Oxford University Computing Laboratory

Computer Manuals

ICL 1900 Scientific and Technical Software
Vehicle (Bus) Scheduling Mk.1

4226

SCIENTIFIC & TECHNICAL SOFTWARE

Vehicle Scheduling

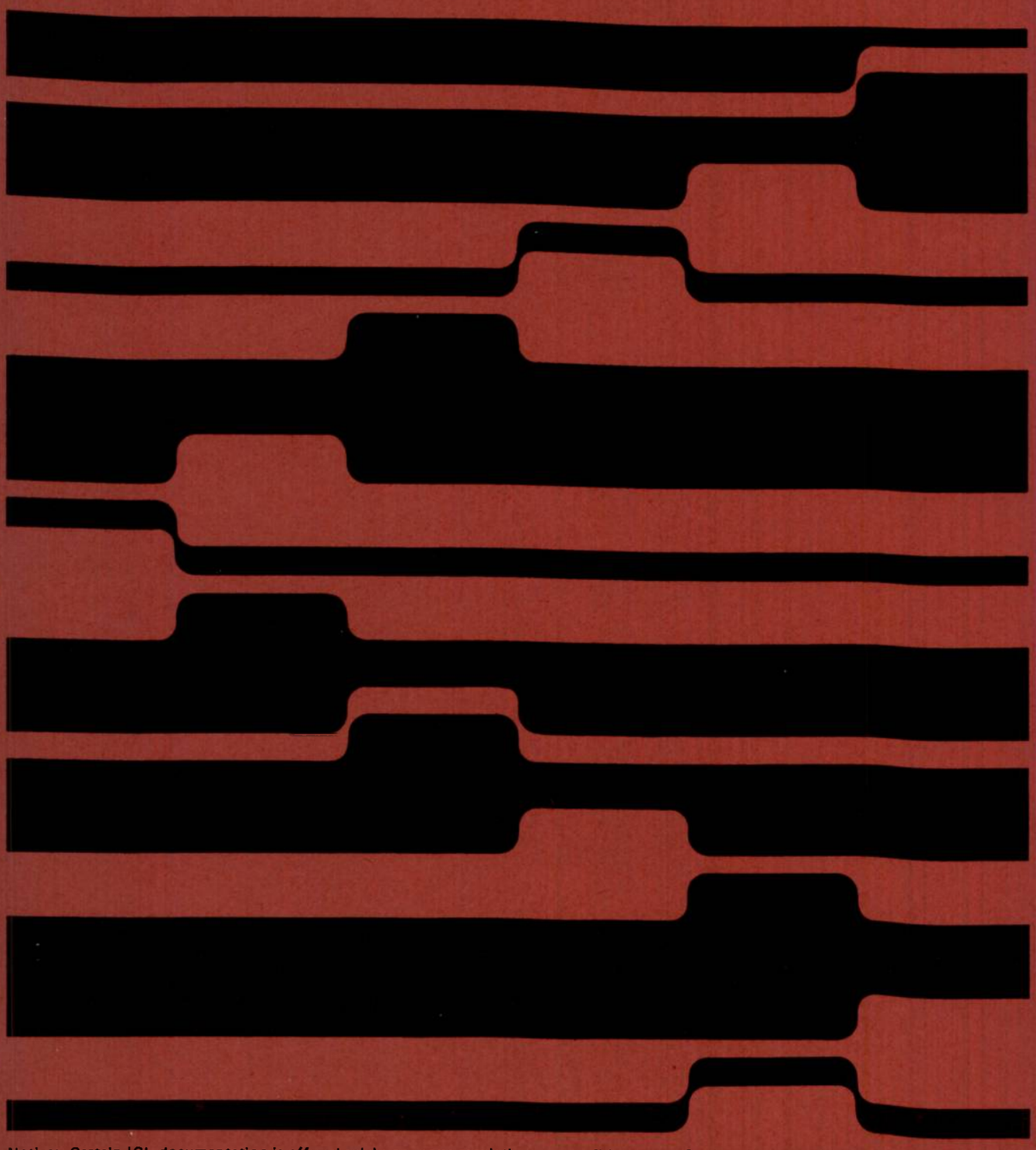
Bus Scheduling Mk.1

1900

ICL

Bus Scheduling Mark 1

1900 Series



ICL International Computers Limited

User notice

Statistical

MANUAL (NOTICE NO.) 25/11/70

4226 BUS SCHEDULING MARK 1 (1)

File one copy of this notice with each of the manuals indicated.

OUTPUT EXAMPLE

The example of line printer output on page 20 of the manual has been incorrectly reproduced. The second line of BOARD 1 should read:

572 DEP FROM 0 17.40 ARR AT 4 17.49 TTT=2 ELT=3

Issued by Technical Publications Service, 30 - 31 Friar Street, Reading, Berkshire. RG1 1JP

ICL International Computers Limited

User notice

MANUAL (NOTICE NO.)

30/12/70

4226

BUS SCHEDULING MARK 1 (2)

File one copy of this notice with each of the manuals indicated.

BUS SCHEDULING #X3MU/2

This new issue of the program corrects the error mentioned in Software Notice 1900/OTHER ENGINEERING APPLICATIONS AND MISCELLANEOUS/19, item 29.

FORM 1/230/45(3.69)

ICL

Bus Scheduling Mark 1

1900 Series

A

Technical Publication 4226

First Edition September 1970

Issued by Technical Publications Service
International Computers Limited
Head Office: ICL House, Putney, London SW15
Produced by ICL Printing Services
at Letchworth, Hertfordshire

Preface

The ICL 1900 Series Bus Scheduling Mark 1 program described in this manual will determine the minimum number of buses required to operate a given timetable, and will produce a detailed schedule of the movements of each bus.

The minimum machine configuration required is:

1900 Series central processor with 16K words of core store

1 80 column card reader or 1 8 track paper tape reader

1 line printer with 120 character positions.

This manual is intended to allow the prospective user to assess the applicability of the program to his needs, and to allow him to make full use of it without any knowledge of computers. Chapter 1 gives an introduction to the facilities of the program. Chapter 2 gives a detailed description of the data which the user is required to provide; Chapter 3 describes the results he will obtain and the facilities for detecting data errors. Chapter 4 is intended primarily for the use of the installation's computer operators and gives information on how to run the program. Chapter 5 gives an example of a run of a small problem. Certain limitations on the size of the problems that the program can handle are given in Appendix 1.

Although this manual refers to buses throughout, the program may also be applied to similar problems associated with railway engines, trains, or other public transport vehicles.

