A QUEUEING-LINEAR PROGRAMMING APPROACH TO SCHEDULING POLICE PATROL CARS

Peter J. Kolesar
Kenneth L. Rider
Thomas B. Crabill
Warren E. Walker

March 1975

P-5260-1

In recent years, a good deal of effort has been devoted to the development and application of quantitative methods to assist police departments in making resource allocation and manpower scheduling decisions. The work of Larson [13] and of Heller [7] are important examples. Surveys and evaluations of much of the pertinent literature in this area are given by Chaiken and Larson [2], and by Gass, Dawson, et al. [5]. This paper can be viewed as part of this general effort. It presents a methodology for creating new patrol car schedules that improve the correspondence between patrol car availability and demands for service. The approach is currently being tested for use by the New York City Police Department.

The level of demand for police service varies considerably through the day. Recognizing this, police departments assign more patrol cars to duty during the busy hours. Figure 1 illustrates both of these points using data from one New York City police precinct. Scheduling decisions, however, have usually been more or less educated guesses. Improvements resulting from better schedules can be quite significant. In an example given below, a schedule derived using the methods presented in this paper maintains performance standards with almost a 25 percent reduction in the number of cars fielded under the traditional schedule.

We view the patrol environment as a complicated multiple-server queueing system. Calls for service--either telephone calls to the radio dispatcher who sends the cars to the scene, or accidents, crimes, and other incidents encountered by the patroling units--are assumed to occur randomly over time. These calls require random amounts of "service time" by one or more patrol cars. When not working on such "jobs" a patrol car is presumed to be either on preventive patrol or out of service for some reason. A variety of mathematical models of this environment can be formulated depending on the objectives of the analysis. Our objective is to use queueing theory to generate estimates of hourly car requirements needed as input to a scheduling model and then to evaluate the resulting schedule with respect to various measures of patrol system performance. We discuss methods for generating estimates of patrol car requirements in Section I, an integer program for generating schedules given patrol car requirements in Section II, and a time-dependent queueing model for evaluating schedule performance in

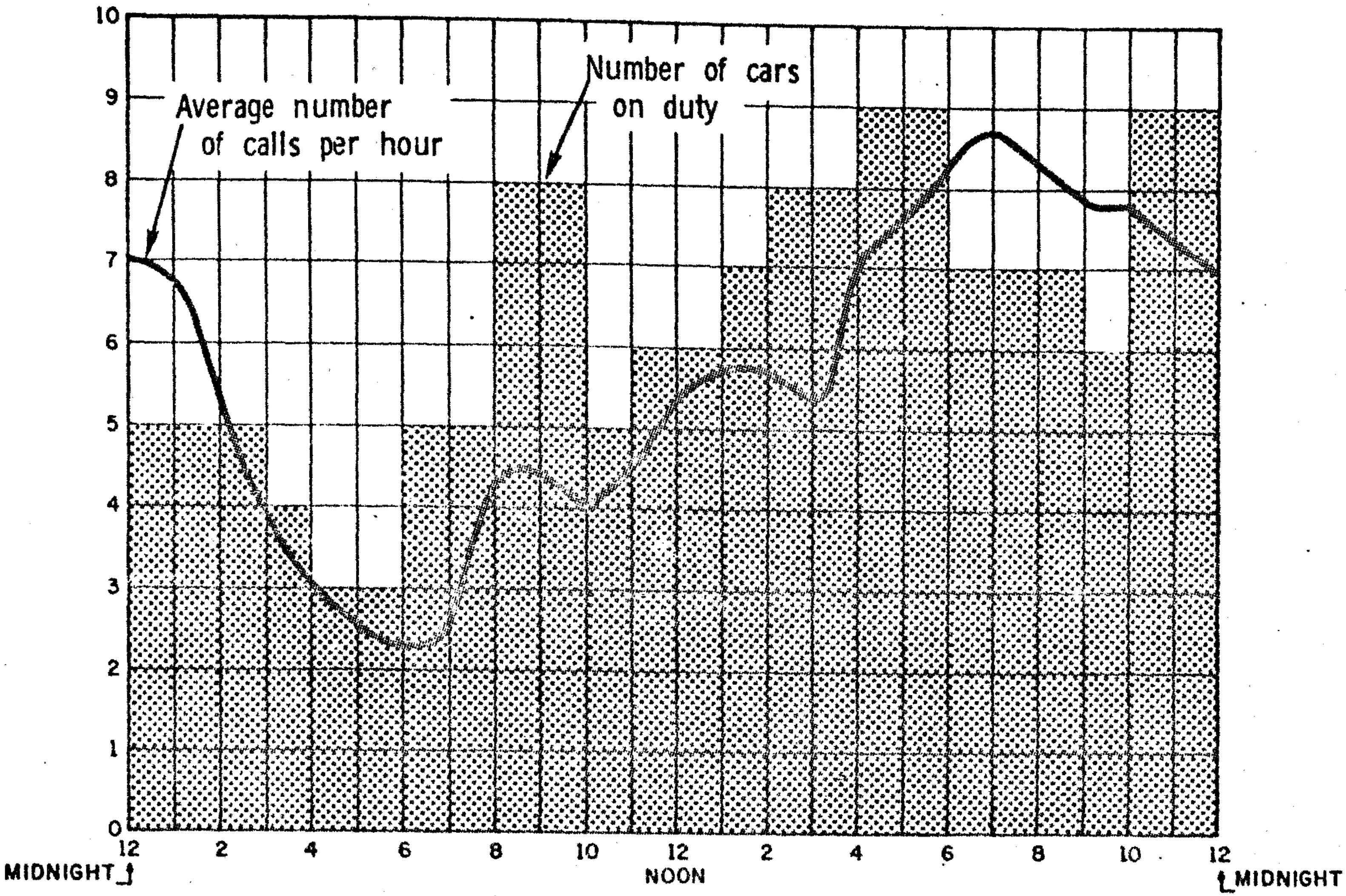

Fig. 1. Call rate and patrol car schedule as functions of time of day in one New York City police precinct

Section III. Section IV uses data from one of New York City's police precincts to show how the method was applied to a sample problem.

