A PATROL CAR ALLOCATION MODEL

Jan M. Chaiken and Peter Dormont

June 1976

P-5594

A PATROL CAR ALLOCATION MODEL

by Jan M. Chaiken and Peter Dormont*

The Rand Corporation
1700 Main Street
Santa Monica, California 90406

June 1976

Abstract

A computer program has been designed for specifying the number of police patrol cars that should be on duty in each geographical command of a city at various times of day on each day of the week. The program is a synthesis of the best features of previous patrol car allocation models, with several improvements, including the capability to prescribe allocations when one tour in each day in each geographical command overlays two other tours. The program was designed to be inexpensive and readily transferable.

*Current address for Peter Dormont: Mathematica, Inc., P.O. Box 2392, Princeton, New Jersey 08540

I. INTRODUCTION

During the last decade, a considerable amount of effort has been devoted to methods for allocating police patrol cars. Out of this work have evolved several computer programs for specifying the number of patrol cars that should be on duty in each geographical command of a city at various times of day on each day of the week. These programs were intended to substitute for the use of "hazard" or "workload" formulas, which are still widely popular although their failings have been pointed out repeatedly [2,7,16,21].

Most of the programs were based on, or were similar to, either the resource allocation system of the St. Louis Police Department [27] or a program designed by Richard Larson [21]. The most widely known program based on the St. Louis system was the Law Enforcement Manpower Resource Allocation System (LEMRAS), a proprietary IBM package [13] that was withdrawn at the end of 1974. Larson's program spawned the following offspring:

- The Police Resource Allocation Program (RAP), a proprietary program of Urban Sciences, Inc. [31]
- A New York City Police Department RMP (Radio Motorized Patrol) allocation program written by Richard Mudge at The New York City-Rand Institute [28]
- A program designed for the Los Angeles Police Department by a UCLA class "Public Systems Analysis" [1]
- A program written for the Rotterdam Police Department [26].

While all of the programs were similar in many ways, each one had several minor features that were considered either especially desirable or particularly inadequate by some analysts or police departments. These features related to the program's mode of operation (batch or interactive), input requirements, assumptions underlying its calculations, or capabilities to take certain performance measures into account. As a result, police departments considering a patrol car allocation program had several competing alternatives, none of which was entirely satisfactory.

After carefully reviewing all the patrol car allocation programs that had been used by police departments up to late 1974, we designed a general-purpose program that we call PCAM (Patrol Car Allocation Model). This model incorporates, by user option, nearly all the features present in earlier programs, together with several improvements. In addition, we followed certain principles that, based on our review, appeared likely to enhance transferability of a new model. First, it could not be proprietary or restricted in any way. Second, it had to be written in a language that could be compiled on nearly any computer system and was likely to be familiar to programmers in municipal government. We chose FORTRAN. Third, it had to operate in either batch or interactive mode, at user option. Fourth, it could not require large amounts of core storage or lengthy processing times. Fifth, it had to adapt flexibly to varying terminology (such as precinct, division, district, area, bureau, sector, station, and foreign-language words that mean the same thing). Sixth, it had to have a complete, detailed user's manual permitting applications by police departments without outside assistance.

In the first four months after its release, PCAM had been adopted by most users of the earlier programs and by several additional departments. By January 1976, it was in use at Larson's nonprofit firm Public Systems Evaluation (for applications in Wilmington, Delaware), at The Institute for Public Program Analysis (for training purposes and possible application in several cities), and at police departments in Seattle, Atlanta, Toledo, Minneapolis, the Netherlands, and Edmonton, Alberta. In addition, data bases were being prepared in anticipation of its use in Los Angeles, New York, and Jacksonville, and the program was made available to time-sharing customers of Urban Sciences, Inc., and Compu-Serv Network, Inc.

In this paper we present a brief history of patrol car allocation programs to show how our model was synthesized from prior models and then describe its features, capabilities, and algorithms. Further details are available in the documentation of the Patrol Car Allocation Model, which includes an executive summary for police administrators and planning officers [4], a user's manual [5], and a programmer's manual providing installation instructions, file specifications, and an annotated program listing [6].

II. HISTORY OF PREVIOUS PATROL CAR ALLOCATION PROGRAMS

Characteristics of the System

All the programs under consideration in this paper allocate patrol cars to independent geographical commands. For clarity of exposition, we shall call these commands "precincts," although terminology varies widely among police departments. A precinct is not the area covered by a single patrol car, but rather is a larger area, ordinarily containing a station house to which the patrolmen report before and after their tours of duty. The important characteristics defining a precinct are (a) that its commander has the capability or authority to decide how many patrol cars will be fielded at various times, and (b) that the dispatchers of patrol cars treat the precinct as an independent command by sending only precinct cars to incidents in the precinct, except under unusual circumstances. Some police departments consist of a single precinct.

Each precinct is modeled as a queuing system in which the servers are the patrol cars and the customers are calls for service (cfs) to the police arising from the precinct. Typical assumptions are that the calls for service can be distinguished by priority level, that each call is served by a single patrol car from the precinct, that calls are placed in queue when all the precinct's cars are unavailable, and that queued calls are subsequently served according to a first-in-first-out discipline within priority levels.

Ordinarily, some or all of these assumptions fail to be precisely correct in practice. Every police department receives at least a few calls that require more than one patrol car to be dispatched. In addition, if a high-priority call arrives when all the precinct cars are busy, it will not actually be placed in queue. Instead, an additional car will be fielded specifically to answer the call, a sergeant's car will be dispatched, a patrol car from a neighboring precinct will be dispatched, a special-purpose unit such as a traffic car or plainclothes unit will be sent to the scene, or some other way will be found to respond to the call.

If these variations from the assumptions in the programs occur infrequently, then they may be ignored without affecting the accuracy of the output substantially. However, if the variations are large, then either the input to the program must be adjusted to account for departmental practices or the output must be interpreted differently. For example, if the department dispatches two cars to every incident and both of them remain at each incident for the same length of time, the output can simply be interpreted as indicating how many pairs of patrol cars should be fielded.

Further assumptions must be made to produce a manageable analytic model of the queuing properties of this system. Generally, the arrival of calls for service in each precinct is described as a time-dependent Poisson process, and the service times in the precinct are assumed to have a negative exponential distribution whose mean may also vary with time but is independent of other characteristics of the system. The assumption of Poisson arrivals is confirmed by data [20], but actual service times are neither exponentially distributed nor identical for all calls. In particular, the assumption that the service time is independent of the system state is not correct, because the service time includes the length of time required for the patrol car to travel to the scene of the incident, which is a function of the number of available servers (patrol cars). Thus, conflicts arise between validity and analytic simplicity of the models; these are resolved in various ways that will be described.

A consequence of these assumptions is that the programs require as input the average call rate and service time for each precinct, as a function of the time of day. The computer programs for patrol car allocation can be distinguished according to whether they do or do not assist the user in estimating these input parameters. Those that make no predictions have sometimes been operated simply by using averages of past data, and sometimes a separate program has been used to make the required predictions. The details of prediction capabilities will be described below in the discussion of individual programs.

