1993

A heuristic approach to the police staff scheduling problem

Charalambos Akis Marangos
Lehigh University

AUTHOR:

Marangos, Charalambos

TITLE:

A Heuristic Approach to the Police Staff Scheduling Problem

DATE: October 10, 1993

A HEURISTIC APPROACH TO THE POLICE STAFF SCHEDULING PROBLEM

by Charalambos Marangos

A Thesis

Presented to the Graduate School

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial Engineering

Lehigh University

1993

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Masters of Science in Industrial Engineering,

date: 8 / 4 / 93

Advisor in charge

IE Department Chairperson

ACKNOWLEDGEMENTS

I would like to thank all the people who assisted me in this study. That includes the professors of the Industrial Engineering Department at Lehigh University, my fellow graduate students, and the officers of the Bethlehem Police Department.

I would like to thank Dr. Laura I. Burke for bringing this problem to my attention and Dr. George R. Wilson for accepting to be my advisor. Dr. Wilson's assistance in formulating the problem as an ILP was invaluable. His steadfast support and direction during the time of the author's study is much appreciated. Thanks go to Dr. John W. Adams and Dr. Robert H. Storer for their insights to the mathematical structure of the problem. I would like to thank Officer Robert Haffner for assisting me in the constraint definition phase of the problem, and for constantly keeping me on my toes with my work by inquiring about my progress during any time around the clock. Many thanks to my colleagues Laura Louise Lansing and Yesim Erke for their insights into the problem and their constant moral support. I thank Margaret P. Church for sharing her knowledge with me about the psychological effects of scheduling. Last, but not least, I thank my parents for their continuous moral support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis presents a heuristic approach to the scheduling of police officers. Schedules produced are cyclic and rotatable, and officers are assigned to work eight-hour shifts, with work stretches greater than two days and not exceeding six days. The days off are in two-day stretches. A manpower allocation problem is solved first, and then a heuristic algorithm is applied to the results in order to produce feasible schedules. The algorithm has three steps. The first step searches for schedules that meet the primary condition of having, on average, two days off in a seven day schedule. In the second step the schedules that satisfy the daily requirements are identified. The third step is an enumeration of the feasible schedules and preferences are assigned to those schedules. The problem has been successfully solved on a personal computer as well as on a workstation. Implementation of the feasible schedules produced is currently being considered by the Bethlehem Police Department.

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

This thesis presents a heuristic approach to the scheduling of police officers. The current schedule is causing a lot of aggravation to the police officers. The schedule is described as 7-2-7-2-6-4. This is read as: seven two seven two six four. The numbers are alternating work days and days off. This means that the officers are working seven days straight, taking two days off, working seven days, taking two days off, working six days and taking four days off. Then, the pattern is repeated.

Two consecutive days off are identified as a pair of off days, while consecutive work days are identified as a work stretch. In addition to the long work stretches of seven days in a row, the second day off pair is split so that one day off is received sometime during the second seven-day stretch. When a pair of days off is split, the off days are called split days off. The split days off are highly undesirable. A work stretch of seven days is draining the officers physically and emotionally. For a nomenclature review see appendix A.

The officers rotate through shifts in the following order: from night to evening to day shifts (N-E-D) in a certain number of days called a cycle. Complaints by the officers include lack of quality time spent with their families, lack of weekends off, decreased attention span when they rotate shifts, and a low esprit de corps. The dissatisfaction of the officers is evident by such indicators as an increased number of sick leaves.

Research about the physical and psychological side-effect of rotating 'backwards' (D to N to E to D ...) through shifts as opposed to rotating forwards (D to E to N to D ...), as well as about long work stretches and single days off give credence to the officers' complaints [Sullivan]. For example, quality sleep is reduced resulting in irritability, poor judgment and sometimes even to clinical depression. The physical effects of bad schedules include stomach and intestinal disorders as well as reduced reaction time, which could eventually be fatal.

Dr. Czeisler, an Associate Professor of Medicine at Harvard Medical School and Director of the non-profit Center for Design of Industrial Schedules, includes among the side effects of bad schedules alcohol and drug abuse, constant sleep deprivation, which causes memory impairment, and performance deficits [FOCUS].

In addition, the federal government has passed legislation requiring that certain guidelines be followed to ensure safety at the workplace [U.S. Dept. of Labor]. It is considered unsafe to have tired officers patrolling the streets; hence, a better schedule is needed.

The general problem of scheduling people for 24-hour operations exists in industry and the service sector. The problem has been researched in the academic community, and solutions have been proposed which use either a heuristic or an exact mathematical approach. Examples of scheduling problems deal with the scheduling of airport controllers [Nanda and Browne], nurses at hospitals [Rosenbloom and Goertzen], telephone operators [Segal], mail and parcel handling companies [Nanda and Browne], police and fire departments [Nanda and Browne], and casino security operators [Panton].

Several interviews with police officers and their union leaders have been conducted. During a period of three years, the problem has been defined in multiple ways. During this time period, several approaches have been addressed independently at Lehigh University by undergraduates and professors alike. Their approaches produced results that solved a different formulation of the problem under consideration. Specifically, the problem was attacked as a four platoon scheduling task.

Three platoons would man the three shifts every day, one platoon per shift, while the fourth platoon would have the day off. A platoon is a group of officers that reports to a specific supervisor. Because of the constraints posed by the officers such as two days off in a row and no more than six day work stretches, only a few alternative schedules were computed and they were rejected by the Police Union and Management alike. The four platoon scheduling problem was abandoned and the scheduling of three platoons will be addressed here.

1.2 ISSUES

The toughest part of solving any problem is first defining it. Police officers were dissatisfied with their schedules, but they did not know how to define the major constraints of the problem. Instead, the author was flooded with various requests and demands about what a schedule should look like and what it should be able to do. Many combinations of these requests made the problem solution infeasible (See the chapter Approach Taken and Computational Investigations). A sorting out stage was required in order to identify the major constraints and to relax others.

An attempt was made to assign weights or factors to the schedules in the optimization function so as to indicate preferences of one feasible schedule over another. This task proved to be a very tedious undertaking. There are a great many opinions as to what makes a good schedule. For example, a schedule could be acceptable if it applies to a Day shift but unacceptable if it applies to a Night shift. Instead of combining all the preferences into one single number, schedules produced are sorted in various ways based on one preference at a time.

The hourly constraints, shown in Table 1, were used to identify the daily staffing requirements, that is, how many people to assign to the various shifts. Calculating these requirements is similar to an allocation problem as is the problem of assigning schedules, even though the former is a much simpler problem to solve than the latter.

It is a general practice to solve difficult problems by breaking them down into smaller manageable pieces. [Panton] for example, constructed several sequences of days on and days off, and then proceeded to combine these sequences/modules into a feasible master schedule which is a schedule for all the employees. The problem at hand was split into two phases.

The first phase allocates daily minimum staffing requirements, and the second phase builds schedules for the officers. The second phase was also split into smaller modules, each one computing a certain part of the problem.

time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
07:00	13	13	13	13	13	13	13
08:00	13	13	13	13	13	13	13
09:00	13	13	13	13	13	13	13
10:00	13	13	13	13	13	13	13
11:00	17	17	17	17	17	13	13
12:00	17	17	17	17	17	13	13
13:00	17	17	17	17	17	13	13
14:00	17	17	17	17	17	13	13
15:00	19	19	19	19	21	17	17
16:00	19	19	19	19	21	17	17
17:00	19	19	19	19	21	17	17
18:00	19	19	19	19	21	17	17
19:00	19	19	19	19	21	21	21
20:00	19	19	19	19	21	21	21
21:00	19	19	19	19	21	21	21
22:00	19	19	19	19	21	21	21
23:00	19	19	19	19	21	21	21
24:00	19	19	19	19	21	21	21
01:00	19	19	19	19	21	21	21
02:00	19	19	19	19	21	21	21
03:00	13	13	13	13	13	13	13
04:00	13	13	13	13	13	13	13
05:00	13	13	13	13	13	13	13
06:00	13	13	13	13	13	13	13

Table 1

Hourly Requirements by day of the week and hour of the day

A scheme for cyclic and rotating schedules had to be devised, and the shift length had to stay at eight hours per day. A cyclic schedule is a sequence of days on and days off that repeats itself every certain number of days. A rotating schedule is a schedule that allows the officers to work on different shifts. Schedules can be separated into two sets: those that include all the cyclic schedules, and those that include only the noncyclic schedules. Because noncyclic schedules provide more flexibility in designing feasible schedules (there is an inherent limitation on the number of schedules that are cyclic *and* satisfy all the constraints) some researchers (e.g., [Burns and Carter]) as well as some police departments (e.g., in New York City) have constructed noncyclic, non rotatable feasible schedules. [Nanda and Browne] have constructed schedules that are fixed, with the shift length increased from eight to ten hours . This approach was discussed but will not be further explored because of union and management objections.

The schedules produced should use the available number of officers. The hiring of temporary staff (temporary law enforcement agents) was out of the question for obvious reasons. In addition, five days, on average, should be worked in a seven day schedule. Any days off should be in continuous two-day blocks and a schedule that would include frequent weekends off was also one of the goals.

A computer program was required by the police scheduling department so that various schedules and various alternatives could be tried out before a particular schedule was implemented. Thus, the speed of the computations in a PC environment was a major factor in searching for a solution algorithm.

Solving the second phase of the problem (the second phase will be illustrated later in this chapter), would yield an ILP with more than 300,000 variables. This size is well beyond the capacity of any software package available today. Constraint collapsing and aggregation routines [Kendall and Zionts] could potentially reduce the problem to a single constraint resource allocation problem. This study, instead, presents a heuristic approach to the scheduling of police officers.

1.3 PROBLEM DESCRIPTION

There are 84 officers to be scheduled. The officers are split into three platoons, each platoon having 28 officers. The three platoons are to be scheduled so that officers are available seven days per week, 24 hours per day, three shifts per day, according to specific hourly requirements. Each shift length should be eight hours. The schedules should be cyclic which means that a certain sequence of working and non-working days repeats itself every certain number of days. The officers should rotate through shifts, for example, if officer A is working Day shift this week, he/she should work on the Evening shift next week. The maximum work stretch should not exceed six days, and it should be greater than or equal to two days.

A mathematical formulation of the problem and its constraints as an integer linear program (ILP) follows.

1.4 MATHEMATICAL FORMULATION

The problem is split into two phases. Phase one computes the lower bound on the number of people required for the three main shifts and any optional swing shifts, as well as the start times for the swing shifts. Swing shifts are shifts that overlap the three main shifts (the Day, Evening, and Night shifts) and are eight hours long (see Figure 1). They are used to compensate for peak demand periods. The name of a swing shift, for example Day Swing, is assigned because it overlaps most of the Day shift. Ties are broken arbitrarily. Officers that man a given swing shift are evenly distributed/assigned from the main shifts that the swing shift overlaps.

The main shifts have fixed starting and ending times. The Day shift begins at 7:00 AM and ends at 3:00 PM. The Evening shift starts at 3:00 PM and terminates at 11:00 PM, while the Night shift commences at 11:00 PM and finishes at 7:00 AM. Phase two builds feasible schedules based on the results from phase one.

