ISSN 1012-277X S.Afr.j.ind.eng.

S A Tydskrif vir Bedryfsingenieurswese, Vol 4, No 1, Junie 1990 pp18-29

SHIFT SCHEDULING FOR A MAINTENANCE WORKSHOP.

M. Sinclair and L. Muller

Department of Industrial Engineering
University of Stellenbosch
7600 STELLENBOSCH

ABSTRACT

A big passenger transport company experienced problems with the utilization of their maintenance crews. Large amounts of overtime were logged, while at the same time a lot of idle time was experienced. This paper describes the development on a microcomputer of a Decision Support System (DSS) to improve their shift scheduling. Central to the DSS is a heuristic algorithm which generates shift schedules. This algorithm is easy to understand and provides good results, while it solves the real life problem rapidly on a microcomputer.

OPSOMMING

'n Groot passasiersvervoermaatskappy het probleme ondervind met die benutting van hulle onderhoudspersoneel. Baie oortyd is gewerk en terselfdertyd is baie ledige tyd opgeteken. Hierdie artikel beskryf die ontwikkeling op 'n mikrorekenaar van 'n Besluitnemingsondersteuningstelsel (BOS) om hulle skofteskedulering te verbeter. Die belangrikste element van die BOS is 'n heuristiese algoritme wat skofteskedules genereer. Hierdie algoritme is maklik om te verstaan en genereer goeie skedules. Terselfdertyd los dit die maatskappy se probleem vinnig op 'n mikrorekenaar op.

1. INTRODUCTION

A big company in the passenger transport sector of the South African economy (to remain anonymous at the request of management) must inspect and service their vehicles on a regular basis. The extent of their business makes it compelling to have maintenance staff available around the clock. The staff work in teams, called crews in the sequel. Shifts can start at any time during the day, as long as each crew works a shift of 8 hours non-stop. At any one time there can be anything from one to three crews on duty. In such a situation it is of course necessary to make sure that the staff is available at the times when the greatest work loads are experienced. In the next section it is shown that there is a discrepancy between the availability of staff and the work load at the company under consideration. This discrepancy often leads to too much overtime being worked, while at other times many of the crews are idle. The company thus loses money through bad shift scheduling.

In an attempt to solve the problem identified in section 2, a microcomputer-based Decision Support System (DSS) was developed to help the supervisor generate shifts which will ensure that the availability of maintenance crews match the work load more closely. Section 3 will be devoted to the description of the core element of the DSS, namely a heuristic algorithm for the generation of shift schedules. In section 4 the rest of the DSS will be discussed. A few concluding remarks will be made in the final section.

2. THE PROBLEM

The company under discussion has a few unique characteristics. Those characteristics relevant to this discussion, are:

(a) Since the vehicles keep to rigid timetables, the arrival times of vehicles for service is known to a fair degree of certainty.

(b) The type of service required (at least as far as the workshop under discussion is concerned) is known beforehand.

(c) The time at which a service MUST be completed is known beforehand.

(d) Vehicles arrive for service and depart for duty around the clock.

(e) Shifts can start at any time of the day.

The present shift schedule is shown in figure 1. The availability of staff is shown in terms of the number of crews scheduled to be on duty during all time intervals.