Another important system characteristic is that, from the point of view of queuing, the number of servers is not constant over time. Rather, patrol cars may be unavailable for reasons other than calls for service (meals, auto repairs, on-view incidents requiring police intervention, special assignments by a commanding officer, and the like). We shall call these events "non-cfs" unavailabilities and describe how they are handled in each of the programs. One approach that has been taken is to ignore non-cfs unavailabilities altogether. However, in this case, the resulting output of the program may bear no relationship to reality [1], in which case it is virtually useless as an aid to planning.

A second approach is to consider non-cfs unavailabilities as if they were calls for service. If the estimates of arrival rates and service times for non-cfs events are accurate, this method tends to work well. However, it is not appropriate to make such estimates for the future by projecting data from the past, because the number of non-cfs events will change if the number of cars on duty is changed. Particularly in departments where patrol cars are unavailable for dispatch during meal times, it is apparent that increasing the number of cars on duty will increase the number of non-cfs events, quite independent of how many there were in the past. The importance of this effect varies from department to department.

A third approach to handling non-cfs events in a patrol allocation program is to assume that cars busy on non-cfs work are not "effectively" present. This means that the number of servers, from the point of view of queuing, is estimated as some fraction of the number of patrol cars fielded in the precinct. The advantage of this method is that the calculations can be performed so as to take into account automatically the change in the amount of non-cfs work that will occur as the number of fielded cars is changed. Details of the method will be given as we discuss the various patrol car allocation programs.

St. Louis Police Department

The computer programs for the St. Louis Police Department were initially proposed and documented by Richard F. Crowther [11] in 1964. (See also Shumate and Crowther [30].) During the four years that followed, these methods were refined, programmed for the department's computer, and applied in one precinct (called "district" in St. Louis) by a project team at the police department [27]. While the total resource allocation project covered many topics, we shall describe only those that were related to determining the number of patrol cars needed in each precinct. These programs performed certain basic functions needed for any patrol car allocation system. They were operated by the department in batch mode on a regular basis for at least five years. The programs had two components, one to predict call rates and service times and the other to calculate queuing statistics.

Prediction. The city was divided into small areas (called Pauly Areas) about the size of several blocks. Dispatchers' records were coded according to the Pauly Area in which the incident occurred, and a program was written to determine the number of incidents in each of eight different categories that occurred in each Area in each hour of the week. Exponential smoothing was used to project these counts into the future, and the service times of incidents were similarly smoothed [25]. Since a precinct can in principle be any collection of Pauly Areas, the hourly call rate in a precinct was estimated by aggregating the call rates for the corresponding Areas, and the service time was estimated as a weighted average.

Queuing. The system was modeled as being in steady state in each hour, with Poisson input and exponential service times whose means were given by the estimates from the prediction program. A program was written to generate tables from Erlang's formula [12] showing the percentage of calls in each tour that would experience a delay for different numbers of servers. (A "tour" is a period of time, commonly eight hours, during which a patrol car may be on duty. The fraction of calls delayed during a tour was estimated as the weighted average of the hourly figures.) Department policy was established that at least enough cars should be fielded to keep the number of calls placed in queue under

15 percent of the total number of calls. By consulting the tables it was possible to determine the number of cars needed to accomplish this objective.

For purposes of comparison with programs to be described below, we shall point out certain details of the St. Louis program. First, the occasional dispatch of more than one patrol car to an incident was handled by counting each dispatch in the data as if it represented an incident. Thus, an incident requiring three patrol cars would count as three incidents. This method appeared satisfactory, and it can be used with any of the patrol car allocation programs.

Second, no attempt was made to take account of non-cfs work in the St. Louis patrol allocation programs. The extent to which this led to actual delays being higher than those predicted by the computer program has not been reported, to our knowledge. However, the department apparently had adequate resources to keep the actual number of calls encountering a queue well under 10 percent.

Third, although calls were divided into categories that could potentially be distinguished by importance or priority, the particular performance measure used (namely, the percentage of calls delayed) does not vary according to the priority of a call. Therefore, there was no operational reason to distinguish among types of calls in the program output.

Finally, the exponential smoothing technique was found to be adequately accurate, through a comparison of the actual number of incidents and service times with the predictions. Apparently the St. Louis Police Department experienced little difficulty in selecting suitable smoothing parameters.

LEMRAS (Law Enforcement Manpower Resource Allocation System)

This IBM software package was based on the St. Louis system and included all of its features, together with a number of improvements [13]. Once again, cities were divided into small areas (which were called "reporting areas" instead of Pauly Areas), and the number of incidents and their service times were predicted by exponential smoothing. Incidents could be divided into a large number of event codes, corresponding to the names given to incidents by dispatchers, and these were aggregated into, at most, 20 "event classes" for purposes of statistical analysis. Each event class could be assigned to one of three priority levels.

In an advancement over the St. Louis system, the LEMRAS program operated on the assumption that calls of a given priority class are not assigned to patrol cars until all higher-priority calls have been assigned. For each specified number of patrol cars on duty, the LEMRAS program estimated the distribution of queuing delays, presented as histograms with five-minute intervals for each priority class. By taking into account the number of calls in each event class expected to occur in each hour, this information was then summarized for each event class on a weekly basis, or whatever was desired by the user. Thus a department using the LEMRAS system could, if it wished, allocate cars to fulfill criteria relating to the proportion of calls delayed and the distribution of delay within priority classes. Some LEMRAS users chose not to take advantage of its capabilities related to priority levels; they simply classified all calls as priority 1. In such applications, the departments had essentially the same patrol allocation system as St. Louis had.

Aside from the priority queuing feature, most of the other improvements in the LEMRAS system were not conceptual in nature but were for the purposes of assisting the user in preparing data for input, providing flexible output formats, etc. Like its St. Louis predecessor, LEMRAS was a batch program. LEMRAS was withdrawn by IBM at the end of 1974 because the program was not compatible with the latest generation of operating systems being marketed by the corporation, and most customers were interested in an on-line interactive program, while LEMRAS operated in batch mode.

Some LEMRAS users developed their own programs to format and print only such LEMRAS output information as was of interest to them. For example, if a department wanted to allocate enough cars to assure that under 10 percent of calls were queued, it might not have any use for tables showing the delays that would occur under allocations that did not meet the objective.

Some LEMRAS users entered all patrol car work, whether for calls for service or not, into the data input and were satisfied with both the predictions and the recommendations for the number of cars to be fielded. Other departments, such as the Los Angeles Police Department

(LAPD) [1], found the predictions for non-cfs work to be frequently very much in error, and therefore did not use them. Even the predictions for call-for-service arrival rates and service times, while usually acceptably accurate, sometimes were incorrect in Los Angeles. This led to some concern that the technique of exponential smoothing was itself inappropriate for the Los Angeles data, but a more likely explanation is that the exponential smoothing parameters had not been set properly, and the city lacked the statistical expertise required to correct the situation. In regard to non-cfs work, as was pointed out earlier, it is conceptually erroneous to try to make predictions from past data. Departments that found their non-cfs predictions satisfactory presumably did not vary the number of cars on duty in a given precinct and tour to any great extent, or for some other reason they were lucky to have a slowly varying pattern of non-cfs work. The LAPD happened not to fall into this group.