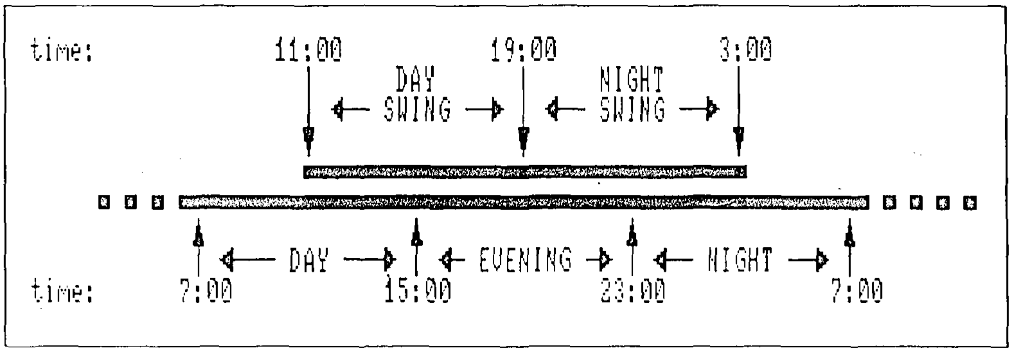

Figure 1

Illustration of the main shifts and the swing shifts

for a 24-hour operation

The staffing requirement constraints were eventually relaxed. This occurred after realizing that the start times of the swing shifts and the number of officers required to work during those swing shift were easily identified by hand, and the Police Department decided that the available officers would be scheduled internally. The computation of the swing shifts will be described in Chapter Three because of the relevance to the scheduling problem.

The results obtained from solving the first phase ILP, are fed into the second phase ILP. The second phase ILP produces schedules for individual officers, at least in theory. The resulting ILP for the second phase is too large for any of the commercial packages available (LINDO, SAS) to solve (see the chapter on Literature Review for possible techniques to solve this ILP). The ILP formulation is included to show a mathematical formulation of the problem.

<u>Phase 1</u>

Assumptions:

- The police force to be scheduled consists of 84 patrol people.
- The force is divided into three platoons of equal size (28 people per platoon) for management purposes.
- Each platoon will be assigned to work during one of the three main shifts and, if needed, for any corresponding

swing shifts. There will be a maximum of three swing shifts, one per main shift.

- The swing shifts may start at any time that allows at least one hour overlap with the corresponding main shift.

Formulation:

let

x_m = the number of officers assigned to main shift m; m = D, E, N (**D**ay, **E**vening, **N**ight)

x_{ij} = the number of officers assigned to swing shift i which begins at hour j; i = D′, E′, N′ ; j = 1, 2, ..., 24

if i=D′ then $24 \le j \le 14$ which define set T_i for i=D′

if i=E′ then $8 \le j \le 22$ which define set T_i for i=E′

if i=N′ then $16 \le j \le 24$, $1 \le j \le 6$ which define set T_i for i=N′

a_{ijt} = 1 if hour t is part of the work period defined by swing shift i that begins at time j

= 0 otherwise

b_{mt} = 1 if hour t is part of the work period defined by main shift m

= 0 otherwise

r_t = the number of officers required during hour t; (Since the demand varies between days (see Table 1), not only within days, a separate problem must be solved for the days where the demand is different).

P = the number of people per platoon (28)

y_{ij} = 1, if swing shift i which starts at time j is assigned to any police officers

= 0, otherwise

· objective function:

$$\min \sum_{m} x_m + \sum_{i} \sum_{j \in T_i} x_{ij} \qquad (1)$$

subject to:

· meet hourly demand

$$\sum_{m} b_{mt} x_m + \sum_{i} \sum_{j \in T_i} a_{ijt} x_{ij} \geq r_t \qquad t=1,2,\ldots,24 \qquad (2)$$

· do not exceed the platoon size

$$x_m + \sum_{j \in T_i} x_{ij} \leq P \qquad (m,i)=(D,D'),(E,E'),(N,N') \qquad (3)$$

· the swing shift should not exceed the platoon size

$$x_{ij} \leq P \cdot y_{ij} \quad y_{ij}=0,1 \quad \forall i, \forall j \in T_i \qquad (4)$$

· assign at most one swing shift per main shift

$$\sum_{j \in T_i} Y_{ij} \leq 1 \quad i=D',E',N' \qquad (5)$$

Phase 2:

let

x_m^d = the minimum number of officers required in main shift m for day d of the week, where $m = D, E, N$ and $d = 1, 2, \ldots, 7$; (1=Monday, 2=Tuesday, etc.; the values are from phase 1)

x_k^d = the minimum number of officers required in swing shift k for day d of the week, where $k = D', E', N'$ and $d = 1, 2, \ldots, 7$; The values are from phase 1.

z_{itj} = 1, if officer i is working day t of tour j
= 0, if officer i is off on day t of tour j

y_{itj} = 1, if day t is the <u>first</u> day of an off-period for officer i on tour j
= 0, otherwise

L = the number of days between shift rotations

T = the length of the scheduling cycle ($T=3nL$ where a multiplier of three is used so that each person rotates through all three main shifts; n is any integer so that T is a multiple of seven, i.e. $modulo(T,7)=0$)

$S(j,t)$ = assigned shift on day t of tour j

w = the set of subscripts representing weekend days during the scheduling cycle

· the objective function seeks to minimize the number of officers on duty and number of days off

$$\min \sum_{i} \sum_{j} \left[\sum_{t \in w} z_{itj} + \sum_{t} y_{ijt} \right] \tag{6}$$

subject to:

· work at most six out of the last seven days (note: T+1, T+2, ..., T+6 are made equal to 1,2,...,6 forming a "wrap-around" schedule)

$$\sum_{\tau=t-6}^{t} z_{i\tau j} \leq 6 \qquad \tau=7,8,\ldots,T+6 \quad \forall i, \forall j \tag{7}$$

· work an average work week of 5 days

$$\sum_{t=1}^{T} Z_{itj} = \frac{5}{7} T \qquad \forall i, \forall j \tag{8}$$

· determine the beginning of an off period for each officer (note: let $T+1$ be changed to 1)

$$z_{i,t-1,j} - z_{itj} \leq y_{itj} \qquad \forall i, \forall j, t=2,,3,\ldots,T+1 \tag{9}$$

· if an officer is working on the last day of a night shift rotation, the next day is the first day of an off period (note: j and t such that $S(j,t)=N$ *or* N' and $t=nL$ where $n=1,2,\ldots,T/L$)

$$y_{itj} - y_{i,t+1,j} \leq 0 \qquad \forall i \tag{10}$$

- each officer on duty will work one shift per day

$$\sum_{j} z_{iij} \le 1 \qquad \forall i, \forall j \tag{11}$$

- meet the minimum number of officers for the main shifts

$$\sum_{i=1}^{3P} z_{iij} \ge x_m^d \qquad d=1,2,\ldots,7 \quad m=D,E,N \tag{12}$$

- meet the minimum number of officers for the swing shifts

$$\sum_{i=1}^{3P} z_{iij} \ge x_k^d \qquad k=D',E',N' \quad d=1,2,\ldots,7 \tag{13}$$

- days off should be in two-day stretches

$$\sum_{l=\tau}^{\tau+2} y_{iij} \le 1 \qquad \tau=1,2,\ldots,T \tag{14}$$

and $\tau = T+1$ *and* $T+2$ *become* $\tau = 1$ *and* 2 *,respectively.*

CHAPTER 2

LITERATURE REVIEW

The book by [Nanda and Browne] has an extensive annotated bibliography on the subject of employee scheduling. Comprehensive overviews of the available solution methodologies have been written by [Bechtold, Brusco, and Showalter], [Burns], and [Burns and Koop].

[Bechtold, Brusco, and Showalter] identify the solution methodologies as LP based or construction. In addition, they classify the labor scheduling research into three categories: <u>days off</u>, where work and non-work days are computed based on a tour that is less than a week long; <u>shift</u>, where the start and end times of shifts and of meal/rest breaks are computed; and <u>tour</u>, which is a combination of the previous two categories. The conclusion was that most of the small problems are solved using optimization techniques, while large problems are most likely to be solved using a heuristic, a practice followed in this thesis.

A formulation of the scheduling problem as an ILP appears in [Bechtold, Brusco, and Showalter] which is a typical formulation of the allocation problem. Bechtold's et al. formulation follows:

$$\text{minimize } Z = \sum_{j=1}^{n} x_j$$

subject to

$$\sum_{j=1}^{n} a_{tj}\, x_j \geq r_t \qquad t=1,2,\ldots,m$$

$$x_j \geq 0 \text{ and integer,} \qquad j=1,2,\ldots,n$$

where

x_j = the number of employees assigned to tour schedule j

r_t = the number of employees required to work in time period t

n = the number of tour schedules to be considered

m = the number of time periods scheduled over the planning horizon

a_{tj} = 1, if the time period t is a work period in tour schedule j

= 0, otherwise

Several heuristics exist in the literature for ways to solve ILP problems related to scheduling. [Morris and Showalter] propose a heuristic to solve such an ILP formulation. The integer constraint is relaxed; the relaxed LP is solved; and, a heuristic is used on the answer. This method would potentially identify "near optimal solutions".

[Tien and Kamiyama] suggest that the ILP used for identifying the manpower requirements could be decomposed into smaller ILP's or transformed into a network flow problem. Transforming the ILP into a pure network flow problem cannot be done for the problem at hand because of the constraint on the work stretch (a work stretch should be no more than six-days long). They decompose the problem into five subproblems, which are used to:

1. determine the temporal staffing requirements
2. determine the total manpower requirements
3. determine any recreation blocks
4. determine a recreation and work schedule
5. determine a shift schedule

The authors concluded that the manpower allocation (subproblem 1) should "be considered separately from the manpower scheduling problem" (subproblems 2-5) [Tien et al., p.280, par. 3]. This has been independently identified by this author.

In this thesis, the second subproblem presented above is the first phase of the problem at hand, while the fourth and the fifth subproblems are integrated into the second phase. The first and the third subproblems are not applicable. Integrating the subproblems is a common practice, as noted in [Tien et al.].

Most of the literature suggests that the scheduling problem be split into modules. [Tien et al.] suggested that the problem be split into two separate problems. [Panton] and [Burns and Koop] suggest that the master schedule be constructed from various modules (sequences of days on and days off).

[Burns and Koop] use sets of modules, called "mini-schedules", which are combined to form a master schedule. The methodology attacks both the shift-changing and the manpower allocation problems. The authors recognized that "in almost all scheduling situations, an even distribution [of weekend days off] is preferred to a skewed one." Their method is to calculate a maximum number of weekends off and then to assign the weekends off evenly throughout the scheduling period. They then proceed to allocate the remaining days off so that the constraints of the problem are satisfied. The even distribution of weekends off was one of the factors used to sort the schedules produced by the algorithm to be presented.