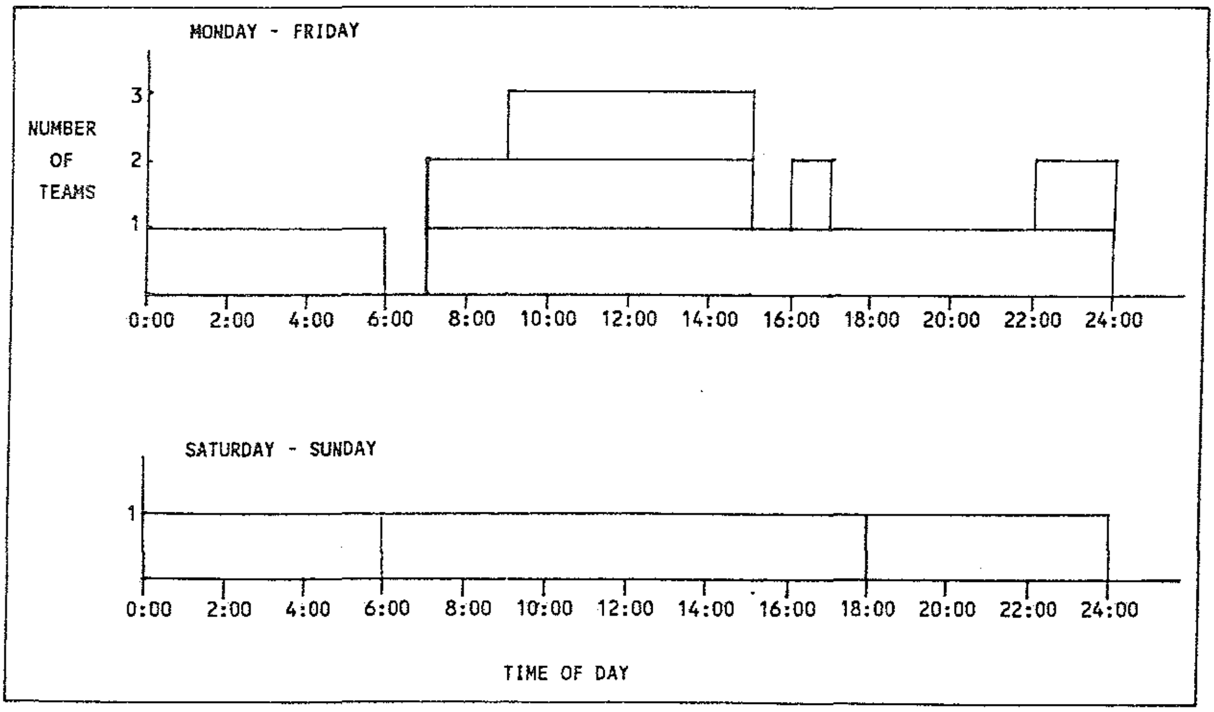

FIGURE 1: Current shift schedule.

The shift schedule represented in figure 1 was generated by hand. In order to evaluate the suitability of this schedule (or any other one) it is necessary to find some similar type of representation for the work load. The characteristics of the business under consideration make it possible to work out for each time interval of each day, a work load in equivalent work hours.

This is more or less the same for each weekday. A typical work load is shown in figure 2.