In Los Angeles, the amount of time devoted to non-cfs work varies from 40 to 60 percent of the total time cars are in the field. This is too large an amount of work to ignore in the program. As a result, when the LEMRAS program was operated using only cfs data, it would specify how many cars should be fielded to assure that under 5 percent of calls would be queued, but the department found that fielding the recommended number of cars led to about 40 percent of calls being queued. The problem was that the LAPD was fielding the number of cars specified by LEMRAS without realizing the distinction between "effective" and "actual" cars. This is simply an illustration of the fact that if a program is used in a way that was not intended, it may fail in dramatic fashion.

Larson's Program

In 1968 and 1969, Richard Larson designed a program for patrol car allocation and applied it, as a test case, to data from New York City [20]. Later, he described the program, together with potential improvements that could be made, in his book Urban Police Patrol Analysis [21]. Larson's program does not perform any estimations of call rates or service times, but requires such information as input. In regard to its queuing formulation, Larson's program is similar to LEMRAS, except that more than three priority levels are permitted, and the program calculates the average length of time a call of each priority level will wait in queue, rather than a histogram of the delay distribution.

The two major advances over LEMRAS incorporated in Larson's program were (1) consideration of performance measures other than queuing delays, and (2) capability to allocate a fixed total number of patrol cars among precincts.

Additional Performance Measures. Larson recognized that queuing delays were not the only measure of performance of a patrol car system, and indeed might be unimportant compared to others. For example, if a precinct were large enough that the average time it took a patrol car to travel to an incident was 15 minutes, it would be of little interest that the average wait in queue was 20 seconds.

Larson discussed in general a variety of performance measures that could be considered, but included only three in his program:

a. Average travel time to incidents,
b. Average patrol frequency (how often a car passes the most heavily patrolled points in the precinct while on preventive patrol), and
c. Patrol hours per outside crime.

These were estimated from approximate analytical models based on principles of geometrical probability.

In one method of using the program, called the descriptive mode, the user could try various numbers of patrol cars in each precinct, and the program would calculate these three performance measures, together with the percentage of calls that would have to wait in queue. If the user had in mind a desired maximum or minimum for some of the measures, he could inspect the tables and see how many cars were needed to accomplish the objectives. Thus, the descriptive mode represented in itself an improvement over the output capabilities of the St. Louis program. In practice, because of additional capabilities of Larson's program, the descriptive mode was mainly used to find out the values of the performance measures for the number of cars currently fielded in each precinct.

A technically modest, but important, improvement introduced by Larson was the capability to permit the user to enter, as input, his desired maximum or minimum for each of the three measures in each precinct. In addition, he could establish administratively a minimum permissible number of patrol cars for some or all precincts. The program would then calculate how many patrol cars were needed in each precinct to meet all the specified constraints, without the user having to inspect a large number of descriptive tables.

Allocation of a Fixed Number of Cars. Larson was the first to recognize the fact that the total number of patrol cars available for fielding in the city was an important consideration in allocating cars to precincts. Therefore, in the prescriptive mode of Larson's program, the user specified the total number of cars to be allocated in the whole city (or some collection of precincts) during the tour in question. The program then allocated cars to precincts in such a way that, first, all the constraints discussed above were met, and, second, the additional cars (if any) were allocated so as to minimize the city-wide average time a call would wait in queue. (Actually, the user could specify weights for each priority level, and the program would minimize the weighted average waiting time.) The optimization was accomplished by a dynamic programming algorithm.

Larson's program did not utilize hourly data varying over a tour, as did the two programs described above, but assumed a steady-state situation with fixed call rate and service time over the period for which allocations were being made. This is a disadvantage, because in many cases call rates vary by 50 percent or more over a tour. If the user operated Larson's program separately for each hour of the tour, he might not be able to vary the number of cars as suggested by the output. On the other hand, if he entered the average call rate for the tour, the resulting output would be less accurate. Larson's program also had no special capabilities for handling non-cfs work, other than by including such work in the call rate and the service time.

This program was written in the Michigan Algorithm Decoder (MAD) language and ran in an interactive mode on the Massachusetts Institute of Technology computer system. It could be accessed from New York by telephone lines, but the NYCPD never used this particular version for any planning purposes. The MAD language was unpopular and was eventually abandoned by MIT, at which time the program "died."

Urban Sciences, Incorporated

Urban Sciences, Inc., rewrote Larson's program in FORTRAN and greatly enhanced its interactive capabilities [31]. This program was made accessible to police departments by contract, but the source code was proprietary. In all conceptual aspects it was identical to the program just described above.

New York City Police Department (Mudge's Program)

This program was written in 1972 by Richard Mudge at The New York City-Rand Institute [28]. While based on Larson's program, Mudge's program was not exactly the same. The two primary differences were:

- Mudge's program would not allocate a specified total number of patrol cars. In prescriptive mode, this program simply calculated the number of patrol cars needed in each precinct to meet constraints entered by the user.
- Mudge's program distinguished between "effective" cars and "actual" cars, as follows. The user specified a fraction (the same for all precincts) representing the fraction of time that cars are busy on non-cfs work. This fraction was used to compute the effective number of cars, which was then rounded to an integer.

Minor differences were as follows: Mudge included more information in descriptive output than was available from Larson's program, and the measures of performance subject to constraint by the user were expanded to include several measures related to queuing. In a sense, this program returned to the philosophy underlying the St. Louis and LEMRAS programs, namely, that a department would want to field enough cars to keep queuing delays under specified limits.

This program also permitted only three priority levels, but the average queuing delay for priority 1 calls was not displayed. Mudge realized that priority 1 calls would be handled in a special way if all the precinct cars were busy, and thus the program's estimates for the delay of such calls would be inaccurate.

Mudge's program is similar to Larson's in that it does not assist the user in predicting call rates or service times and it uses average data for a tour, rather than hourly data. It was written in FORTRAN and was available in two versions, batch and interactive. The NYCPD used this program from time to time over a two-year period for long-term planning purposes.

UCLA Program

As mentioned above, the LAPD had for several years used the LEMRAS program, as modified by its own input and output routines, and was having some difficulty with it. In 1974, a class at the University of California, Los Angeles, prepared a patrol car allocation program for consideration by the LAPD [1]. It was based on the Mudge and Larson programs. In common with the Mudge program, it permitted the user to specify constraints on queuing delays as well as other performance measures. In common with the Larson program, it permitted the user to allocate specified total resources. The primary differences between this program and the other two are as follows:

- The UCLA program allocated car-hours across tours instead of cars across precincts. This means that the user specified the total number of car-hours available in a precinct during a day, and the program prescribed how many cars should be on duty during each tour. Or, alternatively, the user specified constraints on performance measures and the program prescribed how many cars are needed in each tour, adding these to show total car-hours in a day for the precinct in question. This facility permits the number of hours in a tour to differ among tours.