Contents

Illustrations

Chapter 1 Introduction

The main aim of any transport operator is to keep his costs down while maintaining a good service to the public. The program described in this manual is designed to assist in the achievement of this aim; it does this by calculating the minimum number of buses required to operate a given timetable without any reduction of the service. Associated with this optimum number of buses, a schedule of the individual bus movements is produced.

Use of the program can lead to a reduction of operating costs in two ways. Firstly, given a large timetable, it may not be possible by manual methods to compute the actual minimum number of buses required, since the computation is complex, excessively laborious and time-consuming for human beings. Secondly, the program is so fast that it enables the user to investigate the effects of many more variations of the basic problem in a much shorter space of time than is possible by purely manual methods. Thus further possible reductions of the transport fleet can be investigated quickly by making small changes to a proposed timetable. The program is therefore seen as a powerful tool at the disposal of the human bus scheduler, enabling him either to reduce costs or to improve the service.

The size of fleet required by any transport operator is determined by his peak period requirements, and so it is expected that the program will be used mainly for these periods. However, longer periods of up to several days may be investigated should the user require to do so (subject to the various restrictions given in Appendix 1, page 21).

METHOD USED BY THE PROGRAM

To use this program it is not necessary to have any knowledge of computers or of programming; nor is it necessary to have a detailed knowledge of the calculations by which the program reaches its results. The brief outline description of the method which follows should be sufficient to allow the user to take full advantage of the facilities offered by the program. If he is experimenting with a number of different timetables, this description should help him to see what modifications in his timetable are likely to be worth trying out on runs of the program.

The maximum possible number of buses that could be used in scheduling a given timetable would be achieved by assigning a different bus to each journey on the timetable. The program reduces this maximum number by forming *links* between journeys that can be assigned to the same bus. After completing its first journey, a bus may start a new journey from the same terminus. Alternatively it could be moved to a new terminus in an *out-of-service transfer* and then begin its next passenger-carrying journey there. Each link made in one of these ways reduces the total number of buses required by one, and the method used by the program involves calculating which of various possible schedules uses the maximum number of links (and therefore the minimum number of buses).

The way in which this link maximization can be achieved is not normally unique: several different schedules each using the same minimum number of buses could usually be produced. The program prints out one of these possible schedules and gives for each bus a list of passenger-carrying journeys together with details of the time spent by the bus in out-of-service transfers and in any waiting between journeys.

DATA REQUIREMENTS

As mentioned above, the user must supply the timetable he requires to have scheduled. In order to allow the program to schedule out-of-service transfers, it is also necessary to provide a list of the times taken to move buses between all pairs of termini. Alternatively, if the user does not wish to permit any out-of-service transfers between certain specified pairs of termini, he is provided with a facility to ban them. By banning many transfers in this way the user may reduce the efficiency of the final schedule, and so he should keep such banned linkages to the minimum that he considers essential.

The only other data that the user is required to provide are four parameters which control the running of the program. Two of these allow the program to calculate the amount of space that will be required to solve the problem; the other two are concerned with output and with the banning of out-of-service transfers as specified in the previous paragraph. The data is all described in greater detail in Chapter 2.

In general the user can arrange matters in such a way that the information which he is required to input about out-of-service transfers is constant and can be used for several runs of the computer with a different timetable each time.

RUNNING AND OUTPUT

The information required to operate the program is given in Chapter 4. The program reads the data that has been submitted and then prints output as described in Chapter 3. An example of this output for a successful run is given in Chapter 5 and is principally intended to illustrate the output format.

The most important section of the output resulting from a successful run of the program is the *board list*. The *board* is the name given to the sequence of journeys allocated to one bus. The starting point and destination of each journey are listed, together with the departure and arrival times; the times taken up by out-of-service transfers and waiting are also listed.

Other forms of output which may be requested from a successful run are also mentioned in Chapter 3. If the program is unable to run successfully, error messages will be output which will help the user to discover the cause of failure.

LIMITATIONS

The size of problems that can be solved depends on the computer in use. The problem size depends on the number of termini which may be used and the number of journeys that appear in the timetable. For example, if the minimum size of central processor (16K) is used, problems involving 150 termini and 200 journeys or 50 termini and 300 journeys could be solved. A central processor of 48K could solve a problem of 250 termini and 500 journeys.

These and other limitations on the data that can be handled are gathered for ease of reference in Appendix 1, which is of great importance to the intending user. In particular he should take note of Figures 4 and 5, which are described in the Appendix. These Figures show the limitations on numbers of termini and journeys in graphical form. The user is advised to check his data against these graphs before running the program to ensure that his problem is not too large for his computer. The Figures are only approximate; and if his problem comes close to being too large he should use the formulae for problem size which are also quoted in Appendix 1.

SUMMARY

The object of the program is to allow the transport operator to reduce his fleet of buses while maintaining his existing timetable. Alternatively, he may be able to conduct experiments to find a better timetable and to provide a better service to the public without increasing his fleet. These uses depend on the computer's ability to calculate an optimal schedule for a given timetable. A human operator could perform the same calculations, but to do so would take an unreasonable time; thus in all but very small problems a manually produced schedule will be non-optimal. The ability to reach optimal schedules in a few minutes of computer time makes it reasonable to try out the effects of experimental timetables.

If, as is probable, the user is scheduling a period of time during the day (for example one of the peak periods), he will have to consider the interface between the schedule as produced by the program and any schedule he may have for that part of the day preceding the period considered by the program. The data input does not include details of the whereabouts of buses at times before the beginning of the run, and so the user will have to ensure that he can arrange to send buses to the starting point of each board in time for the departure of the first journey.

Chapter 2 Input

The first part of this Chapter describes in general terms the data required to run the program. Suitable data sheets for submitting the data for punching are also shown. A description of the punching conventions that must be followed when preparing the data is given at the end of the Chapter, on page 9.

GENERAL CONSIDERATIONS

The data consists of three blocks of information:

1 the parameters

2 the timetable of journeys

3 the out-of-service inter-termini times

which must be input in that order. These three blocks are considered separately later in this Chapter on pages 5, 7 and 7 respectively.

There are also three important preliminary considerations with regard to the data:

1 Identifying code numbers must be provided for termini and timetable journeys (see below).

2 Times must be provided in a standard way (see below).

3 The problem should not be too large for the computer to solve (see page 5).

Terminus numbers and journey numbers

Each terminus and each passenger-carrying journey on the timetable must be identified for the purposes of the program by a code number which the user must allocate.

TERMINUS NUMBERS

The code number for each terminus must be a unique non-negative integer. It is preferable, but not essential, that these numbers should start at zero and that they should be consecutive, since this results in the minimal core store usage. In the out-of-service inter-termini times block data must be included corresponding to all terminus numbers from zero to the highest terminus number used, whether or not the numbers correspond to termini actually used in the problem that is being solved.

