.5
125	100	100	100	92.9	100	94.4	100	100	91.7
150	100	100	100	83.3	100	84.2	100	88.9	76.9
175	100	100	100	85.0	95.0	87.0	91.3	81.0	56.3
200	100	90.5	100	87.5	91.7	83.3	75.0	52.2	26.3
225	95.7	91.3	100	85.2	92.6	63.0	70.4	38.5	14.3
250	96.0	92.6	92.9	82.8	89.7	51.6	64.5	21.4	4.5

Summary of the results obtained from loading rule Number 3. (fix-code, arrival order)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	88.2	100
100	100	100	55.8	100	100	85.7	95.8
125	100	100	73.9	100	100	85.2	92.3
150	98.0	96.0	87.2	92.9	100	68.8	90.6
175	84.0	81.9	85.9	91.3	100	60.0	80.5
200	83.5	70.8	90.6	85.2	100	44.9	69.6
225	79.1	62.8	91.6	82.8	98.2	40.7	60.4
250	74.8	58.0	91.3	76.1	98.4	40.0	56.9

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
125	100	100	100	100	100	100	100	100	100
150	100	100	93.8	94.4	100	94.7	100	100	100
175	100	100	94.4	95.0	75.0	82.6	100	85.7	81.3
200	89.5	90.5	90.9	87.5	62.5	79.2	100	82.6	68.4
225	87.0	91.3	75.0	85.2	55.6	81.5	96.3	73.1	66.7
250	84.0	77.8	75.0	79.3	58.6	80.6	90.3	60.7	54.5

Summary of the results obtained from loading rule Number 4. (fix-code, function, arrival order)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	88.2	100
100	100	100	55.8	100	100	90.5	95.8
125	100	100	73.9	100	100	88.9	92.3
150	98.0	96.0	87.2	92.9	100	68.8	93.8
175	90.9	83.3	87.4	91.3	100	60.0	82.9
200	83.5	71.4	91.4	88.9	100	49.0	69.6
225	78.7	62.4	91.0	84.4	98.2	42.6	64.2
250	72.8	56.8	89.3	78.9	96.8	41.7	60.3

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
125	100	100	100	100	100	100	100	100	100
150	100	100	93.8	100	100	94.7	100	94.4	100
175	100	100	94.4	100	85.0	82.6	100	81.0	75.0
200	94.7	95.2	90.9	87.5	79.2	75.0	100	73.9	52.6
225	91.3	91.3	79.2	88.9	63.0	77.8	96.3	61.5	57.1
250	92.0	88.9	67.9	79.3	62.1	71.0	80.6	60.7	50.0

Summary of the results obtained from loading rule Number 5. (function, fix-code, arrival order)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	94.1	100
100	98.0	96.2	53.6	91.3	95.2	95.2	91.7
125	97.6	95.9	70.9	90.3	96.3	77.8	88.5
150	92.0	90.4	82.1	78.6	94.4	65.6	78.1
175	89.1	82.9	86.9	73.9	92.9	62.5	73.2
200	80.5	71.7	91.7	66.7	75.5	46.9	60.9
225	73.8	64.3	93.8	59.4	70.9	48.1	54.7
250	68.8	59.8	94.0	54.9	68.3	38.3	46.6

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	90.9	100	100	100	100	87.5
125	100	100	100	92.9	100	100	100	94.1	91.7
150	100	100	93.8	88.9	100	84.2	100	88.9	84.6
175	100	100	94.4	90.0	95.0	82.6	95.7	85.7	56.3
200	100	90.5	95.5	95.8	95.8	83.3	79.2	52.2	26.3
225	100	91.3	95.8	88.9	96.3	81.5	63.0	30.8	9.5
250	100	92.6	96.4	82.8	93.1	61.3	54.8	21.4	9.1

Summary of the results obtained from loading rule Number 6. (shortest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	98.0	98.1	25.8	93.3	93.3	81.8	90.9
75	98.7	98.7	38.2	94.4	93.8	76.5	93.3
100	94.0	91.8	51.2	78.3	85.7	71.4	87.5
125	91.2	87.4	64.6	71.0	74.1	66.7	84.6
150	88.7	82.8	75.2	66.7	77.8	56.3	71.9
175	83.4	71.6	75.1	63.0	69.0	55.0	73.2
200	76.5	58.1	74.3	59.3	63.3	51.0	71.7
225	72.9	51.5	75.1	56.3	61.8	46.3	67.9
250	71.6	49.7	78.1	52.1	63.5	43.3	63.8

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	83.3	100	100	100	100	100	100	100	100
75	88.9	100	100	100	100	100	100	100	100
100	77.8	100	80.0	90.9	100	100	100	92.9	100
125	70.0	91.7	75.0	92.9	100	94.4	100	94.1	91.7
150	76.9	87.5	75.0	83.3	89.5	94.7	100	94.4	92.3
175	70.6	82.4	75.0	85.0	85.0	91.3	91.3	90.5	81.3
200	68.4	71.4	68.2	70.8	70.8	91.7	87.5	87.0	68.4
225	73.9	69.6	62.5	74.1	70.4	77.8	81.5	80.8	61.9
250	72.0	63.0	60.7	75.9	72.4	77.4	83.9	71.4	63.6

Summary of the results obtained from loading rule Number 7. (function, shortest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	90.9
75	100	100	38.6	100	100	88.2	93.3
100	97.0	94.6	52.8	87.0	95.2	85.7	87.5
125	96.0	93.9	69.4	87.1	92.6	74.1	84.6
150	90.7	85.7	77.8	73.8	94.4	68.8	75.0
175	87.4	80.3	84.2	73.9	83.3	62.5	75.6
200	78.5	69.7	89.2	63.0	75.5	53.1	58.7
225	72.0	61.8	90.1	48.4	70.9	46.3	58.5
250	68.4	57.7	90.5	52.1	65.1	43.3	51.7

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	90.9	100	100	100	92.9	87.5
125	100	100	100	92.9	100	100	100	88.2	83.3
150	100	100	93.8	88.9	100	78.9	94.4	88.9	69.2
175	100	100	94.4	95.0	95.0	82.6	87.0	81.0	50.0
200	100	90.5	90.9	91.7	95.8	83.3	79.2	47.8	21.1
225	100	91.3	91.7	74.1	92.6	74.1	66.7	34.6	19.0
250	100	92.6	89.3	82.8	93.1	51.6	64.5	28.6	4.5

Summary of the results obtained from loading rule Number 8. (fix-code, shortest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	88.2	100
100	100	100	55.8	100	100	85.7	95.8
125	100	100	73.9	100	100	81.5	92.3
150	94.7	88.9	80.7	92.9	100	68.8	93.8
175	90.3	81.0	83.5	91.3	100	60.0	82.9
200	83.5	68.0	87.0	88.9	98.0	49.0	71.7
225	77.8	60.3	87.9	85.9	96.4	42.6	62.3
250	74.0	55.6	87.5	77.5	95.2	43.3	62.1

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
125	100	100	100	100	100	100	100	100	100
150	100	100	93.8	100	84.2	94.7	100	94.4	84.6
175	100	100	94.4	100	70.0	82.6	100	85.7	81.3
200	78.9	90.5	86.4	91.7	66.7	79.2	100	82.6	73.7
225	82.6	87.0	70.8	92.6	59.3	77.8	100	65.4	61.9
250	80.0	81.5	67.9	75.9	62.1	80.6	93.5	57.1	63.6

Summary of the results obtained from loading rule Number 9. (fix-code, function, shortest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	88.2	100
100	100	100	55.8	100	100	90.5	95.8
125	100	100	73.9	100	100	88.9	92.3
150	96.7	92.6	84.1	92.9	100	65.6	93.8
175	90.9	83.2	87.2	91.3	100	62.5	82.9
200	83.5	70.6	90.4	88.9	100	49.0	69.6
225	77.3	61.7	89.9	85.9	98.2	44.4	62.3
250	73.6	57.2	89.9	78.9	96.8	38.3	58.6

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
125	100	100	100	100	100	100	100	100	100
150	100	100	93.8	100	100	94.7	100	94.4	84.6
175	100	100	94.4	100	85.0	82.6	100	81.0	75.0
200	94.7	95.2	90.9	87.5	75.0	75.0	100	73.9	57.9
225	95.7	91.7	79.2	85.2	59.3	74.1	88.9	61.5	57.1
250	96.0	88.9	67.9	79.3	62.1	71.0	83.9	57.1	54.5