The methodology that we have developed for creating and evaluating schedules is iterative. An overview is provided below.

- *Specify the Policy Constraints on the Schedules.* That is, what tours of duty and mealtime breaks are permitted? (See Section IV.)
- *Generate Estimates of Hour-by-Hour Car Requirements.* This can be done using one of several mathematical models together with, and modified by, police judgment. (See Section I.)
- *Obtain an Optimal Schedule.* Using the specifications made above, an integer linear program is generated and solved that satisfies all of the constraints using the fewest number of cars. (See Section II.)
- *Evaluate the Schedule.* Using a time-dependent queueing model, detailed information is provided on the levels of police service that would result over the day if the schedule were implemented. (See Section III.)
- *Revise the Constraints.* Since approximate models were used to generate car requirements, the schedule generated may not actually meet all the performance levels desired. Moreover, there may be performance measures of interest besides those explicitly used in estimating the patrol car requirements. This should be remembered when evaluating the resulting schedules. Based on this evaluation, the requirements and constraints of the integer linear program can be revised and the problem resolved. This process may be repeated as often as necessary. (See Section IV.)

I. ESTIMATING PATROL CAR REQUIREMENTS

Using almost any measure of effectiveness, patrol performance is improved by increasing the number of patrol cars on duty. Queueing delays and response times decrease, while car availability and the hours spent on preventive patrol increase. Most other indicators of the quality or quantity of patrol service exhibit improvement where the number of cars is increased. In this section we show, as an example, how one specific measure of service performance is quantitatively related to the number of cars fielded, and how this relationship--really an approximation--can be used to generate estimates of the number of cars required at each hour of the day to obtain desired police performance levels. These requirement estimates can then be used as input to the integer programming model described in Section II. Similar methods can be used to derive car requirements based on other queueing-related service measures.

The Probability That All Cars Are Busy

The relationship between the number of cars on duty during hour t and α, the probability that all the cars are busy, is complicated. The complications arise from the fact that we are dealing with the multiple-server queueing situation, in which the call rate, the number of cars on duty, and other factors are time-dependent. Suppose that the hourly call rates, λ_t, and service rates, μ_t, are known, and suppose also that α has been specified. We want to find r_t, the smallest number of cars to place on duty during hour t so that, given λ_t and μ_t, the probability that all r_t cars are busy at a random epoch during hour t is less than α.

This problem is difficult to solve, the primary difficulty being that the demand for service is not stationary in time, and so r_t depends on λ_t and also on λ_{t-1}, r_{t-1}, λ_{t-2}, r_{t-2}, ..., etc. A solution can be obtained using the time-dependent queueing model, which is discussed in Section III, in an iterative trial and error fashion: Guess at the values of r_t; run the model; correct the values of r_t on the basis of the run, etc. In most cases, however, using the easier to calculate long-run (or stationary) state probability distribution, computed with parameters appropriate only to hour t, will provide excellent estimates of the required number of cars, r_t. There are two reasons why this is so. First, although queues building up in hour t may carry over into hour t + 1, if α is small the probability of a queue will also be small. Therefore, there will be little carry-over, and hours t and t + 1 will be approximately independent. Second, although the call rate, λ_t, changes over time, the changes in successive hours are, in general, not large.

Let $a_j(t)$ denote the stationary probability that j jobs are in the system during hour t, given that there are n_t cars on duty, that the call rate is λ_t, and that the service rate is μ_t. Define $\rho_t = \lambda_t/\mu_t$. Then, using results for the M/M/n queueing model, $a_j(t)$ is given by:

$$a_j(t) = \begin{cases} \dfrac{\rho_t^{\,j}}{j!}\, a_0(t), & 1 \le j \le n_t \\[2ex] \dfrac{\rho_t^{\,j}\, a_0(t)}{n_t^{\,(j-n_t)}\, n_t!}, & j > n_t \end{cases}$$

and

$$a_0(t) = \left[\sum_{j=0}^{n_t-1} \frac{\rho_t^{\,j}}{j!} + \frac{\rho_t^{\,n_t}}{n_t!} \left(\frac{n_t\,\mu_t}{n_t\,\mu_t - \lambda_t} \right) \right]^{-1} .$$

Given a value of α, r_t is the smallest value of n_t such that

$$\sum_{j=0}^{n_t-1} a_j(t) > 1 - \alpha .$$

II. AN INTEGER LINEAR PROGRAM FOR GENERATING SCHEDULES

Suppose that the number of patrol cars required in the field during each hour of the day has been specified. (This may have been done using the formulae in Section I.) Suppose also that there is a set of feasible tour start times, S, given by the police department, all of which we assume start on the hour. Each tour of duty lasts eight hours. We also assume that every tour includes a one-hour mealtime, which also begins on the hour. There may be constraints on the earliest and latest hours of a tour that can be used for a meal break; let $e(t)$ and $\ell(t)$, respectively, denote these values for a tour starting at hour t. Then the earliest possible mealtime for cars working tour t starts at hour $t + e(t)$, and the latest possible mealtime begins at $t + \ell(t)$. (These assumptions are made for clarity of exposition only and are not intrinsic to the model. The model can easily handle tours starting on the half hour, quarter hour, etc., as well as tours and mealtimes of any length.)

Consider a 24-hour problem, and let r_t denote the (integral) number of cars required during hour t ($t = 0,1,\ldots,23$; hour 24 = hour 0), where hour t runs from t to t + 1 clock hours. The decision variables of the programming problem are:

n_i = the number of cars assigned to work the tour starting at hour i, $i \in S$.

m_{ij} = the number of cars working tour i that are assigned to mealtime at hour j, where $i+e(i) \leq j \leq i+\ell(i)$ and $i \in S$.