It is recommended that all the termini which the user might wish to include in his timetables should be numbered. A constant inter-termini time block may then be produced and used for several runs involving various sets of journeys on various timetables, even though some of the termini are not used in a particular run. This will result in useful savings of the user's data preparation time at the expense of slight wastage of core store on individual runs.

For example, if the user has fifty termini which he might wish to use, these should be numbered from 0 to 49. He will set up an out-of-service inter-termini times block giving information for all fifty termini. He may then use this block as input for one run in conjunction with a timetable that does not use (for example) terminus number 35, in a second run with a timetable that does not use termini 3, 9 or 47 and so on.

JOURNEY NUMBERS

Journey numbers must be unique positive integers of not more than five digits. They need not be consecutive, and there is no advantage in their being consecutive.

Times

In the timetable section of the data, times are expressed in hours and minutes; later times must always be expressed as higher numbers than earlier times. Hence, if the period extends beyond midnight, times after midnight must be expressed as times in excess of twenty four hours.

ICL

1900 Bus scheduling

Data sheet 1

Job reference

Prepared by

Date

Page

Punching instructions

All values are integer numbers. Punch as seen, ignoring empty spaces, but insert one space, tab or blank column between successive values in any line.

Total number of journeys	Highest terminus number plus one	Output parameter	Banned linkage time/indicator

Notes for schedule

1 If no intermediate output or data listing is required, make output parameter 0.
2 *If data listing required but not sorted departures, make output parameter* 1.
3 If maximum output required, make output parameter 101.
4 Banned linkage time must not exceed 202 in any circumstance.
5 Insert four values above, and check that problem size does not exceed the available core store.

Figure 1 Suggested data sheet 1

For example, if the period under consideration extends over two days, nine o'clock in the morning of the first day would be represented as 9 hours 0 minutes, half past three in the afternoon of the first day as 15 hours 30 minutes, nine o'clock in the morning of the second day as 33 hours 0 minutes, and half past three in the afternoon of the second day as 39 hours 30 minutes.

The maximum timetable time that can be handled by the program depends on the size of the problem, according to the formula quoted in Appendix 1, page 23. Times of less than about 100 hours will always be acceptable, and this should be more than adequate for most bus scheduling problems.

In the out-of-service inter-termini time block, times are expressed in minutes.

In common with all other data, times must be expressed as integers; that is, fractions of minutes are not allowed.

Problem size

The problem size is a function of the number of termini and the number of journeys and is given by the formula quoted in Appendix 1, page 21. The maximum problem size that can be solved depends on the core size of the computer; Figure 4, page 22, and Figure 5, page 24, show approximately the numbers of termini and journeys that can be handled by the various 1900 Series computers.

For the purpose of calculating the problem size, the numbers of termini and journeys are as defined in the section on parameters.

THE PARAMETERS

Four parameters are required to control the running of the program; these are:

1 total number of journeys

2 maximum number of termini

3 output parameter

4 banned linkage time

A suitable data sheet for submitting these parameters for punching is shown in Figure 1, page 4.

Total number of journeys

This is the number of passenger-carrying journeys to be considered by the program. It will be equal to the number of records in the timetable section of the input data, excluding the terminating record of zeros. This parameter, together with the maximum number of termini, allows the program to compute the precise amount of core store required for solution of the problem. The limitations on the number of journeys are detailed in Appendix 1, page 21.

Maximum number of termini

This is the highest terminus number plus one (to allow for terminus number zero). As discussed on page 3, this may include termini not actually used in the current problem. The limitations on the number of termini (Appendix 1, page 21) refer to this maximum number, and not to the actual number used in a particular problem.

Output parameter

This specifies which of the optional types of line printer output are required. There are four possible types of output:

1 board list

2 error messages

3 data listing

4 time-sorted arrival and departure lists

These are described in Chapter 3. The board list is always printed, unless data errors prevent a successful run of the program. If this is all that is required, the output parameter should be zero.

Types 3 and 4 are optional output. Type 3 may be requested by making the units digit of the output parameter non-zero; type 4 may be requested by making the hundreds digit of the output parameter non-zero. The tens digit of the parameter is non-significant. Thus the following parameter values cover all possibilities:

ICL

1900 Bus scheduling

Data sheet 2

Page

Insert seven zeros after last journey if on this page.

Scheduled journey number	Departure			Arrival		
	Terminus number	Time		Terminus number	Time	
		Hrs.	Mins		Hrs.	Mins

Use one space, tab or blank column, between items in each row. Punch seven values in each line or card.

Figure 2 Suggested data sheet 2

0 gives board list only

1 gives board list and data listing

100 gives board list and arrival and departure lists

101 gives board list, data listing and arrival and departure lists

Banned linkage time

This must be a positive integer not greater than 202; it must be greater than any of the out-of-service inter-termini times which are to be permitted. This means that the largest permissible out-of-service transfer time is 201 minutes. If an out-of-service transfer is to be forbidden between any pair of termini, the banned linkage time is the figure which is entered at the appropriate position in the out-of-service inter-termini times block. For further details see page 9.

THE TIMETABLE OF JOURNEYS

This is the timetable of passenger-carrying journeys for which the user wishes to assign buses.

It is important to realise that the expression *arrival time*, as used in this section, refers not to the time at which the bus actually completes its journey but to the time when it is ready to leave for its next journey. Any time required to turn the bus round and any rest time the crew are required to take must be taken into account.

One record must be provided for each journey. The records may be submitted in any order, except that the last record must be a special dummy record. Each record consists of seven values:

1 journey number (in range 1 to 99999, see page 3)

2 departure terminus number (see page 3)

3 departure time: hours (see page 3)

4 departure time: minutes

5 arrival terminus number

6 arrival time: hours

7 arrival time: minutes

which must be submitted to the computer in that order.

Thus the line:

87 24 8 35 16 8 58

means that journey 87 leaves terminus 24 at 8.35 and travels to terminus 16; the bus is ready to leave terminus 16 again at 8.58.

Times in hours may exceed twelve or even twenty four (see page 3). Minutes and hours values must both be given even if either is zero.

The last record of the timetable block must consist of seven values, all zero. If the optional data listing on the line printer is requested by the output parameter (page 5), this closing record does not appear in the output. The number of records excluding this closing record should be equal to the total number of journeys quoted in the parameters block.

A suitable data sheet for submitting this data for punching is shown in Figure 2, page 6.

