HEURISTIC WORKFORCE SCHEDULING

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

By

William Daniel Culver

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

Georgia Institute of Technology

December, 1975

HEURISTIC WORKFORCE SCHEDULING

Approved:

Richard H. Deane, Chairman

Leon F. McGinnis, Co-Chairman

Russell G. Heikes

Robert G. Parker

Date approved by Chairman: 10/30/75

ACKNOWLEDGMENTS

The author is very grateful to his advisor, Dr. Richard H. Deane, for suggesting the thesis topic and guiding its initial development; and to his co-advisor, Dr. Leon F. McGinnis, for assuming the role of advisor for the latter stages of development. Appreciation is extended to Dr. Robert G. Parker and Dr. Russell G. Heikes for their helpful comments and suggestions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

The purpose of this investigation was directed at determining heuristic methods for developing daily employee work assignments to meet multiple objectives of management and labor. The research is particularly directed at those work environments where manpower requirements fluctuate over a 24-hour day and a 7-day week. Management requires that the manpower requirements be met economically while the employee desires a "reasonable" schedule with respect to work hours and days off.

The research included the development of several measures of manpower requirements variability to be used in predicting excess manpower usage. In addition, two types of heuristic scheduling procedures were developed, one for scheduling complete tours and one for two-phase scheduling through the steps of shift-allocation and tour-building. A limited empirical study of these was performed.

CHAPTER I

INTRODUCTION

In many large scale business organizations manpower requirements may fluctuate widely over a 24-hour day and a 7-day week. Examples include telephone information operators, which must be scheduled around the clock; airlines, which provide reservation or information capabilities; and police or fire stations. There exists in these situations a need for a method of developing suitable individual daily work assignments in an efficient manner.

Suitable assignments are those which meet the multiple objectives of both management and employees. These objectives may often be conflicting in nature. For instance, management requires that manpower requirements be met; preferably the work-force level should always exactly match the work requirements period by period. Meeting this objective, however, would often require employees to work interrupted, or split shifts. The employee, however, desires a "reasonable" schedule with respect to work hours (and days off), e.g., he would prefer an uninterrupted daily work schedule (except for meals and breaks). In addition, there are days such as Saturday and Sunday which are generally preferred as desired "days off" by employees. Some sort of equity is thus required in assigning "days off" in employee schedules. Furthermore, individual employees may have schedule preferences due to family circumstances.

In addition, a seniority system may exist such that management may deem it desirable to give priority to the scheduling of employees in order of their seniority.

There are three primary areas of interest in managing a work force on a daily basis: 1) development of demand forecasts for the system, 2) development of employee requirement levels, and 3) development of individual schedules with respect to working hours and days off.

Forecasting may be accomplished in a number of ways. It could be directly in terms of employee requirements, or in terms of a related unit of work which can be converted to employee requirements. Examples might be telephone call volumes to develop telephone operator requirements or patient demand to develop nurse requirements.

The company will then be interested in devising a set of individual work schedules which in general, will provide that number of employees which most closely fits the requirement levels generated. The assignment of the individual work schedules requires the use of some sort of priority system. There can be a completely static solution (meaning a fixed schedule which does not change over time) for each employee only if the demand pattern cycle length is always one week and the demand load does not vary from cycle to cycle. These are rather restrictive assumptions, and even if there is ample justification for them, it may prove infeasible for some reason to generate and maintain a truly static schedule. (Perhaps even those employees with "good" permanent schedules would prefer an occasional change.)

Nevertheless, a static scheduling approach, with some modifications, is one approach to some scheduling problems.

The inflexibility inherent in static schedules makes it logical to seek methods for generating alternative schedules of a dynamic or changeable, nature. It is reasonable to consider a solution approach based upon a procedure which results in some degree of shift rotation. Rotation of shifts implies at least partial equality of schedule desirability due to its essential no-priority nature.

Since each employee feels the need to be treated fairly, it is essential to develop a means of achieving some degree of equity in assigning schedules.

There are three possible methods of achieving equality. The first consists of cycling each employee through the same series of shifts and days off as every other employee over an appropriately chosen cycle length. This is called a rotation cycle. However, while this is "fair" in that each employee works the same schedule as every other employee, but at different points in time, it may be highly inefficient in achieving employee satisfaction in terms of shift desirability. In addition, rotating shifts may create personnel adaptation problems. Many people cannot adjust their sleeping habits to working the midnight tour one week and the daytime tour the following week.

Another type of non-permanent priority system would exist if there were a completely random priority level assigned, which changed at fixed time intervals. This is an example of another non-preferential schedule assignment procedure because each employee has the same probability of receiving a given position in the tour-choosing queue.

An alternative approach which yields schedules which are fair in the sense that each employee has the same voice in choosing his tour and which simultaneously provides for a higher overall level of employee satisfaction by more closely matching an individual's tour requirements with his desires is one using a rotational priority concept. This approach assigns a cycling or rotating priority level to each employee over an appropriately chosen time interval. Thus while an employee may have the lowest priority at the beginning of the cycle, he has the highest priority at the end of the cycle. This method insures that no employee has any permanent priority over any other and thereby creates equality among the employees.

In general, the no-permanent-priority systems result in "fair" scheduling. At the opposite extreme from these "fair" schedule assignment procedures is the pure seniority system, in which the person with the most seniority has first choice. Such a system may, however, tend to result in fixed schedules which create preferred positions.

Since it may be a bit extreme to assign those employees with the lowest priority the most undesirable schedules week after week, it seems reasonable to consider a modification of this procedure. In particular, a compromise approach based on some suitable condition of rotation or rotation priority with a pure seniority system suggests itself. In this manner, those employees with the least seniority could occasionally work a more desirable schedule. Of course, a suitable weighting factor for each of the two system and cycle lengths must be determined, based on company policy.

The company's determination of manpower requirements will usually involve compromises between two major conflicting measures -- service and cost.

As in any industrial scheduling problem, it is important to define objective measures and constraints for the employee shift scheduling problem. Management and labor will naturally have different and sometimes conflicting objectives and constraints in mind.

One of management's primary objectives in a service organization is the realization of what it defines as "suitable" service to the customer at an economic cost. Since this thesis is directed primarily at service operations, in which the production of services cannot be inventoried or deferred nor can the demand rate for service be adequately controlled, the "ideal" workforce size should vary from hour to hour and day to day as demand fluctuates. Of course it is obviously not feasible in most practical situations to maintain a workforce which varies drastically from hour to hour. It is inherently necessary to offer a work schedule to an employee which has reasonable (or absolute) continuity with respect to an 8 hour work day. That is, the shift is not interrupted for more than two or three hours once it starts. Thus, in a practical environment, management will be forced to maintain an excessive workforce during a particular hour of the day in order to guarantee sufficient manpower to cover requirements during a preceding or succeeding hour.

Given two schedules which meet the established demand schedules, management would generally prefer the one which results in lower costs. This usually means the one requiring fewer employees. On the other hand, an individual would prefer that schedule, assuming equal pay, which better allows him to schedule outside activities (whether they be eating, sleeping, hobbies, or whatever else he likes or needs to do in his spare time) around it. Since individuals may have different preferences it is difficult to make a general statement about group preferences.

The minimization of the workforce level (in terms of man-hours) consistent with suitable service, then, is the criterion of major concern to management in achieving economic costs. But it is this minimization of workforce size that creates part of the conflict between management and labor. Some of the employees of such a minimum-size labor force would undoubtedly be required to work split shifts and tours with irregular shift start times.

This leads to a discussion of the objectives of the employees. Frequently, even their goals will prove to be conflicting, e.g., perhaps they are interested in as many three-day weekends as possible, but do not wish to work more than five consecutive days. Other examples of objectives held by employees are the desire for consecutive days off, single days off on weekends, early or late morning starting times, straight shifts, split shifts, afternoon or night shifts, and lunch and break times spaced evenly throughout the day. They might also be interested in the ability for individuals to trade assignments if this is mutually agreeable.

Some of these objectives may be treated either in the objective function or the set of constraints of a mathematical programming formulation of the problem. The point of view one takes determines the final formulation.

It is important to recognize that if all constraints to the employee scheduling problem are considered as rigid, a feasible or practical solution may not be possible. In such cases it is often necessary to turn a rigid constraint into part of the objective function. Consider the very important management constraint that all forecasted customer demands be met. This absolute constraint may be turned into a "goal" by merely placing a realistic penalty on any demands which are not met by the schedule. Similarly, the constraint that an employee should have his shift start time changed no more than "χ" times per month. This may be made into a "goal" by simply placing a penalty charge in the objective function for any start time changes over "χ" per month for any employee.

Thus it may be important to recognize that some of the "constraints" as posed to the scheduler should really be viewed as goals. This suggests strongly the use of a type of "goal programming" technique which may be used interactively by the decision maker to determine an optimal schedule.

While there are few absolutely necessary constraints with respect to schedules, for the purposes of this thesis all tours are constrained to consist of 5 shifts per week, allowing 2 days off per week. All tours start at the same time every day, there are no split shifts in any of the tours, all shifts are eight hours long, and all demands are met.

The attributes of the employee scheduling problem can be summarized as below:

1. Work load demand varies over a 24-hour day and a 7-day week.

2. There is a demand for immediate service that cannot be backlogged.

3. The demand load is met by employees who work according to a fixed schedule of daily shifts, i.e., the employees are not "on call" all the time.

4. There is a finite set of fixed work patterns constrained by legal requirements, work force or union approval, and practicality.

5. The work shifts may overlap in time.

6. The objective of the schedules is to most nearly match the work load demand pattern with the pattern of workers provided by the schedule.

The development of individual days off and working hours should be accomplished by a method which promotes employee satisfaction subject to management's imposed constraints. The problem is thus one of multiple objectives or goals. Any method utilized to generate work schedules should automatically produce those having several general properties of desirable schedules. Such properties include straight shifts, daytime work, evenly spaced breaks, etc.

The purpose of this thesis is to develop and test alternative heuristic methods of generating employee schedules that satisfy both management and the employee. For practical purposes it is assumed that the demand pattern is deterministic and given.

In order to facilitate understanding it is necessary to define some terms as they are used throughout this thesis. A shift is a single day's work. It generally consists of approximately 8 hours of work uninterrupted except for lunch and short breaks, in which case it is a straight shift. However, it may be interrupted for a longer period, in which case it is a split shift. Overlapping shifts are those which have some, but not all, duty periods in common. Two overlapping shifts generally have start times less than 8 hours apart.

A tour is a set of five shifts (to allow two days off per week) which comprises some employee's work for the week. They generally, but not always, will start at the same time every working day.

A rotation cycle results in each employee's working through the same series of shifts and days off as every other employee over a chosen cycle length. Employees assigned to a rotation cycle work one set of shifts for one time period, another set the next time period, etc. throughout the entire cycle length. These are rotating shifts.