Summary of the results obtained from loading rule Number 10. (function, fix-code, shortest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	94.1	100
100	98.0	96.1	53.5	91.3	95.2	95.2	91.7
125	96.8	94.8	70.1	87.1	96.3	77.8	88.5
150	93.3	90.4	82.1	78.6	94.4	62.5	75.0
175	88.0	81.4	85.2	73.9	90.5	60.0	73.2
200	80.0	71.1	91.0	66.7	75.5	49.0	58.7
225	73.3	63.2	92.2	57.8	72.7	46.3	54.7
250	69.6	59.7	93.8	54.9	68.3	41.7	50.0

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	90.9	100	100	100	100	87.5
125	100	100	100	92.8	100	100	100	88.2	91.7
150	100	100	93.7	88.8	100	84.2	100	88.8	84.6
175	100	100	94.4	95.0	95.0	82.6	91.3	85.7	33.3
200	100	90.5	95.5	95.8	95.8	83.3	79.2	47.8	26.3
225	100	91.3	95.8	85.2	96.3	77.8	63.0	34.6	9.5
250	100	92.6	96.4	82.8	89.7	61.3	64.5	21.4	9.1

Summary of the results obtained from loading rule Number 11. (longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	92.0	93.2	12.1	80.0	75.0	100	100
50	94.0	95.7	25.1	80.0	86.7	90.9	100
75	96.0	97.1	37.6	83.3	87.5	88.2	100
100	94.0	96.3	53.7	73.9	85.7	90.5	83.3
125	92.0	95.2	70.4	71.0	77.8	85.2	69.2
150	81.3	86.2	78.3	47.6	63.9	75.0	56.3
175	79.4	85.0	89.2	45.6	59.5	57.5	48.8
200	66.0	77.5	99.1	29.6	40.8	30.6	34.8
225	54.7	68.1	99.3	21.9	34.5	24.1	17.0
250	48.0	62.9	98.9	12.7	31.7	21.7	17.2

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	33.3	100	100	100	100	100	100	100	100
50	66.7	100	100	100	100	87.5	100	100	100
75	77.8	100	100	100	100	92.3	100	100	100
100	77.8	80.0	100	100	100	85.7	100	100	100
125	60.0	75.0	100	100	100	83.3	100	100	100
150	46.2	43.8	75.0	83.3	94.7	78.9	100	100	100
175	52.9	35.3	72.2	80.0	95.0	82.6	91.3	95.2	100
200	42.1	47.6	59.1	62.5	83.3	75.0	75.0	78.3	84.2
225	26.1	39.1	54.2	58.3	63.0	51.9	59.3	69.2	76.2
250	36.0	33.3	53.6	55.2	65.5	35.5	38.7	60.7	68.2

Summary of the results obtained from loading rule Number 12. (function, longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	98.7	92.9	35.9	94.4	100	88.2	73.3
100	96.0	95.3	53.1	82.6	95.2	95.2	91.7
125	95.2	94.1	69.5	80.6	96.3	85.2	88.5
150	93.3	92.2	83.8	78.6	94.4	75.0	65.6
175	89.7	87.0	91.3	76.1	85.7	62.5	70.7
200	78.5	73.4	93.9	59.3	75.5	49.0	58.7
225	71.1	64.5	94.0	50.0	69.1	40.7	54.7
250	65.6	60.8	95.5	47.9	61.9	31.7	46.6

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	92.3	100	100	100
100	100	100	100	90.9	100	85.7	100	100	87.5
125	100	100	100	92.9	100	88.9	100	88.2	91.7
150	100	100	100	83.3	100	84.2	100	88.9	84.6
175	100	100	100	85.0	95.0	87.0	95.7	81.0	62.5
200	100	95.2	100	83.3	97.5	79.2	79.2	60.9	15.8
225	95.7	95.7	100	77.8	85.2	77.8	70.4	30.8	0.0
250	96.0	96.3	100	79.3	82.8	58.1	41.9	25.0	4.5

Summary of the results obtained from loading rule Number 13. (fix-code, longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	88.2	100
100	100	100	55.8	100	100	85.7	95.8
125	100	100	73.9	100	100	85.2	92.3
150	98.0	96.3	87.2	92.9	100	75.0	87.5
175	92.0	85.7	89.9	91.3	100	62.5	82.9
200	84.5	73.1	93.6	87.0	100	40.8	71.7
225	78.2	63.7	92.9	84.4	100	37.0	62.3
250	74.0	59.6	93.7	77.5	98.4	30.0	58.6

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
125	100	100	100	100	100	100	100	100	100
150	100	100	93.8	100	100	94.7	100	94.4	100
175	100	100	88.9	100	85.0	87.0	100	81.0	87.5
200	94.7	90.5	90.9	87.5	75.0	79.2	95.8	87.0	68.4
225	87.0	91.3	79.2	81.5	63.0	66.7	96.3	73.1	66.7
250	84.0	85.2	75.0	75.9	62.1	64.5	93.5	62.3	59.1

Summary of the results obtained from loading rule Number 14. (fix-code, function, longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	88.2	100
100	100	100	55.8	100	100	90.5	95.8
125	100	100	73.9	100	100	88.9	92.3
150	97.3	95.0	86.3	92.9	100	68.8	93.8
175	92.6	86.2	90.2	91.3	100	57.5	82.9
200	85.0	71.5	91.4	88.9	100	44.9	71.7
225	77.3	63.2	92.2	81.3	98.2	38.9	62.3
250	72.8	57.6	90.5	77.5	96.8	40.0	56.9

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
125	100	100	100	100	100	100	100	100	100
150	100	100	93.8	100	100	94.7	100	94.4	92.3
175	100	100	94.4	100	90.0	87.0	100	81.0	81.3
200	94.7	95.2	95.5	87.5	75.0	75.0	100	78.3	63.2
225	91.3	91.3	75.0	85.2	63.0	74.1	96.3	65.4	52.4
250	88.0	85.2	67.9	79.3	58.6	67.7	93.5	57.1	54.5

Summary of the results ontained from loading rule Number 15. (function, fix-code, longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	94.1	100
100	99.0	97.5	54.4	95.7	100	95.2	91.7
125	97.6	95.9	70.9	90.3	96.3	77.8	88.5
150	93.3	90.4	82.1	78.6	94.4	65.6	78.1
175	90.3	85.1	89.2	73.9	92.9	57.5	73.2
200	80.5	73.3	93.7	66.7	77.6	46.9	63.0
225	73.3	64.2	93.6	59.4	70.9	44.4	56.6
250	68.0	60.3	94.8	53.5	69.8	43.3	48.3

Function Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	90.9	100	100	100	100	100
125	100	100	100	92.9	100	100	100	94.1	91.7
150	100	100	93.8	88.9	100	84.2	100	88.9	84.6
175	100	100	94.4	90.0	95.0	82.6	95.7	85.7	68.8
200	100	90.5	95.5	95.8	91.7	87.5	83.3	56.5	15.8
225	100	91.3	95.8	88.9	92.6	85.2	66.7	26.9	4.8
250	100	92.6	96.4	75.9	93.1	74.2	45.2	21.4	4.5

Summary of the results obtained from loading rule Number 16. (reduced longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	92.0	93.2	12.1	80.0	75.0	100	100
50	94.0	95.7	25.1	80.0	86.7	90.9	100
75	96.0	97.1	37.6	83.3	87.5	88.2	100
100	94.0	96.3	53.7	73.9	85.7	90.5	83.3
125	92.0	95.2	70.4	71.0	77.8	85.2	69.2
150	82.7	84.1	76.4	52.4	66.7	75.0	59.4
175	80.6	83.2	87.2	50.0	61.9	62.5	56.1
200	74.5	73.8	94.4	40.7	51.0	42.9	45.7
225	71.1	63.1	91.9	40.6	54.5	38.9	45.3
250	68.0	60.2	94.7	36.6	46.0	33.3	44.8

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	33.3	100	100	100	100	100	100	100	100
50	66.7	100	100	100	100	87.5	100	100	100
75	77.8	100	100	100	100	92.3	100	100	100
100	77.8	80.0	100	100	100	85.7	100	100	100
125	60.0	75.0	100	100	100	83.3	100	100	100
150	53.8	50.0	72.2	83.3	89.5	78.9	100	100	100
175	58.8	41.2	72.2	75.0	90.0	82.6	100	95.2	100
200	52.6	47.6	68.2	62.5	87.5	79.2	87.5	95.7	84.2
225	43.5	56.5	66.7	59.3	81.5	81.5	92.6	84.6	66.7
250	40.0	48.1	64.3	62.1	82.8	67.7	90.3	82.1	68.2