We wish to find tour assignments $\{n_i\}$ and mealtime assignments $\{m_{ij}\}$ that meet the car requirements using the least number of cars. An integer linear program that accomplishes this objective is:

$$\text{minimize} \sum_{i \in S} n_i$$

subject to

$$\sum_{j=i+e(i)}^{i+\ell(i)} m_{ij} = n_i, \qquad i \in S$$

$$\sum_{\substack{i \in S \\ t-7 \le i \le t}} n_i - m_{it} \ge r_t, \qquad t = 0, 23 \tag{A}$$

$$n_i \ge 0,\ m_{ij} \ge 0, \text{ and integer.}$$

The value of the objective function is the total number of cars used over the day. The first set of constraints assures that every car is assigned a mealtime, and the second set of constraints assures that the actual cars on duty (number of cars assigned minus number of cars on meals) meets each of the hourly requirements.

The resulting integer linear programs can be quite large for reasonable real-world situations. With only three tour start times and four possible mealtimes, there are 15 variables and 27 constraints. This is modest but not trivial for an integer program. With 24 tours, and mealtimes allowed at any hour of a tour, there are 216 variables and 48 constraints. We have also formulated and solved problems for an entire week involving as many as 1344 variables and 336 constraints.

Fortunately, the constraint matrix has a special structure that permits the problem to be solved as a mixed integer program in which only a small number of the variables need be "forced" to be integers. The remaining variables are automatically integral in any optimal solution. This permits use of a standard mixed integer programming code to solve even the largest of our problems in reasonable times. The key idea is contained in the following:

Theorem: Suppose that n_i, the number of cars assigned to tour i, is limited to integer values. Then m_{ij}, the mealtime assignment variables, are automatically integral in any basic solution of (A).

Proof: When the n_i take on fixed values, say n_i^*, the constraints of (A) become

$$\left.\begin{array}{ll} \displaystyle\sum_{j=i+e(i)}^{i+\ell(i)} m_{ij}=n_i^*, & i\varepsilon S \\ \displaystyle\sum_{i\varepsilon S} m_{ij} \leq -r_t + \sum_{\substack{t-7\leq i\leq t \\ i\varepsilon S}} n_i^*, & t = 0,\dots,23 \end{array}\right\} \quad (B)$$

The only coefficients of (B) are +1 and 0, and each of the columns of the associated matrix contains at most two nonzero elements. The matrix, therefore, has a network-like structure that is well known to be totally unimodular. Hence, all of the extreme points of (B) are integer regardless of the values of r_t and n_i^* ([4], p. 70; [8], p. 126).

To solve the smaller problems, the mixed integer program (MIP) subroutines of MPSX [14] can be used in a straightforward fashion. To solve the larger problems we have modified the branching rules in MIP to take advantage of the problem structure. See Appendix A for a description of these modifications. An overview of the MIP-MPSX algorithm is given by Geoffrion and Marsten [6].

III. EVALUATING SCHEDULES

The actual environment in which a schedule may be implemented is complex and random. Nevertheless, a schedule generated using the approach described above is derived from a deterministic integer linear programming model, the inputs to which are determined partly by simple approximate models and partly by police judgments. Before trying any such schedule in the field, we would like to test it using mathematical models that represent more of the complexity of the real world. Two types of models can be used for this purpose. The first is a simulation model of police patrol operations, such as the one described in [10]. A simulation, however, is a relatively expensive and cumbersome tool that requires the collection of a considerable amount of input data to make it work.

The second type of model that can be used for evaluation is a time-dependent M/M/n queueing model. It is not difficult to develop the set of differential equations that describes the system dynamics of the M/M/n queueing system with time-dependent parameters. It is, however, extremely difficult to obtain an analytic solution to the set of equations. We therefore propose numerical integration of the differential equations to obtain such characteristics of the patrol system as the probability distribution of the number of busy patrol cars and the number of calls queued. Our approach is motivated by a recent paper by Bernard O. Koopman [11] in which the efficiency and usefulness of this type of model is illustrated by applying it to the study of air traffic control problems. Koopman also discusses the advantages of this approach relative to the use of simulation.

Our time-dependent queueing model represents a single police precinct. As the different tours and mealtimes commence, the number of police cars available for patroling and for servicing calls will vary. The rate at which calls for service are received also varies during the day, with a peak occurring in the late evening hours and a lull in the morning (Fig. 1). From historical data we can predict the average number of calls for service and the distribution of the number of calls during each hour of the day. Our data also permit identification of daily and seasonal patterns in the call rate. We have determined from these call histories that the arrival of calls for service in any given hour can be well represented as a Poisson process. That is, the probability that a given number of calls will occur during a given hour in a given day is specified by a Poisson distribution with a mean characteristic of that time period.

The model assumes that each call for service is handled by a single patrol car. In practice it sometimes happens that two or more cars are necessary to service a call. Nevertheless, a comparison of results from a stationary M/M/n queueing model (which also incorporates the one car per call assumption) to results from the simulation model (which more closely imitates reality and uses as many cars as is appropriate for the call) indicates that such complications may be neglected without seriously altering the resulting queueing probabilities [9].

While the types of calls (crimes in progress, past crimes, emergencies, accidents, etc.) may vary during the day, we have found that the average service time remains fairly constant. Actual data show that the service times for calls do not have an exponential distribution as assumed by the model, but again a comparison of the M/M/n queueing model to the simulation model, which used empirical--and hence nonexponential--service times, shows good agreement for the prediction of average performance.