In addition the terms manpower allocation, shift scheduling, and employee scheduling are used interchangeably throughout.

The remainder of the thesis is organized as follows: Chapter II is a brief review of relevant literature. Chapter III contains a discussion of the determination of manpower loading requirements based on forecasts and measures of variability of the demand process. Chapter IV contains a mathematical programming approach to the problem and a discussion of the shortcomings of such an approach. Chapter V discusses the approaches actually utilized in solving an example problem. The results are reported in Chapter VI. Finally, Chapter VII contains the conclusions reached and recommendations for future efforts in this area.

CHAPTER II

LITERATURE REVIEW

The initial problem to be solved in employee scheduling is determining which shifts should be worked to provide adequate service at an acceptable cost. However, after this has been decided, there remains the problem of combining these shifts to form tours. The literature seems to deal with one or the other of these problems, but not both simultaneously. In particular, most of the works include an implicit assumption that economically feasible shift start times have already been established. The aim of each of these is to create desirable tours from the pre-determined shifts, which generally are the conventional 8:00 a.m., 4:00 p.m., and 12:00 p.m. shifts. However, in many cases such an assumption will result in unnecessary costs. It may be more economical to use less conventional shift start times, and perhaps even shifts that overlap in time. The first group of works reviewed is of the tour building type, in which desirable tours are to be created from a set of pre-selected shifts. The next group of reviews deals with actually selecting the shifts to be worked.

Healy [7] in an early effort dealing with tour building developed several rules of thumb to assure economical service as well as tour desirability. These rules of thumb, coupled with an enumeration of all desirable tours (meaning those with consecutive days off), make it possible to employ a trial and error procedure to schedule employees, given the shift duration times. The procedure is applicable to situations in which shifts and the number of men per shift are rotated, or those in which shifts are rotated, but the same number of men per shift are needed.

Monroe [15] presents a manual optimization procedure by which the number of tours with consecutive days off is maximized. At most two steps are needed in determining what tours should be formed. Rothstein [16] presents a linear programming formulation of this procedure, which, because of the special structure of the constraints, automatically yield an integer solution.

Tibrewala, et al. [17] present an algorithm to minimize the total number of employees required to satisfy a given demand pattern when the tours include two consecutive days off per week. Baker [3] introduces an algorithm for the same purpose which is not sensitive to the size of the numbers given in the problem, as is the case with the Tibrewala, Phillippe, and Brown algorithm.

Maier-Rothe and Wolfe [14] introduce a system for rotational scheduling of a nursing staff. Alternative tour types were generated and the nurses as a group decided which they preferred.

In an application to a transportation system, Bennett and Potts [3] present an optimization procedure to create tours with such desirable features as Sundays off, three consecutive days off as often as possible, and consecutive day-off pairs as often as possible. This was accomplished by building tours from five shifts whose start times were identical or at least nearly equal.

As previously mentioned, the optimized solution to the employee scheduling problem really starts with a determination of which shifts should be worked. The next group of algorithms deal with this more fundamental problem, referred to as shift allocation.

In one of the pioneer works dealing with the type of scheduling problem of interest Edie [6] mentioned a heuristic method for choosing a set of shifts to be worked. He suggested the use of Gantt-type charts and trial-and-error techniques to juggle workers' start, finish, and break times so that a suitable service level is maintained at an economic cost.

Dantzig [5] suggested the following optimization via a linear programming formulation:

Let X_j = number of shifts of type j required, $j = 1,2,\ldots,m$.

$$S_{ij} = \begin{cases} 1, & \text{if time period i is busy for shift type j} \\ 0, & \text{if time period i is idle for shift type j} \end{cases}$$

D_i = required number of employees working during period i

n = number of time intervals per cycle

$$\text{Minimize} \sum_{j=0}^{m} X_j$$

$$\text{Subject to} \sum_{j=1}^{m} S_{ij}\, X_j \geq D_i \quad (i = 1,2,\ldots,n)$$

$$X_j \geq 0$$

Fractional values for the optimum X_j are possible and rounding (some up, some down) may produce a solution requiring more employees, although the difference is likely to be small.

Dantzig also mentions the a method of reformulating the problem as a standard transportation-type linear programming problem in order to take advantage of special procedures that permit rapid solutions.

Luce [11, 12, 13] developed a heuristic algorithm which assigns shifts which may start at any time and which may overlap. Shifts are chosen one at a time according to a decision rule which attempts to maintain an employee supply schedule which is interval-by-interval proportional to the demand forecast. Luce also proposes a computer-oriented assignment system which builds and assigns tours to individual employees, although again there is the assumption that economically feasible shift start times have been pre-established. The algorithms and some related techniques are explained in greater detail in Chapter VII.

In an effort to determine optimal shift schedules for telephone operators, Henderson and Berry [8, 9] developed a branch and bound algorithm for determining the minimum number of shifts required to meet a 24-hour varying demand. An integer linear programming formulation similar to the formulation proposed by Dantzig is used.

Henderson and Berry have also developed heuristic methods for telephone operator shift scheduling. They first determine what subset of all shifts should be chosen from in assigning shifts. (A shift is defined not only by start and finish times, but also break timings, and lunch period lengths.) There are three heuristic methods introduced for reducing the set of shifts: 1) the first results in a subset of shifts which are very dispersed in terms of start times; 2) the second has start times that are somewhat less dispersed (and it proved much less efficient), and 3) the last was a random shift subset selection procedure.

Next, to assign shifts, a linear programming solution (not necessarily integer) was obtained and modified by a heuristic procedure to obtain an integer solution. Each non-integer variable (representing the number of shifts of a certain type) was rounded up, and then one variable at a time was checked to see if it could be reduced while still maintaining feasibility.

Another heuristic starts from this point and attempts to further reduce the number of operators required by checking all possible ways of removing two of them and replacing them with one operator.

A third scheduling heuristic randomly selects work shifts until all demands are filled. Then the two-for-one exchange heuristic is used to improve this solution. The authors claim very good results in approaching the theoretical lower bound for operators.

Most of the scheduling problems of interest in industry are still solved by trial-and-error techniques. However, some of the newer algorithms (such as that developed by Luce) were developed for specific problems and are being implemented in some places.

In summary, there are two major problems to be solved in order to create a schedule to satisfy a given work load. The first is to determine which shifts should be worked, and the second is to build tours from sets of shifts. Unfortunately, none of the literature treats both problems simultaneously.

CHAPTER III

DETERMINATION OF MANPOWER REQUIREMENTS

Preliminary to the development of a detailed work schedule is the determination of manpower loading requirements. These requirements are derived from forecasts of demands to be made on the system. The demand forecast is the real starting point of the scheduling system. In some cases it will be directly in terms of actual employee hours; however, it will ordinarily be in terms of raw data, e.g., the number of telephone calls of various lengths received per unit of time by a telephone information service. Such raw demands are then converted to some units of work hour requirements.

For example, consider a telephone information service which provides directory assistance to telephone customers. Suppose the forecast predicts that there will be n calls of average length $\bar{\ell}$ over a given time interval. Then one relevant measure of expected work content could be $n\bar{\ell}$. This figure for expected work content could then be converted to expected (or predicted) work requirements in terms of employees.

Management may build a safety factor into the analysis by choosing a desirable staffing level above the predicted requirement. Thus the forecasts and conversion may indicate an expected loading requirement of 10 persons at a certain point in time (i.e., 6:00 - 6:15 a.m. on Monday morning), but management may choose to build a 10% safety factor into the analysis by setting a desired manpower loading of 11 employees for this time period. On occasion, management may also wish to purposefully understaff during certain time periods.

There are a number of methods by which the forecasts can be generated. Techniques such as regression, moving average, and exponential smoothing methods are all useful in developing a distribution pattern of manpower requirements from historical data.

The forecasting system may occasionally receive extraordinary signals as input. This might occur if a telephone information group at one branch office temporarily handles the calls of another group elsewhere. Or perhaps a larger-than-normal demand might be imposed on a telephone system on a holiday. The effect of such one-shot events should be accurately predicted by the forecasting system and yet should have little or no effect on future demand forecasts.

Even though the demand is actually a random variable with a probabilistic distribution, a small variance makes it possible to use a deterministic approximation of the demand. In most cases the simplification introduced by this approach far outweighs the disadvantage of decreased accuracy.

From the raw data forecasts, manpower requirements can be determined. A plot may be drawn of these employee requirements vs. time. An example plot is shown in Figure 1. Note that there may be extreme fluctuations in this plot of employee work requirements. These fluctuations in fact cause great difficulty in the scheduling of shifts. If demand were constant over a 24-hour day and a 7-day week then the manpower requirements would also be constant over this time interval. (See

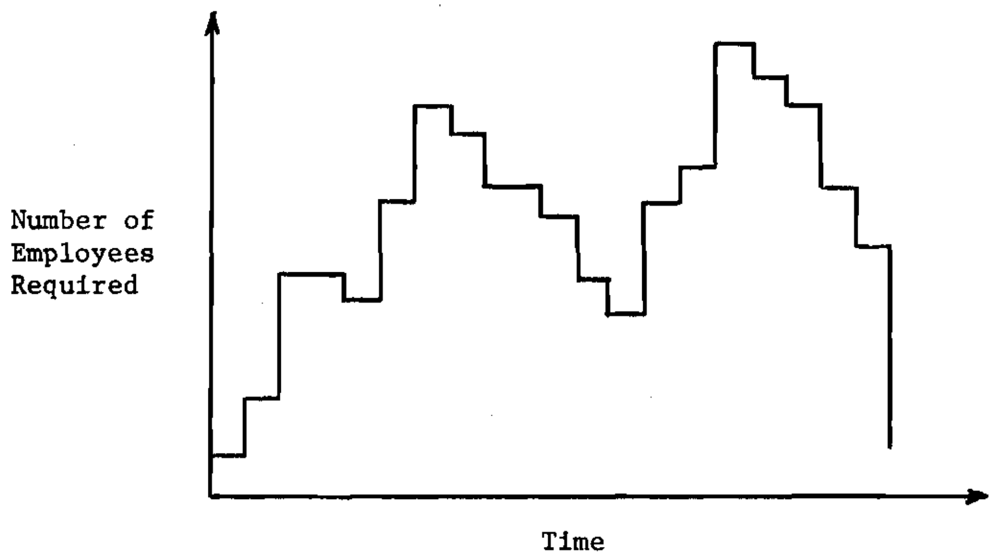


Figure 1. Sample Employee Requirements Graph.

Figure 2(a)). For such a case the shift scheduling problem is trivial.

In general a graph of the demand process vs. time is approximated by a group of step functions. Since one of management's objectives is to minimize the use of excessive manpower, a graph of manpower vs. time resembles this pattern. The next level (although still trivial) of difficulty encountered in scheduling shifts occurs when each step of the demand graph is sustained for integer multiples of the shift length. (See Figure 2(b).)

