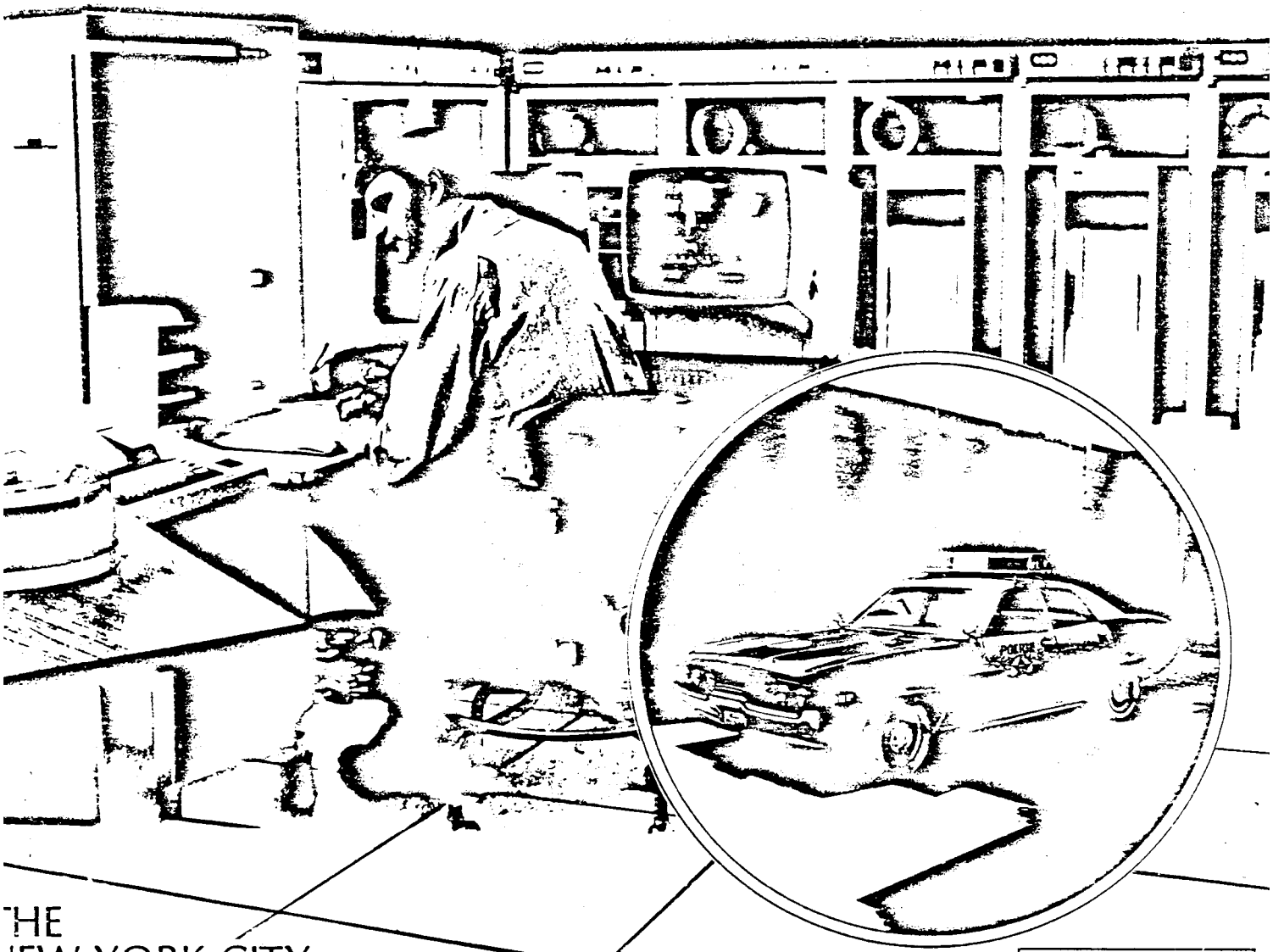
PATROL CAR ALLOCATION MODEL: USER'S MANUAL

PREPARED FOR THE DEPARTMENT OF HOUSING AND
URBAN DEVELOPMENT AND FOR THE DEPARTMENT OF JUSTICE

JAN M. CHAIKEN
PETER DORMONT

R-1786/2-HUD/DOJ
SEPTEMBER 1975



THE
NEW YORK CITY
RAND
INSTITUTE

Rand
SANTA MONICA, CA. 90406

This report was prepared as an account of work sponsored by the United States government, Office of Policy Development and Research, Department of Housing and Urban Development (HUD), under Contract H-2164, and by the Department of Justice, Law Enforcement Assistance Administration, National Institute of Law Enforcement and Criminal Justice, under Grant 75-NI-99-0012. Neither the United States nor HUD, nor the Department of Justice, nor any person acting on behalf of them, makes any warranty, expressed or implied, or assumes responsibility for the accuracy or completeness of the information herein, or assumes liability with respect to the use of any information or method herein.

The New York City-Rand Institute is a nonprofit research institution founded, as its Articles of Incorporation state, "... primarily to conduct programs of scientific research and study, and provide reports and recommendations, relevant to the operations, planning or administration of the City of New York." The Institute was established in 1969 by the City of New York and The Rand Corporation as a center for the continuing application of scientific and analytic techniques to problems of urban life and local government. It is governed by a Board of Trustees appointed jointly by the City and Rand.

Published by The Rand Corporation

PATROL CAR ALLOCATION MODEL: USER'S MANUAL

PREPARED FOR THE OFFICE OF POLICY DEVELOPMENT AND RESEARCH, DEPARTMENT OF HOUSING AND URBAN DEVELOPMENT,

AND FOR THE NATIONAL INSTITUTE OF LAW ENFORCEMENT AND CRIMINAL JUSTICE, L.E.A.A., DEPARTMENT OF JUSTICE

JAN M. CHAIKEN
PETER DORMONT

R-1786/2-HUD/DOJ
SEPTEMBER 1975

THE
NEW YORK CITY
RAND
INSTITUTE

Rand
SANTA MONICA, CA. 90406

PREFACE

This report describes a computer program designed to assist police departments in determining the number of patrol cars to have on duty in each geographical command at different times of the day and week. Chapter I is written for police department administrators and planning officials who wish to understand how the patrol car allocation model can be used in policy analysis. Chapters II and III provide all the information needed to use the program once it has been installed on a computer system. The summary of this report is separately published as

- R-1786/1-HUD/DOJ, *Patrol Car Allocation Model: Executive Summary*.

A third report provides detailed information for data processing personnel to install the program, prepare a data base, and modify the program (if desired):

- R-1786/3-HUD/DOJ, *Patrol Car Allocation Model: Program Description*.

The preparation of this report was supported by the Office of Policy Development and Research of the United States Department of Housing and Urban Development (HUD) under contract H-2164 with The New York City-Rand Institute. Among the objectives of this HUD contract are the development, field testing, and documentation of methods to improve resource allocation procedures in municipal emergency service agencies in the United States.

Design of the computer program described in the report was partially funded by HUD and partially by the National Institute of Law Enforcement and Criminal Justice under grant 75NI-99-0012 to The Rand Corporation. This grant funds a study of computer programs for criminal justice agencies.

This report is part of a series that documents several different deployment models for police, fire, and ambulance agencies. Further information can be obtained by writing to the addresses in Appendix D.

ACKNOWLEDGMENTS

This work was funded by the Office of Policy Development and Research, U.S. Department of Housing and Urban Development, and by the National Institute of Law Enforcement and Criminal Justice. Since the program documented here was based on concepts embodied in previous patrol car allocation programs, we wish to express our indebtedness to the designers of those programs, especially Richard Larson, Richard Mudge, the IBM Corporation, and the Public Systems Analysis class at the University of California, Los Angeles. A component of the model was contributed by Peter Kolesar, and David Jacquette wrote a subsidiary program described in the *Program Description* (R-1786/3).

