COMPUTER MODELS FOR SCHEDULING BUSES AND THEIR CREWS

M.E. PARKER & A. WREN

ABSTRACT: *A group of computer methods which have been developed to solve problems of bus and bus crew scheduling are presented. The major models are those which allocate vehicles to trips and which allocate crews to vehicles. These have been used in a large number of bus undertakings, and considerable savings have resulted. Other models deal with the construction of bus timetables from passenger demand data and the rostering of bus crews.*

COMPUTER MODELS FOR SCHEDULING BUSES

1. INTRODUCTION

The problems of scheduling buses and their crews may conveniently be divided into four parts, alhtough ideally it might be more desirable to treat them as a single problem. We may identify these four parts as:-

(i) Construction of a bus timetable; this involves deciding on the times at which public bus trips are to operate on the various routes of the system, as well as the times at which any special trips are to operate on non-standard routes.

(ii) Bus scheduling; this is the assignment of trips (as produced by the first stage either by computer or by management expertise) to vehicles, and may require adjustment of trip times in order to obtain a more efficient schedule.

(iii) Crew scheduling; here crews are allocated to the bus schedule to ensure that buses are manned at all times without violating any rules of the labour agreement.

(iv) Duty rostering; this stage only applies to undertakings in some countries (not normally in North America) and consists of grouping together five days' duties in order to form a week's work, arranging the weeks into a cyclic pattern and catering for a suitable rotation of rest days.

The bulk of this paper is devoted to parts (ii) and (iii) above, since these are the aspects of most interest to British bus operators, and thus the aspects which we have been most concerned to treat. They are also the most difficult problems to solve. The methods for these developed in the University of Leeds, which are now being marketed in Australia by Nicholas Clark and Associates, are described in section 3 and 4 of this paper, while sections 2 and 5 consist of brief descriptions of the methods used to deal with the other two aspects.

2. CONSTRUCTION OF A BUS TIMETABLE

In the United Kingdom buses, particularly in urban areas, are normally operated at regular headways, which change only a few times in the course of a day (typically, there may be three different headways, operated at peak, midday and evening periods). On top of these regular operations there may be superimposed special trips to supplement the peak hour services, and other trips, perhaps not on regular routes, for school children and other special purposes. The headways for the regular services are determined by the bus operator in the light of his experience of public demand, having regard to the desirability of minimising unnecessary layover at termini; at times it is appropriate to schedule two or more routes having a common terminus simultaneously, making more efficient use of vehicles by switching them from one route to another at that terminus. It is also necessary to have regard to other routes operating in the same area, so that, for example, three routes each with a fifteen minute headway provide a five minute headway along a common stretch of road.

Colleagues in the University of Leeds are currently undertaking research into methods of assisting the operator in the above scheduling task. The plan is to have the computer develop schedules for groups of services, to any headways specified by the operator and having regard to problems of common stretches of road. The operator may then experiment with different possible combinations of headways, the computer indicating the consequences in terms of vehicles numbers and regularity on common stretches of routes, as well as printing the resultant schedules in detail.

At present this research is at the stage where satisfactory schedules are being produced for relatively simple situations.

In many other countries, headways of bus services, and scheduled running times, vary more or less continuously throughout the day. Programs designed to produce timetables and schedules for such situations have been developed by a number of bodies, notably by Elias, (see paper by Elias and Smith, 1970). These programs have generally succeeded in building up bus trips to meet passenger demand, but have often fallen down on the next task of linking suitable one-way trips into efficient schedules for the vehicles. Other colleagues have recently converted Elias's program so that it will treat the two directions of travel separately and are now engaged in linking this with the bus scheduling program to be described in the next section. This will enable an efficient linking of trips to be formed, and will ensure that the computer will indicate where vehicles could be saved by slight revisions to the times of individual trips.

3. BUS SCHEDULING

In this section we deal with the problem of assigning buses to trips. It is often appropriate to consider several routes or even a whole city, or division of a city, simultaneously, interchanging vehicles between routes, and inserting empty journeys between termini, to save vehicles. A bus scheduling algorithm has been developed which is particularly powerful when dealing with situations where buses make special trips during the peak hour, either on regular services or on special services to or from factories or schools. The program does not construct individual trips but assumes that they, and their times, have been provisionally specified, either by a program as outlined in section 2, or more often, by management. It can, however, indicate where retiming or cancellation of trips will produce savings in vehicle numbers.

The algorithm will be described and its use in many different circumstances will be discussed.

3.1 Data

Essentially two sets of data are required for a bus scheduling program of this type. One refers to the individual trips which are to be scheduled, while the other refers to the times which should be allowed for buses to run empty between terminal points.

The data referring to the trips should give the names of the two termini and the provisional times of departure and arrival. It is important that only essential trips are specified, and that any which are present in an existing schedule simply to move vehicles to different termini, should be omitted from the data, so that the computer may choose other movements if appropriate. The trip data may also include, optionally, information on suitable types of vehicle, limits on retiming, and special layover requirements.

The travel times between termini must be estimated with a great deal of care, since if these are too high, the computer may fail to produce an economical solution, and if they are too low the result may be unworkable. It is always recommended, however, that the computer results should be studied carefully by management before implementation, and indeed that after an initial inspection further computer runs should be made. It is therefore probably better to err on the high side when preparing this data. It will be seen that the computer can indicate which connections it would like to make, but cannot because the times specified for those connections are too long. Management may then decide whether any of these times are possibly too high.

Empty running times in practice need to be specified only between points chosen for their geographical proximity, or in order to link different areas by a fast route; other times are initially set to be very large, and the computer calculates approximations to all other pairs using a reduction process based on Floyd's algorithm. (see Floyd, 1962).

3.2 Summary of the Algorithm

The objective of any bus scheduling exercise may vary slightly from one undertaking to another, but certain goals are common to all cases, although they may themselves be contradictory.

1. Minimise the number of buses used.

2. Minimise a function of empty running time, time out of garage, and certain qualitative measures.

In practice, goal (1) will normally take precedence over goal (2), the costs being such that saving a vehicle will be of paramount importance.

The algorithm has already been described by Wren (1972) and an outline only will be given here.

The number of vehicles which are to be used in the schedule may be specified by the user; alternatively, the computer will compute a lower bound to the number required. An initial, usually poor, schedule is constructed using this number and is then improved in various stages.