Assuming that it has been determined that each employee is to handle no more than k work-units, the absolute minimum number of employees (not necessarily an integer number) required to handle an incremental (decremental) step of n work-units of $\frac{n}{k}$. If $\frac{n}{k}$ is an integer quantity and if demand is sustained for an integer multiple of

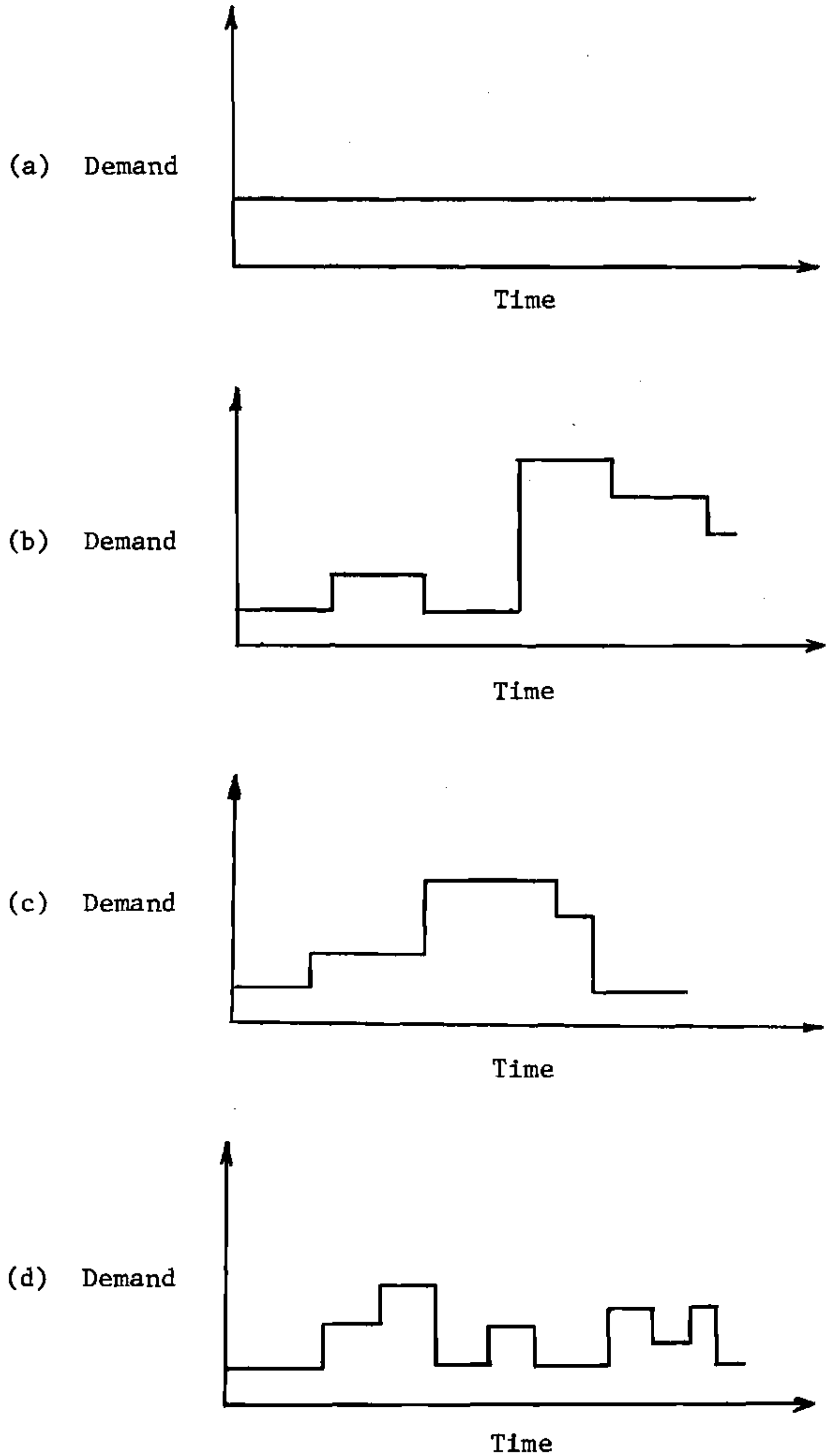


Figure 2. Levels of Scheduling Difficulty.

the shift length, then no excess manpower usage is incurred in transferring $\frac{n}{k}$ additional employees to (from) this step. Even if $\frac{n}{k}$ is non-integer, the shift scheduling problem is trivial for this case.

The next level of difficulty occurs when the demand vs. time graph resembles a rising and then falling staircase but with possibly differing step lengths and heights (see Figure 2(d)). The result is a scheduling problem in which it may be possible to avoid incurring any excess manpower costs, but which is no longer trivial to solve.

Finally, the most difficult problem is one which has a random step pattern (see Figure 2(d)). In many cases excess costs will necessarily be incurred because it is infeasible to match the demand vs. time graph with a manpower requirements vs. time graph that is parallel to it. To do so would require the utilization of split shifts that might prove illegal or impractical.

In order to characterize the nature of the variability of manpower requirements in a given shift scheduling problem, it is useful to develop measures of this variability. One simple measure of this is the number of times the requirements graph crosses the (horizontal) mean manpower requirement line.

Another measure is the relative frequency of the steps encountered. Each step encountered increases the possibility of overshooting the minimum manpower level. As discussed previously the shift scheduling problem is trivial if a constant demand level is maintained or if demand levels remain constant over the entire shift. More frequent changes in demand mean more steps, which may result in more excess manpower usage.

A third measure might be the number of peaks (valleys) above (below) an arbitrarily chosen horizontal line. A peak (valley) is defined as a local maximum (minimum) demand sustained for exactly one time period. The presence of peaks and valleys contribute greatly to the difficulty in minimizing excess manpower.

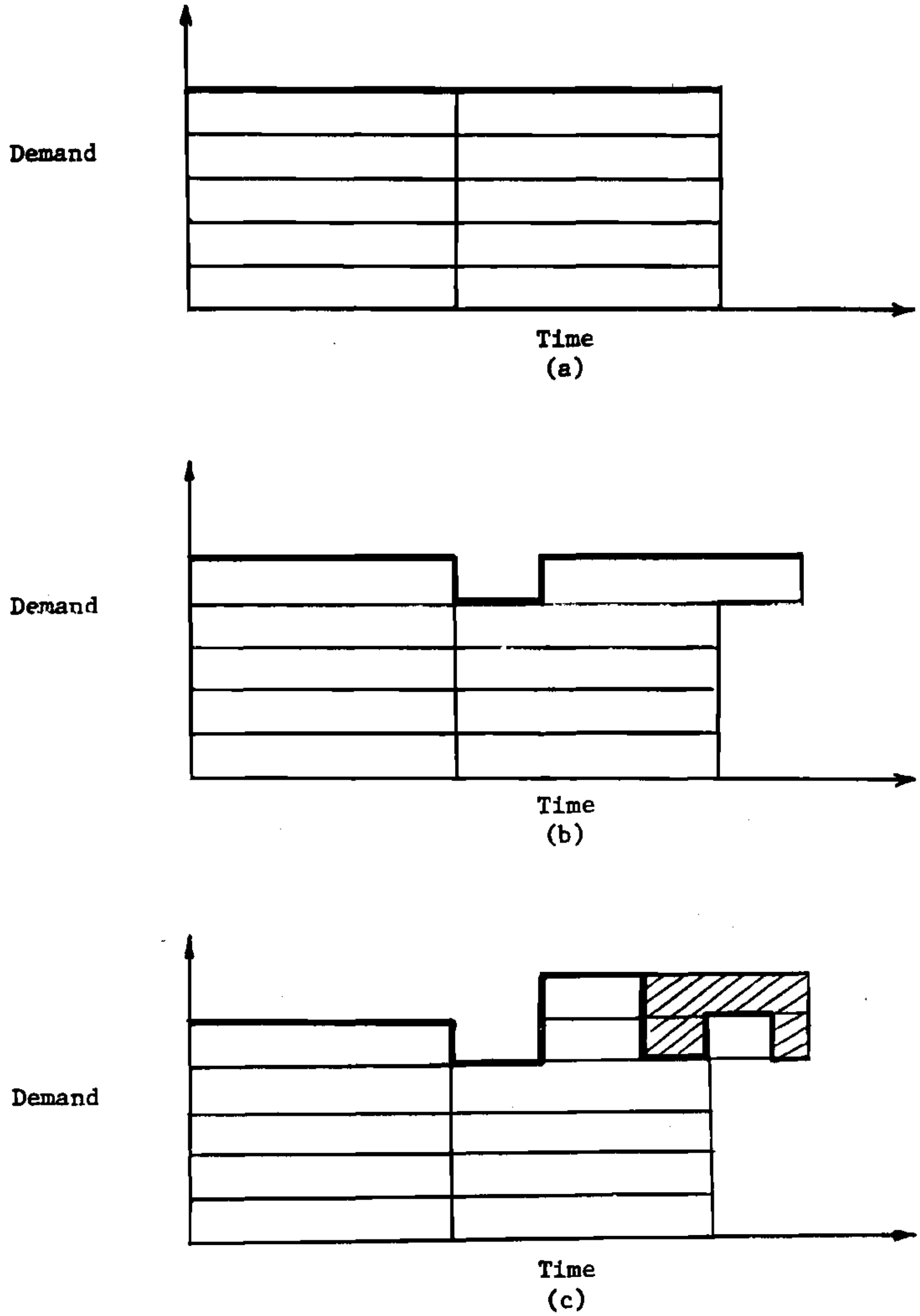
A more elaborate measure of the manpower variability level is the sum of the squares of the deviations from the mean manpower level ($\bar{x}$) of the actual manpower level (X_i). If there are n time periods this could be written as $VL = \sum_{i=1}^{n} (X_i - \bar{x})^2$.

A modification of this same measure is a weighting based on the length of time for which the deviation occurs. If t_i is the length of the time interval over which the deviation occurs then,

$VL' = \sum_{i=1}^{n} \frac{(X_i - \bar{x})^2}{t_i}$. It will be noticed that if the interval length of a deviation is large, then the effect of the deviation is reduced. This is a desirable attribute, since it is the peaks and valleys which occur only for short-time intervals which contribute so much to scheduling difficulties.

One or more of these measures may prove useful in predicting how much excess manpower costs will be incurred. They may also indicate the relative difficulty of the shift scheduling problem for a given demand distribution.

As an indication of how demand variability affects the scheduling of employee shifts, consider the three sample plots given in Figure 3. Note that in Figure 3(a) the demand requirements are fixed at 5 persons



(Cross-hatched area represents idle time.)

(The rectangles represent shifts.)