Comments on interim versions of the program were provided by the Seattle Police Department, the Los Angeles Police Department, and the New York City Police Department, and these were very useful to us in designing the final version.

CONTENTS

PREFACE iii

ACKNOWLEDGMENTS v

FIGURES ix

GLOSSARY xi

Chapter
I. CAPABILITIES AND USES OF THE MODEL 1
Introduction 1
Capabilities of PCAM 2
Typical Applications 6
The Role of Judgment in Using PCAM 13

II. HOW THE MODEL WORKS 16
General Principles 16
Preparation of a Data Base for PCAM 19
Geographical Information 19
Time of Day Information 20
Data for Each Hour 24
Data for Each Shift 25
Data for Each Time Block 26
Unavailability Parameters 27
Calculation of Performance Measures 29
Queuing Delays 29
Utilization 31
Average Number of Cars Available 32
Average Travel Time 33
Average Total Delay 34
Average Patrol Frequency 34
Preventive Patrol as Related to Crime Rates 35
Prescriptive Calculations 36
Meeting Constraints 36
Allocating a Specified Number of Car-Hours 38
Allocating with Constraints 40

III. USER'S GUIDE TO PCAM 41
Overview of Program Operation 41
Data Selection and Modification 42
Descriptive Mode 46
Prescriptive Mode 47
Entering Commands 48
Command Format Conventions 49
Filler Words 50
Program Vocabulary 50

Command Definitions 52
READ 52
LIST 55
DISP 58
ALOC 67
ADD 70
MEET 72
SET 74
WRITE 78
END 79
Error Conditions 79
Selecting an Objective Function 79

Appendix
A. HISTORY OF PATROL CAR ALLOCATION PROGRAMS 83
B. MATHEMATICAL FORMULATION OF PCAM'S CALCULATIONS 96
C. PCAM REFERENCE SHEETS 122
D. ADDRESSES FOR FURTHER INFORMATION 126

REFERENCES 127

FIGURES

1. Example of overlay tour 21

2. Time blocks with an overlay tour 23

3. Example showing calculation of unavailability parameters 28

4. Two sequences for meeting constraints 39

5. Two-step procedure for minimizing an objective function subject to constraints 40

6. Conceptual example of a READ command 43

7. Data selection capabilities of PCAM 45

8. Sample READ command and response 55

9. Sample LIST command and response 56

10. Sample DISP command for Table 1 ordered by tour within day within precinct 60

11. Sample DISP command for Table 1 ordered by precinct within tour within day 61

12. Sample DISP command for Table 2 66

13. Sample ALOC command and response 70

14. Sample ADD command and response 71

15. Sample MEET command and response 74

16. First sample SET command and response 76

17. Second sample SET command and response 77

18. Sample WRITE command 78

B-1. Flow chart for meeting constraints 102

B-2. Initial allocation: no overlay tours 105

B-3. Algorithm for allocating H^* car-hours: no overlay tours 106

B-4. Example of an overlay segment in which the tours do not have the same length 112

B-5. Example of shifts and overlay segments 115

B-6. Potential allocations in the plane with basis M_1, M_2 .. 118

GLOSSARY

ALGORITHM
A procedure for performing a calculation.

ALLOCATE
1. Assign a certain number of cars to each shift.
2. Divide a fixed total number of car-hours among shifts.

AMPERSAND (&)
At the end of a line of PCAM instructions, signifies that the command continues on the following line.

ASTERISK (*)
1. At the start of a line of output from the DISP command, indicates that the tour is overlaid by another tour.
2. In output tables, indicates a limiting constraint.
3. In input commands, represents the current number of car-hours allocated.

AVAILABLE
1. Ready to be dispatched to a call for service.
2. Not engaged in cfs work or non-cfs work.
3. On preventive patrol.

BATCH
A mode of operating a computer program in which all instructions are prepared on cards or other input device prior to program execution, and output is received later from a high-speed printer. Contrasted with INTERACTIVE.

BLOCK, TIME
A period of time (whole number of hours) over which the number of patrol cars on duty does not change. One or two time blocks constitute a tour.

CALL RATE
Average number of calls for service received per hour.

CALL RATE PARAMETER
A parameter for each day in each precinct. When multiplied by the hourly call-rate factor, gives the expected number of calls for service in the hour.

CAR (see PATROL CAR)

CAR-HOUR (ACTUAL)
One patrol car on duty for one hour.

CAR-HOUR (EFFECTIVE)
One hour spent by one patrol car on any activities other than non-cfs work.

CFS
Call for service.

CFS WORK
1. All activities of a patrol car from the time it is dispatched to a call for service until the time it is available again for dispatch.
2. Number of car-hours spent on such activities.

CFS WORKLOAD
1. Loosely speaking, the extent to which cfs work is a burden on a patrol car.
2. Technically, the number of car-hours of cfs work in a given period of time.

COMMAND
1. An instruction to the PCAM program.
2. An administrative unit in a police department that is supervised by a superior officer. (Used in the expression *geographical command.*)

CONSTRAINING MEASURE
Same as LIMITING CONSTRAINT.

CONSTRAINT
A number specified as the largest or smallest value permitted for a performance measure.

CURRENT-DATA
Some or all of the data in DATABASE, which have been read into the computer memory by a READ command and are used and/or modified by PCAM commands.

DATABASE
The data prepared by the user for input into PCAM.

DAY
A 24-hour period used for organizing PCAM data. Not necessarily a calendar day.

DELAY, TOTAL
Sum of queuing delay and travel time. (Same as TOTAL RESPONSE TIME.)

DELIMITER
Any character other than a letter, digit, parenthesis, asterisk, hyphen, period, or ampersand. Examples of delimiters are blanks, commas, colons, and equal signs.

DESCRIPTIVE MODE
Capability to calculate and display performance measures by time of day and geographical command when the numbers of patrol cars on duty in each shift have been specified.

DIVISION
A combination of precincts. Some police departments use the word "division" for a precinct. This is permitted in PCAM by changing the keyword PRECINCT.

EFFECTIVE CAR
The equivalent of a patrol car that does not engage in any non-cfs work.

EXPONENTIALLY DISTRIBUTED
A random variable T is exponentially distributed if there is a parameter μ such that

$$\text{Prob}(T > t) = e^{-\mu t}.$$

The mean of T is $1/\mu$. The assumption that service times for calls to the police are exponentially distributed is not verified by data, but the assumption is technically necessary in PCAM. (This is a source of PCAM's simplicity.)

FIELDED
In the field. A patrol car is fielded if it is on duty.

FILLER WORD
One of the following words, which may be entered in a PCAM command if desired, but will be ignored by the program: FOR, CAR, HOUR, HOURS, TO, ON, BY, DATA.

HOURLY CALL RATE FACTOR
A parameter for a single hour in a single precinct. When multiplied by the call rate parameter for the day, gives the expected number of calls in the hour.

HOURLY SERVICE TIME FACTOR
A parameter for a single hour in a single precinct. When multiplied by the service time parameter for the day, gives the expected service time (in minutes) for calls received during the hour.

INTERACTIVE
A mode of operating a computer program whereby the user enters instructions at a terminal and receives output immediately at the same terminal. Contrasted with BATCH.

KEYWORD
A character string that has a special meaning to the PCAM program. These are either filler words or one of the following: DAY, P, C, T, F, ADD, ALOC, DISP, END, LIST, MEET, READ, SET, WRITE, TOUR (or a substitute provided by the user), DIVISION (or a substitute), PRECINCT (or a substitute).

LIMITING CONSTRAINTS
When meeting constraints, the particular performance measures whose constrained values lead to a need for the largest number of patrol cars. (If these constraints were eliminated, a smaller number of patrol cars would meet all the constraints.)

LIST
Command that causes PCAM to print out the values of the data items associated with all precincts, days, and tours within its scope.

MINIMUM ALLOCATION
The smallest whole number of actual patrol cars that can be assigned to a shift to keep the average utilization of an *effective car* under 1 in every hour.