Summary of the results obtained from loading rule Number 17. (function, reduced longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	98.7	92.9	35.9	94.4	100	88.2	73.3
100	96.0	95.3	53.1	82.6	95.2	95.2	91.7
125	95.2	94.1	69.5	80.6	96.3	85.2	88.5
150	92.0	87.8	79.8	78.6	91.7	62.5	78.1
175	90.3	85.7	89.9	76.1	85.7	62.5	73.2
200	82.0	71.8	91.9	61.1	77.6	49.0	67.4
225	77.8	62.6	91.3	54.7	74.5	46.3	67.9
250	73.2	58.1	91.4	53.5	66.7	26.7	65.5

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	92.3	100	100	100
100	100	100	100	90.9	100	85.7	100	100	87.5
125	100	100	100	92.9	100	88.9	100	88.2	91.7
150	100	100	100	77.8	94.7	84.2	100	88.9	84.6
175	100	100	100	80.0	90.0	87.0	95.7	85.7	75.0
200	100	95.2	90.9	75.0	83.3	83.3	83.3	78.3	47.4
225	87.0	100	87.5	74.1	77.8	77.8	77.8	73.1	42.9
250	88.0	96.3	85.7	72.4	79.3	67.7	77.4	60.7	22.7

Summary of the results obtained from loading rule Number 18. (fix-code, reduced longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	88.2	100
100	100	100	55.8	100	100	85.7	95.8
125	100	100	73.9	100	100	85.2	92.3
150	96.7	91.6	83.2	92.9	97.2	75.0	81.3
175	91.4	83.3	87.3	91.3	97.6	70.0	58.5
200	84.5	71.7	91.7	87.0	95.9	51.0	76.1
225	80.9	63.5	92.6	81.3	94.5	38.9	67.9
250	75.6	58.3	91.7	76.1	93.7	36.7	60.3

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
125	100	100	100	100	100	100	100	100	100
150	100	100	93.8	94.4	94.7	94.7	100	94.4	100
175	100	100	94.4	95.0	80.0	82.6	100	81.0	93.8
200	94.7	35.7	86.4	83.3	70.8	79.2	100	87.0	73.7
225	91.3	95.7	79.2	74.1	70.4	81.5	92.6	76.9	66.7
250	84.0	88.9	78.6	69.0	69.0	67.7	93.5	67.9	59.1

Summary of the results obtained from loading rule Number 19. (fix-code, function, reduced longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	88.2	100
100	100	100	55.8	100	100	90.5	95.8
125	100	100	73.9	100	100	88.9	92.3
150	96.7	91.6	83.2	92.9	97.2	75.0	93.8
175	92.6	84.3	88.4	91.3	97.6	67.5	82.9
200	86.0	71.4	91.4	88.9	95.9	53.1	76.1
225	80.4	62.4	90.9	81.3	94.5	57.4	67.9
250	75.6	57.3	90.1	78.9	93.7	38.3	44.8

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	100	100	100	100	100	100
125	100	100	100	100	100	100	100	100	100
150	100	100	93.8	94.4	94.7	94.7	100	94.4	100
175	100	100	94.4	95.0	95.0	87.0	100	81.0	81.3
200	94.7	95.2	90.9	83.3	75.0	75.0	100	87.0	73.7
225	91.3	95.7	79.2	81.5	70.4	74.1	100	73.1	57.1
250	92.0	92.6	71.4	79.3	62.1	67.7	93.5	64.3	54.5

Summary of the results obtained from loading rule Number 20. (function, fix-code, reduced longest first)

No. of Requests	% of These Scheduled	% of Time Requested Scheduled	Average Machine Utilis. (%)	% of Fixed Conditions Satisfied		% of Prefered Conditions Satisfied	
				M/C	Time	M/C	Time
25	100	100	12.9	100	100	100	100
50	100	100	26.3	100	100	90.9	100
75	100	100	38.6	100	100	94.1	100
100	99.0	97.5	54.4	95.7	100	95.2	91.7
125	97.6	95.9	70.9	90.3	96.3	77.8	88.5
150	92.0	86.0	78.1	78.6	91.7	62.5	81.3
175	89.7	82.2	86.2	73.9	90.5	55.0	78.0
200	82.5	70.8	90.6	68.5	79.6	42.9	69.6
225	79.1	63.1	92.0	65.6	78.2	46.3	69.8
250	74.4	58.0	91.2	62.0	71.4	41.7	65.5

Function / Number of Requests	1	2	3	4	5	6	7	8	9
25	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
75	100	100	100	100	100	100	100	100	100
100	100	100	100	90.9	100	100	100	100	100
125	100	100	100	92.9	100	100	100	94.1	91.7
150	100	100	93.8	83.3	94.7	84.2	100	88.9	84.6
175	100	100	94.4	85.0	90.0	82.6	95.7	85.7	75.0
200	100	90.5	90.9	87.5	87.5	87.5	87.5	78.3	26.3
225	91.3	100	87.5	85.2	85.2	81.5	81.5	69.2	23.8
250	92.0	96.3	85.7	82.8	86.2	71.0	80.6	50.0	13.6

APPENDIX D

FLOW DIAGRAMS FOR THE PROGRAMS DEVELOPED

The flow diagram for the automatic scheduling program was described in Appendix B. The flow diagrams for the other nine programs developed are given in this Appendix.

These are;

(a) on-line programs

i. Program 'INPU'
ii. Program 'AMEN'
iii. Program 'FIND'
iv. Program 'STAT'
v. Program 'UPDA'
vi. Program 'SCHA'

(b) terminal-initiated batch programs

vii. Program 'ANAL'
viii. Program 'PRIN'
ix. Program 'LISTO'

(a) On-line Programs

i. Program 'INPU'

This program reads the relevant data concerning requests for machine time from a VDU, checks it for errors and inconsistencies and writes the details to the data files.

The elements in the program flow chart illustrated are as follows:

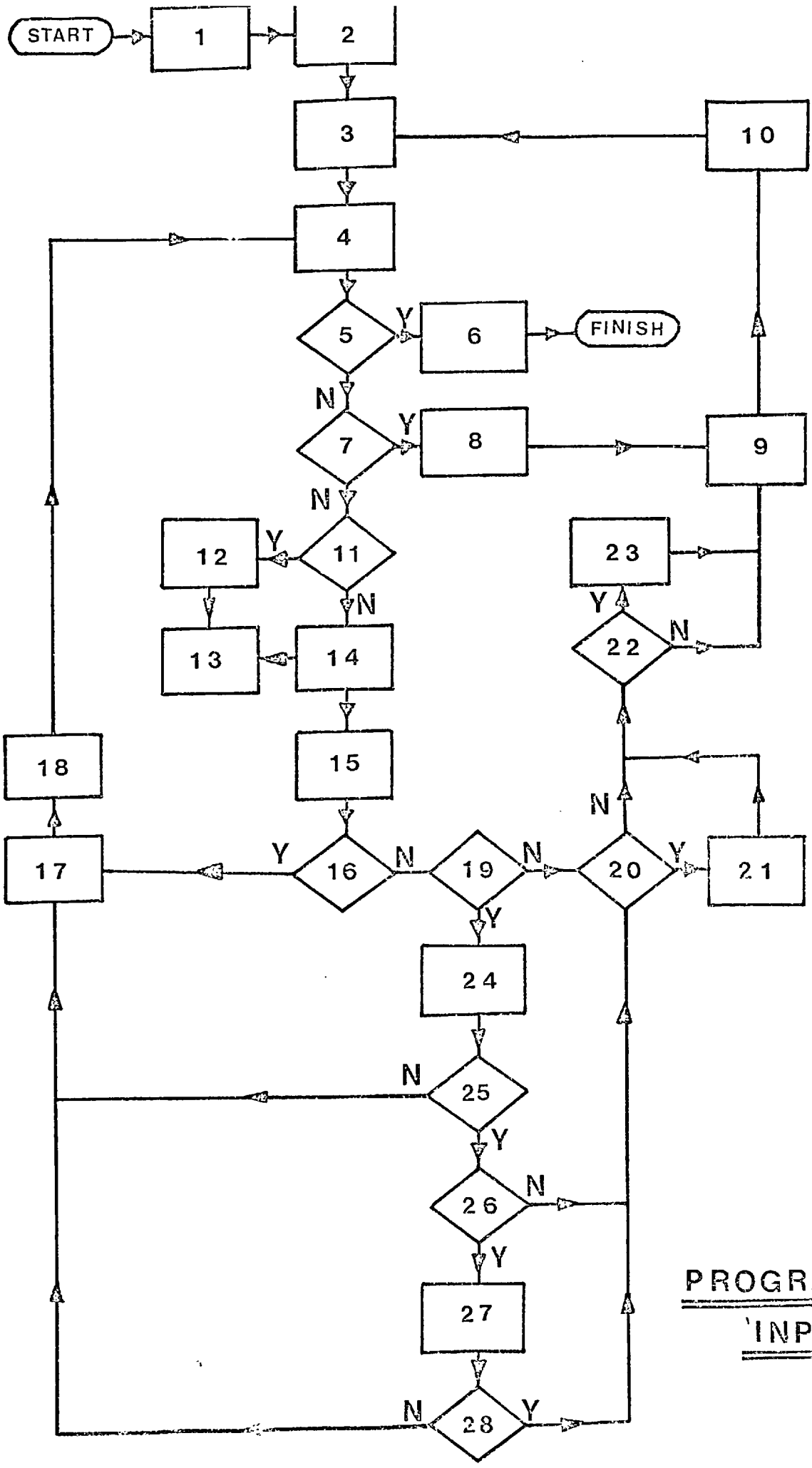
START

1. Defines the files to be used, input and output channels, etc, and writes an introductory message.
2. Reads the number of the last request entered from the files and adds one to give the next request number.
3. Prints out the form to be completed with the input data.
4. Reads the type of request entered (for block bookings etc), together with the number and frequency of repeats for block bookings.
5. Was a message given for the program to end at this point? (yes go to element 6, No go to element 7).
6. Writes the number of requests entered to the VDU together with a terminating message, and stores the number of the last request on the files.