Figure 3. Sample Manpower Loadings.

over the entire planning period. This is a trivial scheduling problem and a suitable manpower loading is given. In Figure 3(b), there is a single "step" in the manpower requirements. This may still be a trivial problem in that only one aberration is required in the schedule. A suitable manpower loading is shown. However as can be seen in Figure 3(c), when alternating "steps" are present in the demand requirements, the scheduling becomes nontrivial. If 8-hour shifts are assumed, then the schedule of Figure 3(c) will necessarily include employee idle time if all demand requirements are met. This implies that more personnel will be required in the schedule of Figure 3(c) even though the total work demands of Figure 3(a), Figure 3(b) and Figure 3(c) are the same.

This is precisely why some measure of demand variable is important. Although two demand schedules may be equal in their total work content, one may require more personnel than the other due to a difference in the nature of the variability of the demand requirements.

CHAPTER IV

PROGRAMMING APPROACH TO EMPLOYEE SCHEDULING

The employee shift scheduling problem has been formulated as a linear programming problem by several researchers.

The basic formulation first proposed by Dantzig [5], is as follows:

Let X_j = number of shifts of type j

C_j = cost of shift type j

D_i = demand for time period i

$$S_{ij} = \begin{cases} 0, & \text{if time period } i \text{ is idle for shift type } j \\ 1, & \text{if time period } i \text{ is busy for shift type } j \end{cases}$$

$i = 1, 2, \ldots, n$

$j = 1, 2, \ldots, m$

where n is the number of time periods

m is the number of shift types

The problem to be solved is:

$$\text{Minimize } \sum_{j=1}^{m} C_j X_j$$

$$\text{Subject to } \sum_{j=1}^{m} S_{ij} X_j \geq D_i \quad i = 1, 2, \ldots, n$$

$$X_j \geq 0$$

The cost coefficients C_j can be actual wage costs based upon length of shift or differential wage rates or can be used as weighting factors to favor certain types of shifts. Only if C_j is constant for each j does the objective function guarantee that a minimum number of shifts will be utilized. If this condition is not met the objective function will instead minimize the total cost of the shifts chosen. However, this is a desirable goal, since presumably the primary reason management desires to minimize the total number of shifts used is to minimize costs.

The constraints insure that all the demands are at least met and possibly overshot. An additional constraint could fix or limit the number of a particular shift type. It would be of the following form:

$$X_j = b_j$$

$$\text{or} \quad X_j \geq \ell_j$$

$$\text{and/or} \quad X_j \leq u_j$$

Several immediate problems develop with the problem formulation. As m and n increase, computation time increases. The obvious solution to this difficulty is to reduce m and n as much as possible. No limit is set on the overshoot above D_i. This calls for a set of slack variables added to the constraints to allow only a specified amount of deviation in any particular time period. The solution is in terms of continuous numbers of shifts when only an integer solution is meaningful. The requirement for an integer solution could be satisfied by using an integer programming technique, or by rounding, truncation or some heuristic modification of the numbers. Optimal integer programming solutions are more accurate, but much more expensive to obtain than the rounded solutions.

A reformulation of the program is as follows:

$$\text{Minimize} \sum_{j=1}^{m} C_j X_j$$

$$\text{s.t.} \quad \sum_{j=1}^{m} S_{ij} X_j \leq D_i + K_i \qquad i = 1, 2, \ldots, n$$

$$\sum_{j=1}^{m} S_{ij} X_j \geq D_i - k_i \qquad i = 1, 2, \ldots, n$$

$$X_j \geq 0 \qquad j = 1, 2, \ldots, m$$

where K_i and k_i are the amounts of over- and under-shoot, respectively, allowed.

For this formulation, there exists the possibility that there is no feasible solution. Thus several attempts with increasing K_i and/or k_i may be necessary to obtain a feasible solution.

Tour-Building

Given a group of shifts which are "efficient" in terms of matching forecast manpower requirements, there still remains the intermediate step of grouping them to form acceptable tours for assignment to individual employees. One method involves the use of an iterative procedure to create acceptable tours from those shifts chosen by a shift scheduling algorithm. The tour generation can be provided by use of the following linear programming formulation suggested by Rothstein, [16] as an extension to Monroe's [15] "arithmetical procedure" for assigning personnel.

Let x_1 = number of Mon. - Tue. regular-day off (RDO) pairs

x_2 = number of Tue. - Wed. RDO pairs

.
.
.

x_7 = number of Sun. - Mon. RDO pairs

b_1 = required RDO's on Tue.

b_2 = required RDO's on Wed.

.
.
.

b_7 = required RDO's on Mon.

u_1 = number of Tue. non-consecutively paired RDO's

u_2 = number of Wed. non-consecutively paired RDO's

.
.
.

u_7 = number of Mon. non-consecutively paired RDO's

d = number of workers assigned to non-consecutively paired RDO's.

Maximize $Z = \sum_{i=1}^{7} x_i$

subject to:

$$
\begin{array}{llr}
x_1 + x_2 & + u_1 & = b_1 \\
x_2 + x_3 & + u_2 & = b_2 \\
x_3 + x_4 & + u_3 & = b_3 \\
x_4 + x_5 & + u_4 & = b_4 \\
x_5 + x_6 & + u_5 & = b_5 \\
x_6 + x_7 & + u_6 & = b_6 \\
x_1 + x_7 & + u_7 & = b_7 \\
x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 & + d & = \frac{1}{2} \sum_{i=1}^{7} b_i
\end{array}
$$

$$
\begin{array}{r}
u_1 - u_2 - u_3 - u_4 - u_5 - u_6 - u_7 \leq 0 \\
- u_1 + u_2 - u_3 - u_4 - u_5 - u_6 - u_7 \leq 0 \\
- u_1 - u_2 + u_3 - u_4 - u_5 - u_6 - u_7 \leq 0 \\
- u_1 - u_2 - u_3 + u_4 - u_5 - u_6 - u_7 \leq 0 \\
- u_1 - u_2 - u_3 - u_4 + u_5 - u_6 - u_7 \leq 0 \\
- u_1 - u_2 - u_3 - u_4 - u_5 + u_6 - u_7 \leq 0 \\
- u_1 - u_2 - u_3 - u_4 - u_5 - u_6 + u_7 \leq 0
\end{array}
$$

$x_i \geq 0 \quad i - 1,2,\ldots,7$

$u_i \geq 0$

$d \geq 0$

All variables integral.

The use of this formulation assures that one important employee objective is met -- that as many consecutive RDO's as possible are assigned. In order to insure that the tours thus generated are acceptable also on the basis of a uniform shift start time, the following iterative scheme is used: The shift scheduling algorithm uses forecast manpower requirements data to determine a "good" set of shifts to closely match these requirements. The tour generation algorithm is used for each shift start time to create tours which not only have a uniform shift start time but which also have the attribute of a maximized number of consecutive RDO pairs.

In some cases there will be a requirement for a non-integer workforce level, for example when the shift allocation algorithm calls for 32 shifts to start at 8:00 a.m. for a week. Since each tour consists of 5 shifts, there will be 6 complete tours with 2 additional shifts called for. A similar problem may also arise elsewhere, such as for the 9:00 a.m. shift. In order to avoid the excess manpower usage that would result from rounding each of these (and others with similar situations) up to the next highest integer number of tours, it may prove beneficial to round down in each case.

After the complete tours are formed, the manpower coverage they provide is subtracted from that required (as forecast). Then the shift scheduling algorithm is applied to the new manpower requirements data to repeat the process. Termination occurs when no complete tours are created twice in succession.

The trouble with this approach is that a very large number of small linear programming problems must be solved, and unfortunately it is impossible to determine the exact number, due to the iterative nature of this approach

In addition, there are computer implementation difficulties caused by the indeterminate nature of some of the right-hand-side constants of the intermediate iterative steps.

Because of the difficulties encountered with a linear programming formulation of the problem, an alternative approach utilizing heuristic procedures for generating tours may be desirable. Such a method should be capable of yielding a first-pass feasible integer solution, unlike the linear programming formulation.

CHAPTER V

HEURISTIC APPROACHES

The difficulties encountered in attempting "exact" solution procedures (such as linear programming) suggest the use of heuristic techniques for determining shift allocations and building tours. In this chapter several alternative heuristic approaches will be described.

At this point, it might prove beneficial to restate the assumptions made and constraints imposed regarding the employee scheduling problem:

1. Work load demand varies over a 24 hour day and a 7-day week.
2. Demand cannot be backlogged.
3. The employees work according to a fixed schedule of daily shifts.
4. There is a finite set of fixed work patterns.
5. The work shifts may overlap in time.
6. The objectives of the schedules is to most nearly match the work load demand pattern with the pattern of workers provided by the schedule.
7. The demand pattern is deterministic and given.
8. All tours include 2 days off per week.
9. All tours start at the same time every day.
10. All shifts are 8 hours long and not interrupted.
11. All demands are met.

There are essentially two approaches to heuristic employee scheduling. The first is to break the problem into two problems - a shift allocation problem and a problem of building tours from the allocated shifts. The second approach, called tour scheduling, simply combines the two steps, i.e., there is no separate shift allocation.

Shift Scheduling - Tour Building (SS-TB)

One measure of progress in allocating employees for a shift scheduling algorithm is the number of workers, W_i, assigned to the i^{th} time interval. The criterion to be minimized in determining a good schedule is excess workforce, given by $\sum_{i=1}^{n} (W_i - D_i)$, where D_i is the manpower demand (or requirements) level during time period i, and n is the number of periods in the schedule length. The desired result is a step function of W_i values which over any interval of time will approximate the step function of D_i values. At any stage in the allocation process, since the shifts are chosen one at a time, there exists some remaining distance between the D_i and the W_i, as in Figure 4.

The main problem encountered in choosing shifts one at a time is overshooting the demand level in buildup and drop-off periods surrounding peak demands. This is due to choosing too many shifts too soon to cover the peaks early in the building process.

In deciding which shift to allocate next, two different approaches will be discussed. The first is essentially a priority selection scheme with several different (calculated) priority indices. The second approach consists of allocating shifts essentially on a "first-demanded, first-supplied" basis.