Let us suppose for the present that we already have this poor bus schedule; it may be poor because it would be expensive to operate, because the crews would not like it, or because the vehicles are expected to do impossible things. (It is easy to construct a poor bus schedule with any given number of vehicles, if one allows buses to travel backwards in time; the ultimate is a schedule with only one bus which covers all the trips of an undertaking in any order, moving backwards through time whenever the departure of one trips precedes the arrival of the "previous" trips, or follows it too closely for the bus to make any necessary dead journey). The improving process will now be summarised, and we shall later see how the poor starting schedule is obtained.

We shall first consider the work of two of the buses in the schedule (assuming for the moment that there are at least two). Each of these buses will have been assigned to that garage which could most suitably operate it. In particular, let us consider two successive trips, A and B, on the first bus, and two successive trips, C and D, on the second bus (remember that the definition of "successive" has been extended to allow the bus to run backwards through time between the trips). (See figure 1). We may now think of changing the schedule so that the first bus, after working trip A works trip D and all subsequent trips previously undertaken by the second bus, while the second bus follows trip C by trip B, etc. We then determine which would be the best garage to serve each of the revised running boards. We are now able to assess the consequences of such a change, and either make the change or restore the original situation accordingly; the general rule is that the change is made unless the overall situation is worsened according to the lights of the undertaking for whom the work is being done. There are basically two types of improvement which may be made; one (figure 1(b)) makes something feasible which was not previously feasible, or at least reduces the extent by which it is infeasible; the other (figure 1(a)) improves a situation which is already feasible, or at least improves the situation in some respect, even if neither the original or changed situation is feasible, as long as the extent of any infeasibility is not exacerbated.

Let us now examine the possibilities in detail:

1. A followed by B and C followed by D are both feasible in the time available, but either A followed by D or C followed by B (or both) is not; no change is made.

2. A-B or C-D or both are infeasible, but A-D or C-B are both feasible; the change is made.

3. Neither the original nor changed situation is entirely feasible; the change is made if the total extent to which buses have to run backwards in time is reduced, or if it is unchanged and some other criterion is improved.

4. A change does not affect the feasibility of the situation. This is by far the most common case, and here account is taken of the overall quality of the original and changed running boards, in terms of empty running and time out of garage.

5. Finally, there are changes which make no difference to any of the above factors. These are implemented initially on a first-in, last-out basis, so as to allow long period of idle time to be created and then used for other purposes. After no more useful changes are found, however, this is changed to a first-in, first-out principle wherever the time span between the earlier arrival and later departure is less than a parameter specified by the user. If this parameter is, for example, specified as 30 minutes, then breaks of under 30 minutes will be smoothed out between bus workings, while those in excess of 30 minutes will be retained and may be used for crew breaks if appropriate.

The program in fact looks at all possible pairs of trips A, B, C & D in this way, even where they are all on the same bus (as may happen in an infeasible situation). It also considers situations where some of these trips are in reality buses in the depot and not yet allocated; for example if A is a bus in the depot, this is equivalent to B being the first trip of the day for the bus; it may then be better for B to follow C, and for D to be the first trip for the bus.

The program repeats the process until no change can be found which improves the situation and at this stage it prints the solution. This solution is the best that the program can find with the chosen number of vehicles and it is therefore presented to management. (Indeed, although this process cannot guarantee optimality, experience indicates that unless the constraints or objective are particularly complex, the solution is truly optimal.) There are three situations which may arise.

1. The solution contains no infeasibilities. Management may wish to accept this or, more likely re-run the program, attempting to find a solution with fewer vehicles.

2. The solution contains some infeasibilities which are not acceptable. An extra vehicle is added and the program restarted.

3. All infeasibilities can be removed by retiming or cancellation of trips, thus making the schedule acceptable. Management may again wish to seek a solution with fewer vehicles.

3.3 Modifications to the Basic Algorithm

Since the details of the algorithm were published in 1972[2] many modifications have been made in order to cater for different bus undertakings' requirements. Some of the principal modifications are described below.

3.3.1 Automatic Retiming: A facility is available as an option whereby users can specify for individual trip times between which there is some latitude in running. Thus, for example, a trip may be timed to leave its point of origin at 1615, but the undertaking may be prepared to have it leave any time between 1612 and 1620 if this would ease scheduling problems. In such a case, the departure time will be given as 1615 in the data, and the trip information will be followed by the figures 3, 5, indicating that the trip may run up to three minutes earlier or up to five minutes later. When this facility is used, the program when it finds it cannot reduce the extent of any infeasibilities further by the process already described, will attempt to ease the situation by retiming various trips between the limits allowed, and will then return to the optimisation process; if after this there are still infeasibilities, further attempts will be made to retime, and the process is repeated until no further improvements can be made. As a final step, the program then investigates all trips which it has retimed, to see whether subsequent operations have removed the need for the retiming; in which case the original time is restored as nearly as possible.

3.3.2 Different vehicle types: With this option, a list of types of vehicle, together with the number of each type available, must be input. For each trip the user should indicate the preferred vehicle type, the other possible types which might be used, showing degrees of preference, and any types that are impossible. At each stage there is a trade-off between the desirability of the individual trips on a board being worked by a particular type of vehicle and empty running time and time out of garage.

3.3.3 Crew scheduling considerations: Various amendments have been made to enable the alborithm to produce certain running boards with the needs of crew scheduling in mind. These are necessarily designed to meet the circumstances of a particular undertaking, and include having peak-only buses out of the garage for as near to a particular length of time as possible, and ensuring that the starting times of morning peak buses and the finish times of evening peak buses are such that they can be worked by split duties.

A similar trade-off to that used when dealing with different vehicle types is employed.

3.4 Comparison with other Bus Scheduling Programs

The program described here, together with two others, was the subject of a comparative study carried out by the SELNEC Passenger Transport Executive, who then operated about 3000 buses in the Greater Manchester area. This has been reported by Dickinson (1973) who concluded:

> "the Leeds University program was the most suitable of the three programs in fulfilling the bus scheduling requirements of SELNEC. It was considered, however, that a number of detailed improvements could still be made. These improvements have in fact now been made to the program and several more are planned."

The SELNEC judgment was based both on quality of results produced in a test exercise and on the ease with which the programs might be adapted to suit further requirements of SELNEC.