THE OUT-OF-SERVICE INTER-TERMINI TIMES BLOCK

This block of information details the out-of-service running times in minutes between all pairs of possible termini. This data allows the program to schedule buses to finish a passenger-carrying journey at one terminus and start its next journey from a different terminus. As discussed on page 3, data must be provided for all termini from number zero to the highest numbered terminus, and for this block only, consecutive numbering is essential. In general, this may be a once-for-all block of data used for several computer runs.

It is assumed that the time required to move the empty bus between two termini is independent of which is the starting terminus. For each terminus, the times are required for out-of-service transfers to all lower numbered termini. Thus the complete information in the block forms a triangular array.

Figure 3 Suggested data sheet 3

ICL

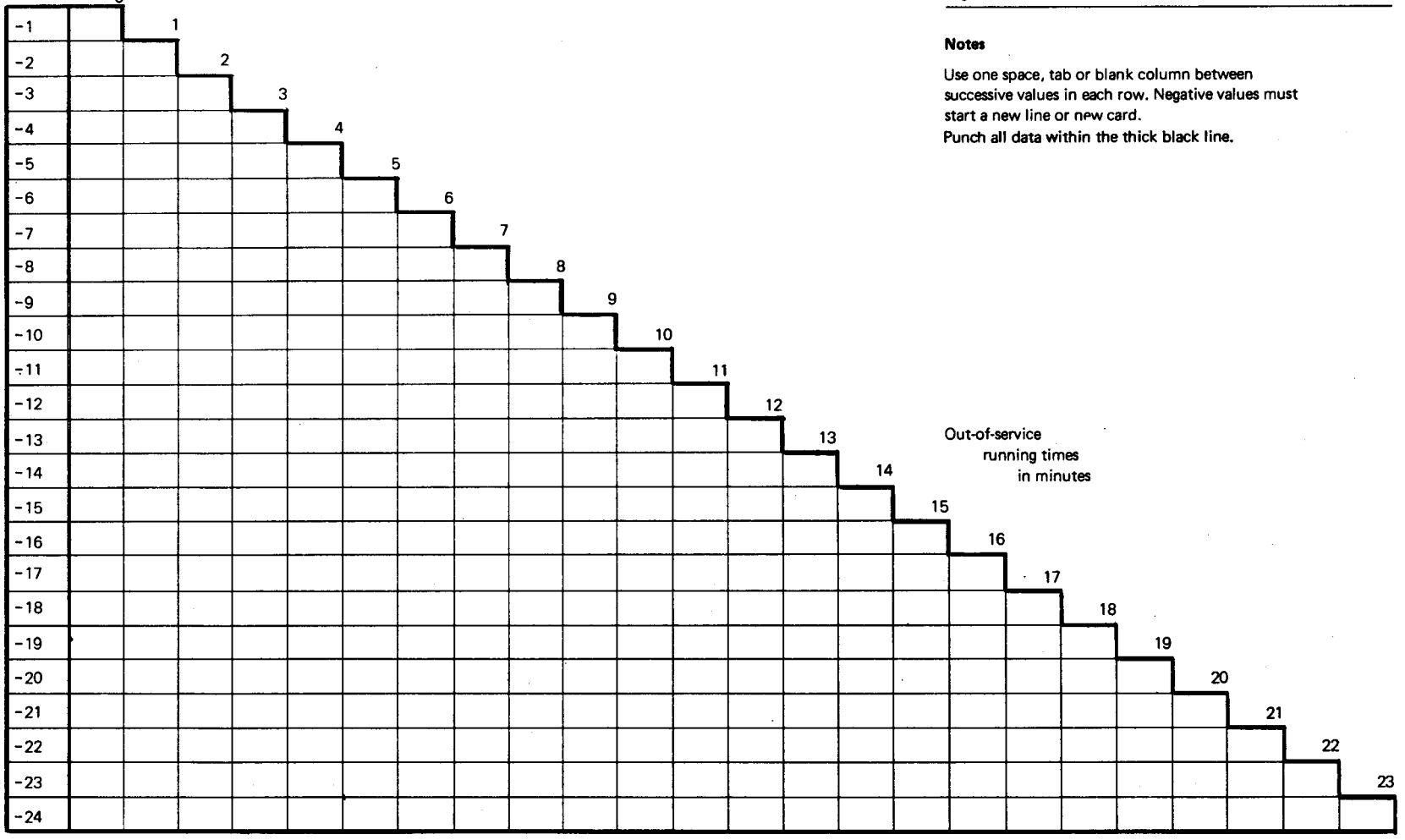
1900 Bus scheduling

Data sheet 3

Page

Notes

Use one space, tab or blank column between successive values in each row. Negative values must start a new line or new card.
Punch all data within the thick black line.



If the user wishes to prohibit any out-of-service journey between two specified termini, then the time entered in the appropriate position must be the banned linkage time defined on page 7, and given in the initial parameters. Times higher than the banned linkage time must not be used.

The set of data for each terminus begins with the terminus number n, preceded by a minus sign; this is followed by the times in minutes for out-of-service transfers between terminus n and each of termini 0, 1, 2,..., (n-1) in that order. Thus no set of data is provided specifically for terminus 0, since transfer times to terminus 0 are all given under the higher numbered termini. The first number in the block will be -1, signalling the start of information concerning terminus number 1.

A suitable data sheet for submitting this data for punching is shown in Figure 3, page 8.

The following example might form part of the out-of-service inter-termini times block for a problem with a banned linkage time of 99 minutes:

```
-1  2
-2  7  4
-3  5  99  6
-4  7  8   99  10
.   .  .   .   .
.   .  .   .   .
.   .  .   .   .
-10 7  4   99  99  6  3  3  2  99  5
etc.
```

This means that the time from terminus 1 to terminus 0 (or vice versa) is 2 minutes,; the time between termini 2 and 0 is 7 minutes; that between 2 and 1 is 4 minutes, etc. Out-of-service transfers are not to be allowed between termini 1 and 3, between 2 and 4, or between other pairs where the figure entered in the table is 99.

If there is a total of n termini defined in the initial parameters, then the program will expect to have completed reading the data when it reads – (n-1) followed by the (n-1) values of the out-of-service inter-termini times between terminus (n-1) and termini 0, 1, 2, ..., (n-2). Any further data will be disregarded, and not read.

DATA PUNCHING FORMATS

Data may be submitted punched on either cards or paper tape, and must be submitted in the following order:

1 parameters

2 timetable of journeys

3 out-of-service inter-termini times

Within each of the three blocks of data, the numbers must be given in free format, with some restrictions. This means that successive values must be separated by at least one space, tab character or newline on paper tape, and on cards by at least one blank column. Each of the three blocks must start on a new line: after a newline character on paper tape or on a new card. Restrictions on the format of the second and third blocks are given in the appropriate sections on page 10.