One limitation of the time-dependent queueing model is that there is no priority structure in the dispatching of calls. As a result, we can only examine overall call delays. Modeling of priority calls is possible, but it would significantly increase the complexity of the computations. It was not undertaken since detailed analysis of delays by call priority can be performed with either the simulation or a stationary M/M/n queueing model. Calls in this time-dependent model are served, therefore, on a first-come-first-served basis. If a patrol car is free when a call arrives, it is dispatched immediately and the call remains in the system only for the length of its service time. (The response time of the patrol car is not explicitly modeled. It would be very difficult to do this in an accurate way without destroying the

Markovian nature of the system, which is essential to our method of solution. Since travel times are short compared to the time spent at the scene of the incident, this approximation is not critical.) During the service interval the patrol car is unavailable for further assignments. If a call arrives when all patrol cars are busy, it waits in queue until all preceding calls have been dispatched. It is then dispatched as soon as the next car becomes available.

The model we have just described is an M/M/n queueing system with time-dependent parameters. In studying this system, we focus attention on the random variable $X(t)$, the number of calls in the system at time t, including those being served by patrol cars and those in the queue. A great deal of information about the performance of the system can be obtained from the Markovian transition probability function $p_{ij}(t_o,t)$, defined as follows:

$$p_{ij}(t_o,t) = P[X(t)=j\,|\,X(t_o)=i], \quad t_o\geq 0, \quad t>t_o, \qquad i,j = 0, 1, \ldots$$

For example, let $n(t)$ denote the number of patrol cars on duty at time t. Then if $X(t)$ is less than $n(t)$, $X(t)$ represents the number of busy cars and $n(t) - X(t)$ represents the number of cars on patrol. If $X(t)$ is greater than or equal to $n(t)$, all cars are busy and $X(t) - n(t)$ represents the number of calls waiting in the dispatching queue. Suppose that at some time we know that there are i calls in the system--that is, $X(t_o) = i$. Given this information about the state of the system at time t_o, we can calculate the following system performance characteristics for any future time t:

The probability that there is at least one call in queue:

$$P[X(t)\geq n(t)+1] = \sum_{j=n(t)+1}^{\infty} p_{ij}(t_o,t) \tag{1}$$

The probability that all cars are busy:

$$P[X(t)\geq n(t)] = \sum_{j=n(t)}^{\infty} p_{ij}(t_o,t) \tag{2}$$

The expected number of calls in queue:

$$EQ = \sum_{j=n(t)+1}^{\infty} [j-n(t)]p_{ij}(t_o,t) \tag{3}$$

The expected number of cars available for patrol:

$$EA = \sum_{j=0}^{n(t)-1} [n(t) - j]p_{ij}(t_o,t) \qquad (4)$$

Before considering how to determine the transition probabilities, we introduce some additional notation:

$\lambda(t)$= the call rate at time t; i.e., the expected number of calls per hour being received at time t, which is the mean of the Poisson process generating the calls. Here we refer to a specific epoch t and its instantaneous call rate $\lambda(t)$. Earlier we used λ_t to refer to the average number of calls during the hour t to t + 1.

μ = the service rate; $1/\mu$ =ES, the expected service time per call. As verified by actual data, we assume in this analysis that μ does not change through time. (Relaxation of this assumption would not appreciably complicate the analysis.)

The transition probabilities satisfy the following system of differential-difference equations. For $t>t_o$,

$$p_{io}'(t_o,t)=-\lambda(t)p_{io}(t_o,t)+\mu p_{i1}(t_o,t),$$

$$p_{ij}'(t_o,t)=\lambda(t)p_{ij-1}(t_o,t)-[\lambda(t)+\mu]p_{ij}(t_o,t)+\mu(j+1)p_{ij+1}(t_o,t), \quad 1\le j\le n(t), \qquad (5)$$

$$p_{ij}'(t_o,t)=\lambda(t)p_{ij-1}(t_o,t)-[\lambda(t)+n(t)\mu]p_{ij}(t_o,t)+\mu p_{ij+1}(t_o,t), \quad j>n(t).$$

These equations cannot be solved analytically for $p_{ij}(t_o,t)$, except for the simplest of functions $\lambda(t)$ and $n(t)$. However, we can solve (integrate) them numerically. Since $\lambda(t)$ and $n(t)$ are periodic functions (repeating themselves every 24 or 168 hours depending on the application) there is a periodic solution that is independent of the initial state i. We wish to find this periodic solution. We denote it by $p_j(t)$, the "long run" probability that $x(t)=j$. ($p_{ij}(t_o,t)$ approaches $p_j(t)$ for large t. [7])

There are an infinite number of equations in (5). In order to solve them numerically we limit ourselves to a finite system of equations that approximates (5) by assuming--not unrealistically--that there is a maximum possible number of calls, m, that can be in the system at one time. In some applications the value of m is dictated by the limitations of the dispatching system. Where such physical constraints do not exist, m is chosen so that the probability of having m or more calls in the system is very small. Hence, we replace (5) by

$$
\begin{aligned}
&p_o'(t)=-\lambda(t)p_o(t)+\mu p_1(t) \\
&p_j'(t)=\lambda(t)p_{j-1}(t)-[\lambda(t)+j\mu]p_j(t)+(j+1)\mu p_{j+1}(t), \qquad 1\leq j\leq n(t) \qquad (6) \\
&p_j'(t)=\lambda(t)p_{j-1}(t)-[\lambda(t)+n(t)\mu]p_j(t)+n(t)\mu p_{j+1}(t), \qquad n(t)\leq j\leq m \\
&p_m'(t)=\lambda(t)p_{m-1}(t)-n(t)\mu p_m(t).
\end{aligned}
$$

A discussion of the numerical methods used to solve this set of m+1 differential equations is given in Appendix B.

IV. A SAMPLE PROBLEM

We illustrate the scheduling methodology described above with a sample problem based on data from a police precinct in New York City. The precinct used in this example cannot be called typical--there is no such thing. It does, however, have characteristics in the middle range of precincts on several measures: physical area, total demand for police service, crime rate, number of cars fielded, etc.

Our data for the hourly call rates are derived from job records collected during one week in August 1972 by the computerized dispatching system used by the New York City Police Department. Based on empirical data, we took the average service time to be 30 minutes (μ = 2 calls per hour).