[Panton] suggests that the master cyclic schedule should be constructed from a library of modules. These modules are built using either an Integer Program or a network flow solution. The number of weeks in the master schedule is equal to the number of employees. An assumption is that the daily number of people required is given. Single days off are allowed, something that this author will try to avoid. The modules can be used independently or put together based on shift-changing constraints; i.e., at least one day off between shift changes should be allocated. The feasibility of a zero-one solution to the ILP is not guaranteed.

[Lowerre] and [Brownell and Lowerre] identify different scheduling-problem formulations based on various constraints. The various combinations of constraints are identified as "policies". Different formulas are used to calculate the lower bounds on the work force for an assortment of policies. The bounds used are quite loose. A typical bound is 2n, where n is the maximum daily requirement. In both papers, work stretches in excess of six days are allowed.

Other authors have also calculated formulas for lower bounds on the work force size. [Burns] and [Burns and Carter] are of the opinion that three lower bounds are sufficient to identify the minimum size of the work force. The formulas are replicated here. A modified version of the first formula so that more weekends off can be allocate, and the original second and third formulas were used to determine the feasibility of the problem at hand (see the chapter on Computational Investigations).

Let:

A = the number of weeks in B weeks that a weekend off is desired

n = the maximum demand on any weekend day: $n = \max(n_{sat}, n_{sun})$

W = size of work force

1. Weekend constraint: The average number of employees available each weekend must be sufficient to meet the maximum demand on any day.

$$W \geq \left\lceil \frac{B \cdot n}{B - A} \right\rceil$$

where $\lceil x \rceil$ *is the smallest integer* $\geq x$.

The idea behind the formulation of the first constraint is that when n people have to work during a weekend, W-n can have the weekend off and this is averaged out for A weekends off in B weeks.

2. Total demand constraint: The total number of employees per week must be sufficient to meet the total weekly demand.

$$W \geq \left\lceil \frac{1}{5} \sum_{i=1}^{7} n_i \right\rceil$$

3. Maximum daily demand constraint: The number of employees must be sufficient to meet the maximum demand on any day.

$$W \geq \max_i \{n_i\},\ i=1,2,\ldots,7$$

where i=1 stands for Monday, 2 for Tuesday, etc.

[Brusco and Jacobs] use a simulated annealing algorithm to solve the cyclic scheduling problem. The heuristic approach is justified because the cyclic staff-scheduling problem presented, which included break periods, had been proven to be NP complete. They formulated the problem to include a cost factor associated with scheduling an employee for a certain schedule. A similar formulation was attempted, but it could not be used, because the structure of the problem at hand could not incorporate a cost factor. A cost could be used if, for example, part time employees could be hired (they cannot), or if the night shift was more expensive to staff than the day shift (they have the same cost).

[Rosenbloom and Goertzen] present an algorithm for scheduling nurses at a hospital. The algorithm consists of three stages.

In stage one, all the possible schedules are generated. The number of possible schedules is rather small because only those schedules that can be part of a cyclic schedule are considered, and their calculation is only carried out once. The calculation of the possible schedules will be lengthy (because it is a combinatorial problem), but the idea of needing to carry out any lengthy computations only once seems quite attractive.

In stage two, an integer program is formulated. The re-usability of the ILP's solution was noted as a positive aspect of the algorithm. "This means that as long as the labor constraints and the minimum daily coverage remain fixed, the entire scheduling problem needs to be solved only once". This author wanted to avoid the use of an ILP because of the inherent problems of ILP's, such as no guarantee for feasibility, lengthy computation times, and an explosion of the size of the ILP once more constraints are added or the scheduling period is increased to three or four weeks.

Finally, in the third stage, the results of the ILP are transformed into work patterns for each of the nurses. The methodology behind [Rosenbloom and Goertzen]'s algorithm was the starting point for the algorithm to be presented in the next chapter.

CHAPTER 3

APPROACH TAKEN

3.1 ISSUES ADDRESSED

The Bethlehem Police scheduling problem was split into two phases. The first phase is the manpower allocation phase where the daily requirement of shifts and people are computed. This requires calculating how many swing shifts will be needed and their starting times, as well as how many people will be assigned to the main and swing shifts. In the second phase the results of the first phase are used to compute rotating cyclic schedules.

Initially the problem was attacked as an Integer Linear Program. This method proved to be acceptable for the first phase, while for the second phase it was not. The first phase was a relatively small ILP with 93 variables. The second phase ILP had a number of variables equal to 84,672 with a scheduling cycle of, say, 28 days and a number of possible tours equal to 18. The size of an ASCII file containing all the constraints was about 32 million bytes long (32 Meg). It was quite obvious that a heuristic was needed. Several approaches were investigated and will be described later in this chapter.

While trying to identify a solution to the problem, the existing number of people had to be utilized. It will be shown that because of the combination of constraints, this requirement could not be satisfied (see the chapter on Computational Investigations).

Split days off had to be avoided. Days off in two-day stretches were used. Again, because of the constraints of the problem (maximum six-day work stretches and days off in two-day blocks), a solution that satisfied all the constraints was hard, if not impossible, to find. Certain assumptions needed to be made, such as the possibilities of hiring more people or of revising the demand during weekend-days.

A cost based optimization function was nearly impossible to assign, and its effectiveness was challenged. It is believed that because of the variation in preferences among employees regarding what constitutes a good schedule, to search for a cost function would be futile. Due to this, a transportation problem formulation was also scrapped from further research since that would require a cost based function.

The algorithm to be presented has its own limitations with respect to the types of problems it can handle, as will be shown later in this chapter.

3.2 ALGORITHMIC APPROACH

The first phase of the problem was discussed above, as well as in the introduction. The results of the first phase ILP, the minimum number of people required for the various shifts, as well as the starting times for the swing shifts are shown in Table 2. Three different ILP's were solved for the periods Monday-Thursday, Friday, and Saturday-Sunday, because the daily requirements varied over these sets of days. The main shifts are indicated with the first letter of the shift, and the swing shifts with the letter primed. The heuristic method for the second phase follows.

The results of the ILP indicate a maximum daily requirement of officers. The requirement for swing shifts is now relaxed, and instead there are only three main shifts to be concerned with. It is clear from the results of the first phase that the swing shifts could be manned evenly from the other two shifts; thus, the swing shifts were not used for the second phase. Instead, they could easily be computed after the schedules are created.

The minimum daily platoon size required per day is the maximum platoon size required on any given day. It will be shown in the next chapter that overstaffing (assigning more shifts than the minimum shift-hours required, shown in table 2) is not as bad as it seems, and it could be beneficial to Police Scheduling.

		Mon	Tue	Wed	Thu	Fri	Sat	Sun
07:00-15:00	D	13	13	13	13	13	13	13
N/A	D′	0	0	0	0	0	0	0
15:00-23:00	E	15	15	15	15	17	17	17
11:00-19:00	E′	4	4	4	4	0	0	0
23:00-07:00	N	15	15	15	15	17	17	17
19:00-03:00	N′	4	4	4	4	4	4	4

TABLE 2

Results of the first phase ILP

The algorithm in phase two has three steps. The first step searches for any schedules that meet the major constraint of having two days off in a seven day schedule, on average. The second step identifies which of these schedules satisfy the daily requirements, and the third step enumerates the feasible schedules in various ways based on preferences. Schedules produced are cyclic and rotatable, and officers are assigned to work eight-hour shifts. Work stretches are greater than two days and do not exceed six days. The days off are in two-day stretches.

[Burns and Carter] have three constraints on the minimum daily manpower required that have to be satisfied in order to find a feasible solution to the scheduling problem. Their constraints indicate the maximum work force required for a given number of weekends off. The modified version of the first constraint and its computation, as well as the calculations for the other two constraints, can be found in the next chapter. The computation of the constraints indicate that a schedule could be found and the algorithm can proceed.

Let M be the maximum demand over any given day and W the size of the available work force. W-M people could then take off the day of the maximum requirement, as it was shown also in [Burns and Carter]. Expanding on this, the ratio M/W is computed, i.e. the ratio of required vs available staff (the

RA ratio). A fraction that is a common multiple of the RA ratio is computed. This second ratio (the ra ratio) should be as close to the RA ratio as possible, and it will be illustrated below.

To compute the ra ratio, the following assumptions are made:

1. The numerator (N) of this fraction should be greater than zero.

$$N \geq 0 \ ; \ N = integer$$

2. The denominator (D) should be greater than one and an exact divisor of W, but three times D should be less than or equal to W, i.e.

$$D > 1 \ ; \ 3 \cdot D \leq W \ ; \ n \cdot D = W \ ; \ n = integer$$

3. The denominator should be a small number, up to and including seven.

$$D \leq 7$$

4. The ra fraction becomes:

$$ra = \frac{N}{D}$$

where $N \leq M \ ; \ D \leq W \ ; \ x < y$

D lines are drawn, each line corresponding to the schedule of one person (see Figure 2). This set of lines will be called a mini-schedule as per [Burns and Koop]. Because D is required to be an exact divisor of the platoon size, multiples of this schedule will be used to account for all the people in the platoon.

The first line, officer 1, will be the schedule for officer 1, the second for officer 2, and so on. At the end of the line the officers rotate cyclically and they start working on their new line/schedule. Cyclically means that officer 1, after the last day of his/her line, becomes officer 2, officer 2 becomes officer 3 and officer 3 becomes officer 1 (see Figure 2).

Each line of the mini-schedule will consist of T columns. Each column is a day of the week, and the first column is a Monday. T has to be a multiple of seven and as noted by various authors ([Panton], [Lowerre], [Brownell and Lowerre], [FOCUS] and [Burns and Koop]) longer than or equal to two weeks.

The requirement that the ra ratio N/D be close to the RA ratio M/W is such that while N people will be required to work, D-N will take the day off, and this will minimize overstaffing.

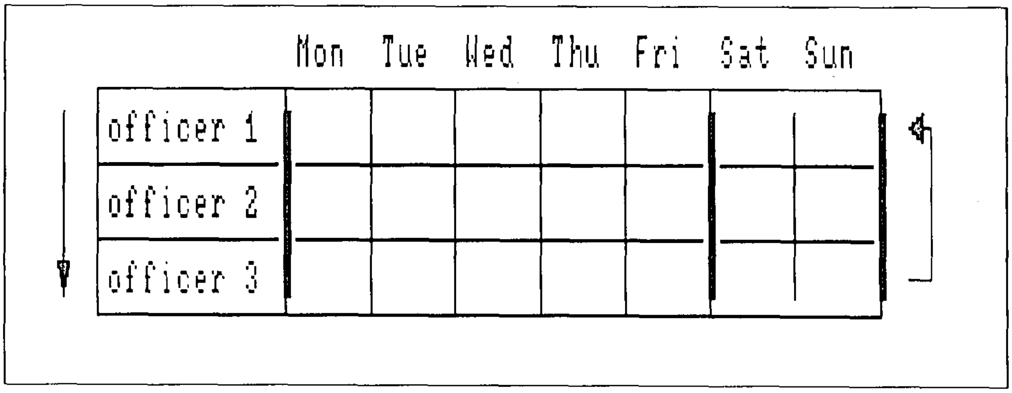

FIGURE 2

Empty mini-schedule for one week per line

and a three week rotation

The requirement that D be less than or equal to seven is because for larger multiples the execution time of the algorithm (several days) was unacceptable. The Police Department required execution times of a day or less.