NEW-DATA
A permanent file which is created by the WRITE command from all or part of CURRENT-DATA.

NON-CFS WORK
1. Any activity of a patrol car that makes the car unavailable for dispatch but was not generated by a previous dispatch to a call for service.
2. Number of car-hours spent on such activities.

OBJECTIVE FUNCTION
The performance measure to be minimized by an allocation.

OPTIMAL
Yielding the smallest possible value of the objective function.

OUTPUT ORDER
A choice of displaying output tables either by tour within day within precinct, or by precinct within tour within day.

OVERLAY TOUR
A tour that begins during one tour and ends during the following tour.

PARAMETER
A number that characterizes a particular hour, block, shift, day, or precinct. See also SERVICE TIME PARAMETER and CALL RATE PARAMETER.

PATROL CAR
A mobile vehicle that can respond to calls for service from the public. Includes vehicles other than automobiles that serve the same function, e.g., scooters.

PATROL FREQUENCY
Number of times per hour that a random point will be passed by a car on preventive patrol.

PCAM
Patrol Car Allocation Model.

PLUS (+)
At the start of a line of output from the DISP command, indicates that the tour is an overlay.

POISSON PROCESS
In the PCAM context, the occurrence of calls for service in a given precinct during a given hour constitutes a Poisson process if there is a parameter λ such that the time between calls has the distribution

$$\text{Prob(time between calls} > t) = e^{-\lambda t}.$$

This assumption is well verified by data.

PRECINCT
A geographical area that is treated as independent from other areas by the patrol car dispatcher. Each patrol car is assigned to an entire tour in one precinct, although it may work in only part of the precinct.

PRESCRIPTIVE MODE
Capability to suggest the number of patrol cars that should be on duty during each shift, so as to meet standards of performance specified by the user.

PREVENTIVE PATROL
The practice of driving a patrol car through an area, with no particular destination in mind, looking for criminal incidents or opportunities, suspicious occurrences, etc.

PRIORITY
Importance of a call for service. PCAM permits three priority levels. Priority 1 calls are so important that the dispatcher will violate ordinary dispatching practices to get a patrol car to respond immediately. Priority 2 calls are important enough that a rapid response is preferred over a slow response. Priority 3 calls can wait in queue without deleterious effect.

QUALIFIER
Phrase(s) associated with a computer command, defining the scope of the command. May be any subset of these phrases, separated by delimiters: 'TOUR=⟨NAMELIST⟩', 'DAY=⟨NAMELIST⟩', 'DIVISION=⟨NAMELIST⟩', 'PRECINCT=⟨NAMELIST⟩'.

QUEUE
In the PCAM context, a collection of calls for service that are waiting to be assigned to a patrol car because no patrol car is available at the moment.

QUEUING DELAY
The length of time a call for service waits in queue.

REGRESSION ANALYSIS
A procedure for fitting a straight line to data so as to minimize the sum of the squares of the deviations of the data from the straight-line estimate.

RESPONSE TIME, TOTAL
Sum of queuing delay and travel time. (Same as TOTAL DELAY.)

SCOPE
The collection of precincts, tours, and days to which the action of a PCAM command applies.

SERVICE TIME
Number of minutes a patrol car will be unavailable from the time it is dispatched to a call until it is available to respond to another call.

SERVICE TIME PARAMETER
A parameter for each day in each precinct. When multiplied by the hourly service time factors, gives the expected service time in each hour.

SHIFT
A particular tour in a particular precinct on a particular day.

SQUARE-ROOT LAW
An equation for the average travel distance D in a region of area A when N patrol units are available:

$$D = \sqrt{\frac{A}{N}} \ .$$

STEADY STATE
In the PCAM context, a situation where the probability of finding n cars available does not change over the course of an hour.

SUPPRESSIBLE CRIMES
Any crimes whose occurrence might conceivably be affected by the amount of preventive patrol. (It is not known whether any crimes are actually "suppressed" by preventive patrol.) The PCAM user is free to define this category of crimes however he chooses.

TIME BLOCK
See BLOCK, TIME.

TOTAL DELAY
Same as TOTAL RESPONSE TIME; the sum of queuing delay plus travel time.

TOUR
A period of time (whole number of hours) beginning when a patrol officer starts work for the day and ending when the officer finishes work. In PCAM, tours are assumed to start at the same time in every precinct on every day (but overlay tours need not be present on every day in every precinct).

TRAVEL TIME

The length of time from the moment a patrol car is dispatched to an incident until the moment it arrives at the scene.

UNAVAILABILITY PARAMETERS

A pair of constants B1 and B2 for each precinct that give the best regression fit to the linear equation

$$\begin{pmatrix}\text{fraction of time}\\ \text{on non-cfs work}\end{pmatrix} = B1 \times \begin{pmatrix}\text{fraction of time}\\ \text{on cfs work}\end{pmatrix} + B2 \ .$$

UTILIZATION

The fraction of time a patrol car is busy on cfs work.

I. CAPABILITIES AND USES OF THE MODEL

INTRODUCTION

The Patrol Car Allocation Model (PCAM) is a computer program designed to help police departments determine the number of patrol cars* to have on duty in each of their geographical commands. Typically, the number of patrol cars needed will vary according to the season of the year, day of the week, and hour of the day. The PCAM program tells a department how to match its actual allocations to these needs, consistent with the overall manpower resources of the department, the levels of performance it desires for patrol cars in responding to calls for service, the hours of the day at which its patrol officers start work, and its dispatching policies.

Although patrol car operations are only part of police work, in most police departments the patrol function consumes over half of the annual budget. Therefore, careful attention to the allocation of patrol resources should be the concern of all police administrators. PCAM provides a tool by which an administrator can establish objectives for the performance of the patrol force and identify those allocation policies that come closest to meeting his objectives. It is intended to substitute for the use of "hazard" or "workload" formulas, which are still widely popular although their failings have been pointed out repeatedly [3,4,10,20].**

This computer program was designed by The New York City-Rand Institute after a careful review of various patrol car allocation programs that have been previously used by police departments.*** Of these, the

---

*By "patrol car" we mean a mobile vehicle that can respond to calls for service from the public. Typical names for a patrol car include "squad car," "radio car," "RMP unit," "black-and-white," and "cruiser." Other vehicles, such as scooters, can be counted as patrol cars in PCAM if they serve the same function.

**Numbers in square brackets identify citations in the alphabetical reference list at the end of this report.

***This review covered all patrol car allocation programs that we were able to locate. A history of the development of such programs is given in Appendix A, together with a description of the capabilities of each program as compared with PCAM.

best known ones are the Law Enforcement Manpower Resource Allocation System (LEMRAS) [9], a product of the IBM Corporation, and the Resource Allocation Program described in Richard Larson's book, *Urban Police Patrol Analysis* [20]. PCAM incorporates many of the features of both of these programs, together with several improvements. In addition, PCAM is available to any police department as a FORTRAN program that can be used "as is" or modified to meet any special requirements of the department, while none of the other programs is generally available at this time.

The PCAM program is designed to run in either batch mode (where the user's input is on cards or a suitable substitute) or in interactive mode (where the user types commands on a teletype or similar terminal and receives output at the same terminal). As the program is distributed, it requires 160K bytes of core storage, but many users, including all departments having eight or fewer geographical commands, will be able to reduce the core requirements if desired.

A copy of the program may be obtained by writing to the address shown in Appendix D. The program is available on cards at a cost of $35, or on tape at a cost of $25, plus $15 if we supply the tape. There is an added charge of $50 for all copies mailed outside the United States.

CAPABILITIES OF PCAM

The Patrol Car Allocation Model has both *descriptive* and *prescriptive* capabilities. The *descriptive* capabilities 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 proposed allocation created by the user, or the particular allocations that are suggested by PCAM when operated in prescriptive mode. This information permits the user to compare allocations and determine which one he thinks is best. The *prescriptive* capabilities of PCAM specify particular allocations that best meet the standards of performance established by the user.