3.5 Implementation

The algorithm has been applied to several different problems for various undertakings. The areas in which it has been used, and the results achieved, will be described, followed by a discussion of the problems which have been encountered.

Yorkshire Traction

Three exercises have been carried out for this company. The first, in which 23 vehicles were saved out of

107, and the second, in which 2 vehicles were saved out of 18, have already been described. (Wren, 1972)

The third exercise consisted of 385 trips from one garage operated by 9 different types of vehicle. Each trip could be retimed by up to two minutes earlier and up to three minutes later. The manual schedule was operated by 42 vehicles. A solution using 39 vehicles was obtained with some retiming and three infeasible connections, all of which were acceptable to management.

Kingston upon Hull Transport

This city operates a fleet of just over 200 buses and we were asked to re-schedule those vehicles which only operated in the peak periods. A few months earlier the city had employed another bus scheduling program, and this had saved them some vehicles, reducing the requirements to 111 buses in the morning and 114 in the evening, over and above those buses which operated all day. We undertook a number of runs of the program, with different numbers of vehicles. Completely feasible solutions were obtained with fewer vehicles than were already operating, and management ultimately accepted "infeasible" solutions in which 107 buses were used in the evening peak and 106 in the morning peak. The infeasibilities were of the order of 1 minute in 20; for example only 19 minutes would be available for a dead journey which had been specified in the data as requiring 20 minutes. This apparently caused no trouble, and the schedules were implemented in May 1972.

Greater Manchester Transport (GMT)

Some of the work carried out for Greater Manchester Transport (at the time known as SELNEC) has already been referred to in section 3.4. Many minor modifications have since been made to produce a bus scheduling program suited to GMT's particular needs and the resultant version of the program has now been sold to GMT who are planning to use it as the basis of their future scheduling system.

Coventry Corporation Transport

This city was faced with a need to revise its schools' schedules very quickly as a result of a re-organisation of education. The pattern of operation was that buses would wherever possible undertake school journeys immediately after the morning peak or immediately before the evening peak, the same vehicles being used for school services and peak hour extra services. The need for a change in the schedule was known well in advance but the final requirements of the Education Authority were not expected to be known until about two or three weeks before the start of the school year. It was therefore decided in April of 1973 to carry out a computer scheduling exercise on the existing school requirements, and as a result of this, the undertaking was shown how a small saving in vehicle numbers could be achieved. It was however not worth their while re-scheduling at that stage and this first exercise served two purposes. First, it established the confidence of the transport department in the program, and secondly, it enabled much of the data to be set up on the computer in preparation for the live scheduling run which was to be done in August. Amendments to the data in respect of the revised requirements were received by us in August and within three weeks of our receipt of the data the buses were operating to our schedules.

Edinburgh City Transport

A pilot exercise was undertaken here with some peak buses. New schedules (with some infeasibilities) were produced using only 47 buses, and these were accepted by management. However, owing to a change in policy, a substantial revision of peak services was then decided upon and the results were not implemented. The undertaking is still interested in using the program and we are hopeful that they may return to us with further data.

Trent Motor Traction

This company had recently been merged with another, and they were interested in considering whether by scheduling the services of the two companies together any buses could be saved. The work currently undertaken by about 120 buses out of 5 depots was presented to the program and the latest schedules produced seemed to indicate that the work could be done by 112 buses. There were however some difficulties with data collection, and if these can be satisfactorily sorted out it is hoped that the number of buses might be reduced still further.

Societe des Transports Intercommunaux de Bruxelles

The Brussels undertaking has spent a considerable amount of effort in developing their own bus crew rostering program and more recently a bus crew duty scheduling program. They had been considering whether they should develop a bus scheduling program, but felt that the time to use such a program would be after they had developed the crew scheduling program, so that they might then use a suite of three programs. Before deciding whether to develop their own bus scheduling program they approached us to see whether ours would be suitable. We entered into this project with some misgivings, since they wanted us to produce a schedule for a single route with a branch at one end, and we felt that probably their existing schedule would be as efficient as possible.

In the event we did produce a schedule with 38 buses where they used 39, the saving being achieved by running one bus empty between the terminus on the branch and the terminus on the main route. This was not implemented directly, but we hope that when Brussels turn to computer scheduling they will use the program again. In the meantime I understand that a consequence has been that the schedulers in Brussels are themselves now looking for situations where buses can be saved by running them empty between prongs of any forked route.

Aberdeen Corporation Transport

Aberdeen wanted us to produce a schedule for a very small number of peak buses (32), and a point of interest here is that the act of collecting data for us showed them how to save six buses themselves. Indeed by the time we received their data they had already implemented their own re-scheduled services using 26 buses. Although the program showed how slight retimings of other trips would reduce the number of buses to 24, these were not acceptable to management.

Other Undertakings

Work is currently in progress for Ulsterbus Ltd. (9 depots spread over a wide area) and Merseyside PTE (2 city depots). The program has also been used as a preliminary to crew scheduling exercises and to assist in developing new bus networks for urban areas.

4. BUS CREW SCHEDULING

In this section we present a suite of programs which has been developed and used on behalf of some half dozen undertakings. The suite consists of an initial program and a series of optimisation programs. The initial program constructs a legal set of duties, while the optimisation programs consist of different algorithms for refining these. The wide differences between union agreements are handled both by parameters and by special routines. Thus, four (or sometimes more) different classes of duty are identified, and the outlines of each class are specified by parameters (e.g. earliest start time, latest finish time, maximum time without a break). The various programs construct tentative duties satisfying these parameters, but enter a special subroutine written for the undertaking concerned in order to determine whether the duties satisfy all the requirements of the undertaking. Whenever it is necessary to determine the cost of a duty, a second subroutine is entered which evaluates the duty according to the wage structure of the undertaking, and according to any rules which may have been evolved for assessing the desirability of a particular type of duty.

4.1 Outline of the Approach

The fact that management-union agreements differ very widely from one undertaking to another makes the development of a generally applicable program complex. Systems analysts studying the problem for the first time usually propose that a list of parameters should be designed to cater for all eventualities. At first sight this is an attractive proposition, but it is one which we have consistently resisted, we believe justifiably, since with every new undertaking we have found at least one constraint which is totally different in nature to any previously experienced.