Each mini-schedule will be used for one shift. There will be no shift change within a mini-schedule. There will potentially be many mini-schedules and thus a number of schedules to choose from. A schedule with an off period at the end of the last line should be selected. That way, after the last line is completed, that officer will move to a different shift, while remaining in the same mini-schedule or moving to a different one.

Next, all possible ways of assigning days off in T days is computed. Only a few of these combinations are candidates for feasible schedules. This is effectively the same as finding all the possible combinations of:

$$\left(\begin{array}{c} T \\ 2 \cdot modulo(T,7) \end{array} \right)$$

Modulo(T,7) is the number of weeks in T and it is multiplied by 2 to identify the days off. On average, two days out of seven are days off.

Out of all the possible combinations only a few exist which satisfy the following constraints discussed earlier:

1. No person is to work more than 6 days straight (maximum work stretch equal to 6).
2. The minimum work stretch is two days.
3. Split days off are not allowed.

In order to compute all the combinations, a lexicographic algorithm was used [Nijenhuis and Wilf]. The algorithm is described next for n items (a_i, i=1,...,k) taken k at a time. For example, if there are three numbers (1,4,5; n=3) to choose two numbers at a time (k=2), then the chosen numbers are stored in *a1* and *a2* as follows (the values of *ai* are changed upon each iteration of the algorithm):

	a1	a2
iteration 1:	1	4
iteration 2:	1	5
iteration 3:	4	5

1. {first time through} $m \leftarrow 0$, $h \leftarrow k$; goto (4);
2. {later entries} If $m \geq n-h$, goto (3); $h \leftarrow 0$;
3. $h \leftarrow h+1$; $m \leftarrow a_{k+1-h}$;
4. For $j = 1,h$; { $a_{k+j-h} \leftarrow m+j$ }; If $a_1 = n-k+1$, final exit;

   EXIT

Initially, the combinations of all the off-days available were computed. The number of iterations for this computation was cut drastically once it was recognized that only modulo(T,7) blocks needed to be arranged, which is the number of weeks per line, since the days off come in two-day blocks (constraint number 3 above). These combinations are saved in a file (comb.out) for further consideration. The algorithm is as follows (comments are enclosed in "/#" and "#/"):

1. Open a file called comb.sch so as to save the schedules produced;

1a. The first combination produced is chosen and it is stored in the first line (Figure 3);

1b. Instead of trying to identify feasible schedules to place in the lines after the first line, the sequence of zeros and ones in the first line is replicated into all

    the lines of the mini-schedule (line one will be denoted by mini-schedule(1), line two by mini-schedule(2), etc.; see Figure 4). This will eliminate the need for having as many files as the number of lines open at any time.

2. CALL SUMIT(mini-schedule);

3. DO $index_1$ = 1,T-1;

    DO $index_2$ = 1,T-1;

    ...

    DO $index_{T-3}$ = 1,T-1;

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
mini-schedule(1) ⇨	1	1	1	0	0	1	1
mini-schedule(2) ⇨							
mini-schedule(3) ⇨							

FIGURE 3

Step 1a of the algorithm

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
mini-schedule(1) ⇨	1	1	1	0	0	1	1
mini-schedule(2) ⇨	1	1	1	0	0	1	1
mini-schedule(3) ⇨	1	1	1	0	0	1	1

FIGURE 4

Step 1b of the algorithm

```
        DO index_{T-2} = 1,T-1;
          DO index_{T-1} = 1,T-1;
            CALL ROTATE(mini-schedule(T));
            CALL SUMIT(mini-schedule);
          END DO
          CALL ROTATE(mini-schedule(T));
          CALL ROTATE(mini-schedule(T-1));
          CALL SUMIT(mini-schedule);
        END DO;
        CALL ROTATE(mini-schedule(T-1));
        CALL ROTATE(mini-schedule(T-2));
        CALL SUMIT(mini-schedule);
      END DO;
              . . .
    CALL ROTATE(mini-schedule(3));
    CALL ROTATE(mini-schedule(2));
    CALL SUMIT(mini-schedule);
  END DO;
  If the combination read in step (1a) is the last one:
    STOP; ELSE, read the next combination and goto
    (1b);
```

The following are the subroutines used above:

```
SUBROUTINE ROTATE(mini-schedule(i))
```

```
/# this subroutine shifts a line of the mini-schedule one
digit to the left (see Figure 5)#/
temp ← mini-schedule(i,1); /#(the second index indicates the
     column of line i of the mini-schedule)#/
mini-schedule(i,j) ← mini-schedule(i,j+1)  for j = 1, 2, ...,
     T-1;
mini-schedule(i,T) ← temp;
RETURN

SUBROUTINE SUMIT(mini-schedule)
/# this subroutine sums one column at a time from the mini-
schedule, until all the columns have been accounted for #/
DO col=1,T;
```

$$\sum_{row=1}^{D} mini\text{-}schedule(row,col)$$

```
     if(Σ ≤ N-1) RETURN;END DO;
CALL PRINTIT(mini-schedule);
RETURN

SUBROUTINE PRINTIT(mini-schedule)
/# this subroutine prints the mini-schedule one line at a time
into the file called comb.sch #/
RETURN
```

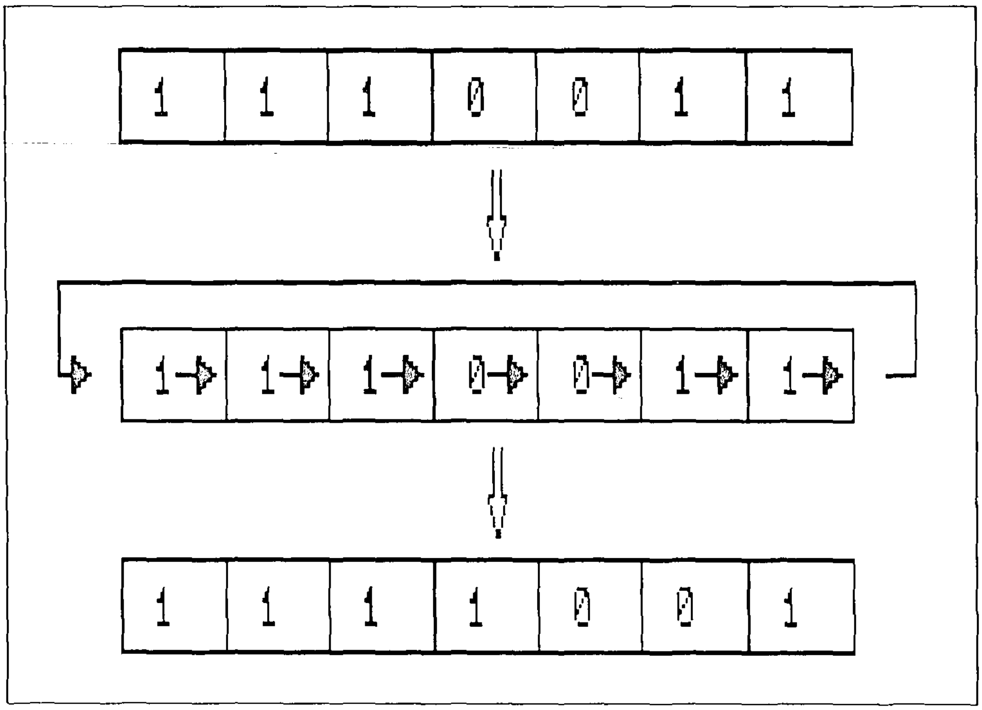

FIGURE 5

Schematic of the Rotate Subroutine

After the algorithm has been executed the resulting schedules in the file comb.sch are selectively copied into different files based on various classification schemes. The schemes could be:

1. Identify all the schedules that have the most weekend days off.
2. Identify all the schedules that have longer than two days off stretches. This could occur when a person is moving from a line in the mini-schedule which ended in an off-day period to a line that begins with an off-day period.
3. Identify all the schedules where the work period between two weekends-off is minimized.

A sample of the mini-schedules produced by applying this algorithm to an ra ratio of 2/3 is included in Table 3 on the next page. A numerical example using the algorithm follows:

00111	10	01111	11	00111	11	10011	11
11001	11	11100	11	11110	01	11100	11
11110	01	11111	00	11111	10	01111	00
01111	10	01111	10	01111	10	01111	10
11110	01	11110	01	11110	01	11110	01
10011	11	10011	11	10011	11	10011	11
10011	11	00111	10	01111	11	00111	11
01111	00	11110	01	11111	00	11111	10
11100	11	11001	11	11100	11	11110	01
11001	11	11001	11	11001	11	11001	11
01111	10	01111	10	01111	10	01111	10
11110	01	11110	01	11110	01	11110	01
11100	11	11100	11	11100	11	11110	01
10011	11	00111	11	00111	11	00111	11
01111	00	11111	00	11111	00	11111	10
11110	01	11100	11	11100	11	11110	01
11001	11	10011	11	10011	11	11001	11
01111	10	01111	00	11111	00	11111	10

TABLE 3

Sample schedules produced

for an ra ratio of 2/3 with a cycle of four weeks

3.3 NUMERICAL ILLUSTRATION

A mini-schedule with three lines and one week per line will be used as an example. The ra ratio will be 2/3. The constraint of having days off in two-day blocks is used. The maximum work stretch will be six days.

The lexicographic algorithm was used to produce the following schedules:

step 1:

0011111
1001111
1100111
1110011
1111001
1111100

It might seem redundant to have all the schedules shown above, since when the first one is rotated, the other ones are produced. But, the first line of the mini-schedules is not rotated. This practice reduces the computation time of the algorithm since only feasible schedules are used in the first line of the mini-schedule (there is no time spent checking for the feasibility of the first line).

The first line is replicated into the other two lines. The mini-schedule becomes:

step 2:

0011111

0011111

0011111

The sum of the first column is less than the numerator of the ra ratio (N=2) so this is an infeasible schedule. The third line is rotated one bit to the left, and the mini-schedule becomes:

step 3:

0011111

0011111

0111110

The first column is summed and since it is less than the numerator (N=2) the third line is rotated again. The third line is rotated a total of seven times, and every time the sum of the first column is less than 2. Now the second row is rotated one bit to the left, and the mini-schedule becomes:

step 4:

0011111

0111110

0011111

Again, the third line is rotated a total of seven times, without finding a feasible schedule. The second line is rotated again, and the mini-schedule becomes:

step 5:

0011111

1111100

0011111

The first line and the second line form a schedule that has a work stretch of ten days. All the rotations of the third line, as well as rotations of the second line, will produce no feasible schedule. The mini-schedule upon terminating the search for a schedule with initial sequence 0011111 is the mini-schedule of step 2 above. The next line is read, and the mini-schedule becomes:

1001111

1001111

1001111

The sum of the first column indicates that this might be a potentially feasible schedule. The second column is summed, and since it is less than two, this mini-schedule is no longer a candidate solution. The third line is rotated seven times, and all rotations produce infeasible mini-schedules. The algorithm repeats itself (rotate the second line twice) until the mini-schedule becomes:

1001111

0111110

1001111

The third line is rotated four times, and the schedule becomes:

1001111

0111110

1111001

The columns are summed and this could be a feasible schedule. The schedule is then searched for work stretches greater than six days, and since the search is negative, this is a feasible schedule.