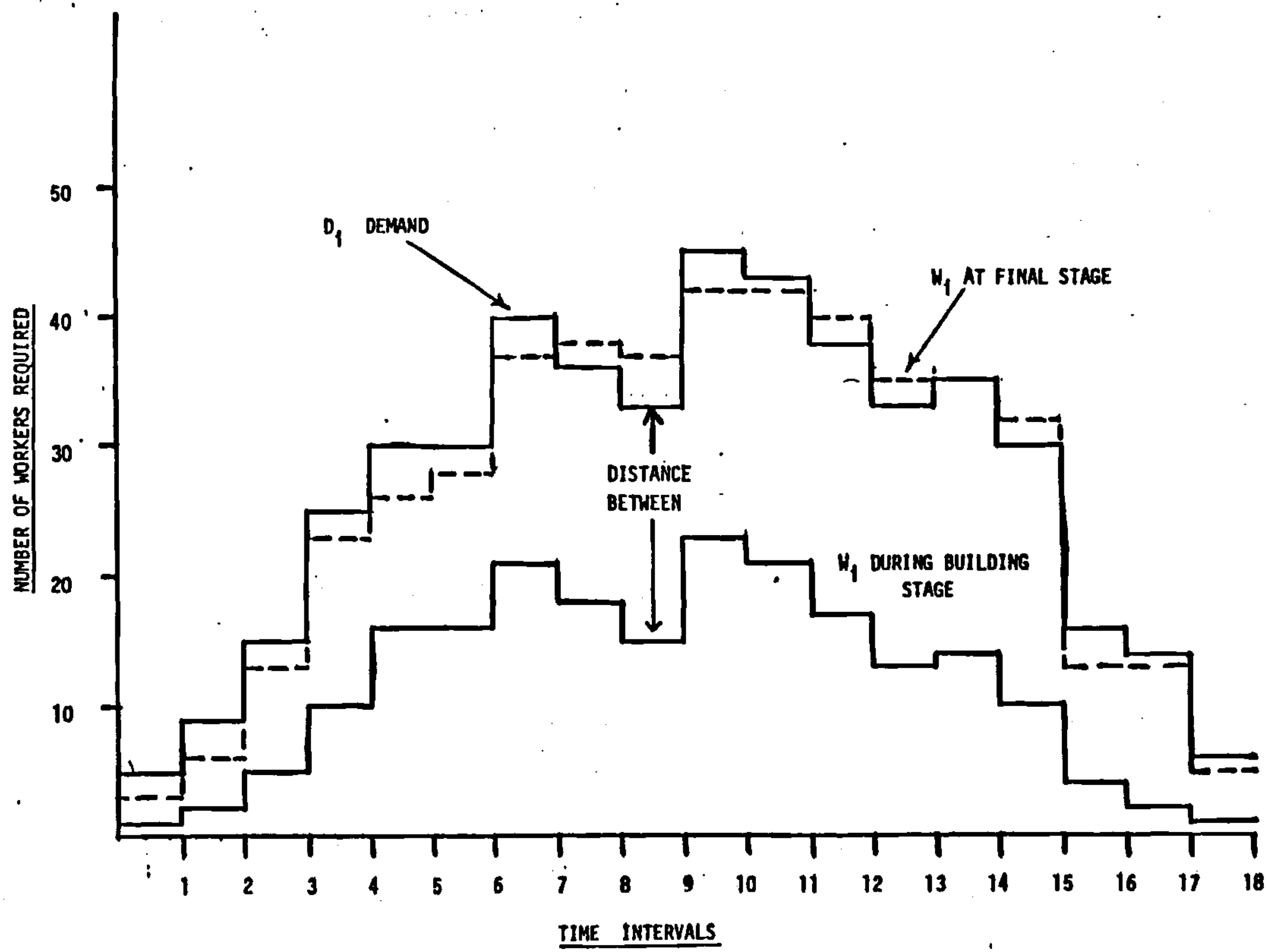


Figure 4. Illustration of the Shift-Scheduling Process (after Luce [11]).

For the calculated priority selection scheme, one way of choosing the next shift is to select that one which maximizes the value of $\sum_{i=K}^{K+7} (D_i - W_i)$ for K = 1,2,...,n-7. (The upper limit of summation is K+7 because all shifts are assumed to cover 8 time periods.) For a shift i, this measure is an indication of the "need" for adding another employee to that shift, since the sum represents the total man-hours of shortage over the time interval covered by that shift. It seems reasonable to allocate a worker where the current need is greatest. The disadvantage of this decision rule is that the size of the peak D_i can influence the early shift choices so much that the problem of overshooting demands occurs.

A refinement which helps to negate the size of the D_i as a factor in shift choice is to choose that shift which minimizes $\sum_{i=K}^{K+7} W_i/D_i$, for K = 1,2,...,n-7. A closely related method consists of temporarily augmenting each shift by one employee. The permanent allocation is made to that shift which minimizes $\sum_{i=1}^{n} W_i/D_i$, where the W_i for the shift are temporarily increased by one unit. The ratio decision rule (with either set of limits of summation) represents a different way of defining a need for more employees over a given time interval. It adds a worker to that shift where the current ratio of supplied employees to demanded employees is least.

Another decision rule could be based on squaring the difference between D_i and W_i. Such a decision rule consists of adding that shift which minimizes $\sum_{i=1}^{n} (D_i - W_i)^2$ after temporary shift allocations have been made or which maximizes $\sum_{i=K}^{K+7} DSQ_i$ for $K = 1,2,\ldots,n-7$.

$$\text{where } DSQ_i = \begin{cases} (D_i - W_i)^2 & \text{for } D_i \geq W_i. \\ -(D_i - W_i)^2 & \text{for } D_i < W_i. \end{cases}$$

These rules behave quite similarly and represent the sum of the squares of the differences in demand and supply. This rule, which is intuitively appealing, results in allocating workers first to those time periods during which they appear to be most needed. As can be seen, both rules include a built-in penalty for over-supplying a time interval.

One method of combating the problem of overshooting demands is to attempt to create a step function of W_i values which at any point in the building process is approximately proportional to the step function of D_i values over each time interval. Such a heuristic has been proposed by Luce [11,12,13]. (The strategy of his solution techniques is to choose one shift at a time from the set of all acceptable ones to allocate a group of shifts). As before, the measurement of scheduling progress is the number of workers W_i, assigned to the i^{th} time interval.

In order to combine the advantages of the percentage term W_i/D_i with the smoothing effect of squaring differences in current manpower supply and demand, Luce suggests calculating $P_i = (1 - W_i/D_i)^2$. Then that shift is chosen which maximizes $\sum_{i=K}^{K+7} P_i$, for $K = 1,2,\ldots,n-7$.

Since P_i is squared it will always be positive, a characteristic desirable for $D_i > W_i$, but one which results in uncontrollable shift choices if any W_i becomes greater than the corresponding D_i. Thus the following modified definition is utilized:

$$P_i = \begin{cases} (1 - W_i/D_i)^2 & \text{for } D_i \geq W_i. \\ -(1 - W_i/D_i)^2 & \text{for } D_i < W_i. \\ 0 & \text{for } D_i = 0. \end{cases}$$

For values of $W_i = 0$, P_i will equal 1, the maximum possible value. This feature assures that at least one shift will be chosen which places a worker in each time interval as early in the allocation process as possible. Such a characteristic is particularly valuable in diverting shift choosing activity away from the peak demands early in the allocation process. This helps avoid overshooting later in the process.

A somewhat different shift allocation heuristic procedure of the "first-demanded, first-supplied" type can be based on choosing a starting time period and allocating shifts in that time period until demand is met. The next shift start time is the first subsequent period for which $D_i > W_i$. Shifts are allocated starting in that period until all demand is met. The process is repeated until all demands are met.

A refinement to this procedure is to use each time interval successively as a starting point. That starting point and set of shifts with the least number of elements is then chosen as best, since the objective is to minimize the number of shifts utilized.

The intuitively appealing scheduling rules mentioned can be formalized into several shift scheduling algorithms, as follows:

Sequential Demand Satisfaction Algorithm

Let X_t = the number of shifts starting at time t

n = the total number of time periods in the cycle

1) Choose time period T = 1 as a starting point
2) If $W_t < D_t$ increase X_t until $W_t = D_t$ for $t = T$.
3) Increment t and repeat 2) until $W_t \geq D_t$ for $t = T,2,\ldots,n$ and $t = 1,2,\ldots,T$.
4) Increment T and repeat steps 2) and 3) until T = n.
5) Select the shift allocation scheme for which $\sum_{t=1}^{n} X_t$ is a minimum.

Luce's Shift Selection Algorithm

Let

$$P_i = \begin{cases} (1 - W_i/D_i)^2 & \text{for } D_i \geq W_i. \\ -(1 - W_i/D_i)^2 & \text{for } D_i < W_i. \\ 0 & \text{for } D_i = 0. \end{cases}$$

n = the number of time periods.

1) Maximize $\sum_{i=K}^{K+7} P_i$ for $K = 1,2,\ldots,n-7$.
2) Assign the shift for which ΣP_i is maximized.
3) Repeat (1) and (2) until $W_i \geq D_i$ for $i = 1,2,\ldots,n$.

The Difference Squared Shift Selection Algorithm

Let
$$DSQ_i = \begin{cases} (D_i - W_i)^2 & \text{for } D_i \geq W_i \\ -(D_i - W_i) & \text{for } D_i < W_i. \end{cases}$$

n = number of time periods

1a) Minimize $\sum_{i=1}^{n} (D_i - W_i)^2$

1b) Maximize $\sum_{i=K}^{K+7} DSQ_i$ for $K = 1,2,\ldots,n-7$.

2) Schedule the shift which extremizes (optimizes) the appropriate measure.

3) Repeat (1a) or (1b) and (2) until $W_i \geq D_i$ for $i = 1,2,\ldots,n$.

Note that there are two alternatives regarding the limits of summation for this shift selection decision rule. Perhaps the most obvious is to sum the argument for an 8-hour interval (representing a typical 8-hour working day) and select that distinct interval which extremizes this sum. The other approach is summation of the argument over the entire cycle length and selection of that temporarily augmented shift which results in the appropriate extreme value. Of course, it may be necessary to extremize a measure in the opposite sense to accomplish similar results if the limits of summation differ. For example, maximization of $\sum_{i=K}^{K+7} (D_i - W_i)^2$ for $K = 1,2,\ldots,n-7$ behaves similarly to minimization of $\sum_{i=1}^{n} (D_i - W_i)^2$ (after temporary augmentation of each shift). To understand why this is so, one merely needs to consider the idea behind the decision rules. It is desired to allocate a worker to that shift which needs him most. Maximizing $\sum_{i=K}^{K+7} (D_i - W_i)^2$ essentially adds a worker to the shift which has the most total "shift urgency." On the other hand minimizing $\sum_{i=1}^{n} (D_i - W_i)^2$ adds a worker to that shift which (after being temporarily augmented) results in the least remaining total "cycle urgency." Since both rules are motivated by the same logic, it is reasonable that they behave similarly.

It should be noted that none of the algorithms presented include the technique of back-tracking. The use of this technique would require discarding certain of the shifts chosen after the first pass at the problem and resuming calculations to determine which, if any, shifts would be necessary to satisfy any remaining demands incurred by discarding the selected shifts. The inclusion of this process could possibly result in substantial manpower savings.

After deciding which set of shifts to schedule, it is necessary to combine them in groups of five to form tours.