Table 1 contains λ_t, the call rate, $\rho_t = \lambda_t/2$, the average number of busy cars, and r_t, the number of cars required so that the system is unclogged at least 90 percent of the time. That is, r_t is the smallest number of cars needed during hour t so that the probability of at least one car being available to respond to a call is at least 0.9. The values of r_t were estimated using the stationary M/M/n queueing model, as discussed in Section II.

Figure 2 shows the schedule that was actually in use in the precinct during the period in August 1972 from which our data come. It uses 24 cars over the three eight-hour tours of a day. Because of low car availability during the early morning hours, the schedule produces periods in which there is a very high probability that no cars will be available to answer a call for service. The line on the graph that shows the probability that there are no cars on patrol was obtained from the time-dependent queueing model.

In order to obtain schedules with better performance characteristics, we set up and solved several integer linear programs. A description and evaluation of some of the resulting schedules follows.

<u>Integer Linear Program 1 (Standard Tours with Standard Mealtimes)</u>

In this case, we restricted ourselves to the tour start times and mealtimes generally used in the New York City Police Department. There are three permitted tour start times--0800,1600, and 2400 (or 0000) hours--and mealtimes can be taken between the second through the fifth hours of the

Table 1

HOURLY CALL RATES, EXPECTED NUMBER OF BUSY CARS, AND NUMBER OF CARS REQUIRED FOR SAMPLE PROBLEMS

t	λ_t	ρ_t $(\lambda_t/2)$	r_t
0	9.8	4.9	9
1	9.6	4.8	9
2	8.7	4.4	8
3	7.4	3.7	8
4	6.7	3.4	7
5	5.3	2.7	6
6	4.1	2.1	5
7	3.3	1.7	4
8	2.5	1.3	4
9	2.5	1.3	4
10	2.9	1.5	4
11	3.8	1.9	5
12	4.3	2.2	5
13	5.0	2.5	6
14	5.9	3.0	6
15	6.6	3.3	7
16	7.8	3.9	8
17	8.6	4.3	8
18	9.4	4.7	9
19	9.8	4.9	9
20	10.2	5.1	9
21	10.4	5.2	9
22	10.2	5.1	9
23	10.0	5.0	9

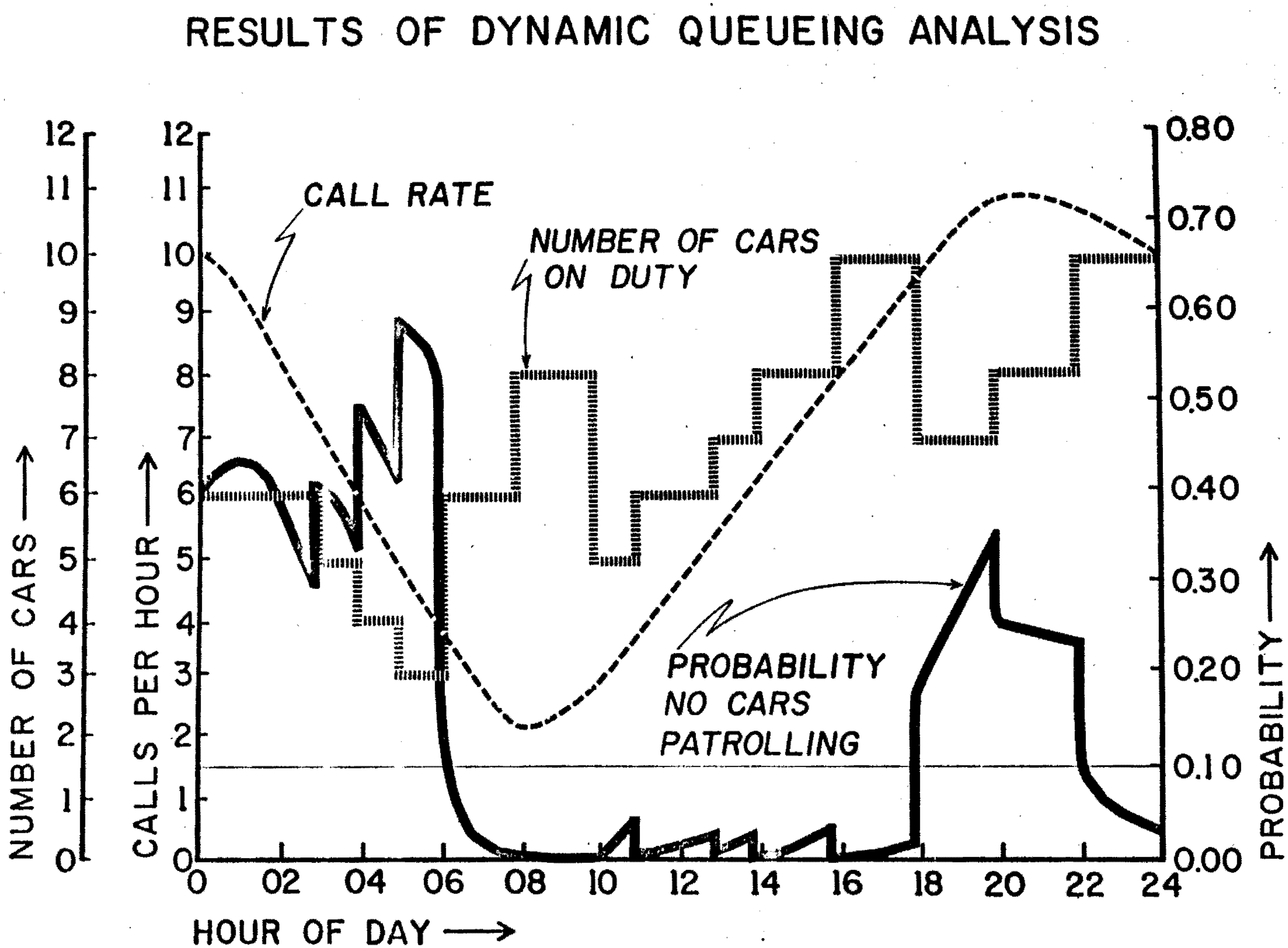

Fig. 2. An actual NYPD schedule using 24 cars

tour. There are 15 variables and 27 constraints in the linear program. Using the car requirements, r_t, shown in Table 1, we obtain the following optimal integer solution, which uses 29 cars:

(i) Tour starting at 2400 hours

cars assigned = 10

mealtimes assigned:	1	cars	at	0200
	2	"	"	0300
	3	"	"	0400
	4	"	"	0500

(ii) Tour starting at 0800 hours

cars assigned = 7

mealtimes assigned:	2	cars	at	1000
	2	"	"	1100
	2	"	"	1200
	1	"	"	1300

(iii) Tour starting at 1600 hours

cars assigned = 12

mealtimes assigned:	3	cars	at	1800
	3	"	"	1900
	3	"	"	2000
	3	"	"	2100

The characteristics of this schedule are illustrated in Fig. 3. Here again, the probability that there are no cars on patrol was computed using the time-dependent queueing model.

Integer Linear Program 2 (All Possible Tour Start Times and Mealtimes)

This program allows a tour to start at the beginning of any hour of the day, and a unit's mealtime can be taken during any hour of the tour. The solution to this problem provides the smallest possible number of cars that could be used to meet the specified requirements. Any additional restrictions, such as prohibited hours for starting mealtimes or tours, will produce a requirement for at least as many cars as the solution to this program.

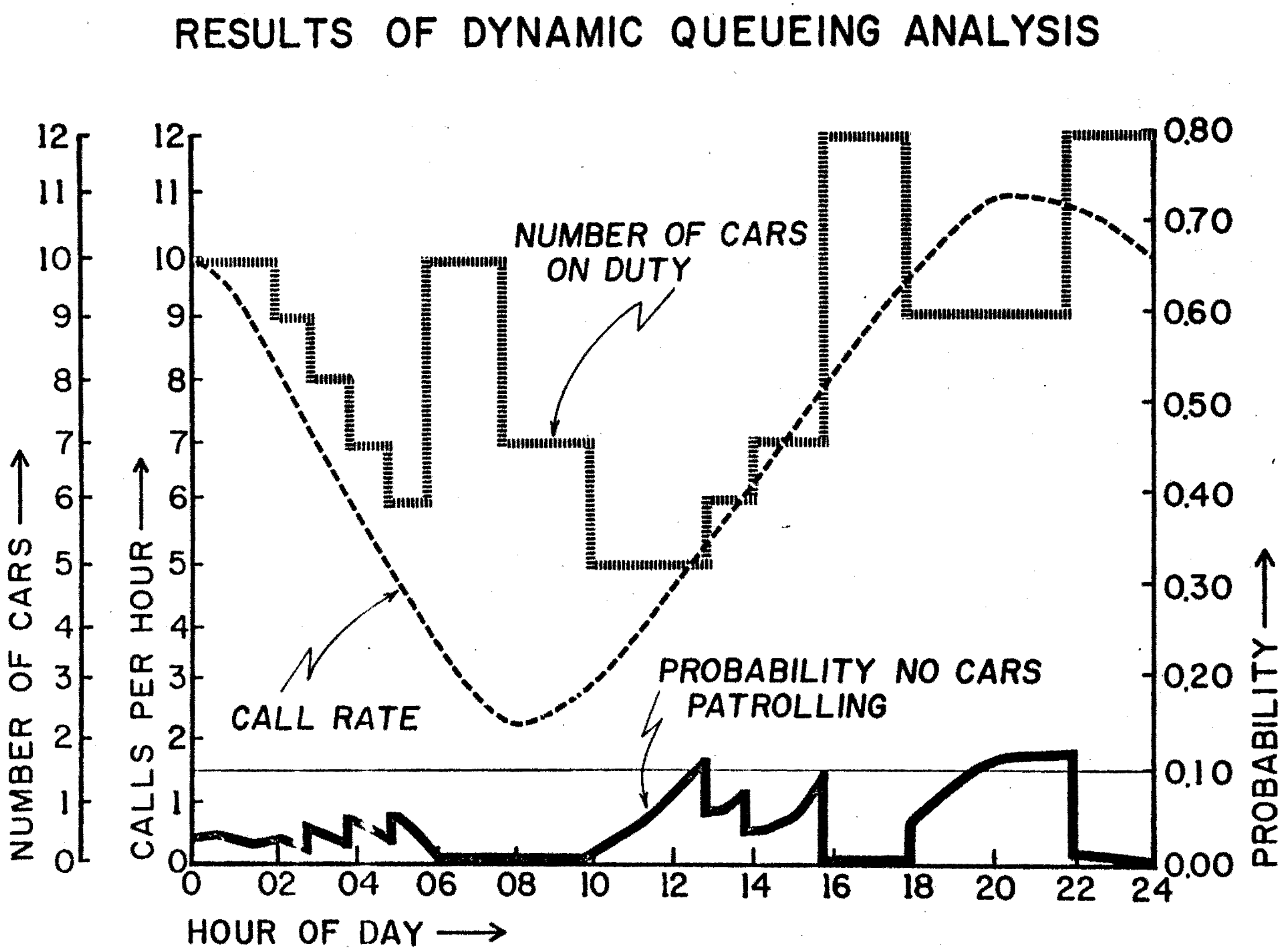

Fig. 3. A computer generated 29 car schedule with 3 tours and $\alpha = 0.1$

An optimal integer solution was obtained that requires 24 cars over the day, but uses 13 different tour start times--an administrative nightmare to implement. This schedule is not illustrated here.