In order to produce some form of compromise, it was decided that we should use parameters where possible, but be prepared to introduce features into the program to suit individual clients as necessary. At first sight, this would seem to conflict with our aim of producing a general program. However, the approach adopted has been to have the program attempt to form duties according to specified parameters which are sufficiently general as not to exclude any possible duties. When a duty is found which passes the conditions implied by the parameters, a special routine is entered which will only validate the duty if indeed it does satisfy all the conditions of the undertaking concerned. At a later stage it will be necessary to judge between two possible sets of duties, and here again it is appropriate to appeal to a special routine which will assess the cost of any duty (including any "cost" attributable because the duty includes some undesirable feature).

Our philosophy is therefore to have a set of parameters which are of general application, together with two subroutines which are written explicitly for a particular client. The first of these, which we call "valid" validates any proposed duty, and the second "wages", tells us its cost. These routines each require about eighty lines of ALGOL, but many of these can be carried over from one version to another, so that little effort is required initially to adapt the program for a new client. The more difficult task is usually to determine from the client exactly what he requires.

Although we have set ourselves this goal of having a general program which, by means of a set of parameters and two special routines, can handle the problems of any undertaking, it must be confessed that we still find that new features have to be introduced into the program in some cases to deal with particular circumstances. These features are usually to help the initial program to produce good schedules despite the client's particular quirks, and we are gradually finding that the need for such program modification is becoming less. From time to time we ensure that any such modifications made for particular clients are generalised so that the standard program will produce good schedules for any one of the clients we have met. This means, in effect, that we are gradually expanding our list of parameters.

We have found it convenient to classify duties according to the types normally considered by a bus undertaking. For each of these types we have a parameter list consisting of eight entries, namely:

the earliest time that the crew can sign on;
the earliest time the crew can take over a bus;
the latest time the crew can leave a bus;
the latest time the crew can sign off;
the maximum length of time without a meal break;
the maximum length of time worked in a duty;
the maximum spreadover of a duty;
the minimum time paid for a duty.

Originally we had four types of duty:

1. Early straight duties which normally start before the morning peak, and work until sometime in the afternoon with a break (paid or unpaid) as near to the minimum as possible;

2. Late straight duties which finish by taking a bus into a depot after about 2200;

3. Middle straight duties which finish in the earlier part of the evening, having started around, or before, noon;

4. Spreadover duties which cover both peaks and have a long break in the middle.

Later we had to deal with undertakings which normally had certain pieces left out of the schedule to be worked as overtime, and we included such pieces as a fifth duty type. More recently, we have allowed a sixth type for one undertaking where evening peak work may be associated with certain early duties for overtime working; this sixth duty then has to incorporate the features of early duties, but has to cater for the conditions binding these overtime pieces to early duties (for instance, there must be a certain minimum time between the end of the early duty and the beginning of the overtime piece).

In addition to the parameters associated with particular types of duty, there are certain parameters which govern the number of duties of each type, and other general features.

The overall approach of the method may therefore be seen as one of constructing tentative duties according to a set of parameters, and of carrying out further tests according to some specially written routines. Within this approach we distinguish a first stage in which we construct a schedule which satisfies all the constraints of the undertaking from the subsequent stages, in which we try to improve on this in some way. Before describing the functioning of the various stages, however, it will be appropriate to describe briefly the data requirements.

4.2 Input Data

In addition to the parameters mentioned above, it it necessary to describe the bus schedule for which the duty schedule is to be constructed, and the time allowances that must be given to crews at the start or finish of a duty or when changing buses.

First, it will be recognised that there are a limited number of points at which a crew may join or leave a bus; we call these relief points. The garage will always be a relief point, as will any other point where a bus may be left unattended by a crew. In addition, there will be agreed relief points on every route (but often only one on any route). It does not follow that because a certain point is the relief point for one route it will be a relief point for another route that passes it. The total work of any bus may be broken down into a number of indivisible units, in the course of which a crew cannot be relieved. The beginnings and ends of these units are the times at which the bus passes a relief point applicable to its route, and are known as relief times.

The data includes a list of names of all relief points used only so that the output is easily understood. The points are numbered, and these numbers are used in the remainder of the data and in the program itself.

There is an entry in the input for each relief time. The relief times are grouped by bus, and the first time for any bus will normally be the time when it leaves the garage; the exception is where a portion of the bus is covered by a crew from another schedule, in which case that portion is excluded from the data, and the remainder is treated as if it were a separate bus, starting on the road. The data for a particular relief time is as follows:

> route number (optional); this is useful where several routes are being scheduled together and there are constraints or preferences affecting the way different routes are treated or mixed;
>
> running board (block) number;
>
> relief time;
>
> relief point number.

It is necessary to know the time which must be allowed, at any relief point, for a crew to sign on and, if appropriate, to travel from the signing-on point to the relief point. Similar information refers to signing-off, and these constitute two lines of data. A third line gives the nearest canteen to each point.

Three square tables are necessary, each with a row and column for each relief point. The first gives the minimum time which must be given if a crew is to come off a bus at one point, travel if necessary to another point, and join another bus, without a meal break; we call these <u>join-up times</u>. The second table gives similar information for the cases where a meal break <u>is</u> taken; we call these <u>break times</u>. A certain portion of a break may be paid irrespective of whether the whole break is paid, for example where the relief point is far from a canteen, or where other travelling is involved; these <u>paid times</u> are specified in the third table.

4.3 Initial Stage

As has already been mentioned, we start by construcing a schedule which satisfies all the constraints. An initial program constructs a schedule of valid duties, leaving some pieces of work for later treatment if necessary. A second program deals in a rather rough way with these pieces of work which have been left out. (They will be treated more thoroughly in the optimisation stages). Throughout the description of this initial stage it is to be understood that the computer will construct tentative duties, or parts of duties, satisfying the broad parameters in the data, and will then use the valid routine to determine whether in fact these duties would be legal.

The first act of the initial program is to determine, for every bus that leaves the garage before the morning peak, the latest time that the crew that takes the bus out of the garage may leave the bus. To illustrate the different functions of the list of parameters and the valid routine, we shall explain in some detail how this is done. The list of relief times for the bus is scanned, starting with the earliest, until one is found which would, according to the parameter list, exceed the maximum length of time without a break for an early duty. The previous relief time is then taken, and the valid routine checks the stretch of work from the time the bus leaves the garage until this time, to determine whether it can be part of an early duty; if it cannot, the next earlier time is taken, and so on, until a valid stretch is found.