The algorithm terminates once all the lines in step 1 are tested for feasible mini-schedules.

CHAPTER 4

COMPUTATIONAL INVESTIGATION

The problem has been decomposed into two modules called phases. The first phase is an ILP, and the second phase is a heuristic. An empirical study of each phase is reported in this chapter.

Phase 1

The ILP, which was presented in the introduction, is formulated so as to produce start times for three main shifts and three swing shifts, as well as to compute the number of people for each shift. The Day, Evening and Night shifts had fixed start times, that is, 7:00, 15:00, and 23:00 o'clock, respectively. The swing shifts were allowed to start any time that would cause at least one hour of overlap with the respective main shift. For example, the Day swing shift was allowed to begin anytime from midnight (24:00) until 14:00 o'clock which translates into 15 possible start times. Similarly, the two other swing shifts had 15 different start times too.

The first phase was solved using LINDO. Had it been necessary to avoid using a commercial package, a simple branch and bound algorithm could have been used (see [Press et al.]). The hourly demand, as seen in Table 1 in Chapter 1, indicates that

Monday through Thursday the demand pattern is the same. Friday has a different demand, and Saturday and Sunday have the same demand. Thus, the ILP was solved three times with different parameters each time. Solving the ILP took less than 30 seconds on a RISC/6000 workstation. The results of the three ILP's appear in Table 4.

The first column of Table 4 has the possible starting times for the shifts. The first row of the table has the seven days of the week. The numbers inside the table identify how many employees should be scheduled to start an eight-hour shift at the time indicated by the first column of the row where each number lies.

In studying the results of the three ILP's, it was noted that the start times of the swing shifts correspond to the start times for the swing shifts the police department currently uses. Also, the results indicate the minimum number of people required per day, by summing every column that corresponds to a day of the week.

The minimum number of people required for every day appears in Table 5. Note that the maximum requirement in Table 5 (55 officers) is not the upper limit on the work force size. Officers have to take days off; and hence, the size of the work force has to be larger than 55.

	mon	tue	wed	thus	fri	sat	sun
7:00	13	13	13	13	13	13	13
8:00							
9:00							
10:00							
11:00	4	4	4	4	4		
12:00							
13:00							
14:00							
15:00	15	15	15	15	17	17	17
16:00							
17:00							
18:00							
19:00	4	4	4	4	4	4	4
20:00							
21:00							
22:00							
23:00	15	15	15	15	17	17	17
24:00							
1:00							
2:00							
3:00							
4:00							
5:00							
6:00							

Table 4

Solution of the first phase ILP

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
minimum staffing requirement	51	51	51	51	55	51	51

Table 5

Minimum staffing requirements per day

based on the results from phase 1

Phase 2

The total number of work-hours required is the sum of the daily requirements times the shift length of eight hours, and it is 2888 work-hours.

The first constraint presented by [Burns and Carter] and explained in the Literature Review chapter is as follows:

$$W \geq \left\lceil \frac{B \cdot n}{B - A} \right\rceil$$

By using a work force of W=84, B=4 weeks, and n=55 people, a feasible schedule would have a maximum number of weekends off

$$A = \lceil 1.38 \rceil = 1$$

When the constraint is modified so that the weekend is defined as any of the Friday-Saturday, Saturday-Sunday, or Sunday-Monday blocks, B becomes the number of weekend days (one weekend = two weekend days) as opposed to weeks. Thus, the result of A=1 means that one weekend-off can be guaranteed in two weeks (note that this guarantee does not hold if the various constraints on work stretches and days off are too tight). So, the assumption will be that one weekend in two weeks will be off, <u>on average</u>. Using B=4 and A=1, the minimum work force is $W \geq 73$, which is within the size of the current work force.

The second constraint is

$$W \geq \left\lceil \frac{1}{5} \sum_{i=1}^{7} n_i \right\rceil$$

which computes to W $\geq$ 51, which is sufficient given the current size of the work force.

The third constraint specifies that the size of the work force has to be greater the maximum daily demand, i.e.:

$$W \geq \max_i \{n_i\},\ i=1,2,\ldots,7$$

Hence, W $\geq$ 55, which is also sufficient for a work force of 84 officers. This number is used to calculate the RA ratio.

This ratio becomes 55/84. The numbers that divide 84 evenly are 1, 2, 3, 4, 6, 7, 12, 14, 21, 28, 42, 84. Of these numbers only the numbers 2, 4, 7 and 14 divide the platoon size evenly and will be used as denominators. The following ratios, with numerators ranging from 1 to 14, are computed:

(55/84=0.65476)

3/4=0.75

5/7=0.71429

6/7=0.85714

10/14=0.71429

11/14=0.78571

12/14=0.85714

13/14=0.92857

Of these ratios 5/7 seems to be the closest to 55/84. But, using this ratio and a line length of four weeks would result in rotation that would take place every 4x7=28 weeks. This means that the schedule will become more or less fixed. It will require a rotation through shifts once every six or seven months, and the Police Department did not welcome the idea of such long rotations.

On the other hand, the greater the cycle, the longer the computation time on the computer. When a seven-line schedule with one week per line was attempted, the completion time (real time) was a few hours on a RISC/6000 work station. When 23 lines (one week per line) were used, it took more than a week to complete, and it did not identify any feasible schedules.

Also, trying a different number of weeks per line indicated that a long rotation is not likely to provide a feasible schedule if a shorter rotation fails to do so. For example, one of the ra ratios attempted was 6/7. The algorithm was repeated four times. Each time the mini-schedules had lines of one, two, three and four weeks respectively. No feasible schedules were produced for this ra ratio.

The ratio 3/4 with four weeks per line was used which would have a rotation every 16 weeks, but because of the constraints of the problem, no solution was found.

There are a few things that can be done to correct this infeasibility. One way is to modify the daily demand. When the demand for the weekend days is observed it is evident that a minimum of 19 people is required for the E and E′ shifts as well as for the D and D′ shifts. This will create an RA ratio per shift of 19/28; 19 is a prime number; thus, a schedule would be required where the rotation between shifts is 28x7= 196 weeks which is unacceptable, given the long time the algorithm needs to execute.

If the ratio is modified to 20/28, this translates into 5/7 which is the same ratio as above. If the daily requirement for people is reduced by two people on the second swing shift of Friday, Saturday and Sunday, then the ratio becomes 2/3 for which a schedule can be found, as is shown in table 6 on the next page.

Note: Table 6 contains five different schedules. Each schedule is for three officers. Each officer will be give a line of a

0111110
1111001
1001111

1111001
1100111
0111110

1111001
1100111
0111110

1111001
1001111
0111110

1100111
0111110
1111001

TABLE 6

Sample schedules produced for an ra ratio equal to 2/3 with one week per line (rotation is three weeks) given schedule to work. For example, the last schedule means that officer 1 will work, starting from the first line, the schedule 1100111 0111110 1111001, officer 2 will work the schedule 0111110 1111001 1100111, and the third officer will work the schedule 1111001 1100111 0111110. All three officers start work on a Monday, and at the end of their schedule (three weeks) they repeat it.

Note that the results in Table 6 indicate that split days off are a necessity. No schedules were produced for the ra ratio=2/3 with one-week line without a split-day off pair.

To summarize, in order to use the 2/3 ratio there is a need to reduce the work force required over Friday and the weekend days. Alternatively, six more people should be hired to increase the platoon size to 30. That would bring the RA ratio to 19/30. Another possibility would be to round the numerator to the composite number 20/30, which comes back to the 2/3 ratio.

Another option would be to relax the constraint of having days off in two-day stretches. Implicitly this constraint was relaxed as it was seen in Table 6. Split days off were created upon rotation of the lines. If this constraint is directly relaxed, then more schedule lines will be available when searching for a feasible schedule.

The following ra ratios were tested for a schedule after relaxing the constraint of days off in two-day blocks: 5/6, 3/4, 19/23, 2/3, 9/11, and 4/5. These ratios test for understaffed and overstaffed situations. An explanation for the use of the ra ratios follows:

- The 5/6 ratio would require the platoon size to decrease from 28 to 24.
- The 3/4 ratio would require the platoon to decrease to 24 officers and would reduce the maximum daily requirement per shift to 18 officers from 19.
- The 19/23 ratio is used for a hypothetical platoon of size 23.
- The 2/3 ratio is used for either a daily shift requirement of 18 people and 27 officers in a platoon, or 20 people and 30 officers in a platoon.
- The 9/11 ratio was used for a platoon size of 22 and a maximum daily requirement per shift of 18 officers.
- The 4/5 ratio is used for a platoon of 25 patrol people and a maximum shift requirement of 20 officers.

The algorithm was tested for lines being one and two weeks long for the above ratios. Another constraint is added to reduce the number of schedules since when the two day block constraint is relaxed, the number of lines built *and* accepted will be approximately

$$\binom{T}{w}$$

where w is the number of days off that can be assigned in T days. This is a very large number for lines larger than a couple of weeks. The constraint is that the maximum work stretch between split days off would be less than or equal to four days. A potential sequence for a two-week-line is 11011101111001 while the sequence 11011111011100 is not acceptable.

The results from the above runs were negative. Results were produced only for the 2/3 ra ratio. By observing the infeasible schedules it seemed obvious that work stretches of seven days had to be used. Also, the use of all the feasible sequences of days on and days off might produce a feasible solution.

Another attempt to find feasible schedules was to create the lines file without the previous constraint. It was carried out for one- and two-week lines. The algorithm was repeated for the previously mentioned schedules. Again, feasible schedules were found only for the 2/3 ratio.

Since small rotation periods and small ra ratios seem to not yield good results, the 5/7 ratio (mentioned above) with lines ranging from one to four weeks was attempted. It took the algorithm less than an hour to complete the one- and two-week lines, while it took several days of number crunching for the three- and four-week lines. Feasible schedules were produced. Sample schedules are shown in Table 7 on the next page with Saturday and Sunday highlighted for four week lines.

The computation times (real time) on the RISC/6000 work stations for the various ra ratios ranged from a couple of minutes for ra ratios using one- or two-week lines and a denominator less than or equal to 7, to several days for larger ratios and three- or four-week lines.

To summarize the results: Several ra ratios were attempted. The numerators ranged from 2 to 19 and the denominators ranged from 3 to 23. Feasible schedules were produced for the following ra ratios: 2/3, 3/5, 5/7.

Yet another option seems to be to increase the number of people in the E and N shifts and to reduce the number of people in the D shift. This would be acceptable had the problem not required rotating shifts.