- The UCLA program operated on the assumption that the amount of non-cfs work performed by a car would vary according to the amount of cfs work. (This was found to be true in Los Angeles, by analysis of available data.) The relationship between the fraction of time busy on non-cfs work and the fraction on calls for service was modeled as a linear equation, separately for each precinct, using data from the precinct [1]. The conversion between "effective" cars and "actual" cars was then calculated from the linear equation. While the linearity of this relationship is simply an empirical observation, one can understand that there must be some relationship by realizing that patrol cars cannot engage in non-cfs work unless they are otherwise available. The more free time a patrol officer has, the larger will be the number of non-cfs events that come to his attention.

This program was written in PL/I and operated in batch mode. It did not make predictions of call rates or service times, which were available from LEMRAS in any event. However, it accepted as input hourly data rather than averages for a tour. It did not have descriptive capabilities, although the output displayed the performance measures for the recommended allocations.

Interim Version of PCAM

During the process of programming PCAM, an interim version of the program was provided to the New York City Police Department and the Seattle Police Department [29]. This program was an improvement over Mudge's program in that it would allocate a specified number of cars as well as determine the number of cars needed to meet constraints. It also included many of the technical improvements incorporated in the final program, including a linear relationship between non-cfs work and call-for-service work.

However, it was limited to allocations across precincts (i.e., it would not allocate car-hours across tours), and it used average call rates and service times for tours rather than hourly data. The interim version was available only as an interactive program. This model was used in Seattle for over a year, where it was validated against actual data for travel times and the fraction of calls entering queue.

Dynamic Queuing Model

All the programs described above assumed the system to be in steady state, either for each hour or for an entire tour. Kolesar, Rider, Crabill, and Walker [18] developed a dynamic queuing model that eliminates this assumption. It calculates time-varying queuing statistics by numerical integration of the differential equations for a system having time-dependent Poisson arrivals, exponential service times, and a time-varying number of servers. This program is especially useful for analysis of tour starting times and scheduling of meal hours, but it is too elaborate to form part of an inexpensive patrol car allocation program. Since the dynamic queuing model does not calculate performance statistics such as travel time and preventive patrol frequency, it is not in itself a suitable substitute for any of the programs that follow the principles elucidated by Larson.

Fortunately, by comparing the output of the dynamic queuing model with calculations performed by assuming steady state in each hour, it has been found that both methods produce approximately the same average statistics for an entire tour [17]. This is because the coupling between tours is quite weak. As a result, allocations derived by incorporating the dynamic queuing model in a patrol car allocation program would be identical, for all practical purposes, to those derived by hourly steady-state calculations. Therefore, we took the latter approach for our own model.

III. CAPABILITIES AND APPLICATIONS

The above history indicates a variety of technical reasons why no patrol car allocation model has as yet achieved general acceptance.

Some models were implemented to suit the requirements of one department, with no consideration given to generality. (In Mudge's program, for example, one has to modify the source code in order to change the number of precincts, and in the UCLA program the values for constraints on performance measures are in the source code.) Prescriptive and descriptive capabilities present in some models were lacking in others. Programs were written in computer languages or dialects for which translators are not widely available, or the source code was kept proprietary.

While there are many obstacles to implementation of computer models in police departments that have nothing to do with the characteristics of the models themselves [8], in this case we felt that a general-purpose model would enhance the chances of implementation. Our work consisted of determining which features of the previously existing models were the most useful, identifying desirable capabilities that did not exist in previous models, and combining all of these in a package that could be used on most computer systems and would be easy to install and run. Our Patrol Car Allocation Model incorporates, by user option, nearly all the features of the programs described in the previous section, except that it will not predict call rates or service times.

We provided for wide usability by writing the program in a standard version of FORTRAN without recourse to language features peculiar to one computer system or compiler. Ease of installation was accomplished by a system of dynamic allocation of array storage which allows the program to adjust array dimensions for a particular city and type of analysis. The program was made readily usable by providing for user control through a sequence of easily learned natural-language commands that can be entered at an interactive terminal or on punched cards. In addition, the program provides for the inclusion of terminology used by a particular department in commands and output table headings. The amount of core storage required by the program varies according to the size of the user's data base but is generally under 160K bytes on an IBM System 370 machine. The cost for typical runs of the program is well under $10.

In the rest of this section we describe the capabilities and functions included in the PCAM program, and examples of applications of the model.

Descriptive Capabilities

Descriptive capabilities of the Patrol Car Allocation Model permit displaying quantitative information about any allocation of patrol cars by time of day and geographical command. This information may refer to the current allocation, any allocation proposed by the user, or allocations that are suggested by the program when operated in prescriptive mode. This information permits the user to compare allocations and determine which one he thinks is best.

When the model is operated in descriptive mode, it can provide the following information for each tour in each precinct:

- The number of patrol cars assigned
- The average fraction of time patrol cars are busy on calls for service (actual utilization)
- The average number of cars available (not busy on either cfs work or non-cfs work)
- Preventive patrol frequency
- Average length of time from the dispatch of a patrol car until its arrival at the scene of an incident (travel time)
- The probability that a call will enter queue
- The average time in queue of calls, by priority level
- The average total response time (time in queue plus travel time).

The model provides for great flexibility in the selection and summarizing of information that is displayed. Information can be selected by precinct, time of day, day of week, or any combination thereof. Thus, the user can examine performance measures for all precincts for one tour on a particular day or look at one precinct for all tours of a day or several tours of each day of the week, etc.

The output information is calculated from simple analytical models that are described in the User's Manual [5]. For example, preventive patrol frequency in a precinct is calculated from the formula originally developed by Larson [21]. The average travel time is calculated from a relationship known as the square-root law [19], which is a function whose variables are the area of the precinct, the effective number of patrol cars on duty, and the effective travel speed of the cars. This relationship has been validated against both real and simulated travel-time data [14,19]. The model's calculations of queuing statistics when there is no non-cfs work are based on an M/M/N formulation. These statistics have been validated against data from a simulation model [19] which itself has been validated against real data in New York City [10]. When non-cfs work is present, an adjustment is made to queuing statistics as described in Section IV, below. A limited validation of these adjusted statistics against real data has been conducted in Seattle, but further experience in other cities is required before this part of the program can be considered fully validated.

Prescriptive Capabilities

The Patrol Car Allocation Model allocates car-hours to shifts, where a shift is defined as a combination of a specific tour on a specific day in a specific precinct. The purpose of allocating car-hours rather than cars is to permit tours to have any duration desired by the user, not necessarily all the same. If all tours have the same length, the user can allocate cars, rather than car-hours, to shifts by adding one line to the source program.

The two basic prescriptive capabilities of the model are (a) determining the minimum number of cars that must be on duty in each shift to meet constraints on performance measures specified by the user, and (b) allocating a user-specified total number of car-hours among shifts so as to minimize an objective function. A variant of the second capability permits the user to add a specified number of car-hours to a previously determined allocation. Thus, minimization subject to constraints, which was accomplished in a single step in Larson's program, requires two steps when operating the PCAM program. This separation into two steps was designed to permit flexibility. For example, it is easy for the user to specify that each shift is to be allocated at least as many cars as are currently present; this capability permits allocation of added manpower, such as a newly graduated class of recruits.

The performance measures subject to constraint by the user are all the descriptive output items listed above, except for actual utilization. For example, the PCAM program will specify the minimum number of cars needed in each shift so as to keep the fraction of calls that are queued under .2 and the average travel time under 8 minutes.