As an example, let us suppose that, in general, a crew can work for 4½ hours without a meal relief, but cannot work more than three hours at a stretch on route 17 unless the portion of work starts after 0700. Suppose also that the following are the relief times of a particular bus on route 17 in the morning:-

0644 0712 0831 0915 0950 1045 1130

The program will work forward from 0644 until it encounters the time of 1130, which is more than 4½ hours from the start; it will then move back to 1045, and will call the <u>valid</u> routine to check whether the stretch from 0644 to 1045 is valid as part of an early duty. Finding that this is not valid, it will next check the stretch 0644 to 0950, and this check also failing, will try 0644 to 0915 which is acceptable. It should be noted that the number of occasions when the time chosen by reference to the parameter fails the validity check is usually small, so that the above process saves the computer frequently having to perform the full validity check, which may be time-consuming.

We refer to the set of latest relief times now formed as <u>marked times</u>; they are not necessarily going to be used, since we may ultimately use an earlier time. The only cases when we stipulate that a marked time must be used is where the bus finishes at that time (e.g. when it is the time the bus returns to the garage after the peak), or, after some duties have been formed, when that time has already been chosen to start the second part of a duty for another crew; such a marked time is said to be fixed. If any buses leave the garage after the end of the morning peak, but before noon, their starting times are also marked. The computer then checks to see whether enough of the stretches leading up to marked times would be valid in spreadover duties (which may have a lower limit on the amount of time in a stretch). If not, it tries to adjust the marks accordingly.

The function of the marked times is to indicate where a crew should leave a bus or join a bus, and the marked times are sorted into chronological order.

A similar exercise is then performed for buses which work over the evening peak. Where the bus runs into the late evening, the marked time is set as the earliest time the crew which takes the bus into the garage may join it (usually some time after the evening peak). Where the bus returns to the garage immediately after the evening peak, two times are marked, namely the time it returns to the garage, and the earliest time the last crew must join it if it is to form the last piece of work for the crew. It will be noted that these marked times are now indicating where evening meal breaks might be taken, or where an afternoon break might end on a middle or spreadover duty. These marked times are now sorted into reverse chronological order.

The process by which duties are formed will now be described in outline, but it should be appreciated that there are many complexities in it, which space will not allow us to mention. These complexities involve special actions to deal with cases which may only rarely arise, but which have to be catered for. It is hoped that their omission will be excused.

The program first builds up early duties (although an option to allow it to start with late duties is now being incorporated). It takes each marked time, in order, and examines it to determine whether it is fixed (i.e., must be used). If it is not fixed, this implies that the bus continues after the marked time, and that no crew has yet been allocated to the following portion. Since this is then the earliest marked time following which no crew is yet allocated, an additional crew would have to be assigned to take over the bus at that time. It will usually be best to start this crew as early as possible, to provide additional cover over the whole of the peak, and not just over part of it. The marked time is therefore moved earlier, provisionally as far as possible, without leaving an unreasonably small piece of work at the start of the bus.

An attempt is now made to form a duty whose first piece of work is on the bus under consideration, finishing at the provisionally marked time, and whose second piece of work takes over another bus at or before its marked time. Normally a number of such duties may be possible, and the computer will give some preference to taking over the second bus at the marked time, since the next duty formed is likely to be relieved later, and may then be too late to take over at this marked time. The computer will also, in making its choice, take into account various other preferences affecting the nature of the duty being formed. Having found the "best" duty using the provisionally marked time, the computer will move the provisional mark progressively later until it reaches the original mark, and will choose the best overall duty. It will also consider forming a part-duty which includes a change of buses without a meal break.

Throughout the process of forming early duties, the computer keeps in mind the desired number of spreadover duties, and ensures that no stretch of work is used to form the first half of an early duty if it is a potential first half of a spreadover duty, unless there are sufficient other potential first halves of spreadover duties to make up the desired number. The computer will also refrain from forming an early duty if the best duty it could form still fails some test of efficiency, for example, if the meal break would be longer than some specified limit.

Whenever a duty or part-duty is formed here, the computer will adjust the list of marked times, fixing those used in the duty, and adding a further marked time to the portion of the first bus beyond the time the crew has left it; this indicates the maximum time the next crew can stay on the bus. The list of marked times is then re-sorted, and the next time is investigated. It should be noted that where a part-duty has been formed, the computer will at a later stage consider what should follow the second piece.

The following example of the building up of early duties is illustrated in Figure 2. It is assumed that two spreadover duties are required, and that they cannot start before 0600, that the maximum time without a meal break is 4½ hours, the maximum time in a duty is eight hours, the minimum meal break is 45 minutes, and the minimum time to change buses is five minutes. All relief times are assumed to be at the same point. Each line in the figure refers to a bus, the relief times being shown. The marked times at the start of the process are shown by means of a ring, and the duties ultimately formed are indicated.

(i) The earliest marked time is at 0900 on Bus 1. This is moved earlier, and we shall assume that the earliest reasonable time on that bus is 0730. This is linked to Bus 2 at 0815, and the duty formed then works on Bus 1 until 0730, and then on Bus 2 from 0815 to 1245. The mark on Bus 2 is now moved from 0915 and fixed at 0815. The mark on Bus 1 is fixed at 0730, a further mark being put on at 1145, which is the latest time a crew joining the bus at 0730 can leave it.

(ii) The earliest mark is now 0815 on Bus 2. This is linked to 0910 on Bus 3, rather than the earlier time of 0900 on Bus 4 because the former is marked. The duty will finish at 1230. The mark at 0910 on Bus 3 is now fixed.

(iii) The mark at 0910 on Bus 3 is linked to 1010 on Bus 4, the duty finishing at 1400. The mark at 1010 is fixed.

(iv) The mark at 0915 on Bus 6 (which is already fixed because it is the end of the bus) is linked to 0920 on Bus 5 without a break, continuing until 1030. Note that this possibility of joining to Bus 5 at 0920 could not be used at the previous stage, because the crew starting at 0515 on Bus 3 could not have continued until 1030 without a break. The mark on Bus 5 at 0920 is now fixed, and an additional fixed mark is placed at 1030.