All values should be integers; no decimal points should be punched. Apart from the space, tab and newline characters on paper tape, the only characters that should be punched are the ten decimal digits, and the negative signs required in the third block of data only.

The parameters

These must be submitted in the correct order as follows:

1 total number of journeys

2 maximum number of termini

3 output parameter

4 banned linkage time

If the input is on cards, the parameters may be punched on one or more cards; there must be at least one blank column between parameters on the same card.

If paper tape is used, newline characters may be interspersed between parameters; there must be at least one space or tab character between parameters. The fourth parameter must be followed by a newline character before the next block of data is punched.

Timetable of journeys

The information relevant to each journey must be punched on one card (or between two newline characters on paper tape).

Each record contains seven values, which must be punched in the order specified on page 7. The values must be separated by at least one blank column on cards, or by at least one space or tab character on paper tape. Blank cards (or multiple newline characters) may occur between records.

The last card (or line) of this block must contain seven zeros separated by blank columns (or space or tab characters).

Out-of-service inter-termini times

Each negative terminus number must start a new card, or follow a newline character, and such positions within this block are the only positions where negative signs may be punched. There are no other special restrictions. The information following a terminus number may appear on one or more cards (or lines) as required, subject to the usual restriction that at least one blank column, space, tab or newline character must occur between each number.

End of data

Paper tape data must end with at least two newline characters and card data with at least one blank card.

Chapter 3 Output

The output of the program is printed on a line printer. A heading is always printed, and gives the date of the run for reference, and the time if that facility is available on the computer in use. This is followed by the number of journeys and termini involved in the problem and the banned linkage time.

If certain errors have been made in the data, fatal error messages will be printed as described on page ., and the run will terminate. Otherwise the program will carry out the computation and print output as described below.

The main output is the *board list*, which lists the journeys assigned to each bus. If requested by the output parameter specified by the user (page 5) a listing of the data and the sorted arrivals and departures will also be output.

MAIN OUTPUT: THE BOARD LIST

A single board is the sequence of journeys assigned to one bus. The boards are assigned numbers by the program and output sequentially. The output for each board consists of a heading giving the board number followed by the information about the journeys. The information for each passenger-carrying journey is output on one line in the following order:

1 journey number

2 departure terminus number

3 departure time in hours and minutes

4 arrival terminus number

5 arrival time in hours and minutes

6 time for out-of-service transfer preceding this journey

7 extended layover time (that is, any waiting time)

The output line might, for example, be as follows:

178 DEP 7 8.18 ARR 21 8.26 TTT 3 ELT 2

This means that journey 178 started from terminus 7 at 8.18 and arrived at terminus 21 at 8.26. Before the start of this journey, the bus travelled for three minutes from its previous arrival terminus to terminus 7 (the *t*erminus-to-*t*erminus *t*ime) and then waited for two minutes (the *e*xtended *l*ayover *t*ime) until its departure at 8.18.

The first output line for each board will not contain TTT or ELT entries, since the data input does not include details of the whereabouts of buses prior to the run. It is assumed that it will be possible to make buses available at the required starting points in time for the early journeys included in the run.

OPTIONAL OUTPUT

There are two types of optional output and the printing of either or both may be requested by use of the output parameter (see page 5).

Data listing

This optional output is requested by making the units digit of the output parameter non-zero.

In the timetable of journeys block, the journeys are output in the same order as that of input with the exception that the final dummy record of seven zeros is not printed. The information about each journey occupies a separate line of print; it is tabulated and headings are printed so that the meaning of the figures is apparent. In addition to the journey numbers which the user has allocated to the journeys, the program also numbers the journeys consecutively from one and these numbers appear as the first column of the table immediately followed by the user's number.

The out-of-service inter-termini times are printed as a triangular matrix.

An example of the data listing is given on pages 19 and 20.

Time-sorted arrival and departure lists

This optional output is requested by making the hundreds digits of the output parameter non-zero. It may be useful for the investigation of alternative optimal solutions. Separate lists of arrivals and departures are provided in increasing time order.

OTHER OUTPUT

Error diagnostics

NON-FATAL ERRORS

If some of the termini included in the table of inter-termini times (from number zero to the highest numbered terminus) are not actually used in the timetable of journeys, messages will be printed as follows:

TERMINUS MISSING IS *n*

where *n* is the number of the terminus which does not appear in the timetable. As mentioned on page 3, the user is allowed not to use some of his termini in a particular timetable for a particular run of the program. Thus the program does not consider this to be a fatal error, and the program will not be halted. However, the user should check that the contents of such messages are in accordance with his expectations, and he may thus be assisted in avoiding erroneous runs.

FATAL ERRORS

Output of any of the following messages implies that the program has found a fatal error in the data. The program will halt, and the board list will not be produced.

Message	*Meaning*
PROBLEM TOO LARGE	The initial parameters imply that more than 32K core store is required (compact data limitation: see page 21).
BANNED TIME TOO LARGE	The banned linkage time exceeds the upper limit of 202 minutes.
NO MINUS BEFORE *n*	where *n* is a terminus number. A minus sign appears to be missing at the start of the *n*th row of the table of inter-termini times.
TIME TOO LARGE AT *n*	where *n* is a time which has been read from the inter-termini times block. It exceeds the banned linkage time, and this is not permitted.
ERROR IN MATRIX	Negative inter-termini time read, or this data otherwise unacceptable. If no error is apparent, the data should be sent to ICL (at the address on page 16).
ERROR IN LINK SUBROUTINE	Program error suspected; the data should be sent to ICL if no obvious data error.

In general, even if an error is found, the program will attempt to read more data to check for the presence of other errors, and therefore more than one diagnostic message may be output. However, under some circumstances the first data error and the output of the first message may cause subsequent messages. Thus the user is advised to concentrate on the first message in any faulty run.

FORTRAN ERRORS

Other error diagnostics which may be printed, and which are not peculiar to this program, are those appropriate to any 1900 Series FORTRAN object program. These are given in the appropriate 1900 Series FORTRAN compiler manual. (The program was compiled and issued in TRACE 1 mode.) The most likely type of error is that due to either a decimal point or an alphabetic character in the data, in which case the message printed will be:

EXECUTION ERROR 0 'x'

where *x* is the illegal character.

Console typewriter output

On 1904 and faster machines, Executive usually prints out on the console typewriter the core store used by the machine rounded up to the next multiple of either 64 or 128 words, depending on the machine. This information may be of interest for subsequent runs. The information required to interpret this value is given in Appendix 1.