FINISH

7. Is the request to be entered an exact duplicate of the last request entered (Yes - 8, No - 11).
8. Reads details of the last request entered from the files.
9. Writes these details onto the files as the present request
10. Resets the values, stored temporarily, for the next and last request numbers, and adds one to the number of requests entered.

11. Is the 'fixed data' (i.e. the request title, project number etc.) for this request the same as for the last request entered? (Yes - 12, No - 14)
12. Reads the fixed data associated with the last request from the data files.
13. Writes this fixed data and the remainder of the input form for the variable data to the VDU
14. Reads all the data input for this request.
15. Checks the input data for errors
16. Are there any errors in the input data? (Yes - 17, No - 19)
17. Writes an appropriate error message giving the number and type of errors found.
18. Writes the data input for the request onto the VDU to be corrected by overwriting.
19. Is the request given a fixed or preferred start time and/or a fixed or preferred machine number? (Yes - 24, No - 20).
20. Is the request a block booking? (Yes - 21, No - 22)
21. Writes a message to confirm the block booking.
22. Is the request a low priority one? (Yes - 23, No - 9)
23. Writes a message to confirm the low priority request.
24. Compares the fix-code with the start time and machine number input
25. Is the fix-code compatible with the input data? (Yes - 26, No - 17)
26. Is the request to be scheduled at a fixed time on a fixed machine? (Yes - 27, No 20)
27. Sees if the fixed space given is free in the schedules and schedules the request if it is.
28. Has the request been scheduled? (Yes - 20, No - 17)



PROGRAM 'INPU'

(ii) Program 'AMEN'

This program deletes, restores and alters information concerning requests already stored in the data files.

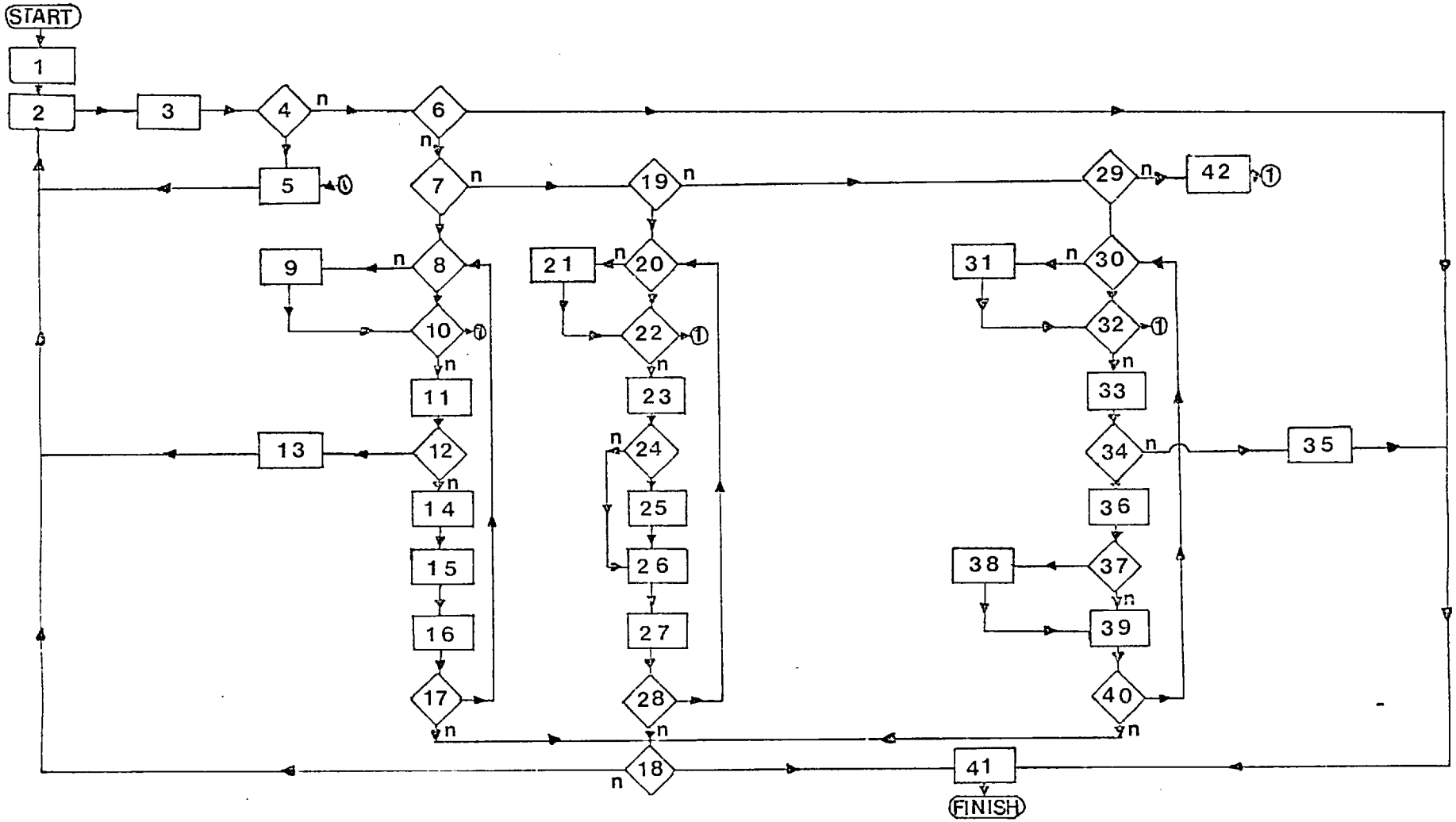
In the flow diagram illustrated the elements are as given below.

START

1. Sets-up the initial conditions for the program.
2. Reads the function wanted and the request number
3. Calculates the number of requests in the file.
4. Is the request number given greater than the number of requests in the files? Yes - 5, No - 6
5. Writes an error message.
6. Is the program to finish? Yes - 41, No - 7
7. Is a request to be altered? Yes - 8 , No - 19
8. Has a request number been given? Yes - 10, No - 9
9. Asks for and reads a valid request number
10. Is this request number greater than the number of requests in the files? Yes - 5, No - 11
11. Reads all the information for the given request from the data files.
12. Is the request labelled as being deleted? Yes - 13, No - 14
13. Writes an error message
14. Writes all the available information for the request onto the VDU
15. Reads the corrected information from the VDU.
16. Updates the data files with this information.
17. Are there any more requests to be altered? Yes - 8, No - 18
18. Is the program to end? Yes - 41, No - 2

19. Is a request to be deleted? Yes - 20, No - 29
20. Has a request number been given? Yes - 22, No - 21
21. Asks for and reads a valid request number.
22. Is this request number greater than the number of requests in the data files? Yes - 5, No - 23
23. Reads the variable data relating to the request from the data files.
24. Has the request been scheduled? Yes - 25, No - 26
25. Deletes the request from the bar-chart record of the schedules.
26. Marks the request as 'deleted'
27. Writes these updatings to the data files.
28. Are there any more requests to be deleted? Yes - 20, No - 18
29. Is a deleted request to be replaced? Yes - 30, No - 42
30. Has a request number been given? Yes - 32, No. 31
31. Asks for and reads a valid request number.
32. Is this request number greater than the number of requests in the data files? Yes - 5, No - 33
33. Reads all the information concerning the request from the data files.
34. Is the request marked as being deleted? Yes - 36, No - 35
35. Writes an error message
36. Writes all the information concerning the request to the VDU
37. Is the request to be scheduled at a fixed time on a fixed machine? Yes - 38, No - 39.
38. Writes a message that program 'SCHA' should be called upon for this request to be scheduled.
39. Updates the data files with the new information.
40. Are there any more requests to be replaced? Yes - 30, No - 18.