The objective functions that can be minimized by the model are:

- Probability of calls entering queue
- Average queue delay for all calls or calls of a specified priority level
- Average total response time.

When minimizing an objective function, the user specifies the subset of shifts to which the car-hours are to be allocated. For example, he can fix the tour and day, in which case car-hours will be allocated across precincts; or he can fix the precinct, in which case car-hours will be allocated across all tours in all days of the week for that precinct; or he can allocate car-hours across precincts, tours, and days simultaneously.

Overlay Tours

PCAM's greatest technical innovation is its ability to deal with overlay tours. That is, it will describe performance measures and prescribe allocations if there is a tour that begins during one "normal" tour and ends during the following tour. For example, if all tours are eight hours in length and begin at midnight, 0800, 1600, and 1900, then the tour from 1900 to 0300 is an overlay tour. (See Fig. 1.) Overlay tours are most commonly used by police departments to synchronize peak manpower with maximum workload when the lengths of tours worked are inflexible.

While a department with an overlay tour can use the earlier programs by artificially imagining shorter "tours" (for example, the time intervals labeled Block 1, ..., Block 5 on Fig. 1), allocations prescribed for these time intervals would not necessarily be compatible. In other words, it is not possible to achieve arbitrary allocations to

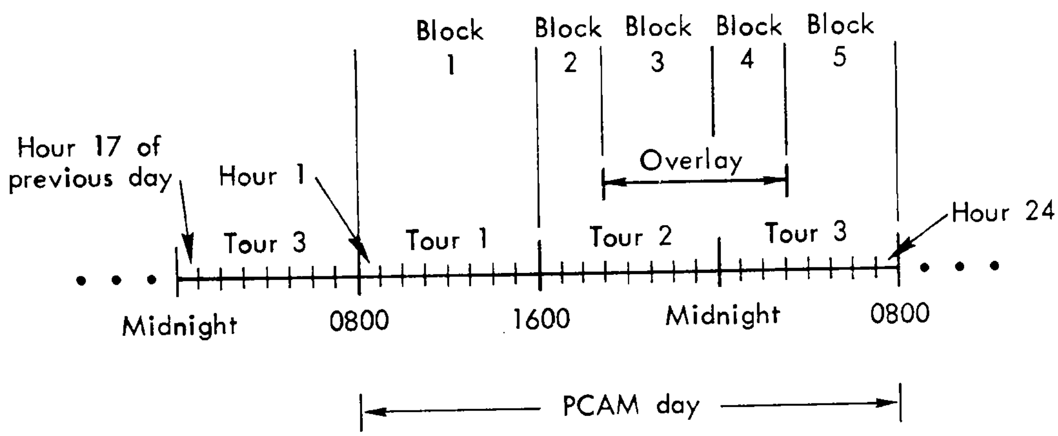

Fig. 1—Time blocks with an overlay tour. A block is a time interval during which the number of patrol cars does not change. Twenty-four-hour "days" are defined in such a way that the overlaid tours (Tour 2 and Tour 3) are both in the same day.

five time intervals by starting patrol cars on duty at four different times. PCAM will recommend only feasible allocations in the case of a single overlay tour, although police departments with more than one overlay must resort to the same "trick" when using PCAM.

In descriptive mode, PCAM computes performance measures taking into account changes in the number of cars on duty caused by the starting and ending of overlay tours. There is no particular difficulty inherent in these computations. In prescriptive mode, however, problems arise. These result from the fact that simple marginal allocation algorithms do not work due to the inseparability of the objective functions for tours involved in overlays. This problem is fully explained in the next section, and an algorithm for solving the allocation problem is described. The algorithm is not optimal under all assumptions, but appears "sensible" in typical applications. More complex algorithms, which could also have solved the optimization problem for multiple overlay tours, were judged too expensive (in terms of computer processing time) for incorporation in the model.

Applications

The primary judgmental problem for police departments using PCAM is selecting suitable constraints and objective functions. Most departments have some relatively large precincts with few calls for service, as well as small, densely populated precincts with many calls for service. Citywide minimization of any queuing statistic tends to concentrate the patrol cars in the precincts with many calls for service, resulting in possibly unacceptable queuing delays and travel times in the precincts that have a large area but few calls for service.

PCAM's facility for setting constraints on performance measures permits the user to introduce aspects of equity into the allocation. In fact, by iteratively restricting the constraint on any one performance statistic, one can achieve an allocation that approximately equalizes that statistic over time and geography. However, the simultaneous equalization of all performance statistics is in general impossible, so the user is forced to consider the trade-offs among performance measures.

Some departments have found that an acceptable allocation is achieved by minimizing total response time (queuing plus travel time) without any constraints. This is because the total response time in a precinct is calculated from its geographical area as well as its call-for-service workload. Moreover, total response time is believed to be correlated with ultimate measures of the quality of police patrol operations, such as the probability that a criminal offender will be arrested at the scene of a crime [9,15].

Once a police department has established the objective functions and/or constraints it wants to use, a variety of applications of PCAM are possible. It can be used during budget preparation to determine the total number of patrol officers a department needs to meet specified performance levels. Once the department's total number of patrol officers has been determined, the program can allocate them among precincts. Either at the same time or later, it can allocate the patrol officers in a precinct to the various tours on different days of the week. It can be used to analyze proposed changes in tour starting times or the possibility of introducing an overlay tour in a department that currently does not have one. It can also indicate the effects of changing the priority structure for calls for service or "screening out" certain calls (refusing to dispatch a patrol car to specific types of low-priority calls).

Since PCAM's calculations are insensitive to the locations of cars within a precinct, the program cannot be used to design patrol areas of police cars. Discussions of suitable models for this purpose and for other analyses of patrol car operations that cannot be accomplished with PCAM are given elsewhere [2,3,8,21,22,23,24].

IV. COMPUTATIONAL ALGORITHMS

In this section we describe briefly the calculations performed by the Patrol Car Allocation Model, with emphasis on the situation when an overlay tour is present, since mathematically this is the only unique feature of the model. Although cars are allocated to shifts by the program, when an overlay tour is present the number of cars on duty can change during the period of time covered by a shift. For example, in Fig. 1 we might have 5 cars allocated to Tour 2 in Precinct 1 on Monday and 3 cars allocated to the Overlay Tour; in this case the number of cars on duty increases from 5 to 8 at 1900 hours, which is in the midst of Tour 2. To discuss these possibilities we use the term time block to refer to a period of time during which the number of patrol cars on duty does not change. Thus, Tour 2 in Fig. 1 consists of two time blocks, Block 2 and Block 3.

Assumptions

For a single hour in a single precinct, calls for service are assumed to arrive according to three independent Poisson processes (representing three priority levels) with sum rate λ and to have identical, independent, exponentially distributed service times with mean $1/\mu$. In a standard steady-state queuing formulation with a fixed number of servers, the mean arrival rates and service time permit calculating any desired queuing statistics for each priority class. (See the PCAM User's Manual [5] for details.)