The calculation of the core store required is the first computation performed by the program after reading the parameters. If for any reason the required amount of core is not currently available the console message:

#X3MU; HALTED:- NC

will appear. Under some circumstances this may happen even though the problem is sufficiently small to be handled by the computer. On observing this message, the operator should take any action that is necessary to make more core available, and will then attempt to restart the program.

Chapter 4 Operating instructions

The bus scheduling program #X3MU is available on magnetic tape or as a binary dump on cards or paper tape.

OPERATING INSTRUCTIONS

Executive priority

The program as supplied has an Executive priority of 50.

Running under Executive alone

	Narrative	*Console message*
1	If the program is on paper tape or cards, load it on the reader and type: where *n* is the loading peripheral unit number.	LO #X3MU *n*
2	Load the card or paper tape reader with the input data.	
3	If the program is in a magnetic tape file called PROGRAM\TAPE, load the tape and then type: A similar message must be typed if the program is held on another tape or on disc.	FI #X3MU #TAPE
4	When the program is loaded, if the data is on cards, type: if the data is on paper tape, type the messages:	GO #X3MU 20 ON #X3MU 1 GO #X3MU 20
5	If no data errors are found, the program will carry out the processing and delete itself with the message:	0#X3MU; DELETED:- 00

Exception conditions

Message	*Reason*	*Action*
HALTED:- CR HALTED:- TR HALTED:- LP	A required peripheral is not available.	Make the peripheral available and restart by typing GO #X3MU.
HALTED:- NC	There is not sufficient core available.	Make sufficient core store available and restart by typing GO #X3MU. If this is unsuccessful, delete the program and return it to the originator.
HALTED:- EE	This message may result from either a data error or a program error.	The program should be deleted and returned to the originator.

In the event of any other message being printed, the meaning of which is not obvious to the operator, the program should be deleted and the job returned to the originator for diagnosis.

If errors are encountered which are not traceable to erroneous data, the data and any associated console and line printer output should be sent for investigation to:

Software Errors Section,
International Computers Limited,
30/31 Friar Street,
Reading, RG1 1JP
BERKSHIRE.

Chapter 5 Example

An example is given on the following pages of the line printer output from a small run of the program. The data was input on punched cards. The example is intended to illustrate the input and output format, and has little other significance.

INPUT

The parameters used for this run were:

```
20  8  1  99
```

signifying that the problem involved twenty journeys and eight termini. The output parameter was 1, and thus the output includes a data listing in addition to the board list, but no time-sorted arrival and departure lists. The banned linkage time was 99 minutes.

The parameters were followed in the input by the first card of the timetable block, which contained the following data:

```
805   5   17   25   2   17   35
```

meaning that the bus assigned to journey number 805 is to leave terminus 5 at 17.25 to travel to terminus 2, and is ready to start its next journey at 17.35. Other timetable journeys followed in similar format finishing with the following two cards:

```
66   3   18   11   7   18   20
 0   0    0    0   0    0    0
```

This was followed by the inter-termini out-of-service times:

```
-1   3
-2   2   5
 .
 .
 .
-7   6   3   4   99   5   2   5
```

The last card of the data following this was blank.

The full data in the timetable and inter-termini out-of-service time blocks can be seen in the listing at the beginning of the line printer output. Note that the journeys in the timetable block are not specified in any particular order, whereas the entries in the out-of-service time block are strictly in order. In the latter block the appearance of 99 at two points in the table implies that out-of-service transfers are not allowed between termini 5 and 3 or between termini 7 and 3.

The main differences between the data as input and as appearing in the line printer listing are as follows:

1 the data is tabulated on output, and headings are printed

2 in addition to the user-assigned journey numbers, the journeys are numbered consecutively on output

3 in the timetable as printed hours and minutes values are separated by points

4 the final input record of the timetable block (seven zeros) is not printed

OPERATING INSTRUCTIONS

As the data was punched on cards, the only operating instructions required were:

```
FI #X3MU
GO #X3MU
```

The program was in a magnetic tape file called PROGRAM X3MU, so no search program name was required in the FIND message.

OUTPUT

The board list contains a series of seven boards, which include all of the twenty journeys included in the timetable of journeys block of data. Thus the minimum number of buses that can be used to satisfy the timetable is seven.

The 1900 Series computer used for the run had no time facility. Thus the time appears in the output as eight zeros.

1900 BUS SCHEDULING DATE 24/03/70 TIME 00000000

20 JOURNEYS 8 TERMINI BANNED TIME IS 99

DATA LISTING

	JOURNEY	DEPARTURE		ARRIVAL	
1	805	5	17.25	2	17.35
2	127	7	17.42	5	17.52
3	364	3	17.56	4	18. 6
4	790	2	17.52	7	18. 6
5	45	4	18. 0	6	18.13
6	233	4	17.33	1	17.41
7	157	5	18.14	4	18.18
8	645	0	18. 4	5	18.11
9	89	6	18. 1	2	18.12
10	23	1	18.12	7	18.19
11	501	1	17.31	3	17.42
12	284	5	17.46	6	17.51
13	77	4	18. 9	1	18.25
14	241	2	18. 7	6	18.20
15	33	6	18.16	3	18.31
16	31	3	17.36	5	17.53
17	131	6	17.49	3	17.59
18	572	0	17.40	4	17.49
19	189	7	17.28	2	17.47
20	66	3	18.11	7	18.20

INTER TERMINI TIMES

-1	3						
-2	2	5					
-3	4	3	4				
-4	3	4	6	2			
-5	3	6	5	99	4		
-6	2	3	7	4	5	4	
-7	6	3	4	99	5	2	5

```
        BOARD LIST          TIME 00000000


        BOARD   1

805     DEP FROM   5  17.25     ARR AT   2  17.35
572     DEP FROM   0  17.40     ARR AT   2  17.49     TTT =  2     ELT =  3
 45     DEP FROM   4  18. 0     ARR AT   6  18.13     TTT =  0     ELT = 11
 33     DEP FROM   6  18.16     ARR AT   3  18.31     TTT =  0     ELT =  3


        BOARD   2

189     DEP FROM   7  17.28     ARR AT   2  17.47
364     DEP FROM   3  17.56     ARR AT   4  18. 6     TTT =  4     ELT =  5
 66     DEP FROM   3  18.11     ARR AT   7  18.20     TTT =  2     ELT =  3


        BOARD   3

501     DEP FROM   1  17.31     ARR AT   3  17.42
790     DEP FROM   2  17.52     ARR AT   7  18. 6     TTT =  4     ELT =  6
 23     DEP FROM   1  18.12     ARR AT   7  18.19     TTT =  3     ELT =  3


        BOARD   4

233     DEP FROM   4  17.33     ARR AT   1  17.41
131     DEP FROM   6  17.49     ARR AT   3  17.59     TTT =  3     ELT =  5
 77     DEP FROM   4  18. 9     ARR AT   1  18.25     TTT =  2     ELT =  8


        BOARD   5

 31     DEP FROM   3  17.36     ARR AT   5  17.53
241     DEP FROM   2  18. 7     ARR AT   6  18.20     TTT =  5     ELT =  9


        BOARD   6

127     DEP FROM   7  17.42     ARR AT   5  17.52
645     DEP FROM   0  18. 4     ARR AT   5  18.11     TTT =  3     ELT =  9
157     DEP FROM   5  18.14     ARR AT   4  18.18     TTT =  0     ELT =  3


        BOARD   7

284     DEP FROM   5  17.46     ARR AT   6  17.51
 89     DEP FROM   6  18. 1     ARR AT   2  18.12     TTT =  0     ELT = 10


OUTPUT COMPLETED     00000000
```