The information provided to the user when PCAM is operated in descriptive mode include the following:*

- The number of patrol cars assigned to each geographical command at each time of day.
- Information about the workload of the patrol cars.
- Information about the amount of preventive patrol engaged in by the patrol cars.
- Average length of time from the dispatch of a patrol car until its arrival at the scene of an incident (*travel time*).
- The percentage of calls that will have to wait in queue until a patrol car is available to dispatch to the incident.
- The average length of time (in minutes) that calls of various levels of importance (or *priority*) will have to wait in queue.
- The average *total response time* (time in queue plus travel time).

In prescriptive mode, PCAM has several capabilities. One of them will tell the user the *minimum number* of patrol cars that must be on duty in each geographical command at all hours of the day to meet standards of performance related to the information listed above. Examples: What is the smallest number of patrol cars needed to assure that no more than 20 percent of calls must be placed in queue? What is the smallest number of patrol cars needed to assure that the average total response time is less than 10 minutes? What is the smallest number needed so that both of these conditions are met?

The second prescriptive capability will tell the user the *"best" allocation of his existing resources* among geographical commands and/or among different times of the day or week. PCAM permits the department to choose among several definitions of "best":

---

*Details will be given in Chapter II.

- The average percentage of calls that must be placed in queue is as small as possible, given existing resources,
- The average length of time calls of a given priority must wait in queue is as small as possible, or
- The average total response time is as small as possible.

The third prescriptive capability is a combination of the two already described. It permits the user to obtain an allocation that (a) meets specified performance standards and (b) is the "best" allocation that can be achieved while meeting those standards.

While these capabilities are quite comprehensive, PCAM cannot tell the police administrator everything he would like to know before making allocation decisions. For example, the reasons an administrator would be interested in reducing response times are to increase the number of on-scene apprehensions of criminals (thereby hoping to decrease crime rates) and to improve public satisfaction with the service provided by the department. But PCAM cannot calculate the number of criminal apprehensions that will result from a particular allocation policy because the information available from past research [5,9] is not adequate to make such calculations, and practically nothing is known about the relationships between response times and crime rates or public satisfaction.

Similarly, PCAM can tell the administrator what will happen to the amount of preventive patrol if he adopts a particular allocation, but it cannot tell him how this will make service better or worse. Recent studies [11] have cast doubt on whether routine preventive patrol has any effect on crime rates, but some departments may want to maintain a certain level of preventive patrol because it serves a traffic control function, permits searching for stolen automobiles, gives police officers an opportunity to provide needed services to the public, or otherwise enhances the police role.

Users of PCAM will also rapidly come to realize that the program cannot relieve them of the responsibility to make difficult decisions about their allocation policy. It is impossible for one allocation to be better in all parts of the city and at all times of day than another

allocation that uses the same manpower resources, so the administrator will have to choose among conflicting alternatives. For example, a change in allocation policy may yield a lower *average* response time than current practices, but then some locations or times of day will have higher response times than they did in the past. Or one allocation might yield better average response time than another, but it has worse imbalances in workload for the patrol officers. These problems of conflicting objectives are not *caused* by PCAM, but the computer program highlights the fact that they exist. Ordinarily they are resolved by carefully reviewing the performance measures for several different allocations in light of the allocation problem being addressed. We shall illustrate this process by giving some examples in the chapter that follows.

Before continuing, however, we wish to point out that PCAM has certain limitations which should be understood by any department considering using this program. First, it is not entirely self-contained, because the user must prepare an appropriate data base for PCAM. Most departments would find it necessary to write subsidiary computer programs to calculate the required input data from dispatchers' cards or similar sources of basic information about the operations of patrol cars. To this extent, PCAM differs from LEMRAS, which did have the capability to calculate many of the input data needed by PCAM.

Second, PCAM is not a complete package for analysis of all questions a department might have about the operations of its patrol cars. For example, PCAM cannot be used to design patrol beats for patrol cars, nor can it analyze the potential advantages of a car locator system or the institution of new policies for selecting which patrol car to dispatch to each incident. For these purposes, other, more complex, models are required. The Rand Corporation can provide police departments with programs and user's manuals for two such models, the Hypercube Queuing Model [2,18] and the Patrol Car Simulation Model [17], as well as this manual for PCAM. Interested readers should consult the documentation of those programs for further information.

If the calculations and analyses that a department wishes to perform match PCAM's capabilities, then PCAM has several advantages in

comparison with a simulation model. First, no special programming skills are required to use PCAM. In fact, after it has been installed on the department's computer, PCAM can be operated by individuals who have no programming experience whatsoever. Second, considerably less data must be assembled for PCAM than for a simulation model. Finally, PCAM is very inexpensive to operate and requires a relatively small amount of computer core storage.

The primary disadvantage of PCAM in comparison with a simulation model is that PCAM's calculations are based on simplifying assumptions that may not represent a department's patrol car operations exactly. However, our experience in several cities suggests that having better estimates of performance measures as calculated by a simulation model would lead to exactly the same allocations as one derives using PCAM, or at most a difference of one patrol car at certain times or places. Indeed, errors that arise in collecting data often lead to greater inaccuracies than the approximations incorporated in the PCAM program.

TYPICAL APPLICATIONS

To illustrate the capabilities of the Patrol Car Allocation Model, we shall present a few hypothetical examples, explaining how the program would be used in each case. The reader should consult the Glossary if he encounters any unfamiliar terms in the examples.

1. Justification of a Budget Request

In past years, Department A was almost always able to find a patrol car available to dispatch immediately to a call from the public. However, during the last year or two, a rapid increase in the number of calls for service has led to a situation where approximately 15 percent of all calls must wait in queue before a car is available for dispatch. The department's chief considers this a marginally acceptable level of performance but is concerned that the expected increase in calls for service next year will create even worse delays. For justification of a request to the City Council for an increase in the authorized strength of his department, the chief would like to know (a) what percentage of calls will have to be placed in queue next year, if there is no increase

in the size of the force and current trends in call rates continue, and (b) how many new patrol officers he would have to hire to assure that less than 15 percent of calls are queued next year.

Let us suppose that City A is divided into 5 patrol districts, each of which has somewhere between 2 and 6 patrol cars on duty, depending on the time of day and day of the week. Patrol officers in City A work one of three "watches," midnight to 8 a.m., 8 a.m. to 4 p.m., or 4 p.m. to midnight.

To apply PCAM to answer the chief's questions, it will be necessary for the Planning Unit of Department A to determine the average number of patrol cars on duty and the average number of calls received in each of the districts during each watch of each day of the week. (Additional input data are needed for PCAM, and these will be described in Chapter II of this report, but the number of cars on duty and the call rates are the key items for this application.) After the data base has been assembled, it would be prudent to have PCAM estimate the percentage of calls delayed and check that this figure agrees with the observed amount (15 percent). If so, this indicates that the data have been prepared properly.

Next, the Planning Unit will have to estimate what the call rates will be next year. Although PCAM cannot help in making these estimates, a suitable approach would be simply to multiply current call rates by a number that incorporates the current annual rate of increase. (For example, if call rates are increasing by 17 percent per year in District 1, the current call rates in District 1 would be multiplied by the number 1.17.)

The Planning Unit now has a choice of preparing a new data base that has the changed call rates in it, or simply entering a command to PCAM that increases the call rates. PCAM will then describe the queuing delays that will occur if the current allocations of patrol cars remain unchanged. Let us suppose the answer is that, on the average, 28 percent of calls will be queued, which is well past the standard set by the chief.

However, this does not quite answer the chief's first question, because it is possible that by changing the number of cars assigned

to certain districts or watches next year, the delays would be reduced. By entering another command to PCAM, it is possible to determine how the *existing* total number of man-hours devoted to patrol should be reallocated among districts and watches so as to have the *lowest possible percentage of calls delayed* next year. Let us suppose the result is that, on the average, 23 percent of calls will be delayed if the cars are reallocated. This answers the chief's question regarding what will happen if he gets no budget increase.