This result led us to ask if it was possible to find a less difficult schedule, but one that would still require only 24 cars. The type of schedule we wanted would have only a small number of tour start times, avoid undesirable start times (e.g., 3:00 a.m.), and have mealtimes scheduled during reasonable hours of the day. By running programs with different numbers of tours and different allowable mealtimes, relying on the results of the previous analysis for insights on what might work well, we obtained the following schedule:

<u>Integer Linear Program 3 (Five Tours, All Possible Mealtimes)</u>

This program has five tour start times--the three current times, 0800, 1600, and 2400; and two additional times, 1200 and 2000. Mealtimes are allowed at any hour during a tour. The optimal integral solution calls for a total of 24 cars (the minimum possible) and the solution is:

<u>(i) Tour starting at 0800 hours</u>

cars assigned = 5

mealtimes assigned: 1 cars at 0800
1 " " 0900
1 " " 1000
1 " " 1200
1 " " 1400

<u>(ii) Tour starting at 1200 hours</u>

cars assigned = 2

mealtimes assigned: 1 cars at 1200
1 " " 1300

<u>(iii) Tour starting at 1600 hours</u>

cars assigned = 8

mealtimes assigned: 2 cars at 1200
2 " " 1700
1 " " 1800
1 " " 1900
1 " " 2200
1 " " 2300

(iv) Tour starting at 2000 hours

cars assigned = 2

mealtimes assigned:	1	cars	at	2000
	1	"	"	2100

(v) Tour starting at 2400 hours

cars assigned = 7

mealtimes assigned:	1	cars	at	0300
	1	"	"	0500
	2	"	"	0600
	3	"	"	0700

This schedule is illustrated in Fig. 4. Comparing these results to the current situation shown in Fig. 2, we see that although both schedules field the same number of patrol cars over a day, the resulting performance characteristics are considerably different. For example, under the current schedule, almost 60 percent of the incoming calls during some hours have to wait in queue before a car is dispatched. Under the schedule resulting from Linear Program 3, the percentage of calls delayed in queue never exceeds 12 percent.

The above are only a few examples from an extensive series of integer linear programs that were solved to test and develop new schedules for the New York City Police Department. The resulting schedules that appeared capable of being implemented were subjected to analysis using the time-dependent queueing model. In addition to the one-day schedules illustrated above, we solved scheduling problems for entire weeks with data from different seasons of the year and from different precincts. We used the techniques presented in this paper to find answers to such questions as:

- What improvements in performance would result if patrol cars could be assigned to start their tours of duty at any hour of the day and take their mealtimes at any hour during the tour?
- What is the best schedule that uses only four tour start times? How much worse is the performance of such a schedule than the performance of the best five-tour schedule? How much better is this schedule than the standard three-tour schedule?
- Since the pattern of calls on weekends is different from weekdays, should the weekend schedules be different?

- Are the patterns of calls for service in different areas similar enough so that principles developed from studying a few precincts can be applied generally?
- How does the number of patrol cars assigned vary with the desired service levels?