Appendix 1 Limitations

PROBLEM SIZE

Number of journeys and termini

The area of core store required by the program is a function of the number of journeys and the maximum number of termini: two of the parameters which form part of the input. The formula for the calculation of core store requirements is given below.

The area of core store available for use of the program is the size of the machine reduced by the amount taken up by Executive (and any other operating systems in use) and any other programs currently in store (in a multiprogramming machine).

The number of journeys and termini is limited, firstly, by the requirement that the core required by the program must not exceed the core available in the computer. A second limitation, resulting from the compact data mode of compilation, is that the core store used by the program must not exceed 32768 words. This will only affect machines larger than 32K.

CORE STORE USAGE

The core storage usage of the program is given by the formula:

$$Q=P+12J+J[(J+23)/24]+[T(T-1)+1]/6+T$$

where:

Q is the core required in words

P is the size of the program, not including data storage. This value will be published at the time of issue, but may be assumed to be approximately 5120 words

J is the number of journeys quoted in the parameters

T is the number of termini quoted in the parameters

In evaluating each term on the right hand side, the results of divisions should be rounded down.

Figure 4, page 22, gives these requirements approximately in graphical form. The axis at the left of the diagram gives the core store requirements for numbers of termini and journeys represented by the points of intersection of the curves which make up the graph. For example, the core store requirements of a problem of 50 termini and 100 journeys can be estimated as follows: the intersection of the curve starting opposite the number 50 on the termini axis and that starting at 100 on the journeys axis is opposite a point just over 7000 on the core store axis. Thus such a problem will require just over 7000 words of core store. For numbers of termini and journeys which do not correspond to curves, the appropriate position between two curves must be estimated; this can be done sufficiently accurately for most purposes.

If a more precise estimate of core store requirement is needed, the formula should be used. For the example given above, of 50 termini and 100 journeys the core store requirement is given by:

$$Q=5120+1200+100(123/24)+(49\times50+1)/6+50$$

$$=7278$$

Note that in the calculation the results of divisions have been rounded down.

CORE STORE AVAILABLE

Figure 5, page 24, shows the effects of the above considerations for the various core sizes available in the 1900 Series. The area enclosed by the axes and the curve for the appropriate size of computer represents the problems that can be solved on that computer.

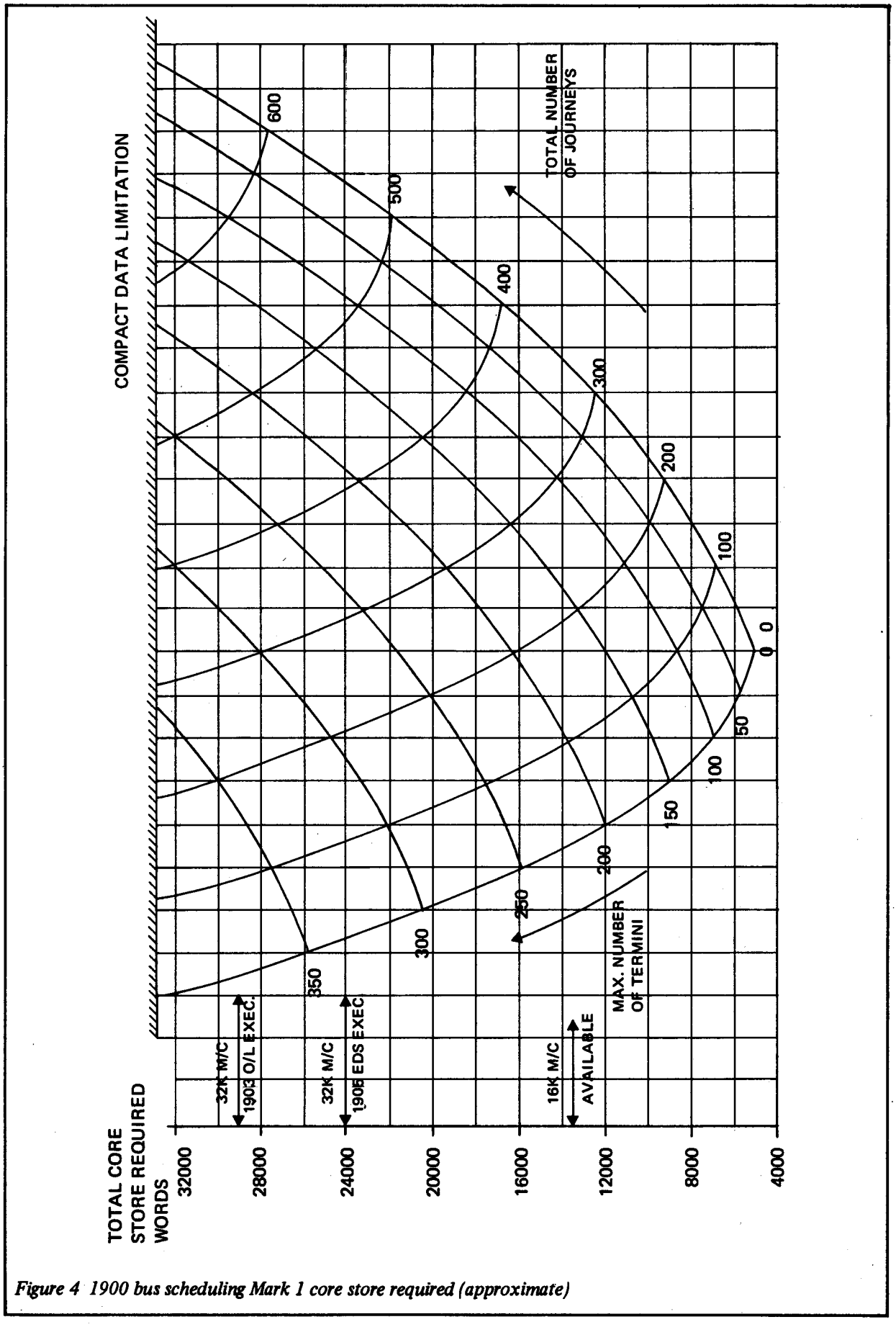


Figure 4 1900 bus scheduling Mark 1 core store required (approximate)

For example, a problem of 150 termini and 200 journeys would be within the scope of any of the computers. A problem of 150 termini and 350 journeys would be too large for a 16K machine but could be solved on a 32K computer. A problem of 150 termini and 600 journeys could be solved only on a 48K computer.