To answer the chief's second question, there are two ways to proceed. One is to note that each new officer hired can increase the number of car-hours provided in a week by about 32. (This reflects Department A's experience that about one day per week is not spent on patrol duty, due to training, court time, vacation, etc., and the fact that each patrol car in City A is manned by a single officer.) Therefore, a series of commands can be entered into PCAM to increase the number of car-hours by various multiples of 32, each time reallocating to achieve the smallest possible average percentage of calls delayed. The smallest multiple of 32 that results in an average of under 15 percent of calls delayed represents the number of officers that would have to be added.

The second way to proceed is to enter a single command into PCAM that asks what is the smallest total number of car-hours needed to assure that under 15 percent of calls are delayed in *every* watch in *every* district. Subtracting the number of car-hours currently fielded from the answer given by PCAM, and then dividing by 32, gives the number of new officers needed. This answer will, in all likelihood, be *different* from the answer given by the first procedure, since the first method tells how many officers are needed to keep the *average* number of calls queued under 15 percent. In this case, some watches in some districts will have more than 15 percent of calls queued; others will have less than 15 percent queued. But the second procedure will assure that in every watch in every district the average number of calls placed in queue will be under 15 percent of the total.

To resolve the difference, the chief will be shown both sets of results. He will decide which figures he would like to present to the

City Council, since this is really a matter for his judgment, and cannot be left to a computer program.

2. Adjusting to a Budget Constraint

Department B is in a much worse situation than Department A. At the present time, over 40 percent of all calls in City B must be placed in queue, and even "emergency" calls experience an average delay of 9 minutes before a patrol car can be dispatched. (In Department B an "emergency" call is one that the department would like to respond to rapidly, but it is not as urgent as a "top priority" call, such as "officer needs assistance.") Moreover, the budget allocated to Department B for next year is the same as for this year. In light of recently negotiated salary increases, this means that Department B will have to reduce the number of car-hours devoted to patrol. Since Department B has already been using a patrol car allocation program, the chief is fairly certain that delays cannot be reduced substantially by reallocation, and therefore he has reluctantly concluded that certain types of calls from the public will have to be excluded from receiving a response by a patrol car next year.

The chief has prepared a tentative list of types of calls that might be excluded from receiving a response, and he would like to know how many of the items on his list will have to be excluded to bring the average delay for emergency calls under 5 minutes.

For this application, the Planning Division in Department B already has the information it needs to prepare a PCAM data base describing the current situation. But, to answer the chief's question, it will be necessary to find out how many calls there are of each type shown on the chief's list. This will have to be done for each of City B's 11 patrol precincts for each eight-hour tour during the week.* In addition, the Planning Division will have to estimate, from next year's

*Department B uses the name "precinct" to refer to the same kind of geographical command that Department A calls a "district," and Department B uses the word "tour" instead of "watch." Since one of PCAM's desirable features is that the user specifies the names that are applicable in his department, we shall use various names in these hypothetical examples without intending to imply any difference in meaning.

budget, how many car-hours of patrol can be provided. Let us suppose the answer is that Department B will be able to field 10,000 car-hours of patrol per week next year.

Now PCAM's data base does not include a count of the number of calls in each precinct and tour according to the type of the call. Instead, calls are aggregated into three priority levels in PCAM's data base: priority 1 ("top priority"), priority 2 ("emergency"), and priority 3 ("all other calls"). If one of the call types on the chief's list were to be eliminated, this would change the total call rate and the fraction of calls that fall into each priority level. The planning officers will have to calculate what these changes would be.

Then a sequence of commands would be entered into PCAM to change the data in a way that imitates the elimination of the first call type on the chief's list. Next, a command would be entered to allocate 10,000 car-hours among all the precincts and tours so as to minimize the average delay for priority 2 calls. If the resulting delay, as estimated by PCAM, is under 5 minutes, then only this one type of call would have to be eliminated. Otherwise, the process would be repeated for the second type of call on the chief's list, and so forth, until the estimated priority 2 delay is under 5 minutes.

At the end of this analysis, the planning officers would know (1) which types of calls will have to be excluded from dispatch, (2) how many officers should be assigned to each precinct next year for patrol car duty, and (3) how many patrol cars should be on duty during each tour in each precinct next year.

3. Reallocation Among Tours

Let us suppose that the chief of Department B approves the plan developed in the example described above. However, six months later, the plan is found to have been unrealistic in two aspects. First, although the public has accepted the fact that patrol cars will not respond to certain types of incidents during the daytime, at night they insist that the police respond, and dispatchers have been assigning patrol cars to the supposedly "excluded" incidents at night. Second, precinct commanders did not reduce the number of patrol cars in

the field when their manpower was reduced, but instead they removed officers from other types of assignments.

So, rather than having 10,000 car-hours of patrol work per week, the department actually has 11,000 car-hours. Moreover, delays in dispatching are rare during the day but frequent at night. What is needed is a reallocation of manpower among the tours. The chief does not want to change the number of patrol officers assigned to each precinct until the next budget cycle, so the total number of car-hours in each precinct must remain the same.

To perform the required analysis, Department B's Planning Division must update the PCAM data base to reflect what has actually happened to call rates in the three priority levels, and also determine the number of car-hours currently fielded in each precinct. Then, for each precinct, the planning officers (or the precinct commander) would enter a small number of commands into PCAM that result in allocating the existing number of car-hours among the tours of the week. This would produce the desired new allocation.

4. Possibility of an Overlay Tour

The officers in City C work tours that begin at midnight, 0800, and 1600. However, most of the call-for-service workload occurs during the hours 1800-0200. The department would like to analyze the possibility of reducing the number of patrol officers on the three existing tours and establishing an overlay tour that works from 1800 to 0200.

Department C already uses PCAM for patrol car allocation, so it only takes a minute or two to change the data base in such a way as to permit an overlay tour. There is no need to collect any additional data. In its previous use of the model, Department C decided that suitable allocations were obtained by minimizing the average response time after meeting two constraints:

- Not more than 20 percent of calls are queued.
- A random point is passed by a car on preventive patrol at least once every four hours.

To allocate the existing patrol resources among four tours instead of three, the department will use the same criteria. This can be done in such a way that the number of patrol officers assigned to each of Department C's five patrol divisions remains unchanged, or the officers can be reallocated among divisions as well as tours.

In the latter case, only three commands need to be entered into the PCAM program. One assures that the constraints are met, and the second allocates the remaining patrol resources so as to minimize average response time. The third command causes PCAM to display the estimated performance measures for the allocation with four tours. By comparison with current performance measures (previously calculated using PCAM) the department can determine the extent to which response time and queuing delays will be reduced and workloads will be balanced if it adds an overlay tour. The same display tables show the number of patrol cars that should be assigned to each tour, so the department will know how many patrol officers would have to be reassigned from each existing tour to the new overlay tour.

5. Adding Resources

Department C is about to graduate a class of 75 new recruits and wants to assign them to patrol car duty. A single command to PCAM will determine how many of them should be assigned to each patrol division so as to minimize average response time. (The constraints are already met.) One more command will display the results.

6. Seasonal Variations

City D has a large recreation area that is busy in the summer and closed in the winter. For this reason, Department D experiences fairly large fluctuations from month to month in the number of calls for service received from different parts of the city. The department adjusts to these variations by having a group of patrol officers (the Mobile Patrol Team) who are not assigned to any of the city's seven patrol districts, but instead move from district to district every month.

Department D uses PCAM to determine how many of the Mobile Patrol Team officers should be assigned to each district each month. Part of

the department's PCAM data base describes the current allocation of those officers who are assigned to patrol districts. Another part of the data base is updated every month to describe the call-for-service workload expected in each hour of the week in every district during the upcoming month. These workloads are predicted by a computer program that was developed by the department and is based on a statistical technique known as "exponential smoothing." (The same method was used in LEMRAS.)