A heuristic tour-building algorithm consists of developing tours on the basis of the number of daily shift allocations for a given shift start time. One tour at a time is formed by designating as days-off those two days (per week) which are assigned the fewest remaining number of shifts for each start time. Then the remaining number of shifts on each of the five working days is decremented by one shift, and the process is repeated until all shifts have been combined to form tours.

What this process basically does is to build one tour at a time from a set of five shifts, each of which is selected from one of the five days of the week with the most remaining shifts to be allocated to tours.

Next, in order to attempt to satisfy the desire for consecutive days off, each tour is inspected to see if the two days off are consecutive. If so, the tour is left intact; otherwise it is broken into its component shift parts and added to a group of other similarly disassembled tours.

After all tours have been inspected and either saved or broken apart, the tour creation algorithm is again applied to the shifts comprising the undesirable (disassembled) tours to re-create tours. The expectation is to create tours with consecutive days off from the group with non consecutive days off.

This entire process is repeated until no new consecutive-day-off tours are created. At this point all the current tours, even though they may have day-off pairs that are non consecutive, are permanent.

Tour Scheduling

Rather than create schedules for employees by first scheduling shifts and then combining these shifts to form acceptable tours, it is possible to schedule tours directly, each with the property that a fixed start time is observed for the entire week and each including 2 days-off per week. In fact, all the assumptions and constraints for the employee scheduling problem hold for tour-scheduling as well as for shift scheduling and tour-building. The solution techniques of such a tour-scheduling algorithm consists of choosing five shifts (one tour) at a time from the set of all shifts to create employee schedules.

This procedure is very similar to the shift-scheduling algorithm in that both attempt to create employee staffing levels which are proportional to manpower requirements levels with a limited and sometimes insufficient number of man-hours. The difference between the two algorithms is that the tour-scheduling algorithm assigns five shifts at a time, in the "best" (according to the decision criterion) places, instead of one at a time assigned by the shift-scheduling algorithm.

The measure of progress is the supply of workers, S_i, assigned to the i^{th} interval. Again the criterion for a good schedule is to minimize $\sum_{i=1}^{n} (S_i - D_i)$, where D_i is the forecast manpower demand level during time period i, and n is the number of time intervals in the cycle length. It is required that the difference between demand and supply be minimized at each step of choosing a tour. As a result it is reasonable to assume that the step function of S_i values will, over any interval of time, closely resemble the step function of D_i values.

At each stage there exists some difference between the D_i and the S_i. Using the final decision criterion of the shift-scheduling algorithm rather than the intermediate refinements, $TP_i = (1 - S_i/D_i)^2$ for $i = 1,2,\ldots,168$ is calculated. The tour chosen is the one which maximizes $\sum_{i=K}^{K+7} TP_i$ for $K = 1,2,\ldots,161$. This tour choice criterion possesses all the advantages of the shift-selection criterion; namely, a partial negation of the D_i as a factor in tour choice in order to avoid over choosing tours to cover peak demand periods, as well as the smoothing effect of squaring $(1 - S_i/D_i)$.

Let

$$TP_i = \begin{cases} (1 - S_i/D_i)^2 & \text{for } D_1 \geq S_i \\ -(1 - S_i/D_i)^2 & \text{for } D_i < S_i \\ 0 & \text{for } D_i = 0 \end{cases}$$

For values of $S_i = 0$, TP_i will equal 1, the maximum possible value. This feature assures that at least one tour will be chosen which places a worker in each time interval as early in the building process as possible. This characteristic helps divert tour choosing activity away from peak demands early in the process and thereby helps avoid overshooting later. Tour selection ends when $S_i \geq D_i$, for $i = 1,2,\ldots,n$.

With a few changes in notation from the above the Tour Scheduling Algorithm can be formalized.

The Tour Scheduling Algorithm

Let d_1 = first day of week off $d_1 = 1,2,\ldots,6$.

d_2 = second day of week off $d_2 = d_1 + 1, d_1 + 2,\ldots,7$.

D_{ij} = manpower demand for time period i, day j

S_{ij} = manpower supply for time period i, day j

m = the number of different shift start times per day.

$$TP_{ij} = \begin{cases} (S_{ij}/D_{ij})^2 & \text{for } D_{ij} \geq S_{ij} \\ -(1 - S_{ij}/D_{ij}) & \text{for } D_{ij} < S_{ij} \\ 0 & \text{for } D_{ij} = 0 \end{cases}$$

1) Maximize $\displaystyle\sum_{l=K}^{K+7} \sum_{\substack{j=1 \\ j\neq d_1 \\ j\neq d_2}}^{7} TP_{ij}$ for $K = 1,2,\ldots,m$.

2) Assign the tour for which $\sum_i \sum_j TP_{ij}$ is maximized

3) Repeat 1) and 2) until $S_{ij} \geq D_{ij}$ for $i = 1,2,\ldots,n$
$j = 1,2,\ldots,7.$

Summary

In this chapter, several heuristic methods for solving the employee scheduling problem have been described. A two phase heuristic is obtained by combining any one of the shift allocation heuristics with the tour building heuristic. The tour scheduling heuristic is a one step procedure.

CHAPTER VI

RESULTS

In order to test the usefulness of the scheduling algorithms presented, a series of problems obtained from data furnished by the Southern Bell Telephone Company was solved. The forecasts of system demand are based on cell volumes observed from historical data. The raw data in the form of call colume figures were converted into manpower staffing requirements levels by application of standard performance conversion ratios. The scheduling algorithms all use these required staffing levels as input. A graph of a typical day's manpower requirements level is shown in Figure 5. Although demand was forecast over each 15-minute interval by Southern Bell, it was decided for the purpose of this thesis to base solutions over hour-long time intervals using the average of ths four 15-minute intervals as the requirement for the 1-hour interval.

Only the data of Problem 1, which is displayed in the Appendix, was actually furnished by Southern Bell. From this data it can be seen that there are usually two pronounced peak demand periods per day, corresponding to peak loads on the system. In addition, call volumes are much higher during the week than on the weekend. Otherwise the demand patterns are similar from day to day. Therefore one would expect that scheduling difficulties in terms of avoiding excess manpower usage would be similar. (See Table 1 for measures of the weekly manpower requirements variability.)

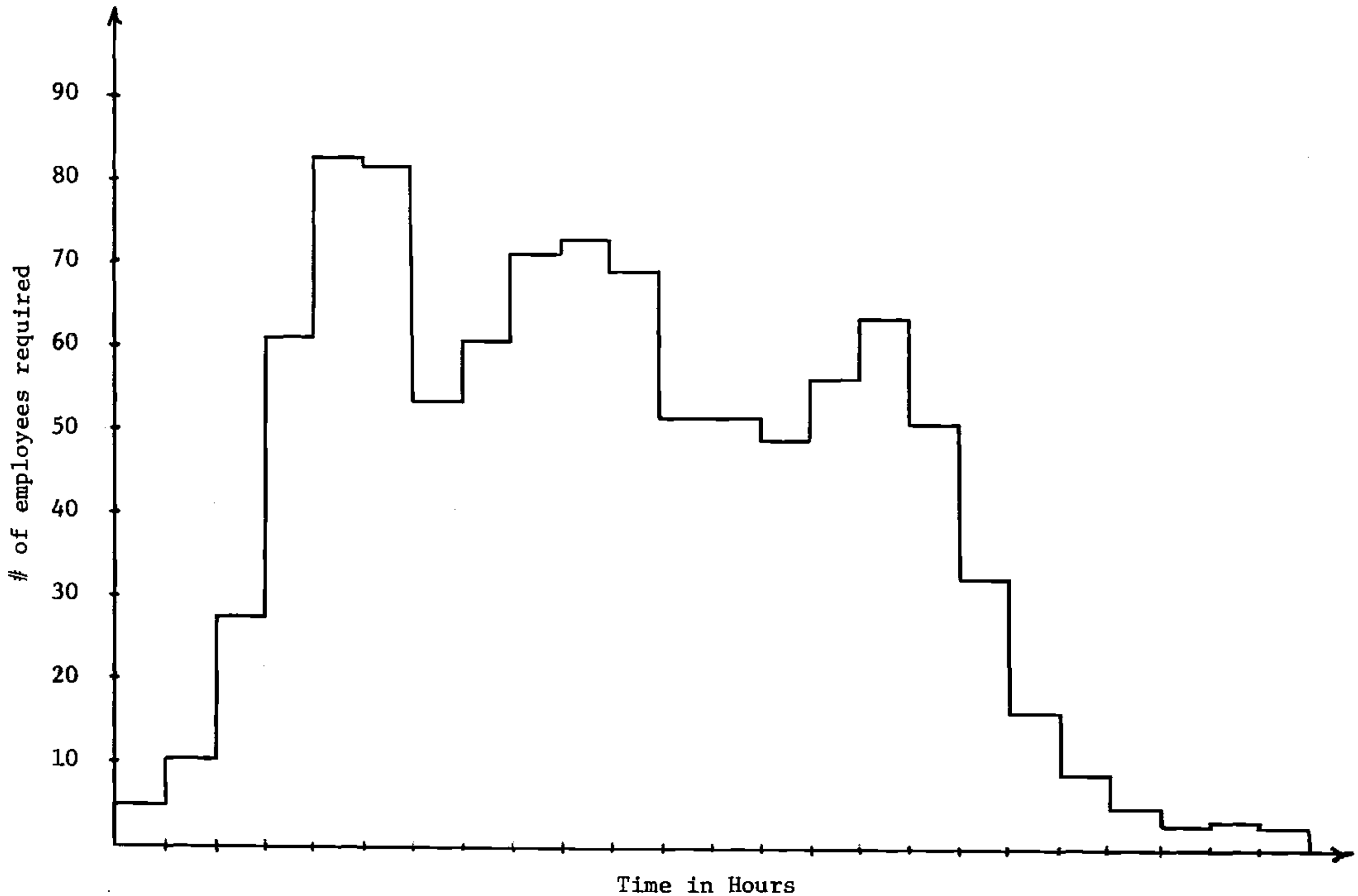


Figure 5. Typical Day's Demand Profile.

The data sets of Problems 2-6 were constructed from the data of Problem 1 by modifying it in such a way that a wide range of values was observed for each of the manpower variability measures. This was expected to simplify the analysis of the usefulness of each measure in predicting manpower staffing excesses for the algorithms presented. (See the Appendix for the date to Problems 2-6 and Table 1 for measure of variability of these problems.)