41. Writes a terminating message.
    FINISH
42. Writes an error message as all possible functions of the program have been checked.



PROGRAM 'AMEN'

(iii) Program 'FIND'

The purpose of this program is to search the data files and find all those requests which have certain named characteristics in common. These are shown on a VDU.

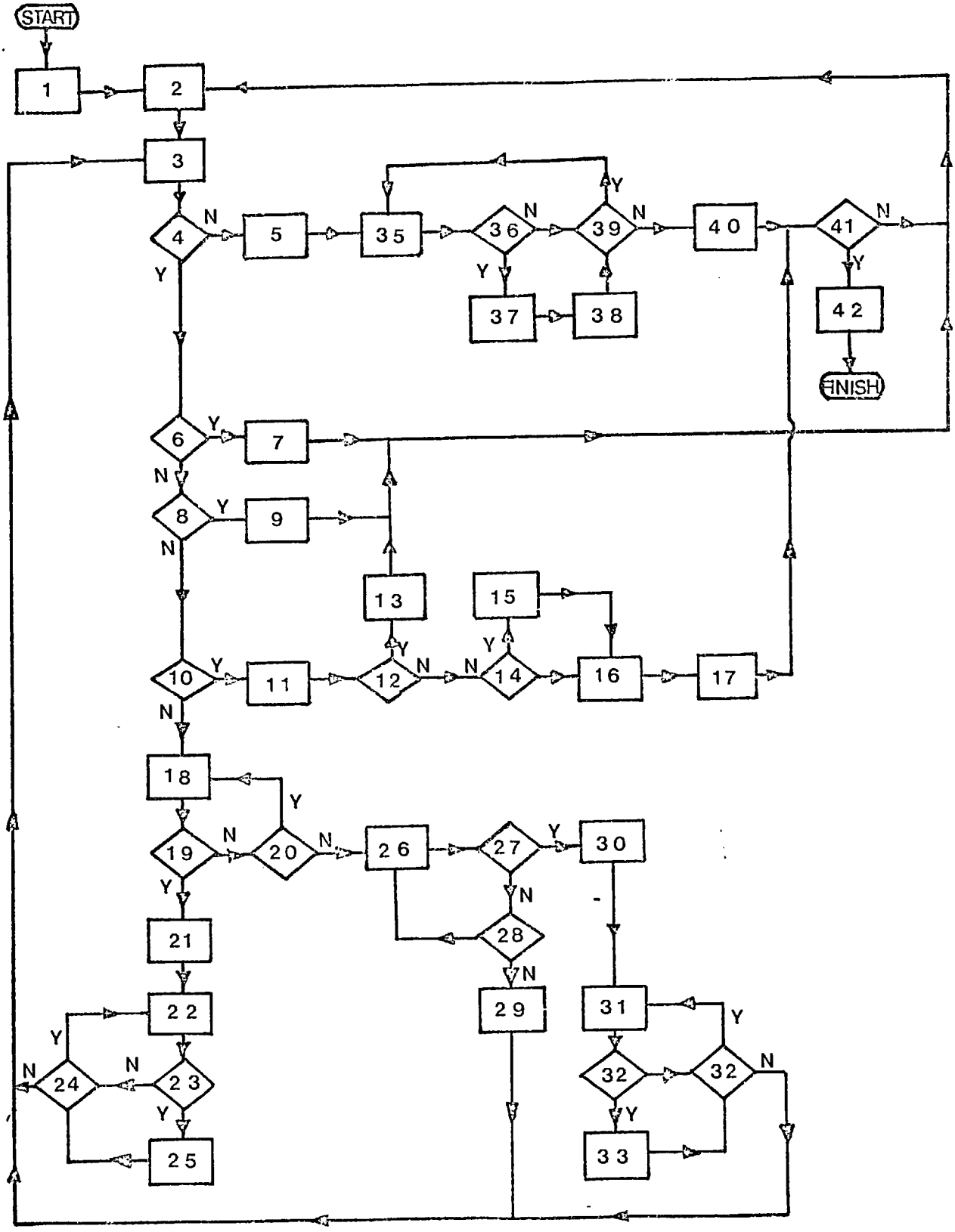
In the flow diagram illustrated the elements are:

START

1. Sets-up the initial conditions for the program.
2. Writes a heading to ask for input.
3. Reads the field to be searched and the value wanted
4. Are there any more searches in this particular series? Yes - 6, No - 5
5. Writes a message giving the number of searches made etc.
6. Are all the requests in the file to be listed? Yes - 7, No - 8
7. Writes a message to use program 'LISTO' for this information.
8. Was the field to be searched omitted? Yes - 9, No - 10
9. Writes a message that the input was blank.
10. Is a request asked for by number? Yes - 11, No - 18
11. Calculates the number of requests in the data file.
12. Is the number of the request asked for greater than the number of requests in the files? Yes - 13, No - 14
13. Writes an error message.
14. Was the number omitted? Yes - 15, No - 16
15. Asks for and reads a valid request number.
16. Reads all the information concerning this request stored in the data files.
17. Writes this information to the VDU.
18. Takes the next field in the file containing 'fixed data' for requests.
19. Is this the field given in element three? Yes - 21, No - 20.
20. Are there any more fields in this file? Yes - 18, No - 26.

21. Sets the search area wanted in the file.
22. Takes the next request.
23. Was this request marked in all the previous searches (if any) in this series, and does it have the value wanted in the search field? Yes - 25, No - 24
24. Are there any more requests in the file? Yes - 22, No - 3
25. Marks the request with the present search number.
26. Takes the next field in the file containing 'variable data' for requests.
27. Is this the field given in element three? Yes - 30, No - 28.
28. Are there any more fields in this file? Yes - 26, No - 29.
29. Writes an error message.
30. Sets the search area wanted in the file.
31. Takes the next request.
32. Was this request marked in all the previous searches (if any) in this series, and does it have the value wanted in the search field? Yes - 33, No - 34.
33. Marks the request with the present search number.
34. Are there any more requests in the file? Yes - 31, No - 3
35. Takes the next request in the file.
36. Was it marked in all the searches in this series? Yes - 37, No - 39.
37. Reads all the information concerning this request from the files.
38. Writes this information to the VDU.
39. Are there any more requests in the file? Yes - 35, No - 40.
40. Writes a message and resets the markers in the files.
41. Is the program to finish now? Yes - 42, No - 2
42. Writes a terminating message.

FINISH



PROGRAM 'FIND'

(iv) Program 'STAT'

This program is a corollary to Program 'FIND', in that information about the number and times of requests asked for and scheduled with each function or fix-code are fed back to the schedulers. Machine utilisation can also be found.

The elements in the following flow diagram are:

START

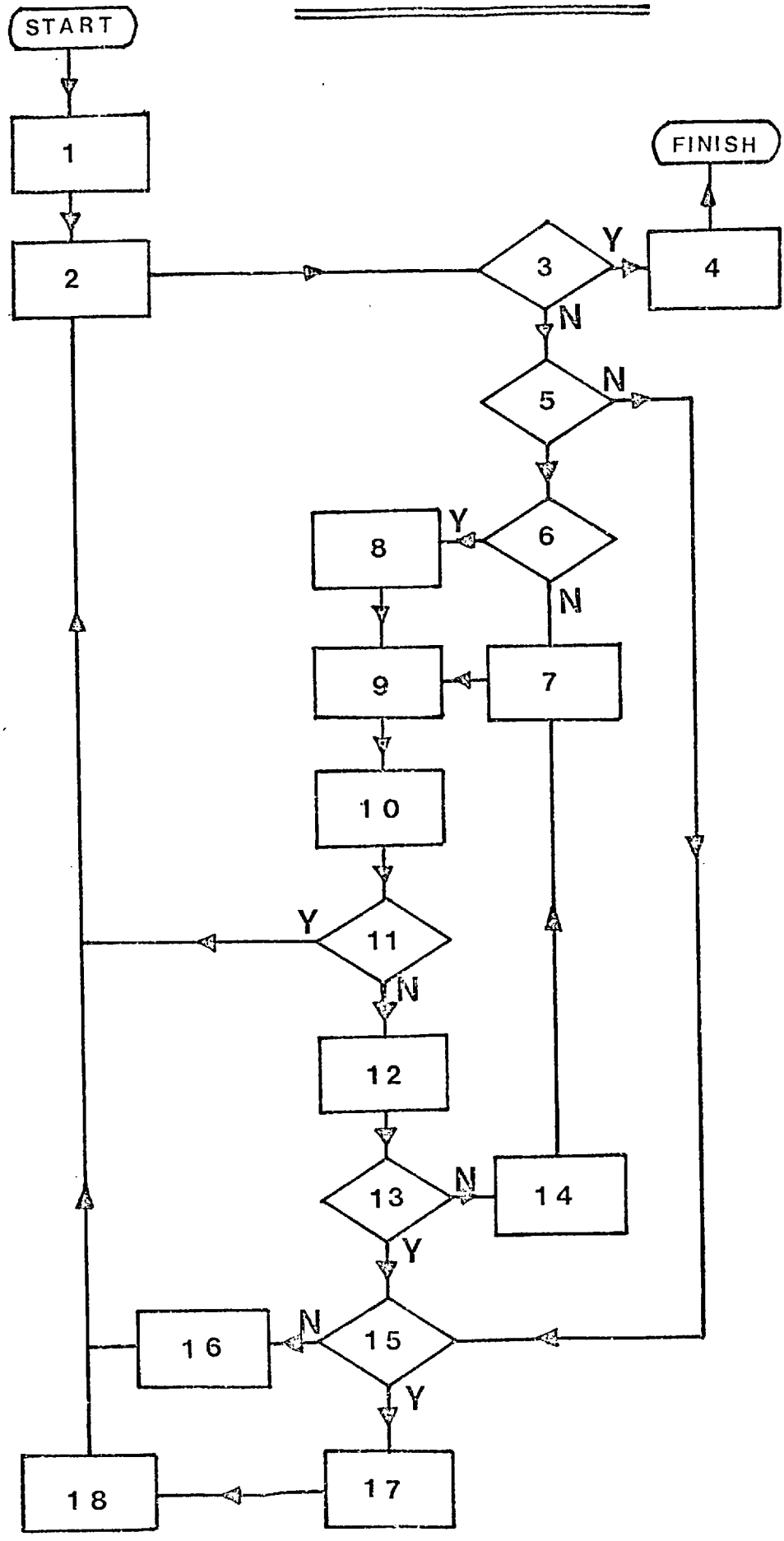
1. Sets-up the initial conditions for the program and writes an introductory message.
2. Reads the schedulers requirements for the program.
3. Is the program to finish at this point? Yes - 4, No - 5
4. Writes a terminating message.

FINISH

5. Is data wanted for each function or fix-code, or is all available information wanted? Yes - 6, No - 15
6. Is all the information wanted, or information for each function? Yes - 8, No - 7
7. Sets the program to find information for each fix-code.
8. Sets the program to find information for each function.
9. Calculates the total number of requests and the times of requests with each of the set function/fix-code, and finds how many of these have been scheduled.
10. Writes the results to the terminal.
11. Was information demanded for only one of function/fix-code? Yes - 2, No - 12.
12. Resets the relevant variables to zero.
13. Has information for each fix-code been sent to the terminal? Yes - 15, No - 14
14. Sets the program so that machine utilisation is demanded when the information for each fix-code has been given.
15. Is information about the machine utilisations demanded? Yes - 17, No - 16

16. Writes an error message for an incorrect instruction.
17. Calculates the utilisation for each machine.
18. Writes these results to the terminal.

PROGRAM 'STAT'



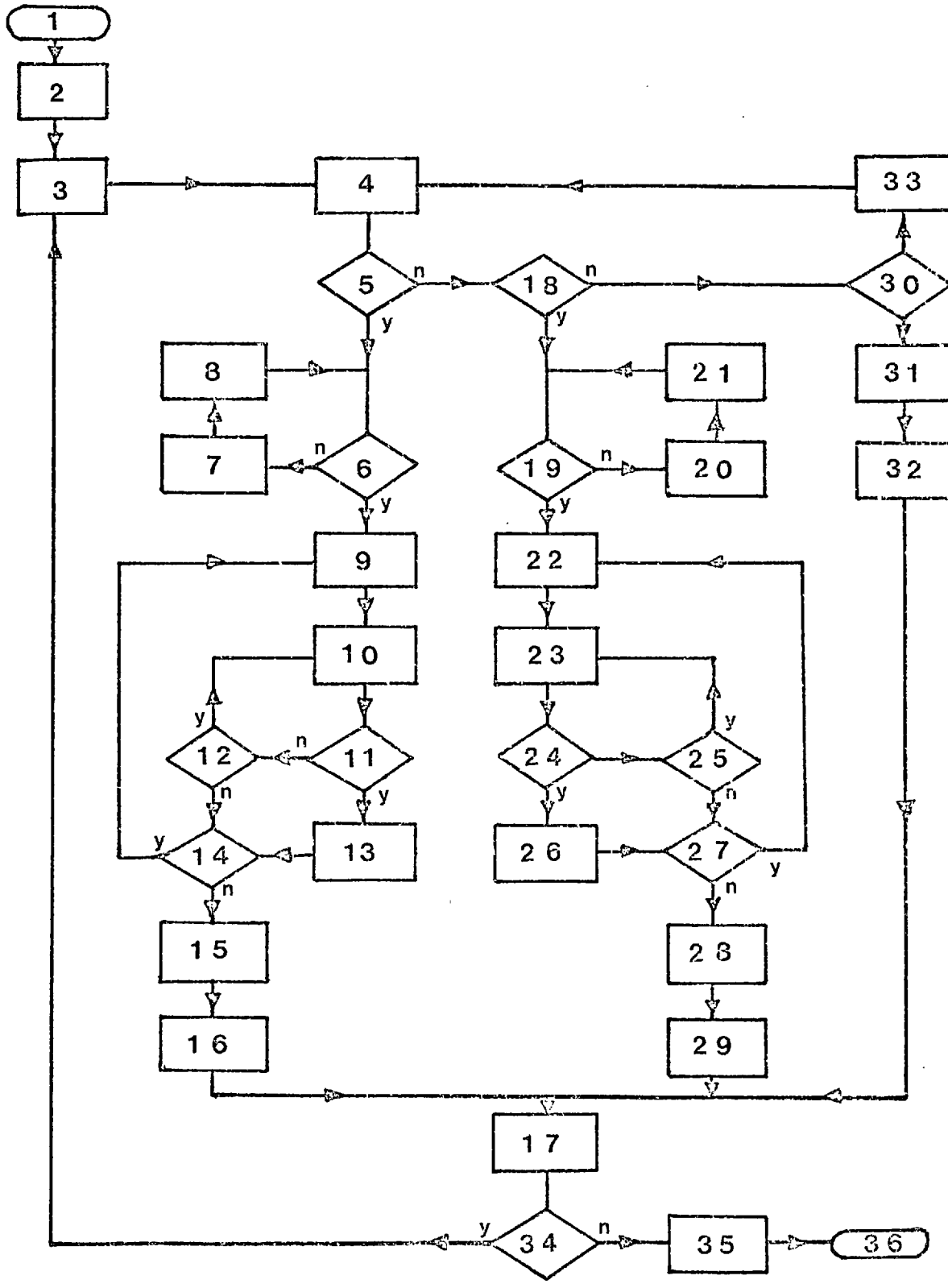
(v) Program 'UPDA'

This program alters, adds to or deletes from the lists of machines associated with each function.

In the flow diagram the elements are as follows:

1. START
2. Sets-up the initial conditions for the program.
3. Creates a matrix containing the machines associated with each function, in the order of preference, as they stand at present.
4. Reads an instruction from the VDU.
5. Is a machine to be deleted from the lists? Yes - 6, No - 18
6. Has the machine number been given? Yes - 9, No - 7.
7. Write a request for the machine number to be entered.
8. Reads the number entered.
9. Takes the next function.
10. Takes the next machine number associated with this function.
11. Is this the machine number to be deleted? Yes - 13, No - 12
12. Are there any more machines associated with this function? Yes - 10, No - 14
13. Deletes the machine number and moves the other numbers to fill the gap.
14. Are there any more functions to be considered? Yes - 9, No - 15.
15. Writes a message to the VDU.
16. Writes the revised list of machines to the VDU for verification.
17. Updates the data files with the new information.
18. Is one machine number to be replaced by another? Yes - 19, No - 30.
19. Are both machine numbers given correctly? Yes - 22, No - 20.