The formula implies that the maximum number of journeys that can be handled on any computer is 674.

Departure and arrival times

The maximum time that can be handled in the timetable of journeys block of data is given by the formula:

$$(J+1)C+J<4{,}194{,}303 \quad \text{(i.e. } 2^{22}-1\text{)}$$

where:

J is as defined in the previous section

C is the time in minutes

For some selected numbers of journeys, the maximum values are as follows:

If $J = 650$, the maximum $C = 6441$ minutes = 107 hours 21 minutes

If $J = 499$, the maximum $C = 8387$ minutes = 139 hours 47 minutes

If $J = 299$, the maximum $C = 13980$ minutes = 233 hours 0 minutes

It is not expected that users will encounter the limitation unless the program is being used to schedule systems other than buses.

Since the arrival time of a journey must be later than its departure time, the above restrictions should strictly be said to apply to the arrival time. The departure time of any journey must be represented by a smaller number than the arrival time.

OTHER LIMITATIONS

Journey numbers

The *journey numbers*, which are assigned purely for identification purposes, should not be confused with the *total number of journeys* which has been discussed in the previous section. The total number of journeys can never exceed 674, but the journey numbers need not be consecutive, and thus may exceed this value. The only limitation is that these numbers must be integers in the range 1 to 99999; within this range numbers may be assigned for the convenience of the user. No duplication of journey numbers is allowed within a run of the program.

Inter-termini out-of-service times block

TERMINUS NUMBERS

The user does not have the same freedom in allotting terminus numbers as he does for journey numbers. Within this block of data, termini must be numbered consecutively from zero.

For a discussion of terminus numbers see page 3.

TIMES

The banned linkage time, as specified in the parameters, must not exceed 202 minutes.

No time in the inter-termini out-of-service times block may exceed the banned linkage time. Out-of-service transfers which are to be permitted must have times smaller than the banned linkage time.

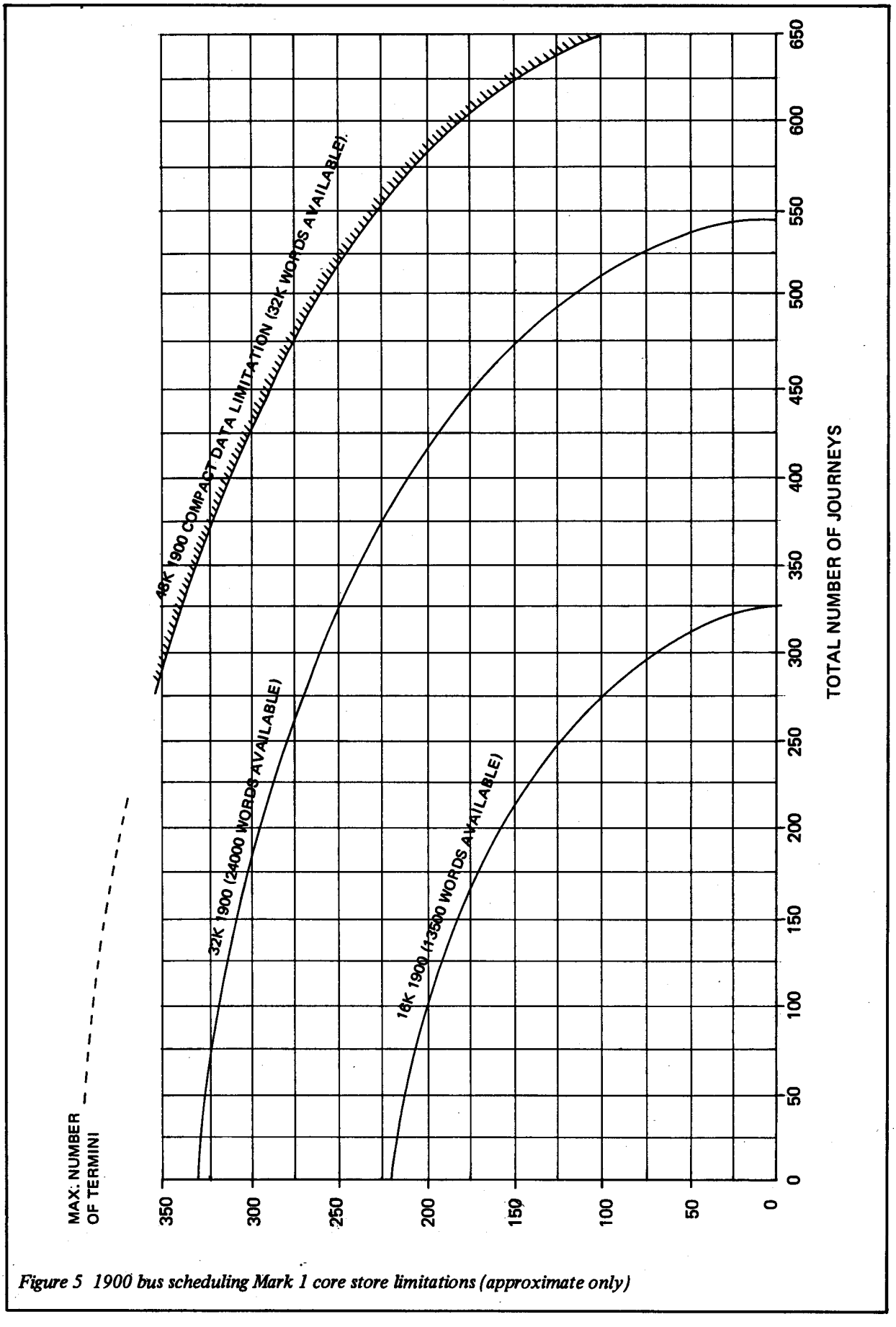


Figure 5 1900 bus scheduling Mark 1 core store limitations (approximate only)

Index