Table 1. Measures of Manpower Requirements Variability

Problem	Mean Crossings	Steps	Peaks	Valleys	$\Sigma(\bar{X}-X_i)^2$	$\Sigma \frac{(\bar{X}-X_i)^2}{t}$	Runs Above Median	Runs Below Median
1	14	150	20	15	15,994	14,336	14	20
2	14	161	24	19	15,979	15,979	16	20
3	14	38	0	0	15,024	3,546	6	14
4	14	21	0	0	880	216	1	14
5	70	74	0	2	9,922	4,467	34	19
6	68	82	1	0	1,978	1,064	40	37

Each problem solved has the same total daily manpower requirements, but the examples differ widely in their measures of manpower variability. Problems 1 and 2 have a large number of steps, peaks, valleys, runs above and below the median and a large VL and VL' (refer to Chapter IV). All the other problems, with the exception of Problem 5, have demand levels that do not fluctuate drastically from the mean daily demand level or from the demand level of the previous time period. They also have fewer steps, peaks, and valleys.

Results for Heuristics

Table 2 compares the solution efficiencies of the various SS-TB algorithms for each example problem. It can be seen that the Sequential Demand Satisfaction Algorithm performed best in each case. Luce's Algorithm was worst except for one case, and the other two algorithms usually performed very similarly to each other.

The best algorithm resulted in excess manpower usage ranging from 1.6% to 30.8% for the shift allocation step. However, for the type of demand experienced by the telephone company, about 16% excess usage was incurred.

The tour-building stage of the algorithm introduces much more required idle manpower due to the constraints that all tours should have a uniform shift start time and two days off during the week. (See Table 3.) For example, assume there is a need for 26 shifts to begin at 8:00 a.m. throughout the week. Since groups of five shifts are combined to form tours, there are five complete tours. However, there is a constraint that all demands are met -- therefore four extra idle shifts must be included to build a complete tour for the single unassigned shift. (A proposed remedy for this is suggested in Chapter VII.) Table 2 displays the results of the tour-building stage.

Ths computer execution time required to obtain a solution with each of the four decision rules, choose the best solution, and build tours from it was about one minute per problem. However, the time

Table 2. Shift-Allocation and Tour Building Results

Problem	Shift Alloc. Rule	Total Man-hrs. Allocated in Shift Sched. Phase	% Excess over Total Work Content	Total Man-hrs. Allocated after Tour Building	% Total Excess	% Tours w/consec. Days Off
1	A. Max $\sum_{i=K}^{K+7} (D_i - S_i)^2$	7456	16.48			
	B. Max $\sum_{i=1}^{168} (D_i - S_i)^2$	7480	16.86			
	C. Luce's	7496	17.11			
	D. Seq. Dem. Satis.	7435	16.15	9320	45.6	59.2
2	A	7464	16.61			
	B	7480	16.86			
	C	7496	17.11			
	D	7443	16.15	9320	45.6	65.2
3	A	7296	13.98			
	B	7312	14.23			
	C	7320	14.36			
	D	7248	13.23	8120	26.8	78.3
4	A	6576	2.73			
	B	6576	2.73			
	C	6648	3.86			
	D	6506	1.64	8800	37.5	55.9
5	A	8472	32.35			
	B	8720	36.23			
	C	8544	33.48			
	D	8372	30.79	11,920	86.2	72.1
6	A	7696	20.23			
	B	7688	20.11			
	C	7720	20.61			
	D	7627	19.15	10,440	63.1	73.9

Table 3. Excess Manpower Percentage Introduced in TB Stage of SS-TB Algorithm.

Problem	% Excess Introduced by TB Stage
1	39.5
2	39.5
3	13.6
4	34.9
5	55.4
6	43.9

Table 4. Tour Scheduling Algorithm Results

Problem	Total Man-hrs. Scheduled	% Excess over Work Content	% Tours with Consec. Days Off
1	8200	28.1	44.4
2	8200	28.1	38.0
3	8480	32.5	41.5
4	7800	21.9	54.4
5	9480	48.1	43.9
6	9400	46.9	49.8

required for scheduling shifts and building tours for the Sequential Demand Satisfaction Algorithm was only about eight seconds. It is the most efficient of the form SS-TB Algorithms both in terms of solution efficiency and computational efficiency.

The effectiveness of the heuristic that builds tours from shifts is shown by its performance in forming tours with consecutive days off. The percentage of tours of this type varies from approximately 56% to 78% according to the problem. (See Table 2 for exact results). Considering that only 28.6% of the possible tour types have consecutive days-off, the simple heuristic seems to be very useful.

The Tour Scheduling Algorithm results are shown in Table 4. It appears that this algorithm is rather inefficient not only in terms of excess manpower usage (the excess ranging from 21.9% to 48.1%), but also with respect to computer execution time. It requires about 3 minutes per problem on a Univac 1108. However, relaxing some of the constraints might result in substantial manpower savings, as discussed in the next chapter.

With respect to the criterion of the number of tours with consecutive days off, the Tour Scheduling Algorithm performs reasonably well. The percentage of tours of this type ranged from 38.0% to 54.4% (see Table 4) for the example problems. However, these results are rather poor compared with the tours created by the shift scheduling and tour building algorithms.

Evaluation of Measures

It is apparent from the sample of problems solved that there are inconsistencies in the values of the measures and the excess manpower required by the heuristics. For example, problem 6 has more steps, peaks and runs above and below the median than problem 5, as well as approximately the same number of mean crossings, and yet it resulted in only approximately .7 as much excess manpower usage as problem 5 did. Some of the variability measures in problems 1 and 2 are roughly equal to those in problem 5. Problems 1 and 2 are even more highly variable than problem 5 in terms of steps, peaks, valleys, and runs above and below the median, and yet problems 1 and 2 required only about .53 as much excess manpower as problem 5.

For the problems solved it appears that the number of steps, peaks, valleys, runs above and below the median, and number of mean crossings are not well correlated with excess manpower required in the heuristic solutions. However, VL and VL' seem to be somewhat more correlated with this required excess. Neither of these is a consistently good indicator, though, as evidenced by the results of problem 6. These two measures are relatively small in this problem, yet there is a 19% excess manpower usage incurred

Evaluations of Heuristics

From the limited computational study performed, neither the shift-scheduling and tour-building algorithm nor the tour scheduling algorithm emerged as clearly superior. The former is more efficient with respect to avoiding excess manpower usage through the shift-scheduling stage. It is also much more effective in producing tours with consecutive days off (see Table 5). Finally it requires much less computer execution time.

However, the tour-scheduling algorithm is generally more efficient in terms of manpower usage levels in the final analysis (see Table 6). From the test results it appears that the tour-scheduling algorithm provides a solution which is more efficient.

Table 5. Relative Efficiency of Algorithm Types in Producing Consecutive-Day-Off Tours. (More efficient = 100%)

		Algorithm Type	
		Shift-Sched. & Tour Building	Tour Sched.
Problem	1	100%	75%
	2	100%	58%
	3	100%	53%
	4	100%	99%
	5	100%	61%
	6	100%	67%

Table 6. Relative Cost of Algorithm Types (Lower Cost = 1.0)

		Shift-Sched. & Tour Building	Tour Sched.
Problem	1	1.14	1.0
	2	1.14	1.0
	3	1.0	1.04
	4	1.13	1.0
	5	1.26	1.0
	6	1.11	1.0

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

Since it is difficult to make meaningful statements about the usefulness of the various measures of demand variability in predicting excess manpower without optimal solutions to the problems, an alternative approach is suggested. By solving a large number of problems it would be possible to determine through multiple regression analysis the effect that each of the different types of variability has in causing excess manpower usage. For example the required excess (RE) is a function of several variables. Thus:

$$RE = f(M_1, M_2, \ldots, M_n, D)$$

Multiple regression analysis could be used to determine an approximate relationship between the RE and the measures presented (and others).

In fact, on reexamining the data, it appears that significant drops in the demand level, such as from 60 workers to 10 (see problem 5), resulted in work schedules with large excess requirements. The magnitude, frequency, and duration of these dropoffs seem to have an effect.

In comparing the SS-TB algorithm with the TS algorithm as presented here, it is apparent that the former has a much greater potential for reducing scheduled excess manpower to an acceptable percentage. This fact, coupled with its much greater efficiency in terms of computer execution time as well as its superior effectiveness in terms of producing a large number of consecutive-day-off tours, makes it a better choice for problem solving, _if_ the inefficiencies introduced in the tour-building stage can be eliminated.

There are, however, several modifications which might result in substantial improvements to either algorithm. For example, scheduling work over a greater cycle length (more than one week) might prove helpful not only in terms of obtaining less excess manpower, but also in obtaining greater employee satisfaction regarding tour types. With such a longer cycle length being scheduled, it would be possible to let employees have an occasional 3-day or 4-day weekend if desirable.

In addition, a provision for split-shifts and non-uniform tour-starting times would relax some of the constraints and should result in better solutions. A complete relaxation of these constraints would not be feasible, but tours with an interrupt interval or non-uniform start times lying in a reasonable range might be acceptable.

Both types of algorithms could possible profit from the use of back-tracking. At some point(s) in the scheduling process some of the earlier parts of the schedule could be reviewed in hopes of placing workers more efficiently, as mentioned in Chapter V.

A suggestion for improving the performance of the SS-TB algorithm is to round _down_ the number of tours called for at each time interval. For example, if there are 22 shifts scheduled to start at 8:00 a.m. throughout the week the tour-building algorithm presently rounds it up to 25 shifts and forms at least 5 tours. (The number of tours of type i required is

$$\max \left\{ \left[\frac{\sum_{i=1}^{24} \sum_{j=1}^{7} \text{Shift}_{ij}}{5} \right], \max_{j}(\text{Shift}_{ij}) \right\}$$

where: $[x]$ = smallest integer greater than x

Shift_{ij} = number of shifts of type i required on day j

However, it might be more efficient to choose 20 of these shifts, build 4 tours from them, and then create a residual demand from the 2 shifts that were not scheduled. If this process were carried out for each of the shift start times throughout the day (24, in this case), then a reiteration to schedule shifts and build tours from the residual demand might result in a more efficient labor utilization scheme.

There should be some qualification of results obtained from the SS-TB algorithm. Tours were built from only the shift allocation rule that required the smallest excess manpower. Building tours from the shift allocated by other rules might yield better final results.

Finally, it may prove worthwhile to consider a rather unconventional approach to the employees themselves. For example, the employees could be provided with a graph of the demand requirements for a unit of time, such as a week. Knowing his priority level, each employee would be free to request his own desired schedule, that is, he could input his choice of shift times for that week, with absolutely no restrictions imposed by the company. (Of course, there would be legal and union constraints.)

As long as the shift request curve is proportional within certain limits to the shift demand curve it should be possible to assign the employee precisely that group of shifts which he requests. It is the fact that what may be a desirable schedule for one employee may be very undesirable for another that permits such an approach to be acceptable. Also, the effect of very unusual shift choices by even a number of employees may be acceptable due to the smoothing effect caused by widely varying employee objectives. In general, for scheduling problems it is the assignment of late tours or shifts that is especially difficult. But this problem can be avoided to some extent by judicious assignment procedures earlier in the process.