00111	10	01111	11	00111	10	01111	11
00111	11	10011	11	00111	11	10011	11
11001	11	11100	11	11001	11	11100	11
11110	01	11111	00	11110	01	11111	00
11111	10	01111	00	11111	10	01111	00
11110	01	11100	11	11110	01	11100	11
11001	11	10011	11	11001	11	10011	11

00111	10	01111	11	00111	10	01111	11
00111	11	10011	11	00111	11	10011	11
11001	11	11100	11	11001	11	11100	11
11110	01	11111	00	11110	01	11111	00
11111	10	01111	00	11111	10	01111	00
11110	01	11100	11	11110	01	11100	11
11001	11	10011	11	11001	11	10011	11

00111	10	01111	11	00111	10	01111	11
00111	11	10011	11	00111	11	10011	11
11001	11	11100	11	11001	11	11100	11
11110	01	11111	00	11110	01	11111	00
11111	10	01111	00	11111	10	01111	00
11110	01	11100	11	11110	01	11100	11
11001	11	10011	11	11001	11	10011	11

TABLE 7

SAMPLE SCHEDULES PRODUCED WITH AN ra RATIO OF 5/7

and four weeks per line

Due to the problems presented above with the structure of the daily requirements and the structure of the platoons, the police force is considering non-rotating shifts. Had non-rotating shifts been used, the size of the platoons would stay the same while the RA ratios for the D, E, and N shifts would become 14/28, 20/28 and 20/28 respectively.

The total number of required work-hours is the sum of all the entries in Table 3, i.e. 2808 hours. The total number of work-hours required, as presented by the solution of the first phase (Table 2) and calculated similarly, is 2888 hours which is a (2888-2808)/2808 = 2.9% excess of work-hours. By using the ratio 2/3 and increasing the platoon size by two people, the overstaffing in work-hours becomes

$$\frac{(60\ people) \times \left(8\ \frac{hours}{day}\right) \times \left(7\ \frac{days}{week}\right) - 2808\frac{people \cdot hours}{week}}{2808\frac{people \cdot hours}{week}} = 19.7\%$$

This number might seem high. But considering that officers need to be on court duty, are required to write reports, need a break for lunch, etc., it makes sense to have these excess work-hours. Having these extra hours is essential so that chores can be completed without reducing the number of patrols. One hour for a lunch break is 12.5% of an eight-hour shift; 19.7% of the same shift is almost 95 minutes which means the additional 35 minutes can be used for non-patrol activities.

Also, vacation times have not been considered. Since the overstaffing seems to be more evident during the D shift, people will be able to take extra days off during that shift without affecting the daily patrol requirements.

When the 2/3 schedules were computed, they were sorted based on the size of the maximum work stretch, the maximum number of two-days off blocks, the time between weekends off, and the most even distribution of weekends off.

Three sample master schedules for the ra ratios 2/3, 3/5, and 5/7, with one week per line, for different police force sizes, are illustrated in Appendix B.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

In this thesis the problem of manpower allocation and scheduling of people for a 24-hour, 3-shift, 7-days per week operation was attacked. The scheduling of the Bethlehem Police Department was used as an example.

A modular approach was taken in order to solve the scheduling problem which introduced the notion of "mini-schedules". A similar approach was taken in the references: [Panton], [Burns and Koop], [Tien et al.].

A key point behind the construction of the mini-schedules is that while the first line of the schedule consists of a particular sequence of days on and days off, the remaining lines contain the *same* sequence but shifted. This is less time consuming in searching for a feasible schedule and more flexible than if the possible sequences were stored in separate files for each of the lines. This way, schedules that might have been rejected because by themselves they might be infeasible, now are accepted because they are examined as part of a sequence (the rotation through the lines).

The problem was split into two phases. The first phase used an ILP formulation to solve for the manpower allocation problem.

It identified the daily staffing of the shifts, the number of swing shifts required, and the starting times for the swing shifts.

The second phase used a heuristic to schedule the available officers. The heuristic seems to work well when the ratio of required over available staff can be approximated with a ratio where the denominator is less than or equal to five. This means that the execution time for the algorithm ranged from a couple of minutes to several hours for "small" ra ratios (2/3 to 6/7 with one- or two- week lines). On the other hand, the algorithm took several days to execute for "large" ra ratios (6/7 to 19/23 with three- and four-week lines). This is to be expected because of the combinatorial nature of the problem.

[Rosenbloom et al.] noted that an attractive feature of their algorithm is that it only needed to be carried out once. This was an appealing conclusion which was adapted for this algorithm. The size of the police force is not likely to change drastically within a short period of time. So, the scheduling personnel might try out RA ratios without being stressed for timely results.

Formulas given in the literature were modified for the current problem and used to prove its feasibility [Burns] and [Burns and Carter]. The idea behind the modification was that when the weekend is defined as any of the pairs Friday-Saturday, Saturday-Sunday, and Sunday-Monday, the number of weekends off in a given schedule is doubled.

This algorithm may suggest improvements such as more or less staff, fixed shifts, and the necessity of relaxing certain constraints.

The solution, for a work force of 84 officers, produced a 19.7% excess of work-hours which translates into one hour and 35 minutes that an officer can be absent for a lunch break, etc., without violating staffing requirements.

The algorithm is quite easy to understand and quick to execute for certain configurations of the problem. It can be modified for any number of constraints. It produces mini-schedules that can be combined or replicated for the whole staff, and it can be readily computerized.

The starting times of the main shifts should be reconsidered. By simply plotting the daily demand (see Figure 6), it is evident that a Night shift (or a Day shift) which begins at 3:00 o'clock would produce a more even covering of the daily requirements, thus reducing the number of swing shifts required.

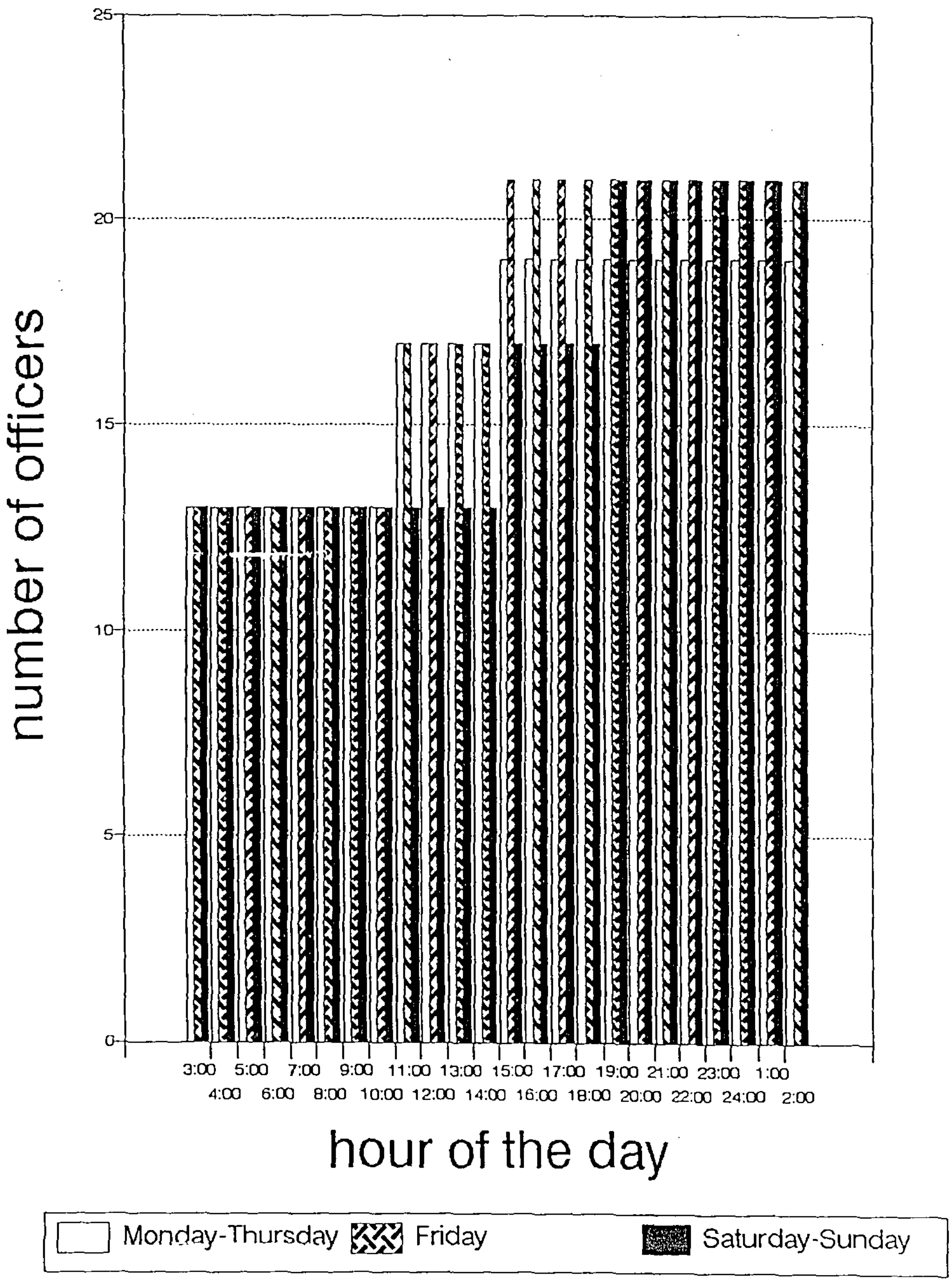

Figure 6

Daily manpower requirements

Further research is suggested to make the algorithm adapt to RA ratios where the numerator is a prime number and to make the algorithm faster for ra denominator values greater than seven. This could be achieved by using a method that limits the iterations of the do-loops if duplicate schedules are produced.

Another suggestion for further research is to create a scheme where the feasibility of the problem can be predicted, before attempting to solve for large ra ratios, based on the constraints of the given problem. This will be helpful when testing for large ra ratios.

BIBLIOGRAPHY

Bechtold, Stephen E., and Brusco, Michael J., and Showalter, Michael J., "A Comparative Evaluation of Labor Tour Scheduling Methods", Decision Sciences, vol 22, pp. 683-696, 1991.

Brownell, William S., and Lowerre, James M., "Scheduling of Work Forces Required in Continuous Operations Under Alternative Labor Policies", Management Science, vol 22, No. 5, January, 1976, pp. 597-605.

Brusco, Michael J., and Jacobs, Larry W., "A Simulated Annealing Approach to the Cyclic Staff-Scheduling Problem", Naval Research Logistics, vol 40, 1993, pp. 69-84.

Burns, R.N. ,"Manpower Scheduling With Variable Demands And Alternate Weekends Off", INFOR, vol 16, No. 2, June 1987, pp. 101-111.

Burns, R.N., and Carter, M.W., "Work Force Size and Single Shift Schedules with Variable Demands", Management Science, vol 31, No. 5, May 1985, pp. 599-607.

Burns, R.N., and Koop, G.J., "A Modular Approach to Optimal Multiple-Shift Manpower Scheduling", Operations Research, vol 35, No. 1, January-February 1987, pp. 100-110.

Denardo, Eric V.: Dynamic Programming, Models and Applications. Prentice Hall, Inc., 1982.

Harvard University News Office for the Medical Area, "Health Benefits of Improved Scheduling, Dr. Czeisler Adjusts Police Shifts to Accommodate Circadian Rhythms", Focus, p. 1, Harvard University: January 28, 1988.

Kendall, Kenneth E., and Zionts, Stanley, "Solving Integer Programming Problems by Aggregating Constraints", Operations Research, vol 25, No. 2, March-April, 1977.