By entering a single command to PCAM, the total car-hours that can be provided by the Mobile Patrol Team are allocated among districts and watches, *as an addition* to the car-hours already provided by the district officers. By displaying PCAM's allocations after the addition, and comparing them with the (fixed) allocation of district officers, the results tell not only the number of Mobile Patrol Team officers who should be assigned to each precinct, but also which watches they should work.

THE ROLE OF JUDGMENT IN USING PCAM

As is illustrated by the above examples, PCAM is not a single-purpose automatic method for allocating patrol resources. Rather, it is a flexible tool that can adapt to a wide range of user requirements. Only the user can specify what type of allocation question is to be answered by PCAM, and only he can establish the standards of performance that PCAM will use in prescribing allocations.

The first few times that a department uses PCAM, it will be natural to try several different standards of performance to see what happens. But it will soon become apparent what standards lead to allocations that are both feasible and acceptable to the department. Thereafter, the department would ordinarily want to use the same standards in every run of the PCAM program.

For example, a department might *wish* it could allocate patrol cars in such a way that less than 5 percent of calls encounter a queue at all times of day in every precinct. But if the current situation is that about 15 percent of calls are queued in nearly every precinct, there will be no way that the desired objective can be achieved with

existing resources. If the user asks PCAM to tell him how many car-hours are needed to assure that under 5 percent of calls are delayed in every tour in every precinct, he will rapidly discover that the answer far exceeds the number of car-hours the department can field. This, then, is an *infeasible* standard, and the department will have to be satisfied with a less demanding standard.

Similarly, a user might *think* he wants his allocation to minimize the citywide total of calls that will be queued. But when he asks PCAM to allocate patrol cars according to this objective, he might find that the resulting allocation, while bringing the number of calls queued to under 7 percent of the total, produces enormous delays in two precincts that have a low rate of calls for service. Such an allocation is *unacceptable*, and the user will have to establish a new standard, perhaps including the condition that no precinct will be permitted to have more than 25 percent of its calls delayed.

In short, the choice of standards and objectives is a matter of judgment that can best be decided in each department by inspecting the allocations that PCAM produces using several different standards.

Another area for exercise of judgment is the preparation of a data base. While PCAM requires certain information to be in the data base, the user may tailor the accuracy of the data to the particular application he has in mind. As an example, if PCAM is to be used for long-term planning related to total size of the force, then rough estimates of next year's call rates will be suitable input data. These need not be broken down carefully by hour; instead, as an approximation, the estimated average hourly call rate in a tour can be entered into the data base for every hour in the tour. However, if the department is considering changing the hours at which tours begin, then it is important to estimate accurate call rates for each hour.

Similarly, if the department is not interested in distinguishing performance measures according to the priority level of the calls, it is free to count all calls as if they had the same priority. This eliminates the necessity for deciding which types of calls belong to priority 1, 2, and 3, and collecting separate statistics for the three priority levels.

Other shortcuts in preparation of the data are possible if the department is not concerned with the accuracy of certain performance measures, and these will be indicated where appropriate in Chapter II.

II. HOW THE MODEL WORKS

In this chapter we shall discuss the data base required for PCAM, tell how the program calculates the performance characteristics it displays, and describe the procedures it uses to prescribe allocations. The discussion is nontechnical, with details relegated to a companion volume, the Glossary, and the appendices. We begin with some general terminology and principles that will be referred to throughout the chapter.

GENERAL PRINCIPLES

To simplify the discussion, we shall use the term "precinct" to refer to an independent geographical command that is variously called a precinct, division, district, area, beat, or sector. 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 that (a) its commander has the capability or authority to decide how many patrol cars will be fielded at various times, and (b) the dispatchers of patrol cars treat the precinct as an independent command by sending only its cars to incidents in the precinct, except under unusual circumstances.

Some police departments are small enough that they do not have separate geographical commands. For them, PCAM can be used to determine how the total number of patrol cars they field should vary by time of day. Such departments should think of themselves as a single "precinct" for purposes of the discussion that follows.

The Patrol Car Allocation Model operates on the principle that calls for service to the police requiring the dispatch of a patrol car will be assigned to a single car in the precinct of occurrence, if one is available. If a call arrives when all cars in the precinct are busy, PCAM assumes that the call will be placed in queue until one of the precinct cars is available for dispatch. If several calls are already in queue when another call arrives, PCAM assumes that the order in

which the waiting calls will be assigned to cars depends on their relative priority or importance. PCAM allows three priority levels. All priority 1 calls in queue will be dispatched before any call of priority 2, and all priority 2 calls will be dispatched before priority 3.

Ordinarily, none of these assumptions is precisely correct in practice. Every department receives at least a few calls that require more than one patrol car to be dispatched. In addition, if an extremely urgent 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 program occur infrequently, then they may be ignored for all practical purposes. However, if the variations are extreme, 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 a department dispatches two cars to some fraction of its calls for service, it is appropriate to count each dispatch in the input as a separate call for service. On the other hand, if a department *always* dispatches two cars to each incident, and they stay there for the same length of time, it would be more accurate to count each call once and interpret the output to indicate how many *pairs* of patrol cars should be fielded, i.e., the numbers should be doubled. If a department would *usually* dispatch a car from a neighboring precinct rather than place a call in queue, then the term "precinct" has not been defined properly for that department, and larger areas should be considered precincts.

PCAM assumes that patrol cars can be busy on two types of activities. One, called *cfs work*, results when a patrol car is dispatched to a call for service. All other activities that prevent a patrol car from being dispatched to a call constitute *non-cfs work*, which is also sometimes called downtime.* These activities include meals, auto

*However, the word *downtime* suggests that no useful police function is performed during this time, which is not necessarily the case.

repairs, on-view incidents requiring police action, special assignments by commanding officers, and the like. When a patrol car is not busy on either cfs work or non-cfs work, it is assumed to be on preventive patrol. This is ordinarily not exactly correct, since patrol cars may be engaged in nonmobile work while still available for dispatch. However, it is a reasonable approximation for comparing the amount of preventive patrol among tours or precincts.

Data collected in several cities show that non-cfs work consumes at least 20 percent of a patrol car's time, and sometimes as much as 60 percent, so its effect on unavailabilities of patrol cars is as great as that of cfs work, or even greater. Therefore, if an attempt is made to use PCAM without determining the amount of non-cfs work, the output from the program will bear little relationship to reality and will therefore be virtually useless as an aid to planning. In the absence of any information about the amount of non-cfs work, it is better to make an educated guess than to ignore the problem.

One method for taking non-cfs work into account is to consider every incident that causes a patrol car to be unavailable as if it were a call for service. If the department's estimates of non-cfs work are accurate, this method will result in a good match between PCAM's calculations of performance measures and the actual situation in the field. However, *the method is not recommended*, because it is extremely difficult to estimate what will happen to non-cfs work if the allocation of patrol cars 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 amount of non-cfs work, quite independent of how much non-cfs work there was in the past.

The recommended method for taking non-cfs work into account in PCAM is to assume that each patrol car will spend a certain fraction of its time on non-cfs work. For example, suppose the fraction is 1/3 in one precinct. Then if 6 patrol cars are on duty in the precinct, there will be 2 car-hours of non-cfs work per hour, but if 9 cars are on duty, there will be 3 car-hours of non-cfs work per hour. A study conducted in cooperation with the Los Angeles Police Department [1]

showed that the fraction of time spent on non-cfs work actually varied with the amount of cfs work per car, so PCAM permits the user to include two *unavailability parameters* in the data base; these tell PCAM how the fraction of time spent on non-cfs work is related to cfs work. We shall describe later how the department can determine its unavailability parameters.