RESULTS OF DYNAMIC QUEUEING ANALYSIS

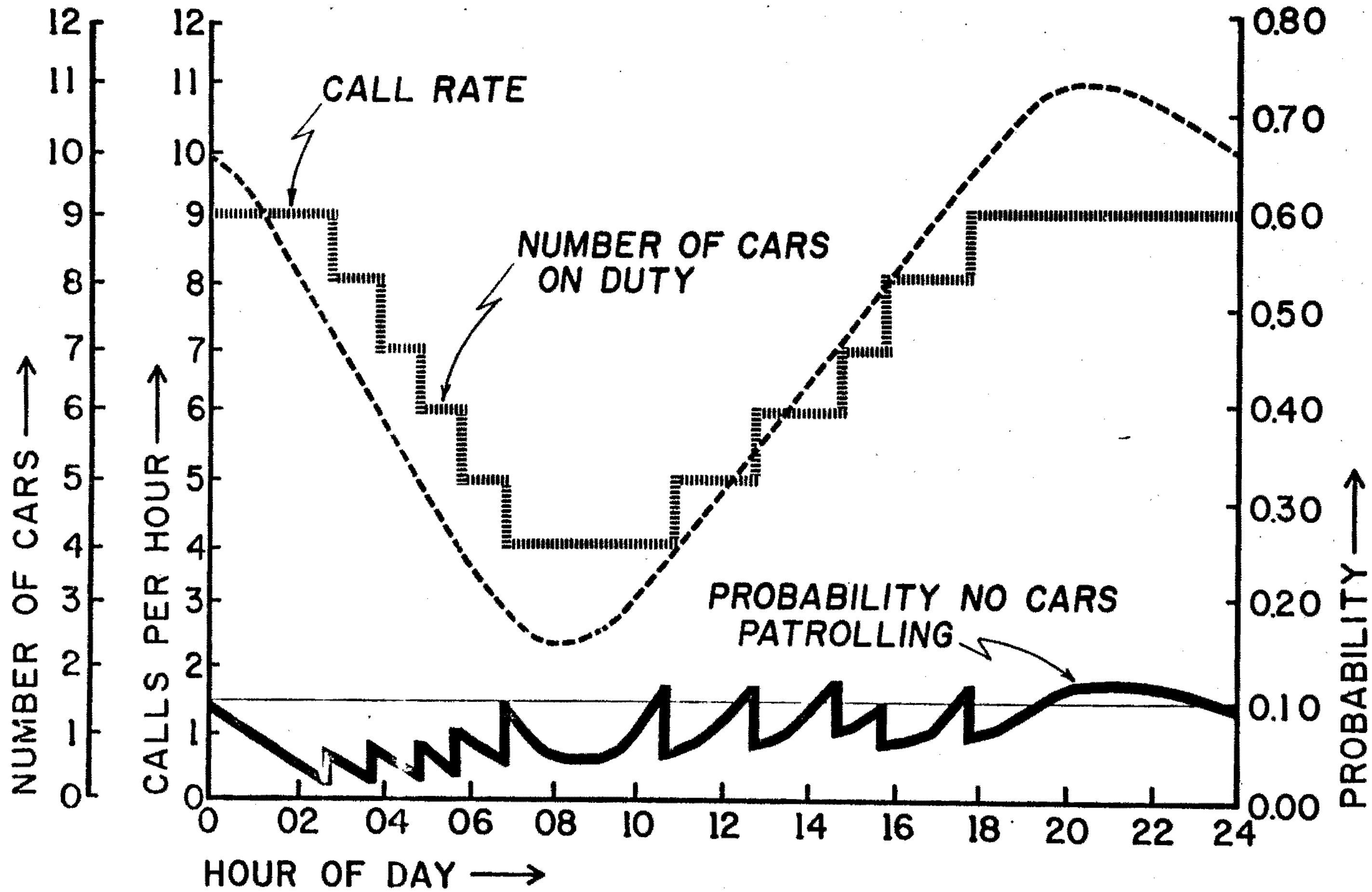

Fig. 4. A computer generated 24 car schedule with 5 tours and $\alpha = 0.1$

Appendix A

An IBM program product, the Mathematical Programming System Extended (MPSX) with Mixed Integer Programming (MIP) was used to obtain integer solutions to the Scheduling problem [14]. MIP uses the branch and bound method of solving mixed integer programming problems, and allows the user to choose a standard solution strategy or to implement a strategy appropriate to the structure of the problem.

We modified the standard solution strategy because of several special features of the Scheduling problem. In our problems, all feasible integer solutions have integer-valued objective functions, and the optimal integer solution is, in general, very close in value to the optimal solution to the problem when integer constraints are relaxed. Also, there may be many alternative optimal solutions, but for our purposes, we often do not need to enumerate these alternatives. Finally, the standard MIP strategy makes heavy use of the "pseudocosts" of altering variable values and of the variable weights in the objective function. Since all of the integer variables in our objective function have the coefficient unity, some modification of the branching rules seemed desirable.

When the search for an optimal integer solution commences, there is already available an optimal solution to the problem obtained by relaxing the integer constraints. We take advantage of the fact that the integer solution is often close to this "continuous solution" by placing all nodes whose objective function value is at least 4 cars more than the continuous solution into an inactive state. If there is no integer solution this close to the continuous solution, a new set of nodes with values no more than 8 cars higher than the continuous solution is made active. This process continues until an integer solution is found. In practice, an integer solution has always been found among the first set of nodes.

When a feasible integer solution is found, all nodes with a value lower by less than 1 are placed in an inactive state and all nodes with values worse than the solution are dropped. Because we know that any integer solution must have an integer functional value, we know that the nodes just made inactive cannot lead to an integer solution better than that just obtained. The nodes are not dropped, however, since the last solution might be optimal and we may want alternative optima for some problems.

As the search progresses, the best functional value among all the nodes, which is usually noninteger, will become larger. If we obtain an integer solution that has a functional value equal to the best functional value rounded up to an integer, we know that the solution is optimal. If only one optimal solution is desired, the search can stop at this point. If alternative optima are wanted, the search continues.

During the search process the variable expected to give the greatest expected functional deterioration is chosen as the branching variable, and all variables with current values $[x]+.2<x<[x+1]-.2$ are given priority for branching consideration. The node with the best functional value is choosen in order to obtain a "bushy" tree. Since we know that the optimal integer solution will usually be close to the optimal continuous solution and since we also know that the optimal solution can be no smaller than the best functional value rounded to the next highest integer, this procedure ensures a quicker "proof of optimality" even though it may take longer to find the first integer solution.

While the standard solution strategy would have been sufficient to obtain the desired solutions, the revised strategy allows a quicker proof of optimality and a more predictable behavior of the search process for our particular problem.

Appendix B

An IBM program product, the Continuous System Modeling Program (CSMP) [3], was used in the numerical solution of the time-dependent queueing equations. CSMP permits description of the queueing model with FORTRAN-like statements, and the user can select among various numerical integration techniques.

We selected the fourth-order Runge-Kutta method with a variable step size. Two difficulties peculiar to the family of equations we were solving led to this choice. First, Runge-Kutta was chosen because there is a discontinuity in the equations at each tour change or scheduled meal. There are many numerical integration methods that evaluate the equations to be integrated several steps ahead and use the results to estimate higher-order derivatives, which are in turn used to accelerate the integration.* Unfortunately, the estimated derivatives are incorrect when a discontinuity exists in the equations. The Runge-Kutta techniques, in contrast, concentrate on a single step and do not look ahead. Therefore, if a single step does not pass over a shift change, there will be no discontinuity in the calculations.

Second, a variable step size was chosen because after a shift change there may be a quick transient response in the solution. The intervals over which strong transient responses occur constitute only a small fraction of the entire interval of integration. A fixed step procedure would have required an uneconomically small step size over the entire time period of the calculations to insure the numerical stability of the solution immediately after the shift changes. Therefore a variable step procedure was chosen with the constraint that a step could not straddle a shift change.

During the numerical evaluation of the equations, the state probabilities were constrained to be greater than or equal to zero. A check was made of numerical accuracy by summing the probabilities. At no point in the solution did the sum deviate from unity by more than 0.0014.

---

*See [12] for a discussion of alternative numerical integration techniques.

As initial values for the integration, we used the steady state solution for the M/M/n system with the midnight parameter values. The integrations were run for a two-day interval and the solutions for the first day were compared to the solutions for the second day. After a sufficiently long time the effects of the initial distribution should disappear and one might expect that two solutions of the equations separated by 24 hours would be close. In all cases, the probabilities converged to the periodic solution well before the onset of the second day. Of course, we did not learn this until the second day had been solved. The values of $p_j(t)$ for the second day were then used as the periodic solutions to the equations.