Lowerre, James M., "Work Stretch Properties for the Scheduling of Continuous Operations Under Alternative Labor Policies", Management Science, vol 23, No. 9, May, 1977, pp. 963-971.

Morris, James G., and Showalter, Michael J., "Sample Approaches to Shift, Days-Off and Tour Scheduling Problems", Management Science, vol 29, No. 8, August 1983, pp. 942-950.

Nanda, Ravinder, and Browne, Jim. Introduction to Employee Scheduling. pp. 9-10, New York: Van Norstrand Reinhold, 1992.

Nijenhuis, Albert, and Wilf, Herbert S.: Combinatorial Algorithms, for Computers and Calculators. New York: Academic Press, Inc. 1978.

Panton, David M., "On the Creation of Multiple Shift Continuous Operation Cyclic Rosters Under General Work Force Conditions", Asia-Pacific Journal of Operational Research, 8, (1991), p.189-201.

Press, William H., and Flannery, Brian P., and Teukolsky, Saul A., and Vetterling, William T. Numerical Recipes: The Art of Scientific Computing (Fortran Version). Cambridge University Press, 1989.

Rosenbloom, E.S., and Goertzen, N.F., "Cyclic Nurse Scheduling", European Journal of Operational Research, 31, (1987), pp. 19-23.

Segal, M., "The Operator Scheduling Problem: A Network Flow Approach", Operations Research, 22, pp. 808-823, 1974.

Sullivan, Craig A., "Tactical Sleeping: Sleep Deprivation Can Be As Dangerous As Getting Shot", Police, June 1991.

Tien, James M., and Kamiyama, Angelica, "On Manpower Scheduling Algorithms", Siam Review, vol 24, No. 3, July 1982.

United States Department of Labor, Employment Standards Administration, Wage and Hour Division, "Fair Labor Standards Act of 1938, As Amended", Washington, D.C: WH Publication 1318, 1990.

Winston, Wayne L. Operations Research: Applications and Algorithms. Boston: PWS-Kent, 1987, pp. 391-404.

APPENDIX A

NOMENCLATURE

Cycle: The length, in days, of the cyclic schedule.

Cyclic schedule: A sequence of days on and days off that repeats every certain number of days.

Days off: The days a person is not assigned to work. A zero will indicate days off.

Days on: The days a person is assigned to work. Days on will be indicated with the number one.

Fixed schedule: A schedule where people do not rotate through shifts.

Main shift: Either one of the Day (D), Evening (E) or Night (N) shifts. Main shifts have most of the people working.

Modulo arithmetic: The remainder of the division of two numbers. If the divisor is greater than the dividend, the modulo is the positive difference of the two numbers.

RA ratio: The ratio of the required over the available people that can work on a given day.

ra ratio: A common multiple of the RA ratio, with different characteristics at different times..

Rotatable schedule: A schedule where at each complete rotation a person is assigned to a different shift.

Shift: A period of the day, that has a certain starting and ending time, during which a person is to work. The length of a typical shift is eight hours.

Split days off: A block of off-days that is non-continuous, such as two single days off separated by a work stretch. The schedule 1110110 has split days off.

Swing shift: A shift that does not have the same start time as either of the three main shifts but has the same length as the main shifts. It is used to compensate for peak period demands. The swing shifts are denoted by the letter of the main shift they overlap with a prime (D′, E′, N′). A maximum of only three swing shifts will be used.

Tour: A certain sequence of either main or swing shifts, or a combination of both. For example, L days on D', L days on E, and L days on N' would be a particular tour.

Week: A seven day period which consists of the days Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday, in this order.

Weekend: Any of the following two-days off stretches: Friday-Saturday, Saturday-Sunday, and Sunday-Monday.

Work stretch: A number of days on with no days-off in it. For example, the pattern 111110011111000 is a cyclic schedule, read "5-2-5-3", with a cycle of 15 days and a work stretch of five days.

APPENDIX B

SAMPLE MASTER SCHEDULES

Sample master schedules are produced for the ra ratios 2/3, 3/5, and 5/7. The size of the police force is 81, 75, and 63 respectively. The mini-schedules used, have the days on identified with a "1", and the days off with a "0". In the master schedule, instead of "1" for days on, lower case letters of the three main shifts are used. A "d" means that an officer will work Day shift on a particular day; an "e", Evening shift; and, an "n", Night shift. Swing shifts will be calculated internally by the Police Department.

The columns of the master schedules are days of the week, and they are identified with the first letter of the day. All the mini-schedules used have one week per line.

The master schedules are split over several pages. In order to read the schedules, the pages that make up a schedule have to be placed next to each other.

Master Schedule 2/3

This schedule is for 81 officers. There are 27 officers per shift. The mini-schedule used is:

0111101

1101011

1011110

to construct a master schedule consisting of nine individual lines of shift assignments and days off patterns (Tables 8 and 9). The following assignments can be made:

Officers 1 through 9 will work according to line 1

Officers 10 through 18 will work according to line 2

Officers 19 through 27 will work according to line 3

Officers 28 through 36 will work according to line 4

Officers 37 through 45 will work according to line 5

Officers 46 through 54 will work according to line 6

Officers 55 through 63 will work according to line 7

Officers 64 through 72 will work according to line 8

Officers 73 through 81 will work according to line 9

	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS
line 1 :	0dddd	0d	dd0d0	dd	d0ddd	d0	0eeee	0e	ee0e0	ee	e0eee	e0
line 2 :	dd0d0	dd	d0ddd	d0	0eeee	0e	ee0e0	ee	e0eee	e0	0nnnn	0n
line 3 :	d0ddd	d0	0eeee	0e	ee0e0	ee	e0eee	e0	0nnnn	0n	nn0n0	nn
line 4 :	0eeee	0e	ee0e0	ee	e0eee	e0	0nnnn	0n	nn0n0	nn	n0nnn	n0
line 5 :	ee0e0	ee	e0eee	e0	0nnnn	0n	nn0n0	nn	n0nnn	n0	0dddd	0d
line 6 :	e0eee	e0	0nnnn	0n	nn0n0	nn	n0nnn	n0	0dddd	0d	dd0d0	dd
line 7 :	0nnnn	0n	nn0n0	nn	n0nnn	n0	0dddd	0d	dd0d0	dd	d0ddd	d0
line 8 :	nn0n0	nn	n0nnn	n0	0dddd	0d	dd0d0	dd	d0ddd	d0	0eeee	0e
line 9 :	n0nnn	n0	0dddd	0d	dd0d0	dd	d0ddd	d0	0eeee	0e	ee0e0	ee

Table 8

Master Schedule for an ra ratio equal to 2/3

(continues on the next page)

MTWTF	SS	MTWTF	SS	MTWTF	SS
0nnnn	0n	nn0n0	nn	n0nnn	n0
nn0n0	nn	n0nnn	n0	0dddd	0d
n0nnn	n0	0dddd	0d	dd0d0	dd
0dddd	0d	dd0d0	dd	d0ddd	d0
dd0d0	dd	d0ddd	d0	0eeee	0e
d0ddd	d0	0eeee	0e	ee0e0	ee
0eeee	0e	ee0e0	ee	e0eee	e0
ee0e0	ee	e0eee	e0	0nnnn	0n
e0eee	e0	0nnnn	0n	nn0n0	nn

Table 9

Master Schedule for an ra ratio equal to 2/3

(last page)

Master Schedule 3/5

This schedule is for 75 officers. There are 25 officers per shift. The ra ratio is 3/5. The mini-schedule used is:

0011111

1001111

1100111

0111110

1111100

to construct a master schedule consisting of 15 individual lines of shift assignments and days off patterns (Tables 10,11, and 12). The following assignments can be made:

Officers that work according to line 1: 1 - 5

Officers that work according to line 2: 6 - 10

Officers that work according to line 3: 11 - 15

Officers that work according to line 4: 16 - 20

Officers that work according to line 5: 21 - 25

Officers that work according to line 6: 26 - 30

Officers that work according to line 7: 31 - 35

Officers that work according to line 8: 36 - 40

Officers that work according to line 9: 41 - 45

Officers that work according to line 10: 46 - 50

Officers that work according to line 11: 51 - 55

Officers that work according to line 12: 56 - 60

Officers that work according to line 13: 61 - 65

Officers that work according to line 14: 66 - 70

Officers that work according to line 15: 71 - 75

		MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS
line	1 :	00ddd	dd	d00dd	dd	dd00d	dd	0dddd	d0	ddddd	00	00eee	ee
line	2 :	d00dd	dd	dd00d	dd	0dddd	d0	ddddd	00	00eee	ee	e00ee	ee
line	3 :	dd00d	dd	0dddd	d0	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee
line	4 :	0dddd	d0	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	0eeee	e0
line	5 :	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	0eeee	e0	eeeee	00
line	6 :	00eee	ee	e00ee	ee	ee00e	ee	0eeee	e0	eeeee	00	00nnn	nn
line	7 :	e00ee	ee	ee00e	ee	0eeee	e0	eeeee	00	00nnn	nn	n00nn	nn
line	8 :	ee00e	ee	0eeee	e0	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn
line	9 :	0eeee	e0	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	0nnnn	n0
line	10 :	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	0nnnn	n0	nnnnn	00
line	11 :	00nnn	nn	n00nn	nn	nn00n	nn	0nnnn	n0	nnnnn	00	00ddd	dd
line	12 :	n00nn	nn	nn00n	nn	0nnnn	n0	nnnnn	00	00ddd	dd	d00dd	dd
line	13 :	nn00n	nn	0nnnn	n0	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd
line	14 :	0nnnn	n0	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	0dddd	d0
line	15 :	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	0dddd	d0	ddddd	00

Table 10

Master Schedule for an ra ratio equal to 3/5

(continues on the next page)

MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS
e00ee	ee	ee00e	ee	0eeee	e0	eeeee	00	00nnn	nn	n00nn	nn
ee00e	ee	0eeee	e0	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn
0eeee	e0	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	0nnnn	n0
eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	0nnnn	n0	nnnnn	00
00nnn	nn	n00nn	nn	nn00n	nn	0nnnn	n0	nnnnn	00	00ddd	dd
n00nn	nn	nn00n	nn	0nnnn	n0	nnnnn	00	00ddd	dd	d00dd	dd
nn00n	nn	0nnnn	n0	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd
0nnnn	n0	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	0dddd	d0
nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	0dddd	d0	ddddd	00
00ddd	dd	d00dd	dd	dd00d	dd	0dddd	d0	ddddd	00	00eee	ee
d00dd	dd	dd00d	dd	0dddd	d0	ddddd	00	00eee	ee	e00ee	ee
dd00d	dd	0dddd	d0	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee
0dddd	d0	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	0eeee	e0
ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	0eeee	e0	eeeee	00
00eee	ee	e00ee	ee	ee00e	ee	0eeee	e0	eeeee	00	00nnn	nn

Table 11

Master Schedule for an ra ratio equal to 3/5

(continues on the next page)