It is very difficult to prescribe suitable _a priori_ guidelines to guarantee acceptable results using the approach of schedule generation by employees. It does seem possible, however, to advance some appealing heuristic rules. It is logical to maintain from some point fairly early in the schedule assignment process a manpower assignment level which is approximately proportional to, during every time interval, the manpower demand level.

Presumably, there will be a need to become increasingly more discriminating in granting requests as the assignment process continues. There are three reasons for this. 1) The total man-hours necessary dictated by the demand forecast process will of course serve as a lower bound for the number of man-hours assigned. Generally, it will be necessary to overshoot this mark, for it is the theoretical minimum, however, as more and more assignments are made it is desirable to more closely approximate the demand curve, because eventually it must be met as closely as possible. Hence, there is a need for more restrictive selections. 2) The employee requests will be reviewed in some priority order. For this reason it is reasonable to allow those with higher priorities somewhat greater freedom of choice. 3) If no control is maintained over the shift assignment process, it is possible that a large group of granted requests on particular days such as Saturdays, or particular start times, such as 8:00 a.m. would pose problems in creating reasonable schedules for many of the remaining employees.

One solution approach to the "ending" problem involves allowing some percentage of the employees complete freedom in determining their schedules, that is, perhaps the first 30% could have exactly the schedule they request. Of course, this number could be extended if the manpower assignment graph were a close enough proportional approximation to the manpower demand graph. Then, the remaining employees could be assigned schedules developed by more conventional means.

Of interest also is the strategy to be used by the employees in requesting schedules. It will probably prove beneficial to allow each employee to make a first- and second-choice schedule request. If the first must be rejected for some reason, then rather than be faced with the "leftover" shifts at the end of the assignment process, the employee would still have a chance of being assigned a schedule which, to him, has some element of desirability.

It will require experience based upon past "playings" of this assignment "game" as well as a knowledge of one's priority position and the manpower demand levels for the week to make an intelligent strategy. Consider for a moment the dilemma facing the employees. An employee having a reasonably high priority level may use his first choice to request a schedule which to him is highly desirable. However, there is a possibility that this request may be denied. Thus he must decide upon a suitable second choice. Does he use it to request a schedule which, though different from the first, is still relatively desirable to him, but which may include shifts which have proved popular with fellow employees or for which there has been little forecast demand? Or does he use something akin to a maxi-min strategy, in which he chooses a marginally acceptable schedule which includes shifts for which there is either high forecast demand or low employee demand, with the expectation that "surely" the second choice will be granted? Or does he choose a compromise approach that lies somewhere between these extreme cases? Much the same dilemma faces an employee with a low priority level, even for his first choice.

Future research could be conducted either in the area of good employee strategies or in developing suitable guidelines that would prevent the assignment process from going out of control.

In conclusion, there definitely exists a need in certain industries for the ability to schedule workers by methods other than Gantt-chart-type trial-and-error techniques and to assign them to shifts other than the conventional 8, 4, and 12:00 shifts. The research reported in this thesis attempts to meet this need in two ways: 1) by determining a means of predicting excess manpower usage by examining the usefulness of several measures of manpower requirements variability; 2) by developing two types of heuristic scheduling procedures, one for scheduling complete tours, and one for shift-scheduling and tour-building.

APPENDIX

DEMAND DATA OF SAMPLE PROBLEMS

Problem 1

Hour

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	5	10	32	73	93	92	63	69	86	87	88	56	53	55	57	68	53	31	16	8	5	3	3	3
2	5	11	31	68	88	88	59	63	76	70	72	53	49	50	52	59	52	32	16	9	5	4	3	3
3	4	11	27	62	83	82	54	61	72	74	69	52	52	49	56	63	52	32	16	9	5	3	4	3
4	5	11	28	68	80	79	56	61	71	74	71	55	53	54	57	68	49	30	17	9	6	4	3	4
5	6	12	30	64	84	83	57	66	83	85	76	57	54	50	49	51	42	30	18	12	8	6	5	4
6	4	7	15	32	42	43	41	38	36	35	35	36	42	41	43	43	37	31	16	10	7	5	5	3
7	4	6	12	18	26	29	34	33	31	28	31	31	38	42	46	56	52	34	15	7	5	3	3	3

Problem 2

Hour

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	5	10	32	73	93	92	63	69	86	87	88	56	53	55	57	68	53	31	16	8	5	4	3	2
2	5	11	31	68	88	89	59	63	76	70	72	53	49	50	52	59	52	32	16	9	5	4	3	2
3	4	11	27	62	83	82	54	61	72	74	69	52	53	48	56	63	52	32	16	9	5	3	4	3
4	5	11	28	68	80	79	56	61	71	74	71	55	53	54	57	68	49	30	17	9	6	4	3	4
5	6	12	30	64	84	83	57	66	83	85	76	57	54	50	49	51	42	30	18	12	8	6	5	4
6	4	7	15	32	42	43	41	38	36	35	36	35	42	41	44	43	37	31	16	10	7	5	4	3
7	4	6	12	18	26	29	34	33	31	28	31	32	38	42	46	56	52	34	15	7	5	3	2	3

(Continued)

Problem 3

Hour

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	16	16	16	85	85	85	67	67	87	87	87	56	56	56	56	56	56	33	7	7	7	7	7	7
2	16	16	16	83	83	83	60	60	72	72	72	52	52	52	52	52	52	31	7	7	7	7	7	7
3	14	14	14	76	76	76	57	57	71	71	71	54	54	54	54	54	54	32	7	7	7	7	7	7
4	15	15	15	73	73	73	60	60	72	72	72	56	56	56	56	56	56	35	7	7	7	7	7	7
5	16	16	16	77	77	77	62	62	82	82	82	51	51	51	51	51	51	29	8	8	8	8	8	8
6	9	9	9	38	38	38	38	38	38	38	38	38	38	38	38	38	38	38	10	8	8	8	8	8
7	8	8	8	29	29	29	29	29	29	29	29	29	42	42	42	54	54	24	24	4	4	4	4	4

Problem 4

Hour

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	40	40	44	44	47	47	47	50	50	50	50	50	50	50	50	50	50	50	50	40	40	40	40	40
2	38	38	38	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	40	40	40	40	40
3	35	35	35	35	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	40	40	40	40	40
4	36	36	37	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	44	40	40	40	40	40
5	38	38	44	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	44	40	40	40	40	40
6	10	11	11	35	35	35	35	35	35	35	35	35	35	35	35	35	35	35	15	15	15	15	15	15
7	9	10	10	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	13	13	13	13	13	13

Problem 5

Hour

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	10	10	60	60	45	90	90	45	70	70	45	45	45	45	70	70	45	45	50	50	13	13	13	10
2	10	10	60	60	40	40	85	85	40	40	65	65	40	40	60	60	40	40	49	49	10	10	10	10
3	10	10	55	55	40	40	85	85	40	40	65	65	40	40	60	60	40	40	45	45	10	10	10	5
4	10	10	60	60	40	40	85	85	40	40	65	65	40	40	60	60	40	40	47	47	10	10	10	9
5	10	10	60	60	40	40	85	85	40	40	65	65	40	40	65	65	40	40	51	51	10	10	10	10
6	10	10	30	30	25	25	35	35	25	25	40	40	26	26	40	40	25	25	40	40	25	10	10	10
7	10	10	25	25	20	25	35	35	20	20	40	40	22	22	50	50	22	22	25	25	25	8	8	8

Problem 6

Hour

Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	40	40	45	46	40	40	45	45	40	40	67	67	40	40	67	67	40	40	45	45	40	40	45	45
2	35	35	44	44	35	35	45	45	40	40	60	60	35	35	55	55	35	35	45	45	35	35	45	45
3	22	25	44	44	35	35	45	45	40	40	60	60	35	35	55	55	35	35	45	45	35	35	45	45
4	30	35	44	44	35	35	45	45	40	40	60	60	35	35	55	55	35	35	45	45	35	35	45	45
5	40	40	46	46	35	35	45	45	40	40	60	60	35	35	55	55	35	35	45	45	35	35	45	45
6	10	10	30	30	25	25	30	30	26	26	35	35	25	25	45	45	25	25	35	35	30	25	10	10
7	10	10	30	30	20	20	25	25	20	20	35	35	20	20	45	45	20	20	40	40	20	20	9	8

BIBLIOGRAPHY

1. Abernathy, W. J., N. Baloff, and J. C. Hershey, "The Nurse Staffing Problem: Issues and Prospects," Sloan Management Review, Vol. 13, No. 1, Fall, 1971.

2. Arabeyre, J. P., et al., "The Airline Crew Scheduling Problem: A Survey," Transportation Science, 3, 140-163, May 1969.

3. Baker, Kenneth R., "Scheduling a Full-time Workforce to Meet Cyclic Staffing Requirements," Management Science, Vol. 20, No. 12, August 1974.

4. Bennett, B. T., and R. B. Potts, "Rotating Roster for a Transit System," Transportation Science, Vol. 2, No. 1, Feb., 1968.

5. Dantzig, G., "A Comment on Edie's 'Traffic Delay at Toll Booths,'" Operations Research, Vol. 2, 1954.

6. Edie, L. C., "Traffic Delays at Toll Booths," Operations Research, Vol. 2, No. 2, 1954.

7. Healy, W. C., "Shift Scheduling Made Easy," Factory, Vol. 117, No. 10, October, 1959.

8. Henderson, W. B., and W. L. Berry, "Determining Optimal Shift Schedules for Telephone Traffic Exchange Operators," Paper No. 507, Krannert Graduate School of Industrial Administration, Purdue University, April, 1975.

9. Henderson, W. B., and W. L. Berry. "Heuristic Methods for Telephone Operator Shift Scheduling: An Experimental Analysis," Working Paper No. 20, Center for Business and Economic Research, The University of Tennessee.

10. Klasskin, Phillip M., "Operating to Schedule Operators." Telephony, July, 1973.

11. Luce, B. J., "A Shift Scheduling Algorithm," 44th National Meeting of ORSA, November, 1973.

12. Luce, J., "Employee Assignment System." Joint National Meeting of ORSA/TIMS, April 1974.

13. Luce, B. J., "Dynamic Employment Planning Model," 44th National Meeting of ORSA, November, 1973.

14. Maier-Rothe, C., H. C. Wolfe, "Cycle Scheduling and Allocation of Nursing Staff," Socio-Economic Planning Sciences, Vol. 7, No. 5, October, 1973.

15. Monroe, G., "Scheduling Manpower for Service Operations," Industrial Engineering, Vol. 2, No. 28, August, 1970.

16. Rothstein, M., "Scheduling Manpower by Mathematical Programming," Industrial Engineering, April, 1972.

17. Tibrewala, R., D. Phillippe, and J. Browne, "Optimal Scheduling of Two Consecutive Idle Periods," Management Science, Vol. 19, No. 1, September, 1972.