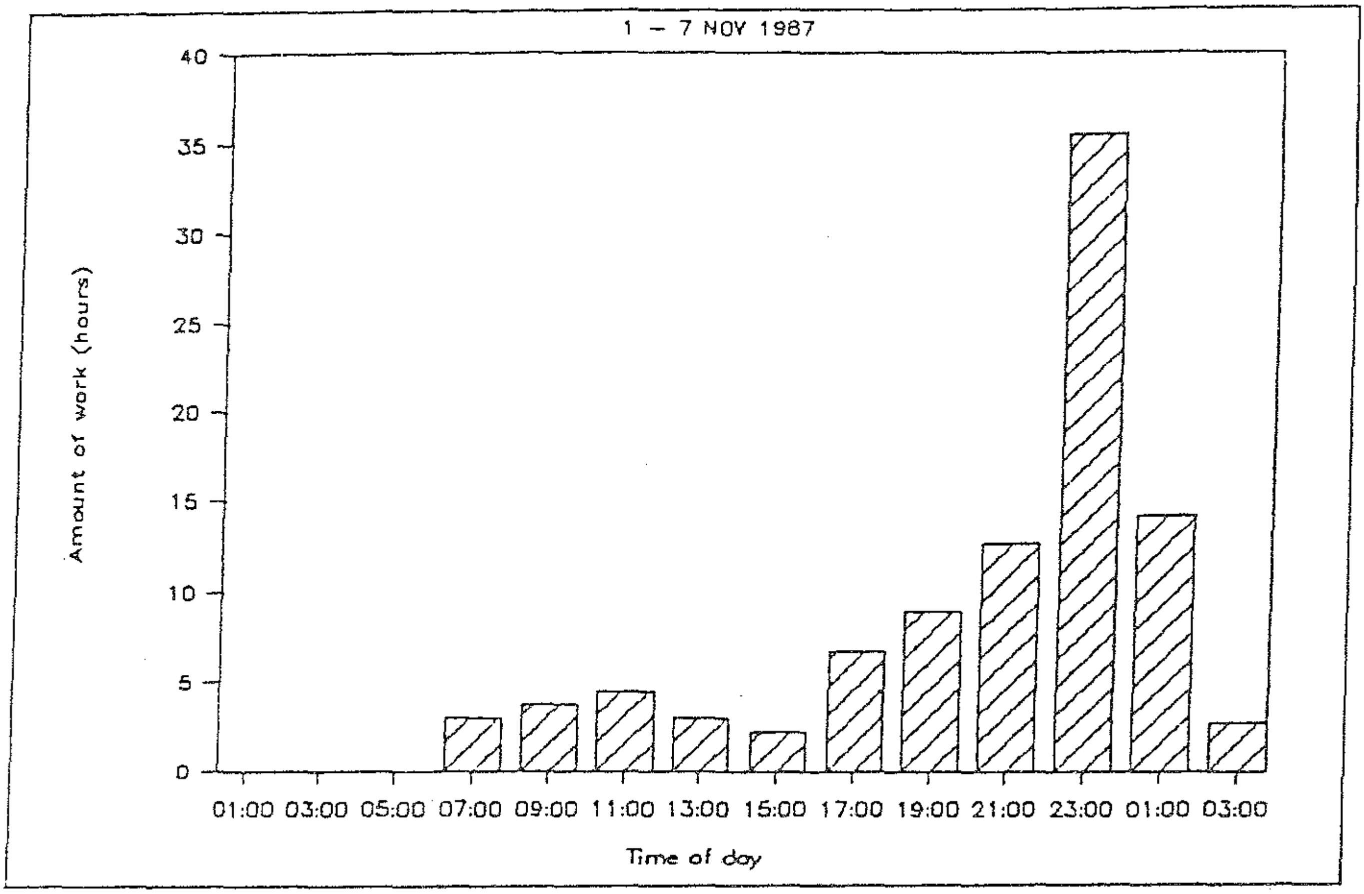

FIGURE 2: Work load on a typical week day.

Figure 1 shows that there are three crews available between 09:00 and 15:00 each weekday, while there is only one crew on duty between 17:00 and 06:00 the next morning. Figure 2 shows that the peak work load is experienced between 16:00 and 02:00. There is thus only one crew on duty during the period in which the peak load is experienced, while up to three crews are on duty during the quiet period in the day.

It is obvious from these figures why in one month 27344 man hours idle time were recorded, while at the same time a substantial amount of overtime was worked. Waiting time results from the unavailability of vehicles to service while maintenance staff were on duty. Overtime results because a maintenance crew must finish the service on a vehicle once it has started the service.

The problem is thus to supply the supervisor with some kind of decision support in order to generate shift schedules which can lead to improved utilization of the maintenance crews.

3. SOLUTION OF THE PROBLEM.

Two possible solution approaches were considered. The first possibility was to adopt a flexitime system. It was felt, however, that given the rigidity of the time tables followed by the vehicles and the fact that at least one full crew must be on duty at any time throughout the day, such a system would not be appropriate. The second possibility was to develop some kind of computerized decision support. This was the line of attack followed in the end. Given the circumstances in the company, the following constraints were placed on the computerized system:

(a) The system must run on a relatively inexpensive microcomputer.
(b) The system must be user friendly.
(c) The system must be able to supply decision support, not make the final decision.

It was decided to concentrate on the scheduling of shifts and not the allocation of specific crews to certain shifts. The reason for this was that, since the timetables for the vehicles remained fixed for at least three months at a time, a good shift schedule would have to be found at most once every three months. Flexible rotation of the crews through the shift schedules could be used to take into account factors (mainly human factors) which could not be built into a computer model.

Since it was expected that a shift schedule would stay valid for at least three months at a time, a reasonable amount of trouble could be taken to construct a good schedule. As a first step, a literature survey was carried out. The problem under consideration falls into the category of crew scheduling. Many publications on this problem exist (see Bodin, et al [1] for a survey). All of these make use of models which are difficult to solve on the type of microcomputer specified above. For example, Paixao, et al [2] use a linear integer programming model, Gianessi, et al [3] use a travelling salesman model, and Marsten, et al [4] use a set partitioning model

Given the restriction on the computing power available, it was decided to use a heuristic method to solve our problem. The heuristic procedure used in this study was adapted form the one described in De Sousa, et al [5]. The reasons the latter method could not be used as it stood, are two-fold. Firstly, the method was still under development at the time of publication and full particulars were not available. Secondly, the problem as considered by De Sousa, et al was not constrained by the requirement that shifts must cover an uninterrupted eight hour time interval.