To model the stochastic variation in the number of servers due to non-cfs work, the fraction of time each of N patrol cars will be unavailable on non-cfs work is assumed to be a function $U(\lambda, \mu, N)$. For queuing purposes, the number of servers is then $n = (1 - U)N$. We refer to n as the "number of effective cars" and N as the "number of actual cars." If n is not an integer, queuing statistics for n servers are estimated by linear interpolation of steady-state statistics for $[n]$ servers and $[n] + 1$ servers, where $[n]$ denotes the integer part of n. These calculations cannot be performed unless $\lambda/[n]\mu < 1$.

In accordance with the findings of the UCLA class, the function U is assumed to have the form

$$U(\lambda, \mu, N) = B_1 \lambda/N\mu + B_2.$$

The constants B_1 and B_2 are determined separately for each precinct from data showing the actual fraction of calls delayed in queue for various achieved values of λ, μ, and N. This is accomplished by numerical inversion of Erlang's formula for the probability of delay, which determines the number of effective cars.* In this way the constants adjust for non-cfs unavailability (whether non-cfs events are recorded by the police department or not) and also for the inaccuracy introduced by assuming identically distributed exponential service times. In other words, the constants B_1 and B_2 automatically assure that the calculation of queuing statistics in the model will match the true performance of patrol cars, at least in regard to the probability that a call enters queue.

Meeting Constraints

When the user specifies constraints on performance measures, the program assures that the constraints are met in every time block specified. This is accomplished by a simple iterative procedure in which the number of cars is increased by 1 in each step. The initial assignment is either the current allocation or the minimum number of cars needed to keep $\lambda/[n]\mu$ under 1 in each hour, depending on instructions from the user.

Once the required allocations to blocks are determined, they are converted to an allocation to shifts with the following properties:

1. At least as many cars are assigned to each block as are required.
2. The shift allocation consumes the smallest possible number of car-hours consistent with (1).

Allocating a Specified Number of Car-Hours

To allocate car-hours across shifts, the user of the model specifies the total number of car-hours to be allocated, the shifts over which the allocation is to take place, and the objective function F to

---

*A computer program to perform this inversion, which is not part of the Patrol Car Allocation Model, is listed in the Program Description [6].

be minimized by the allocation, which, as mentioned earlier, may be chosen as one of the following:

F_1 = average fraction of calls queued
F_2 = average waiting time in queue
$F_{2,p}$ = average waiting time for priority p calls, p = 1, 2, 3
F_3 = average total response time.

The user also specifies whether car-hours are to be allocated in addition to those already allocated, or whether the allocation is to begin as if no cars were currently allocated.

The program then follows a heuristic algorithm that is intended to minimize F by allocating an integer number of cars to each shift in such a way as to consume all the car-hours specified.* However, the algorithm has been proved optimal only in the cases (a) when there are no overlay tours or (b) when the overlay tour has the same duration as the two tours it overlays. To describe the algorithm, we denote by $B_1, B_2, \ldots, B_K$ the time blocks over which the allocation is to take place. An allocation to shifts induces a specification of the number of cars assigned to each block: $n_1, n_2, \ldots, n_K$. Denote by $F_i(n_i)$ the average value of the objective function F over block B_i when n_i cars are assigned to B_i. Then the objective function F has the following properties:

Property 1. The value of F is a weighted average of the f_i's:

$$F(n_1, n_2, \ldots, n_K) = \sum_{i=1}^{K} w_i f_i(n_i) \Big/ \sum_{j=1}^{K} w_j .$$

---

*In some cases a small number of car-hours may remain unallocated if they are not enough to equal one car working for one shift. Ordinarily a police department would have no use for noninteger allocations to shifts, which is why PCAM allocates integers. However, if the initial allocation has noninteger allocations (e.g., it may be an average of actual allocations over several weeks), the user can, if he wishes, ask PCAM to add integers to the existing allocation, resulting in a noninteger allocation.

Here w_i is the total number of calls in block B_i when $F = F_1$, F_2, or F_3, and w_i is the total number of priority p calls in block B_i when $F = F_{2,p}$.

Property 2. Each f_i is convex decreasing. More precisely, if $n < n'$ then $f_i(n') < f_i(n)$, and if $n < n' \leq n'' < n'''$, then $f_i(n'') - f_i(n''') \leq f(n) - f(n')$.

With No Overlay Tours. When there are no overlay tours, every shift is the same as a time block, so the shifts are $B_1, \ldots, B_K$. The model's allocation algorithm begins with an initial allocation n_1, n_2, ..., n_K that is the same as is calculated when meeting constraints and depends on whether the user wants to start with the current allocation or not.

Then the model calculates, for each shift B_i, a number Δ_i representing the amount by which the weighted objective function will improve per car-hour if one car is added to shift B_i:

$$\Delta_i = w_i(f_i(n_i) - f_i(n_i + 1))/h_i,$$

where h_i is the number of hours in shift B_i. The algorithm adds one car to the (or a) shift having the largest value of Δ_i and then repeats the process until all the car-hours are consumed.

It is well known that this iterative process (marginal allocation) leads to an optimal solution, because the objective function is separable and convex. However, a proof for this particular case is also given in the User's Manual [5].

With Overlay Tours. To describe the difficulties with overlay tours, we shall indicate the problems that would arise if we attempted to use the procedure that we have just described for the case of no overlays. First, it is possible to determine an initial allocation of cars to time blocks exactly as before, but then this initial allocation might not be feasible. This means there might be no way to achieve the indicated assignments to blocks by starting cars to work at the beginning of tours. Among the feasible allocations that have at least as many cars in each block as are needed for the initial allocation, some have fewer car-hours than others. Among those that have the smallest possible number of car-hours, some may have a lower value of the objective function than others. In short, some care had to be exercised in selecting the initial allocation.

Second, if we add cars iteratively to blocks so as to minimize the objective function, we again encounter the problem that the resulting allocation may not be feasible. If we convert this optimal allocation into a feasible one, we find (a) the feasible allocation may have more car-hours than we intended to allocate, and (b) there is no guarantee that the feasible allocation is optimal for the number of car-hours it does have.

If, on the other hand, we attempt to add cars iteratively to <u>shifts</u> instead of time blocks, it turns out that the marginal allocation procedure described above <u>does not work</u>. To be more specific, it is <u>not</u> true, in the case of overlays, that the optimal allocation of N + 1 cars can necessarily be found by starting with the optimal allocation of N cars and adding one car to one shift. The reason the method fails in this case is that the objective function is no longer separable with respect to the decision variables, which are the numbers of cars assigned to each shift. For example, suppose two shifts are on duty during block B_i. Then block B_i contributes $w_i f_i(N_1 + N_2)/\sum w_j$ to the objective function, where N_1 is the number of cars assigned to one of the shifts in the block, and N_2 is the number of cars in the second. This cannot be expressed as the sum of a function of N_1 and a function of N_2.

Third, if the overlay tour does not have the same length as the tours it overlays, the standard definition of the word "optimal" will lead to unsatisfactory allocations. Figure 2 illustrates this problem by showing the fraction of calls delayed for various allocations in an example precinct having 4 tours, one of which is an overlay. The lengths of the tours in the overlay segment* are as follows: Tour 1 is 6 hours long, Tour 2 is 10 hours long, and the overlay tour is 12 hours long. From the figure, it can be seen that the minimal allocation to the

---

*An <u>overlay segment</u> is a collection of three shifts, one of which is an overlay and the other two of which are overlaid.