(v) The mark at 0920 on Bus 5 is linked to 1030, also on Bus 5, the duty finishing at 1415.

(vi) The next mark is at 1010 on Bus 4. However, the early work on Bus 4 is one of the only two portions which start late enough to form part of a spreadover duty (the other is the portion on Bus 1 from 0730), and this must be left aside.

(vii) The next mark is at 1030 on Bus 5, and this is linked to 1145 on Bus 1, forming a three-part duty which finishes at 1500.

(viii) The next, and last, mark is at 1145 on Bus 1, but again this has to be part of a spreadover duty.

We have now formed five early duties, and have left aside two portions which can form the first parts of spreadover duties.

A similar process is then applied to the evening buses, working backwards into the afternoon. Here care has to be taken when moving marked times later into the evening that no opportunities of using the evening peak part of the bus as the latter part of a spreadover duty are lost.

By this time most early duties, all late duties, and some middle duties will have been formed. There will remain certain work during the peaks, and possibly other work during the day. The peak work is now examined to determine which portions can be paired to form spreadover duties, and any work still unallocated during the morning peak is formed into an early duty in its own right, with, if possible, a further piece of midday work attached.

If by this stage the desired number of duties has been reached or exceeded, the initial program stops. Otherwise it will form unassigned work into duties of one piece, starting with the latest finishing work until the desired number is reached.

A second program is now entered which tries to fit in any unallocated work (possibly as third portions of duties), without rearranging anything already grouped together. Any work which still cannot be fitted in is then designated as a duty of a special fifth type. This type is costed very highly in the optimisation processes.

4.4 Refining Stages

The routines used in the refining stages may be broken down into two groups. The first of these consists of those designed to try to decrease the number of crews or of unassigned portions of work (fifth type of "duty"). The routines in the second group attempt to reduce the cost of the system, including any notional cost of undesirable features. There is no specific order in which the various routines must be used, and experience shows that some of the routines may be more appropriate for one undertaking than for another. However, we normally use the first group of routines immediately following the initial stage, and then follow this with some routines from the second group, after which we may return to a routine from the first group to see whether the number of crews can be reduced further. Some routines may be used several times in the process.

4.4.1 Routines to reduce numbers of duties: These routines are concerned only with reducing the numbers of proper duties or of unallocated portions of work; they do not take into account the cost of the resultant schedule. The full process of trying to reduce the number of crews can be very time-consuming, and we have spent considerable effort evolving methods which are compromises between searching for all possibilities, however remote, and saving computer time.

Reduce This routine first checks to see whether the target number of duties has been exceeded, and only takes any action if this is so. It has been our experience in some cases that undertakings prefer to have a solution which uses more crews than the minimum, in order not to increase unduly the average work content of duties, and for this reason we do not reduce the number of duties below this target. In this routine, each duty is considered in turn, and an attempt is made to redistribute its work among the other duties. Sometimes it would be necessary to take work out of another duty in order to make room for the redistributed work, and this is done if this piece of work itself can find a home elsewhere.

Maxspread This routine tries to decrease work in duties of low spreadover by increasing the work in duties of higher spreadover so that gaps are created to help REDUCE get rid of duties.

Only 5 alter This routine only considers the unassigned work (fifth duty type). It tries to reduce the unassigned stretch of work as much as possible by putting as many pieces of it on to any other duty, or duties, that will take the pieces. If all the work cannot be completely got rid of in this way, the remaining stretch of work is swopped with any other smaller stretch of work on any other duty (other than a type 5 duty). This swopped stretch is then checked to see if any of the work involved can be moved and so on until no further reducing can be done.

4.4.2 Routines to reduce cost: The routines in this group appeal to the wages routine in order to determine the cost of any possible revision of services. For this purpose, the wages routine calculates not only the cost in terms of payments to crews, but also a notional cost associated with any undesirable features in a duty; the sum of these two costs is the objective which the routines try to minimize. The formula by which the notional cost is calculated is usually determined initially after consultations with the schedules department of the undertaking concerned. However, it may be changed if, after inspecting the first computer produced schedules, the undertaking notices that some undesirable features are still present.

Halves This routine cuts each duty into two portions at a suitable break point, and forms two lists of half duties, as follows:-

Duty type	List one	List two
Early	1st half	2nd half
Split	1st half	2nd half
Middle	1st half	2nd half
Late	2nd half	1st half

This grouping has been chosen as that which gives the maximum number of possibilities of pairing members of one list with members of the other. The problem of determining the optimum pairing is an assignment problem, but in order to reduce the amount of computer time needed for its solution, the current solution is taken as the starting solution for an optimization method which is a modification of the standard method for solving the transportation problem. Where there is a restriction on the number of duties of one type there might be a danger of this routine, which cannot look at over-all numbers, causing a violation of this restriction. In order to overcome this problem, the routine is in these cases called twice. As an example of this, let us consider the case where the number of spreadover duties is restricted, say to ten, and let us suppose that there are eight spreadover duties in the current solution. The cost array is built up for the first pass row by row, and impossible costs are set against the formation of spreadover duties in all but the eight rows corresponding to current spreadover duties, and the first two other rows in which spreadover duties might be formed. In the second pass the costs are built up column by column in a similar manner.

<u>Recut</u> This takes the work of each vehicle in turn, and considers all the times when crews are changed on that vehicle.

If by moving one of these changeovers to another relief time the cost is reduced, such a move is made. In some cases, this move will actually be to another relief time which is itself a changeover time; then one of the two crews will be working for a zero time on that bus, and in fact will have the number of portions in his duty reduced by one.

Change ends This routine takes a small piece of work around the middle of the day off each duty, and solves the assignment problem of reallocating these to the duties in the optimal way. The pieces removed will generally be the last parts of early duties, the first parts of middle or late duties, and some part out of the middle of split duties. Where a duty is currently in three parts, a whole part will be removed if possible; otherwise a piece of work is cut off one portion of work.

Stretchswop This routine tries to level out duties by looking at each individual duty in turn and seeing if firstly, by removing any stretch and putting the stretch on another duty, and secondly by exchanging any stretch for any other stretch in any other duty, the cost is lowered.