The following solution approach was followed:

3.1 Work units:

In Bodin, et al [1] and Paixao, et al [2] uninterrupted units of work were identified and then combined to form a shift or service. The unit of work which cannot be interrupted in our case refer to the service to be performed on a vehicle.

3.2 Time intervals and work load per time interval:

Both Bodin, et al [1] and Paixao, et al [2] divide the total time over which scheduling must be done into smaller time intervals for the sake of simplicity. The work load is then calculated per time interval. In this study the 24 hour day is divided into half hour units. This is of course a compromise between accuracy (shorter time intervals) and ease of solution (longer time intervals). For each time interval in every day, the total work load (converted into equivalent number of crews needed) is calculated.

The next step is to convert the work load for each separate day into a single work load for a "typical" day, which can then be used for the heuristic procedure. This was done by considering each half hour time interval and assigning the maximum work load in that time interval for all days of the week to the typical day. Thus a single work load distribution is obtained for the typical day. The shift scheduling will be done to cover that work load. The idea was to make sure that no day will be shortstaffed. Another reason was that crews will work according to a specific shift time table for at least one full week at a time. By allocating them the same shift each day for a reasonable period at a time, their activities after work would be disrupted less than otherwise.

3.3 Heuristic method:

The heuristic used can be classified as a greedy heuristic. At any stage of the procedure, the remaining work load of the "typical" day to be covered by the remaining number of crews, is available for each half hour time interval. One shift is scheduled during each iteration. The method runs through the day from beginning to end, starting the eight hour shift at the beginning of each half hour time interval. Each time the total amount of work load covered by the shift being scheduled to start at a specific time, is calculated. In the end, the shift is scheduled to start at that time where it can cover the biggest amount of the work load. The remaining work load for the half hour intervals affected is reduced, the number of crews available is reduced by one, and the procedure is repeated until no crew remains unscheduled.

For a more formal, structured exposition of the algorithm, see the appendix.

4. IMPLEMENTATION ISSUES.

The procedure discussed in the previous section concentrated on the basic issues involved in solving the problem. In this section the emphasis is on implementation. In the spirit of all DSS's, the system is highly interactive. The final DSS consists of four main steps, namely information input, information processing, determination of the work load and the calculation of shift starting times. A short discussion on each of these steps follows. This section is concluded with a discussion of the computer program.

4.1 Information input:

The user is given the option of using the existing data or reading in new data. The following types of data are required for each service:

(a) The number of vehicles for which data must be read in.

(b) The number of maintenance crews available.

(c) The shift length.

(d) The type of service. Services at different levels of complexity are catered for.

(e) The scheduled arrival time of the vehicle.

(f) The scheduled departure time of the vehicle.

(g) The day of the week.

Some inputs have a default value. These include the length of a service type, the delay from the arrival of a vehicle until the service can begin and the time necessary from the completion of a service until the departure of the vehicle.

4.2 Information processing:

The second step is to process the data into work units as discussed above. For each service the day, type, starting and ending times are determined. All other data are irrelevant as far as the solution procedure is concerned. The services are then processed into work units for each half hour time interval.

4.3 Determination of work load:

After the previous step, the maximum work load in each half hour time interval is calculated. Thus one day is constructed with the work loads in each half hour time interval large enough to cover the requirements of every day of the week.

4.4 Calculation of shift starting times:

If the user so chooses, the first three crews can be scheduled directly to cover the full 24 hour period (assuming a shift length of 8 hours). The work load is then updated accordingly and the rest of the scheduling is done as described above. The scheduling can be repeated with different values for shift length and number of crews, until the decision maker is satisfied with the result.

4.5 The computer program:

The computer coding for the DSS was done in Turbo PASCAL. The system can be used on any IBM compatible microcomputer with a 20 MB fixed disk. The system begins with an introduction which explains the aims of the system to the user. It also mentions that information is available which can be used without any inputs from the user.