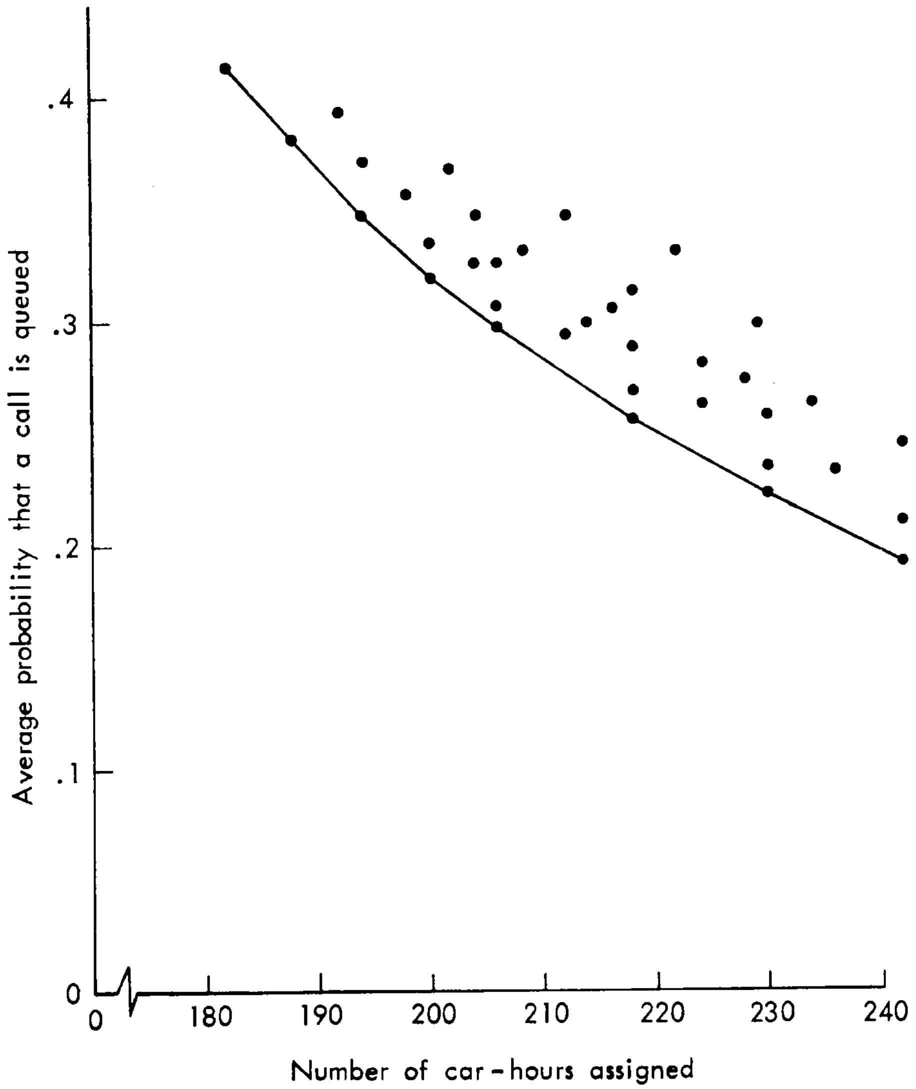

Fig. 2—Average queuing probability for an overlay segment in which the tours do not have the same length. The call rates and service times in each hour were determined from actual data in a test city. The smallest number of car-hours needed to keep effective utilization under 1 in each hour is 182. The other points on the graph correspond to allocations having between 182 and 240 car-hours. Some allocations having a large queuing probability were not graphed.

overlay segment requires 182 car-hours, the next feasible allocation requires 188 car-hours, and the following one requires 192. Since there is only one feasible allocation with 192 car-hours, it might be said that it is the "optimal" allocation of 192 car-hours. However, no police department would be interested in this allocation, because a <u>smaller</u> number of car-hours (namely, 188) can be allocated to give a <u>lower</u> value of the objective function F_1. Also note that there is an allocation of 212 car-hours that has a lower value of the objective function than any allocation of a smaller number of car-hours, and yet it does not look "desirable."

Basically, "desirable" allocations are those that lie on the piecewise linear curve shown on Fig. 2. This curve can be defined as the graph of the maximal convex function ϕ such that $\phi(x)$ is less than or equal to the value of the objective function for every feasible allocation of x car-hours. Then the problem of finding the optimal allocation of H car-hours can be stated as follows: Choose an allocation of H' car-hours for which (1) the value of the objective function is $\phi(H')$, and (2) H' is as large as possible, subject to the constraint $H' \leq H$. We did not solve this problem in full generality. Instead, we developed an algorithm that is optimal when the overlay tour is the same length as the overlaid tours (the most common case). In other realistic cases that we have tested (including that shown in Fig. 2) where tours in an overlay segment differ in length, the algorithm recommends allocations that lie on the maximal convex function ϕ. However, it is not difficult to generate unusual examples (such as an overlay tour that is half as long as the overlaid tours) where the algorithm fails.

The algorithm we have developed works in the following way. The <u>initial allocation</u> for each time block is found as in the case of no overlay tours. Then the initial assignment to blocks is converted into an allocation of cars to shifts with the following properties:

1. Every block has at least the number of cars in the initial assignment.
2. The number of car-hours assigned to an overlay segment is as small as possible, consistent with 1.

3. The value of the objective function is as small as possible, consistent with 2.

This is accomplished essentially by finding one shift allocation that meets condition 1 and then searching through all allocations that have the same number of car-hours or fewer car-hours and also meets condition 1.

Then the algorithm iteratively adds car-hours by checking (a) the change in the weighted objective function per car-hour added, assuming that one car is added to each shift in turn, and (b) for each overlay segment the change in the weighted objective function when one car is added to each of the overlaid tours and (simultaneously) one car is removed from the overlay tour. As an example, suppose the algorithm has proceeded to a point where an overlay segment has 8 cars assigned to tour 1, 6 cars to tour 2, and 4 cars to the overlay. The algorithm would then calculate the change in the weighted objective function per car-hour added for the following four possibilities:

Tour 1	Tour 2	Overlay
9	6	4
8	7	4
8	6	5
9	7	3

A proof that this algorithm is optimal when the overlay tour has the same duration as the tours it overlays is given in the User's Manual for the Patrol Car Allocation Model [5].

Potential Future Improvements

Some police departments have begun to adopt ten-hour tours. Typically these departments have a tour that overlaps another tour but is not an "overlay" because during some of its hours it is the only tour on duty. For example, Tour 1 could be from 0600 to 1600, Tour 2 from 1600 to 0200, and Tour 3 from 2000 to 0600. While Tour 3 overlaps Tour 2, during the hours 0200-0600 it is the only tour on duty. The current version of the Patrol Car Allocation Model cannot allocate cars in such arrangements, but the required modifications are not conceptually difficult.