20. Writes a request for the machine numbers to be entered from the VDU.
21. Reads the machine numbers entered.
22. Takes the next function.
23. Takes the next machine number associated with this function.
24. Is this the machine number which is to be replaced? Yes - 26, No - 25
25. Are there any more machine numbers associated with this function? Yes - 23, No - 27.
26. Replaces the old machine number with the new one.
27. Are there any more functions to be considered? Yes - 23, No - 28.
28. Writes a message to the VDU.
29. Writes the revised list of machines to the VDU for verification.
30. Is a summary of the machines associated with each function wanted? Yes - 31, No - 33.
31. Writes this information to the VDU.
32. Reads the revised machine order from the VDU, if any alterations have been made.
33. Writes an error message for an incorrect instruction.
34. Are there any more changes to be made? Yes - 3, No - 35.
35. Writes a terminating message.
36. FINISH



PROGRAM 'UPDA'

(vi) Program 'SCHA'

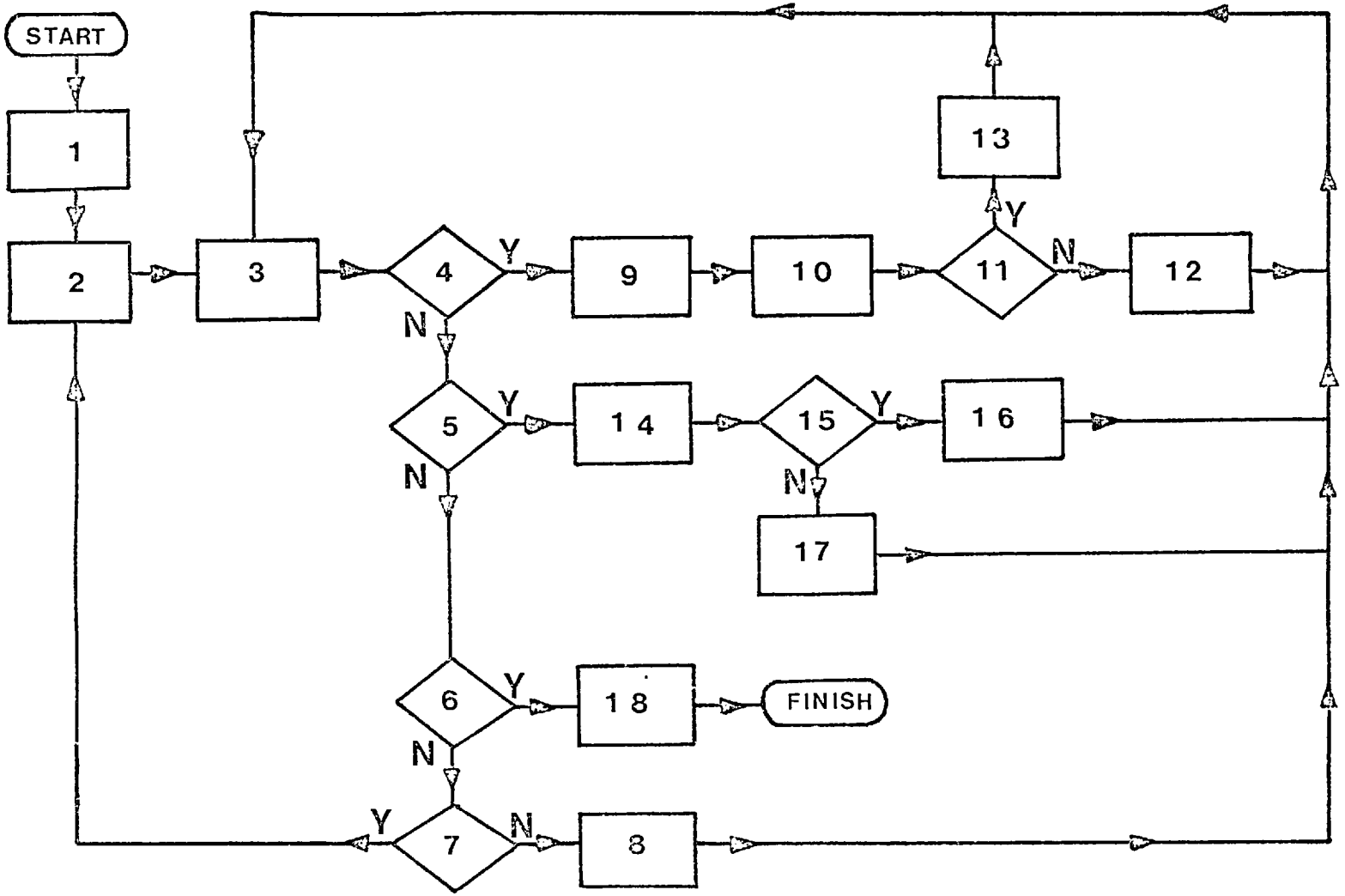
This program makes alterations to the schedules which have been automatically produced by the program 'SCHI'. The operator directs the program to add and subtract requests as required.

The elements in the flow diagram are as follows:

START

1. Set-up the initial conditions for the program.
2. Displays on the VDU the schedules for those machines which have been specified by the scheduler
3. Reads the next function wanted in the program
4. Is a request to be added to the schedule? Yes - 9, No - 5
5. Is a request to be subtracted from the schedule? Yes - 14, No - 6.
6. Is the program to finish at this point? Yes - 18, No - 7
7. Are any more machine schedules to be displayed on the VDU? Yes - 2, No - 8
8. Writes an error message for an incorrect instruction.
9. Reads the relevant data for the request from the data files, and writes a message to the VDU if a special code is present.
10. Checks the information present for the request, and asks for any which is missing, and determines if the request can be scheduled.
11. Can the request be scheduled at the specified place? Yes - 13, No - 12
12. Writes a message that the request cannot be scheduled in the place specified.
13. Schedules the request in the gap, updates the data files and writes the altered schedule to the VDU for verification.
14. Reads the information concerning the request from the data files.

15. Has the request been scheduled? Yes - 16, No - 17
16. Deletes the request from the schedules, updates the data files and writes the new schedule to the VDU for verification.
17. Writes a message that the request has not been scheduled.
18. Writes a finishing message to the VDU.
    FINISH



PROGRAM 'SCHA'

(b) Terminal-Initiated Batch Programs

(vii) Program 'ANAL'

This program analyses the results of the schedules produced by program 'SCHI'. The numbers and times requested and scheduled are calculated for each function and fix-code, machine utilisations etc. are also given.

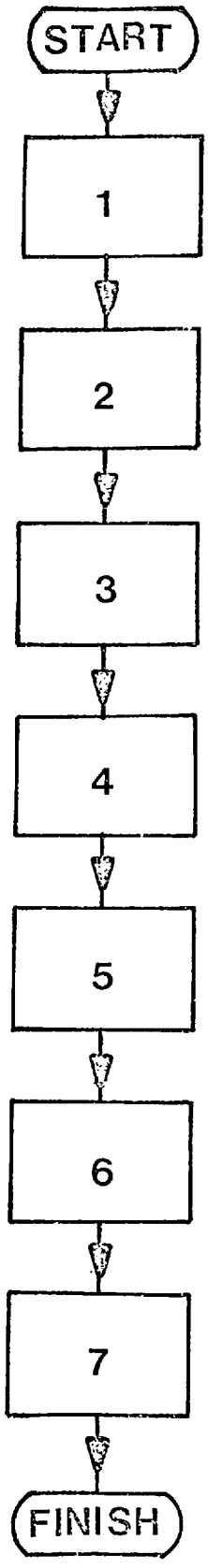
The scheduled data can either be compared with the original input data, or with the slightly modified data given after the scheduling run (for instance a request for a m/c pair may be altered to two requests, each on one machine).

The elements in the flow diagram are as follows:

START

1. Sets-up the initial conditions for the program.
2. Reads all the information given for the input data and calculates the number of request and times required for each type of request.
3. Reads all the information given for the scheduled data and calculates the number of request and times scheduled for each type of request.
4. Calculates and prints the results of the scheduling for each function.
5. Calculates and prints the results of the scheduling for each fix-code.
6. Calculates and prints the machine utilisations.
7. Calculates and prints the totals (numbers and times) for the scheduling run.