Next the user is asked whether he/she needs any help in the use of the program. If help is needed, three pages of information are displayed.

The user can then look at and change the data already in the data base. A description of the data is given in section 4.1. After the processing of the data (see section 4.2), the user again has the option of viewing the information. The maximum work load which will be used by the scheduling program is shown graphically on the monitor.

The final step is to do the shift scheduling. The user is asked to supply the number of crews available and the shift length. The algorithm then determines the shift schedules as discussed above.

Two routines finalise this step. First, the schedule is compared with the work loads of the individual days to evaluate its usefulness. The amounts by which the scheduled work availability underestimates the work load for each day is calculated and summed for the whole week. The decision maker can use this number as an indication of the usefulness of the schedule. Secondly, the scheduled shifts are shown graphically on the monitor.

This step can be repeated with different shift lengths and number of crews, until the user is satisfied.

5. CONCLUSIONS.

The timetables for the company's operations usually stay the same for three months at a time. Thus it will be necessary to recalculate the shift schedules in about three-monthly intervals.

It was found that the calculated shift schedule resulted in a total shortage of 182 half hour units per week for the situation at present, whereas the original schedule had a shortage of 222 units, an improvement of 18%. The program is, as can be expected, very sensitive to variation in the work load pattern during the week.

If the user wants a different shift schedule for some days (e.g. weekends), those days can of course be considered separately.

A typical run of the system takes about 30 seconds for the information processing step (excluding input and output) and about 10 seconds for each run of the shift scheduling program. Thus a quick, reasonably successful DSS has been created for the problem of shift scheduling.

REFERENCES:

[1] Bodin, L ; Golden, B ; Assad, A ; Ball, M (1983): "Routing and scheduling of vehicles and crews - the state of the art." Computers and Operations Research, Vol. 10, No. 2, pp 117 - 138.

[2] Paixao, J ; Branco, I M ; Captivo, E ; Pato, M V (1986): "Bus and crew scheduling on a microcomputer." In OR Models on Microcomputers (Coelho, J D and Tavares, L V, eds.), Elsevier Science Publishers, Amsterdam, pp 79 - 95.

[3] Gianessi, F ; Nicoletti, B (1979): "The crew scheduling problem: A travelling salesman approach." In Combinatorial Optimization (N Christofides ; A Mingozzi ; P Toth ; C Sandi), John Wiley and Sons, New York, pp 389 - 392.

[4] Marsten, R E ; Shepardson, F (1981): "Exact solution of crew scheduling problems using the set partitioning model: Recent successful applications." Networks, Vol. 11, pp 165 - 177.

[5] De Sousa, J P ; Guimaraes, R C (1986): "A decision support system for vehicle and crew scheduling in urban transport services." In OR Models on Microcomputers (Coelho, J D and Tavares, L V, eds.), Elsevier Science Publishers, Amsterdam, pp 65 - 78.

APPENDIX: The shift scheduling algorithm.

In order to give a more formal exposition of the algorithm used for the scheduling of shifts, some notation must be introduced. In the discussion in the main text, the day is divided into half hour time intervals. Let $[t_i ; t_{i+1}]$, i = 1, ... 48, indicate these time intervals, and let w_j denote the remaining work load in each of these time intervals (measured in number of crews needed). Denote the number of crews available by N. Then the algorithm can be given in a semi-structured code as follows:

```
1.  Initialize w_j and N.  Clear all shift allocations.
2.  FOR i := 1 TO N DO
3.    t_0 := 0
4.    Load := 0
5.    FOR j := 1 TO 48 DO
6.      Sum := 0
7.      FOR k := j TO j+15 DO {See remark below}
8.        Sum := Sum + w_k
9.      END FOR (k)
10.     IF(Sum > Load) THEN
11.       Load := Sum
12.       t_0 := t_j
13.       J := j
14.     END IF
15.   END FOR (j)
16.   Allocate crew i to the shift beginning at time t_0.
17.   Update w_j, for all j in {J ; J+15}
18. END FOR (i).
```

Remark: Note that step 7 will need some kind of wrap-around operation to make provision for shifts which run past 24:00.