More challenging is to handle the allocation problem for police departments that have complex arrangements of several overlays. For example, eight-hour tours might start every four hours, such as midnight, 0400, etc. In this case every tour can be viewed as an overlay. Designing a suitable patrol car allocation procedure for such departments requires the development of a computationally efficient algorithm for optimizing an objective function having the form described in this paper. Since allocation models are usually operated under severe constraints of core storage and computer run time, general solutions using nonlinear integer programming packages tend to be impractical. However, formulating the problem as one of nonlinear integer programming might lead to insights and simplifications from which a suitable algorithm could be devised. Such an algorithm would also presumably handle a single overlay tour whose duration differs from that of the overlaid tours, a situation which is solved heuristically, but not necessarily optimally, in the current model.

ACKNOWLEDGMENTS

Design and documentation of the Patrol Car Allocation Model was funded jointly by the Office of Policy Development and Research, U.S. Department of Housing and Urban Development, under contract H-2164, and the National Institute of Law Enforcement and Criminal Justice, Law Enforcement Assistance Administration, Department of Justice, under grant 75-NI-99-0012. Since the PCAM program is based on concepts embodied in previous work, we wish to express our indebtedness to the designers of earlier patrol car allocation programs, especially Richard Larson, Richard Mudge, the IBM Corporation, and the Public Systems Analysis class at UCLA. A component of the model was contributed by Peter Kolesar, and David Jaquette wrote the subsidiary program that calculates unavailability parameters by inversion of Erlang's formula. Extremely useful comments on interim versions of the program and draft documentation were provided by Richard Larson, Juan Pineda, the Seattle Police Department, the Los Angeles Police Department, the New York City Police Department, and the referees for this journal.

REFERENCES

1. An Analysis of the Patrol Car Deployment Methods of the Los Angeles Police Department, Engineering School Report by Public Systems Analysis class, University of California at Los Angeles, 1975.
2. Chaiken, Jan M., Patrol Allocation Methodology for Police Departments, The Rand Corporation, R-1852-HUD, 1975.
3. Chaiken, Jan M., Hypercube Queuing Model: Executive Summary, The Rand Corporation, R-1688/1-HUD, 1975.
4. Chaiken, Jan M., and Peter Dormont, Patrol Car Allocation Model: Executive Summary, The Rand Corporation, R-1786/1-HUD/DOJ, 1975.
5. Chaiken, Jan M., and Peter Dormont, Patrol Car Allocation Model: User's Manual, The Rand Corporation, R-1786/2-HUD/DOJ, 1975.
6. Chaiken, Jan M., and Peter Dormont, Patrol Car Allocation Model: Program Description, The Rand Corporation, R-1786/3-HUD/DOJ, 1975.
7. Chaiken, Jan M., and Richard Larson, "Methods for Allocating Urban Emergency Units: A Survey," Management Science, Vol. 19, 1972, pp. P-110-130.
8. Chaiken, Jan M., et al., Criminal Justice Models: An Overview, The Rand Corporation, R-1859-DOJ, 1975.
9. Clawson, Calvin, and Samson Chang, Impact of Response Delays on Arrest Rates, Seattle Police Department, 1975. (Submitted to Management Science.)
10. Crabill, Thomas B., Peter Kolesar, and Warren Walker, Validation of a Police Patrol Simulation Model, The Rand Corporation, P-5464, 1975.
11. Crowther, Richard F., The Use of a Computer System for Police Manpower Allocation in St. Louis, Missouri, Department of Police Administration, Indiana University, 1964.
12. Erlang, A. K. All of Erlang's papers are available in English translation in E. Brockmeyer, H. L. Halstrom, and A. Jenson, The Life and Works of A. K. Erlang, Copenhagen Telephone Company, 1948. Erlang's formulas for the M/M/N queue are given in practically every text on queuing theory and in the PCAM User's Manual, Ref. 5.

13. IBM Corporation, LEMRAS Application Description Manual, Document H20-0629; Law Enforcement Manpower Resource Allocation System (LEMRAS) Program Description Manual, Program 5736-G21, Document SH20-0695-0.

14. Ignall, Edward, Peter Kolesar, and Warren Walker, Using Simulation to Develop and Validate Analytical Emergency Service Models, The Rand Corporation, P-5463, 1975.

15. Isaacs, Herbert, "A Study of Communications, Crimes, and Arrests in a Metropolitan Police Department," Task Force Report: Science and Technology, a report to the President's Commission on Law Enforcement and Administration of Justice, U.S. Government Printing Office, Washington, D.C., 1967.

16. Kakalik, James S., and Sorrel Wildhorn, Aids to Decisionmaking in Police Patrol, The Rand Corporation, R-593-HUD/RC, 1971.

17. Kolesar, Peter, private communication.

18. Kolesar, Peter, et al., "A Queueing-Linear Programming Approach to Scheduling Police Patrol Cars," Operations Research, Vol. 23, 1975, pp. 1048-1062.

19. Kolesar, Peter, and Edward Blum, "Square-Root Laws for Fire Engine Response Distances," Management Science, Vol. 19, 1973, pp. 1368-1378. (Revised and updated version, Square-Root Laws for Fire Company Travel Distances, The Rand Corporation, R-895-NYC, 1975.)

20. Larson, Richard C., Models for the Allocation of Urban Police Patrol Forces, MIT Operations Research Center Technical Report No. 44, 1969.

21. Larson, Richard C., Urban Police Patrol Analysis, MIT Press, Cambridge, Mass., 1972.

22. Larson, Richard C., "A Hypercube Queuing Model for Facility Location and Redistricting in Urban Emergency Services," Computers and Operations Research, Vol. 1, 1974, pp. 67-95.

23. Larson, Richard C., "Approximating Performance of Urban Emergency Service Systems," Operations Research, Vol. 23, 1975, pp. 845-868.

24. Larson, Richard C., Hypercube Queuing Model: User's Manual, The Rand Corporation, R-1688/2-HUD, 1975.

25. McEwen, J. Thomas, "A Mathematical Model for Prediction of Police Patrol Workload," paper presented at TIMS/ORSA Joint National Meeting, May 1968.

26. McEwen, J. Thomas, and R. C. Larson, "Patrol Planning in the Rotterdam Police Department," Journal of Criminal Justice, Vol. 2, 1974, pp. 235-238.

27. McEwen, J. Thomas, Project Director, Allocation of Patrol Manpower Resources in the Saint Louis Police Department, Vols. I and II, report of the St. Louis Police Department, 1968.

28. Mudge, Richard, "A Description of the New York City Police Department RMP Allocation Model," The New York City-Rand Institute, unpublished.

29. Mudge, Richard, and Peter Dormont, "A Description of the New York Police Department RMP Allocation Model," The New York City-Rand Institute, December 1974, unpublished.

30. Shumate, R. P., and R. F. Crowther, "Quantitative Methods for Optimizing the Allocation of Police Resources," J. Crim. Law, Criminology, and Pol. Science, Vol. 57, 1966, pp. 197-206.

31. Urban Sciences, Inc. (177 Worcester St., Wellesley, MA 02181), "Police Resource Allocation Program (RAP)," September 1972.