4.5 Experience

We now have experience in using the program on behalf of seven undertakings, with widely differing constraints, and in some cases with different constraints in separate areas of one undertaking. In six cases, comparisons are available with manually produced schedules, and typical results for these undertakings are displayed in Table 1. The individual experiences will be discussed during the presentation.

4.6 General

So far we have not specifically highlighted the objectives of crew scheduling by computer. When we entered into an arrangement with the National Bus Company in 1970, we did, however, set out what seemed to be three alternative objectives:

(i) To compile, if possible, better schedules than are compiled at present;

(ii) To compile schedules comparable with those compiled manually at present, but with a speed which would enable schedules to be revised and tested more easily;

(iii) To compile schedules which are not as good as manually produced schedules, but which can speedily be altered by an experienced scheduler to give the desired quality.

It is clear that the second of these objectives has been met (and therefore the third is redundant); in some cases the first has also been met, but it is accepted that the quality of manually produced schedules is such that often no improvement can be expected.

5. DUTY ROSTERING

There has been little demand for a computer program to form rotating rosters of bus crew duties in the United Kingdom, although such rosters are in widespread use. One problem is that bus operators have not been able to specify precisely what is required of a roster; where the requirements have been precisely specified, the problems have turned out to be so insignificant that the use of a computer would have been extravagant.

Nevertheless, a former colleague while working with us in the University of Leeds, succeeded in developing a program which produced a rotating roster based on the ideal requirements of one bus undertaking. The method was a variation of that of Bennett and Potts (1968), fitting a given set of duties to a given pattern of days off and days on which particular types of duties should be worked. In practice, there was no way in which the two patterns could be reconciled (there were, for example, more duties than slots in the roster, and decisions had to be made as to which duties to leave out). At the time that the work was being undertaken the bus operator was in the process of changing the pattern of rosters, and it did not prove possible within the time span of the project to obtain a definitive statement of the requirements.

The program written at that time is still available and could readily be adapted to the needs of any other organisation.

6. CONCLUSIONS

Bus scheduling and bus crew scheduling by computer are viable propositions. The programs described have produced bus schedules substantially better than those in operation, and the results have been implemented in various locations. The bus crew scheduling programs have also produced implemented results; here it is not normally expected that savings should be achieved, the advantage being in speed of compilation. Nevertheless, there have been cases where savings have been achieved.

Since preparing the bulk of this paper, a live run of the crew scheduling program has been carried out for an undertaking whose name cannot yet be given. This run has treated two groups of schedules for each of two depots, producing results requiring 23, 27, 64 and 66 crews; these compare with manual schedules requiring 24, 27, 66 and 66 crews respectively, an overall saving of three crews out of 183, and a similar saving in cost. The schedules have been accepted by the schedules staff of the undertaking and it is hoped that they will be implemented during April.

7. ACKNOWLEDGEMENTS

The work reported here is the result of substantial research by many present and past members of the Operational Research Unit of the University of Leeds, and we are grateful for all their assistance. In particular, we acknowledge the help of Philip and Barbara Manington, who contributed to the first drafts of sections 3 and 4 respectively. We should also like to thank the staff of many bus undertakings who have taken an interest in our work.

REFERENCES

Elias, S.E.G. and Smith, N.S., Jr. (1970) "The development and demonstration of an automatic passenger counter". *West Virginia University Engineering Experiment Station Bulletin*, no. 94, vol. 1.

Floyd, R.W. (1962). Algorithm 97, Shortest path, *C.A.C.M.* p.345.

Wren, A. (1972) "Bus scheduling - an interactive computer method". *Transportation Planning and Technology*, vol. 1, p.115.

Dickinson, R.W. (1973) "A SELNEC view of computer scheduling". *Proceedings of the Fifth Annual Seminar on Bus Operations Research*, University of Leeds, 1973, p.1.

Bennett, B.T. and Potts, R.B. (1968) "Rotating roster for a transit system". *Transportation Science*, vol. 2, p.14.

	No. of Duties	Hours Paid	Unallocated No. of Pieces	Unallocated Hrs.	Duty Type: Early	Split	Middle	Late
Bristol								
Manual	39	266:02	1	1:17	19	3	5	12
Computer	39	265:21	1	2:00	19	3	5	12
Midland Red								
1 (Manual	46	380:50	-	-	13	19	4	10
1 (Computer	46	379:28	-	-	13	19	4	10
2 (Manual	18	145:09	-	-	1	12	2	3
2 (Computer*	17	142:56	-	-	1	12	1	3
Cleveland								
1 (Manual	57	526:17	6	11:22	24	-	12	21
1 (Computer	57	520:55	3	10:14	25	-	11	21
2 (Manual	128@	1038:49	22	45:06	52	-	31	45
2 (Computer*	128	1020:32	14	27:21	51	-	35	42
Leeds								
1 (Manual	70	628:54	15	30:30	26	21	3	20
1 (Computer	70	641:12	9	23:12	25	23	4	18
2 Computer*	21	201:36	6	15:22	7	8	-	6
3 Computer	32	298:21	4	7:49	10	12	2	8
4 (Computer&	84	949:16	11	41:27	33	24	5	22
4 (Computer&	84	1003:36	8	30:35	30	28	4	22
Glasgow								
Manual	79	695:00	-	-	44	-	6	29
Computer 1	79	706:45	-	-	41	-	9	29
Computer 2	78	688:14	-	-	40	-	9	29
Manchester								
1 Computer	31	237:42	1	2:28	16	-	4	11
2 Computer	102	800:54	3	14:18	34	28	7	33
3 (Manual	130	?	?	?	40	25	33	32
3 (Computer	125	995:41	2	10:42	37	24	32	32

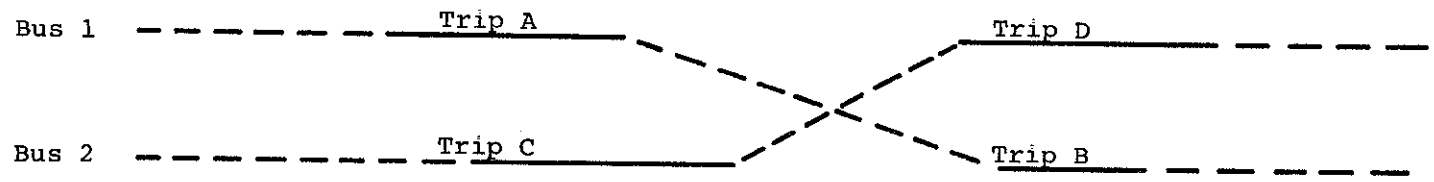
NOTES:

* Schedule implemented by bus operator.