FINISH



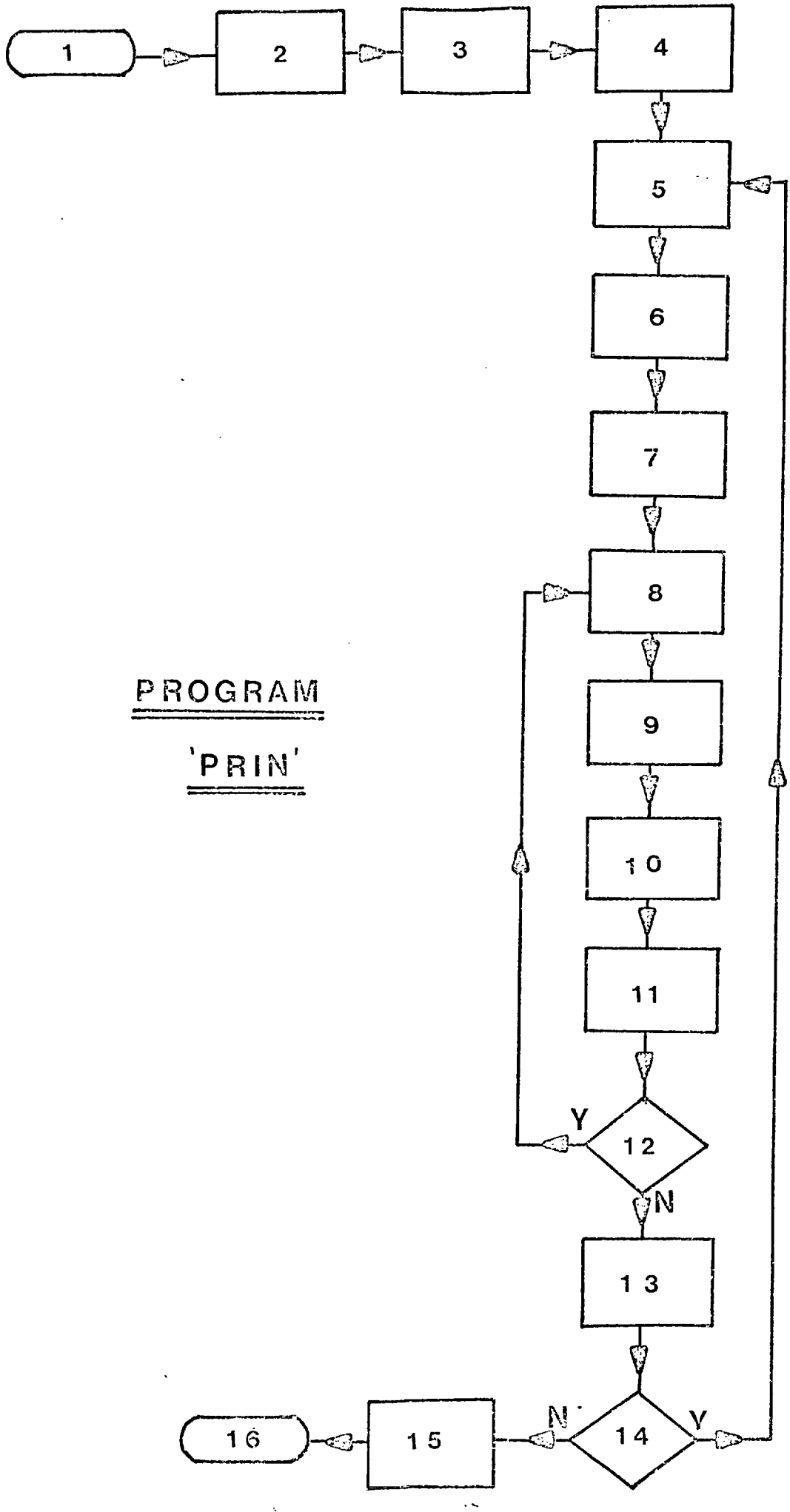
PROGRAM 'ANAL'

(viii) Program 'PRIN'

This program prints the schedules produced in program 'SCHI', and stored in a data file in the form of bar charts.

In the flow diagram given for the program the elements are as follows:

1. START
2. Sets-up the initial conditions for the program.
3. Generates an array of the machines and requests scheduled on them.
4. Writes a general heading for the output.
5. Takes the next machine to be considered.
6. Writes a heading, time scale etc. for this machine.
7. Arranges the requests scheduled on this machine in the order earliest first.
8. Takes the next request scheduled on this machine.
9. Calculates the start, duration, line-up and finish times in terms of ten minute elements starting at 0900.
10. Sets the values of the vectors containing information for this request.
11. Writes these vectors into the output file.
12. Are there any more requests scheduled on this machine? Yes - 8, No - 13
13. Writes a time scale, indicates the unused time etc.
14. Are there any more machines to be considered? Yes - 5, No - 15
15. Lists the uncheduled requests in numerical order.
16. FINISH



PROGRAM

'PRIN'

(ix) Program 'LISTO'

This program lists all the requests stored in the data files for a particular day, the lists being in both numerical and alphabetical order.

In the flow diagram given the elements are as follows:

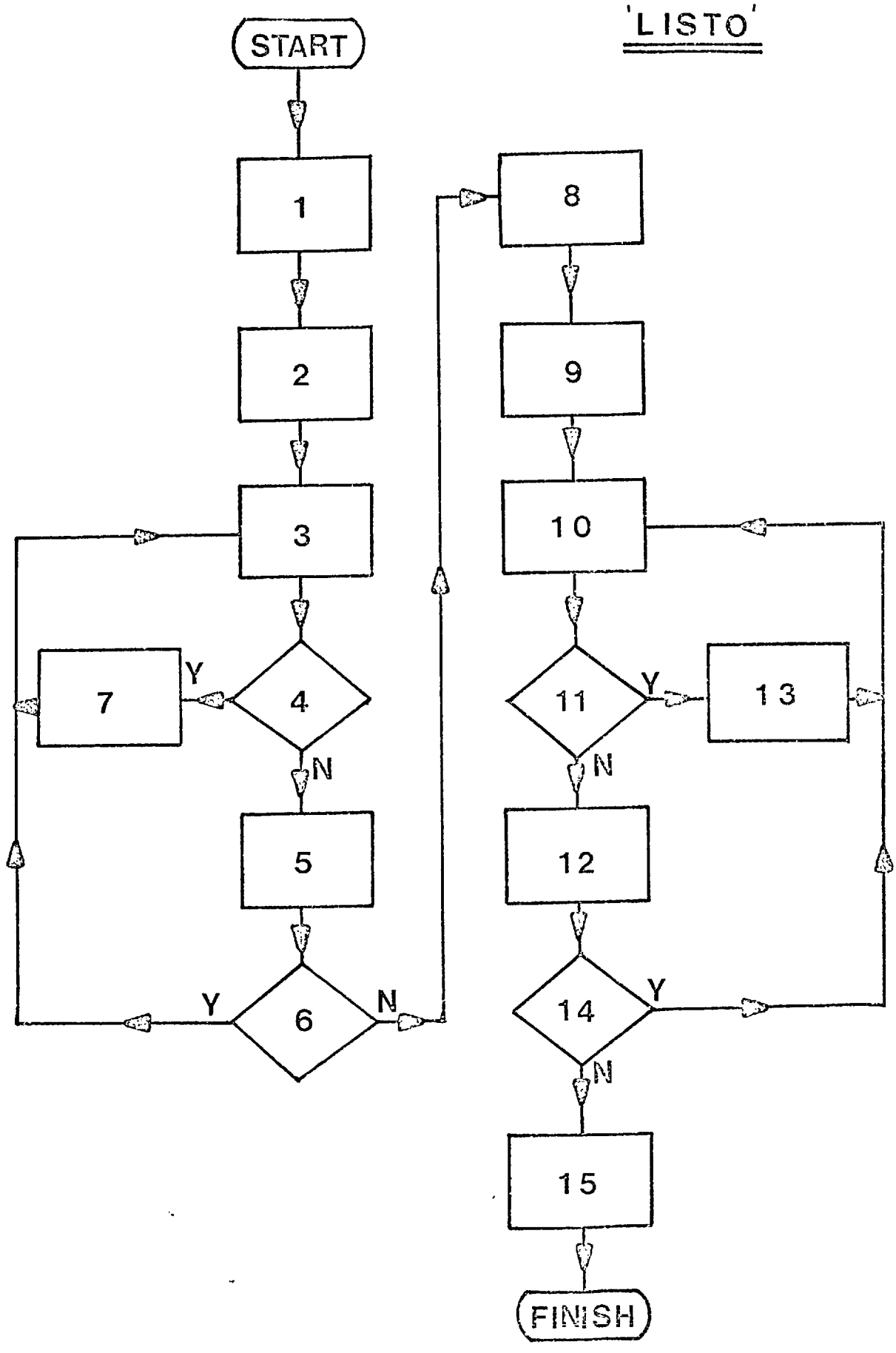
START

1. Sets-up the initial conditions for the program
2. Writes a heading for the numerical listing.
3. Takes the next request.
4. Is this request blank or deleted? Yes - 7, No - 5
5. Writes all the information available for this request into the output file.
6. Are there any more requests in the files? Yes - 3, No - 8
7. Ignores this request.
8. Writes a heading for the alphabetical listing.
9. Arranges the requests in alphabetical order.
10. Takes the next request
11. Is this request blank or deleted? Yes - 13, No - 12
12. Writes all the information available for this request into the output file.
13. Ignores this request.
14. Are there any more requests in the files? Yes - 10, No - 15
15. Writes a terminating message to the VDU.

FINISH

PROGRAM

'LISTO'



TOTAL LENGTH

Introductions	7 pages
Text	202 pages
Appendices etc	61 pages
Total	270 pages