MTWTF	SS	MTWTF	SS	MTWTF	SS
nn00n	nn	0nnnn	n0	nnnnn	00
0nnnn	n0	nnnnn	00	00ddd	dd
nnnnn	00	00ddd	dd	d00dd	dd
00ddd	dd	d00dd	dd	dd00d	dd
d00dd	dd	dd00d	dd	0dddd	d0
dd00d	dd	0dddd	d0	ddddd	00
0dddd	d0	ddddd	00	00eee	ee
ddddd	00	00eee	ee	e00ee	ee
00eee	ee	e00ee	ee	ee00e	ee
e00ee	ee	ee00e	ee	0eeee	e0
ee00e	ee	0eeee	e0	eeeee	00
0eeee	e0	eeeee	00	00nnn	nn
eeeee	00	00nnn	nn	n00nn	nn
00nnn	nn	n00nn	nn	nn00n	nn
n00nn	nn	nn00n	nn	0nnnn	n0

Table 12

Master Schedule for an ra ratio equal to 3/5

(last page)

Master Schedule 5/7

This schedule is for 63 people. The ra ratio is 5/7. Each shift has 21 officers. The mini-schedule used is:

0011111

1001111

1100111

1110011

0111110

1111001

1111100

to construct a master schedule consisting of individual lines of shift assignments and days off patterns (Tables 13 - 16). The following assignments can be made:

Line 1 will be assigned to officers 1 - 3

Line 2 will be assigned to officers 4 - 6

Line 3 will be assigned to officers 7 - 9

Line 4 will be assigned to officers 10 - 12

Line 5 will be assigned to officers 13 - 15

Line 6 will be assigned to officers 16 - 18

Line 7 will be assigned to officers 19 - 21

Line 8 will be assigned to officers 22 - 24

Line 9 will be assigned to officers 25 - 27

Line 10 will be assigned to officers 28 - 30

Line 11 will be assigned to officers 31 - 33

Line 12 will be assigned to officers 34 - 36

Line 13 will be assigned to officers 37 - 39

Line 14 will be assigned to officers 40 - 42

Line 15 will be assigned to officers 43 - 45

Line 16 will be assigned to officers 46 - 48

Line 17 will be assigned to officers 49 - 51

Line 18 will be assigned to officers 52 - 54

Line 19 will be assigned to officers 55 - 57

Line 20 will be assigned to officers 58 - 60

Line 21 will be assigned to officers 61 - 63

	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS
line 1 :	00ddd	dd	d00dd	dd	dd00d	dd	ddd00	dd	0dddd	d0	dddd0	0d
line 2 :	d00dd	dd	dd00d	dd	ddd00	dd	0dddd	d0	dddd0	0d	ddddd	00
line 3 :	dd00d	dd	ddd00	dd	0dddd	d0	dddd0	0d	ddddd	00	00eee	ee
line 4 :	ddd00	dd	0dddd	d0	dddd0	0d	ddddd	00	00eee	ee	e00ee	ee
line 5 :	0dddd	d0	dddd0	0d	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee
line 6 :	dddd0	0d	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	eee00	ee
line 7 :	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	eee00	ee	0eeee	e0
line 8 :	00eee	ee	e00ee	ee	ee00e	ee	eee00	ee	0eeee	e0	eeee0	0e
line 9 :	e00ee	ee	ee00e	ee	eee00	ee	0eeee	e0	eeee0	0e	eeeee	00
line 10 :	ee00e	ee	eee00	ee	0eeee	e0	eeee0	0e	eeeee	00	00nnn	nn
line 11 :	eee00	ee	0eeee	e0	eeee0	0e	eeeee	00	00nnn	nn	n00nn	nn
line 12 :	0eeee	e0	eeee0	0e	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn
line 13 :	eeee0	0e	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	nnn00	nn
line 14 :	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	nnn00	nn	0nnnn	n0
line 15 :	00nnn	nn	n00nn	nn	nn00n	nn	nnn00	nn	0nnnn	n0	nnnn0	0n
line 16 :	n00nn	nn	nn00n	nn	nnn00	nn	0nnnn	n0	nnnn0	0n	nnnnn	00
line 17 :	nn00n	nn	nnn00	nn	0nnnn	n0	nnnn0	0n	nnnnn	00	00ddd	dd
line 18 :	nnn00	nn	0nnnn	n0	nnnn0	0n	nnnnn	00	00ddd	dd	d00dd	dd
line 19 :	0nnnn	n0	nnnn0	0n	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd
line 20 :	nnnn0	0n	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	ddd00	dd
line 21 :	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	ddd00	dd	0dddd	d0

Table 13

Master Schedule for an ra ratio equal to 5/7

(continued on the next page)

MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS
ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	eee00	ee	0eeee	e0
00eee	ee	e00ee	ee	ee00e	ee	eee00	ee	0eeee	e0	eeee0	0e
e00ee	ee	ee00e	ee	eee00	ee	0eeee	e0	eeee0	0e	eeeee	00
ee00e	ee	eee00	ee	0eeee	e0	eeee0	0e	eeeee	00	00nnn	nn
eee00	ee	0eeee	e0	eeee0	0e	eeeee	00	00nnn	nn	n00nn	nn
0eeee	e0	eeee0	0e	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn
eeee0	0e	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	nnn00	nn
eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	nnn00	nn	0nnnn	n0
00nnn	nn	n00nn	nn	nn00n	nn	nnn00	nn	0nnnn	n0	nnnn0	0n
n00nn	nn	nn00n	nn	nnn00	nn	0nnnn	n0	nnnn0	0n	nnnnn	00
nn00n	nn	nnn00	nn	0nnnn	n0	nnnn0	0n	nnnnn	00	00ddd	dd
nnn00	nn	0nnnn	n0	nnnn0	0n	nnnnn	00	00ddd	dd	d00dd	dd
0nnnn	n0	nnnn0	0n	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd
nnnn0	0n	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	ddd00	dd
nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	ddd00	dd	0dddd	d0
00ddd	dd	d00dd	dd	dd00d	dd	ddd00	dd	0dddd	d0	dddd0	0d
d00dd	dd	dd00d	dd	ddd00	dd	0dddd	d0	dddd0	0d	ddddd	00
dd00d	dd	ddd00	dd	0dddd	d0	dddd0	0d	ddddd	00	00eee	ee
ddd00	dd	0dddd	d0	dddd0	0d	ddddd	00	00eee	ee	e00ee	ee
0dddd	d0	dddd0	0d	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee
dddd0	0d	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	eee00	ee

Table 14

Master Schedule for an ra ratio equal to 5/7

(continued on the next page)

MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS	MTWTF	SS
eeee0	0e	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	nnn00	nn
eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn	nnn00	nn	0nnnn	n0
00nnn	nn	n00nn	nn	nn00n	nn	nnn00	nn	0nnnn	n0	nnnn0	0n
n00nn	nn	nn00n	nn	nnn00	nn	0nnnn	n0	nnnn0	0n	nnnnn	00
nn00n	nn	nnn00	nn	0nnnn	n0	nnnn0	0n	nnnnn	00	00ddd	dd
nnn00	nn	0nnnn	n0	nnnn0	0n	nnnnn	00	00ddd	dd	d00dd	dd
0nnnn	n0	nnnn0	0n	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd
nnnn0	0n	nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	ddd00	dd
nnnnn	00	00ddd	dd	d00dd	dd	dd00d	dd	ddd00	dd	0dddd	d0
00ddd	dd	d00dd	dd	dd00d	dd	ddd00	dd	0dddd	d0	dddd0	0d
d00dd	dd	dd00d	dd	ddd00	dd	0dddd	d0	dddd0	0d	ddddd	00
dd00d	dd	ddd00	dd	0dddd	d0	dddd0	0d	ddddd	00	00eee	ee
ddd00	dd	0dddd	d0	dddd0	0d	ddddd	00	00eee	ee	e00ee	ee
0dddd	d0	dddd0	0d	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee
dddd0	0d	ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	eee00	ee
ddddd	00	00eee	ee	e00ee	ee	ee00e	ee	eee00	ee	0eeee	e0
00eee	ee	e00ee	ee	ee00e	ee	eee00	ee	0eeee	e0	eeee0	0e
e00ee	ee	ee00e	ee	eee00	ee	0eeee	e0	eeee0	0e	eeeee	00
ee00e	ee	eee00	ee	0eeee	e0	eeee0	0e	eeeee	00	00nnn	nn
eee00	ee	0eeee	e0	eeee0	0e	eeeee	00	00nnn	nn	n00nn	nn
0eeee	e0	eeee0	0e	eeeee	00	00nnn	nn	n00nn	nn	nn00n	nn

Table 15

Master Schedule for an ra ratio equal to 5/7

(continued on the next page)

MTWTF	SS	MTWTF	SS	MTWTF	SS
0nnnn	n0	nnnn0	0n	nnnnn	00
nnnn0	0n	nnnnn	00	00ddd	dd
nnnnn	00	00ddd	dd	d00dd	dd
00ddd	dd	d00dd	dd	dd00d	dd
d00dd	dd	dd00d	dd	ddd00	dd
dd00d	dd	ddd00	dd	0dddd	d0
ddd00	dd	0dddd	d0	dddd0	0d
0dddd	d0	dddd0	0d	ddddd	00
dddd0	0d	ddddd	00	00eee	ee
ddddd	00	00eee	ee	e00ee	ee
00eee	ee	e00ee	ee	ee00e	ee
e00ee	ee	ee00e	ee	eee00	ee
ee00e	ee	eee00	ee	0eeee	e0
eee00	ee	0eeee	e0	eeee0	0e
0eeee	e0	eeee0	0e	eeeee	00
eeee0	0e	eeeee	00	00nnn	nn
eeeee	00	00nnn	nn	n00nn	nn
00nnn	nn	n00nn	nn	nn00n	nn
n00nn	nn	nn00n	nn	nnn00	nn
nn00n	nn	nnn00	nn	0nnnn	n0
nnn00	nn	0nnnn	n0	nnnn0	0n

Table 16

Master Schedule for an ra ratio equal to 5/7

(last page)

VITA

The author, Charalambos Marangos, was born on June 15, 1967 to Mrs. Violetta Stylianou and Mr. Akis Marangos in Nicosia, Cyprus. His undergraduate study was done at Lehigh University where he received a Bachelor of Science in Industrial Engineering in 1991. He had been awarded a four-year full scholarship by Fulbright/Amideast. While an undergraduate he served in the Student Senate, a student governing body, and the Forum, a student, faculty and staff governing body at Lehigh University. He worked in KRUPP Steel, in Germany, during the summer of 1989. He served as a visiting instructor to Lehigh University during the summer of 1991. He will receive a Masters of Science in Industrial Engineering from Lehigh University in the Fall of 1993. He was awarded Teaching Assistantships in the College of Engineering and in the Industrial Engineering Department throughout his graduate study. He is a member of the Honor Society for International Scholars, Phi Beta Delta, and a member of the Industrial Engineering Honor Society, Alpha Pi Mu. He has worked on numerous projects with various manufacturing companies throughout his undergraduate and graduate study through the CIM Lab at Lehigh University, doing industrial simulations. He is proficient with various computer languages and software. He is planning to work in industry for a few years and then return to academia for a Ph.D.

END

OF

TITLE