PCAM uses the unavailability parameters to convert the *actual* number of cars fielded into *effective* cars. An effective car is the equivalent of a patrol car that spends all its time on cfs work and preventive patrol; it does not perform any non-cfs work. In the example of the precinct where cars spend 1/3 of their time on non-cfs work, when 9 actual cars are fielded there are 6 effective cars. (The number of effective cars does not have to be an integer. Thus, if 8 actual cars are fielded, there will be $5\frac{1}{3}$ effective cars.)

PREPARATION OF A DATA BASE FOR PCAM

The data base for PCAM must be prepared according to a format that is detailed in Chapter II of the companion *Program Description* (R-1786/3). An example of such a data base is shown in Appendix A, "Demonstration Data Base," of the same volume. In this chapter we describe the information that must be collected to prepare a data base.

Geographical Information

As mentioned previously, PCAM imagines the city to be divided geographically into precincts. (However, a city having only one precinct is permitted.) Optionally, the precincts can be considered as belonging to larger geographical commands that we shall call "divisions." If the user wishes to allocate patrol cars within each division, one at a time, or to display summary statistics for each division, this can be a useful feature. Otherwise, the name of the division to which a precinct belongs serves no other function in PCAM; all data must be prepared separately for each precinct, whether or not the user aggregates precincts into divisions.

PCAM permits the user to substitute any words he wants for "precinct" and "division." The words he chooses will appear in the headings

for all tables and will substitute for the words PRECINCT and DIVISION in all the commands to PCAM that are described in Chapter III. For example, if the department chooses the words DISTRICT and BUREAU, then the information printed out by PCAM for each district will appear under the heading "DISTRICT." These words are the first data items provided in the data base.

Each precinct and division must have a *name*, such as MIDTOWN, NORTH, or FIFTH. PCAM permits names to be up to 8 characters long, of which the first must be a letter. The other 7 characters may be letters, numbers, periods, or hyphens, but no blanks are permitted. If a department's precincts and divisions are numbered, it may choose names such as ONE, TWO, THREE, ..., or FIRST, SECOND, THIRD, ..., or P1, P2, P3, ..., or P-1, P-2, P-3, ..., or PCT.1, PCT.2, PCT.3, ... The names selected by the user are entered into the PCAM data base.

Additional characteristics of precincts that must be entered into the data base are their area (square miles) and number of miles of streets to be patrolled. If the department does not know the number of street miles in its precincts, a reasonable approximation is (street miles) = 35 × (square miles). The factor 35 may be different in some cities, depending on how close together the streets are.* The number of street miles is used primarily to calculate relative amounts of preventive patrol and does not have to be exactly correct if the user is not interested in preventive patrol.

Time of Day Information

For purposes of entering data into PCAM, time is divided into 24-hour periods called *days*. However, PCAM's days are not necessarily the same as calendar days, because PCAM requires that every tour must belong entirely to one day. A *tour* is defined to be a period of time during which a patrol car may be on duty. Thus, if a department has some patrol cars that work from 7 p.m. to 3 a.m., a *day* cannot begin at midnight. If, in addition, some patrol cars are on duty from

*In case some locations are patrolled more heavily than others, the user will want to enter the "effective" number of street miles. See Chapter 4 of Ref. 20.

midnight to 8 a.m., then a *day* cannot begin at 3 a.m. either, but might start at 8 a.m. (See Fig. 1.)

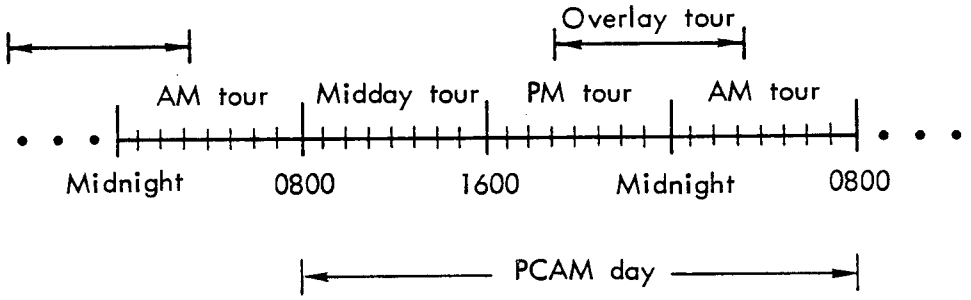


Fig. 1 — Example of overlay tour

The simplest situation is if the department's cars begin work at midnight, 0800, and 1600. Then, every day can begin at midnight, and there are three tours in each day. Any other arrangement of three 8-hour tours is nearly as easy to handle. The starting time of one of the tours is selected as the beginning of the day, and all the others will fall within the day. For example, if tours begin at 0715, 1515, and 2315, the department could choose to have PCAM's day begin with the hour from 0715 to 0815 or the hour from 2315 to 0015. In any event, *the first hour in PCAM's data base is the first hour of PCAM's day*, which may or may not be the hour beginning at midnight.

The PCAM data base may contain any number of days. Common applications would involve a single day (a "typical" or "average" day) or seven days. However, if desired, all the days in a month can be included in the data base.

PCAM allows tours to be any number of whole hours in length, although most applications will involve 8-hour tours. In addition, PCAM will permit *one* tour in each day to be an *overlay* tour, meaning that it begins during one tour and ends during the following tour. For example, if the department's tours begin at midnight, 0800, 1600, and 1900, then the tour from 1900 to 0300 is an overlay tour. (See Fig. 1.)

This department can begin its PCAM *day* at 0800 or 1600, but not at any other time. (In the figure, it is assumed to begin at 0800.) If there is an overlay tour, PCAM does not require that it be present on every day of the week nor in every precinct. For example, it is possible to have an overlay tour only on Fridays and Saturdays in Precinct 4.

If, for some reason, the department does not wish to use PCAM for allocation of patrol cars over entire days, it may enter nonsense data, for example all zeros, for the hours of the day that are not of interest, so long as no attempt is made to have PCAM read the data for those hours.

PCAM requires that all tours begin at the same hour of the day in every precinct on every day. If this is not the case, the department may choose to make a reasonable approximation (for example, if some precincts begin tours at 0745 and others at 0815, it will do no harm to approximate all of these as 0800), or it may choose to prepare separate data bases for each group of precincts having the same arrangement of tours, but then allocations can be made only among precincts in the same data base.

Just as with divisions and precincts, PCAM permits the user to choose his own word to substitute for "tour." For example, the words WATCH, SHIFT, or PLATOON can be selected. If this is done, the selected word will appear in all table headings and will be used in commands to PCAM. However, for the remainder of this report we shall continue to use the word *tour* to refer to a period of time that is the same on every day in every precinct, and we reserve the word *shift* to mean a specific tour on a specific day in a specific precinct.

An important concept in PCAM is a *time block* (or *block* for short). This is a period of time during which the number of patrol cars on duty does not change. If the department has three 8-hour tours each day, then a time block is the same as a tour. However, in the example of the department with 8-hour tours beginning at midnight, 0800, 1600, and 1900, there are five time blocks: 0000-0300, 0300-0800, 0800-1600, 1600-1900, and 1900-2400. (See Fig. 2.)