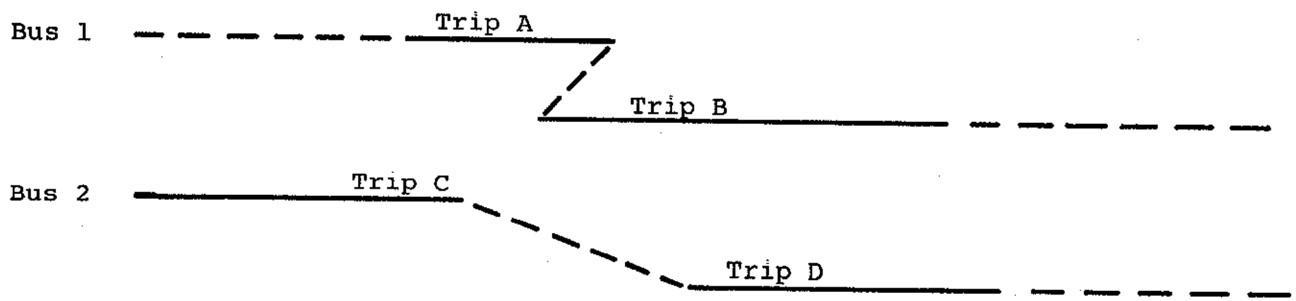
@ This consisted of six different sets of duties ranging in number from 10 to 38, each of which had to be compiled separately.

& These are two computer runs showing the effect of different numbers of split duties.

TABLE 1



(a) Feasible situation.



(b) Infeasible situation (bus runs backwards in time after Trip A).

FIGURE 1

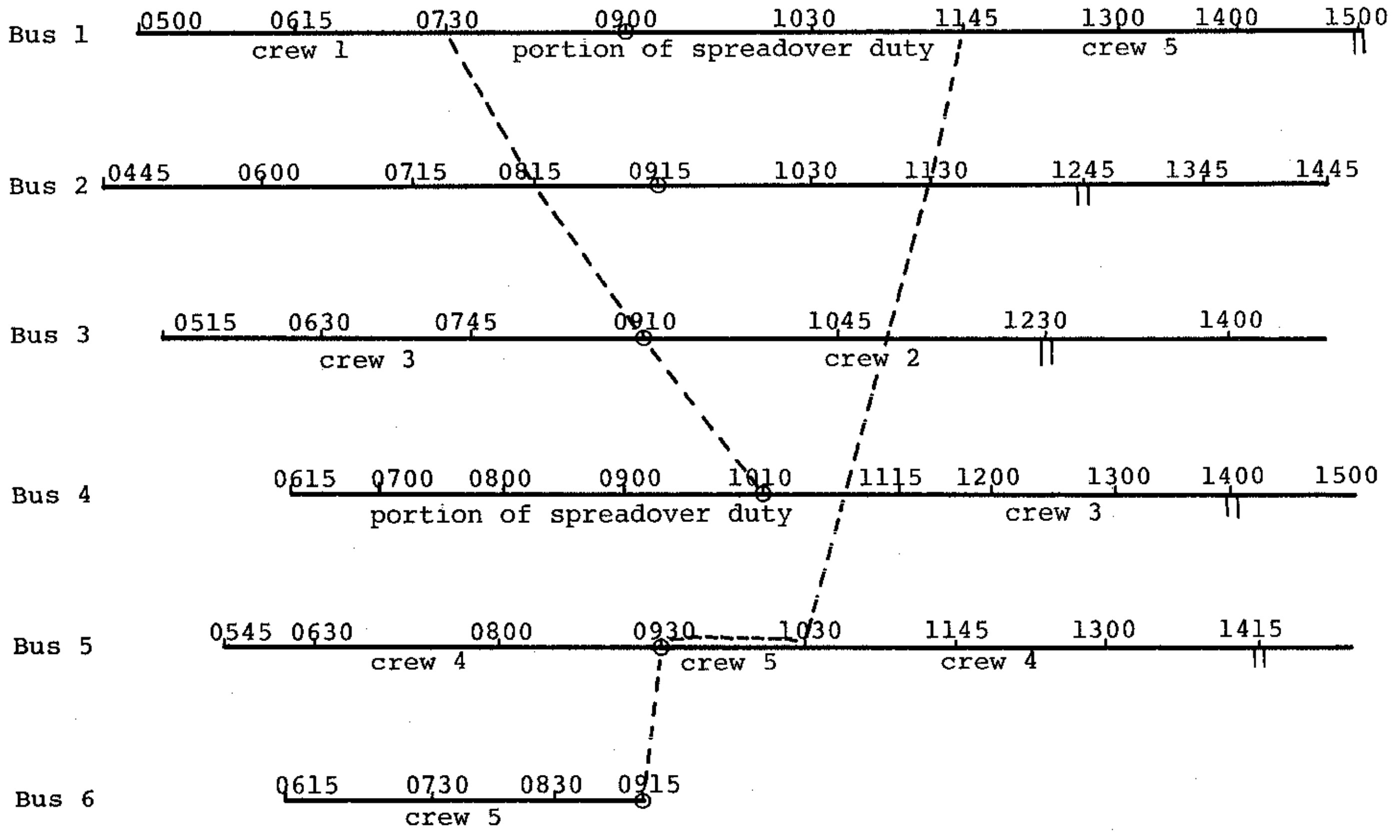

Bus 1
0500 0615 0730 0900 1030 1145 1300 1400 1500
crew 1
portion of spreadover duty
crew 5
Bus 2
0445 0600 0715 0815 0915 1030 1130 1245 1345 1445
Bus 3
0515 0630 0745 0910 1045 1230 1400
crew 3
crew 2
Bus 4
0615 0700 0800 0900 1010 1115 1200 1300 1400 1500
portion of spreadover duty
crew 3
Bus 5
0545 0630 0800 0930 1030 1145 1300 1415
crew 4
crew 5
crew 4
Bus 6
0615 0730 0830 0915
crew 5

FIGURE 2