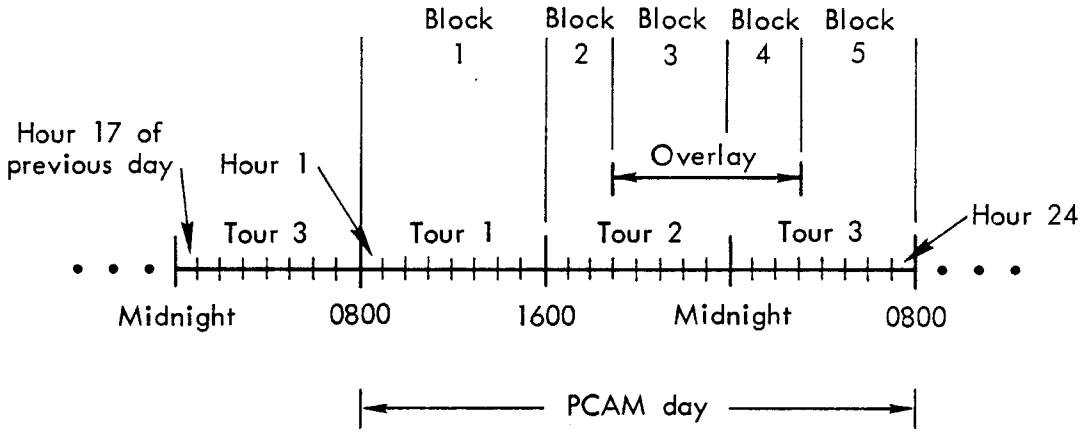


Fig. 2 — Time blocks with an overlay tour

Supposing that this department begins its PCAM *day* at 0800, then the time blocks are described to PCAM as follows:

Block 1	Hours 1-8
Block 2	Hours 9-11
Block 3	Hours 12-16
Block 4	Hours 17-19
Block 5	Hours 20-24

Tours are described in terms of the blocks they include. Continuing the example, the tours in this department are:

Tour 1	Block 1
Tour 2	Blocks 2 and 3
Tour 3	Blocks 4 and 5
Overlay tour	Blocks 3 and 4

To describe temporal characteristics in the PCAM data base, the user enters the number of time blocks in each day (up to 24), the number of tours in each day, the *ending* hours of each block (8, 11, 16, 19, and 24 in the example), and the blocks that constitute each tour.

In addition, every tour and day is given a *name*. As with precincts and divisions, these are limited to 8 characters, of which the

first must be a letter, etc. Examples of permitted tour names are FIRST, SECOND, THIRD, FOURTH, AM, PM, MIDDAY, NIGHT, GRAVEYRD, AFTRNOON, MIDN-8, T9-16, T16-24, and W.8X16. The user is not permitted to call a tour DAY unless he is willing to take special precautions when operating the program (see Chapter III). Examples of suitable day names are MONDAY, MON-TUE, and MARCH27.

Data for Each Hour

For every hour of every day in every precinct, PCAM needs to know the average number of calls for service that are expected to occur (*call rate*) and the average *service time* (number of minutes a patrol car will be unavailable from the time it is dispatched to a call until the time it is available to respond to another call).

PCAM provides that the call rate is entered into the data base by specifying two numbers: a *call rate parameter* for the day and precinct and an *hourly call rate factor*. The product of these two is the call rate in the hour. This arrangement permits flexibility in organizing the data. Examples of how this feature might be used are as follows:

1. The call rate parameter could be the *total* number of calls in the day, and the hourly call rate factor could be the *fraction* of all calls occurring in that hour.
2. The call rate parameter could be the *average* number of calls in an hour of the day, and the hourly call rate factor could be a number near 1.0 that indicates how the number of calls in the hour differs from average. (This corresponds to the numbers calculated by the prediction component of LEMRAS.)
3. The call rate parameter could be the *maximum* number of calls in an hour, and the hourly call rate factor could be a number less than or equal to 1.0 indicating how the hour differs from maximum.
4. The call rate parameter can be 1, and the hourly call rate factor equal to the expected number of calls in the hour. (This method facilitates considering what would happen if call rates change by a certain percentage.)

When running the PCAM program, the user works with a copy of the data base that is stored in the computer's memory and is called CURRENT-DATA (see Chapter III). He is permitted to modify the call rate parameter in CURRENT-DATA, but not the hourly call rate factors. This allows the user to imitate what would happen if there is a general increase or decrease in the number of calls being received, leaving the *relative* proportion in each hour unaffected.

Service times are handled in the same way, with a *service time parameter* for the day and precinct, and an *hourly service time factor* for each hour. Although service times will usually be found to vary from precinct to precinct, a reasonable assumption is that each *type* of call requires the same service time in each precinct, but the *mix* of calls varies among precincts. (For example, some precincts just happen to get more calls requiring over an hour of service time.) Thus, it is sensible to divide calls into 20 or so types, calculate the average service time for each type from *citywide* data, and estimate service times in each precinct according to the fraction of calls of each type. This is the method used in LEMRAS.

Depending on the kinds of data available to the department, the user is free to assume that call rates do not vary over a tour, that service times are the same in all precincts in all tours, etc. However, assuming that call rates do not vary from tour to tour will ordinarily be a very poor approximation.

Data for Each Shift

We define a *shift* to be a particular tour in a particular day in a particular precinct. The data required by PCAM for each shift are as follows:

1. The average number of patrol cars that start work at the beginning of the shift.
2. The average speed (miles per hour) that cars travel when responding to calls for service. While police officers often estimate rapid speeds of response, data in several cities show that this speed averages around 15 mph and

is rarely over 25 mph. Some departments may have conducted experiments in which the travel time and travel distance of responding patrol cars were recorded. If so, they can determine the average travel speed from this information. Otherwise, since the travel speeds are used to estimate average travel times for precincts, it is best to collect some data showing travel times under the current allocation and then simply adjust the input value of travel speed until PCAM's estimates of travel times agree with the actual data.

3. The average speed (miles per hour) that cars travel when on preventive patrol. This is ordinarily in the range of 7-15 mph.
4. Fraction of calls which are priority 1 and fraction of calls which are priority 2. (Recall that three priority levels are permitted by PCAM. Therefore, the sum of these two fractions must be 1.0 or less. PCAM will calculate the fraction that remains for priority 3 calls.) If the department does not wish to separate calls into priority levels, we recommend setting the fraction of calls that are priority 2 equal to 1.0, in which case the fraction of priority 1 calls must be entered as 0.0.

Data for Each Time Block

Some departments feel that a comparison between two precincts in regard to the amount of preventive patrol provided must take into account the relative crime rates in the two precincts. More particularly, only certain types of crimes are considered relevant for preventive patrol; these are usually called "outside" crimes or "suppressible" crimes. If the user wishes to take advantage of PCAM's capability to relate preventive patrol to crime rates, he must enter into the PCAM data base the total number of suppressible crimes expected to occur in each time block of each day of each precinct. If the user is not interested in such comparisons, he may enter nonsense data (anything greater

than zero) for the crime rates, and ignore the resulting output from PCAM.*

Unavailability Parameters

PCAM assumes that each patrol car will be busy on non-cfs work a certain fraction of the time it is on duty. This fraction may vary from shift to shift, but PCAM does not accept data on the unavailability fraction in each shift. This is because data showing non-cfs unavailabilities in the past may be a poor guide to what will happen in the future if allocations change. Instead, PCAM assumes that the fraction of time that a patrol car will spend unavailable on non-cfs work is given by the following equation:

$$\begin{pmatrix}\text{fraction of time}\\ \text{on non-cfs work}\end{pmatrix} = B1 \times \begin{pmatrix}\text{fraction of time}\\ \text{on cfs work}\end{pmatrix} + B2 \ .$$

The *unavailability parameters* B1 and B2 are constants that are entered separately for each precinct. They may not vary from day to day or tour to tour, unless separate data bases are constructed for each day or tour.

The equation shown above was found to be valid in a study of data from the Valley Bureau of the Los Angeles Police Department [1]. To calculate B1 and B2 for each precinct, the user must collect data showing the fraction of time spent on non-cfs work and the fraction of time spent on cfs work for a number of tours in the precinct, and then draw a straight line through a graph showing (fraction of time on non-cfs work) versus (fraction of time on cfs work). The parameter B2 is then the intercept of this line, and B1 is its slope (positive for upward slope, negative for downward). See Fig. 3. Each point shown on this graph represents data from one tour on one day. For example, in the AM tour on July 27, patrol cars in this example spent 58 percent of their time on non-cfs work and 24 percent on cfs work. The remaining 18 percent was spent on patrol.

---

*The user can also rewrite the FORTRAN subroutines that print out the calculations, so that the uninteresting numbers never appear in the printout. See Chapter I of the *Program Description* (R-1786/3).

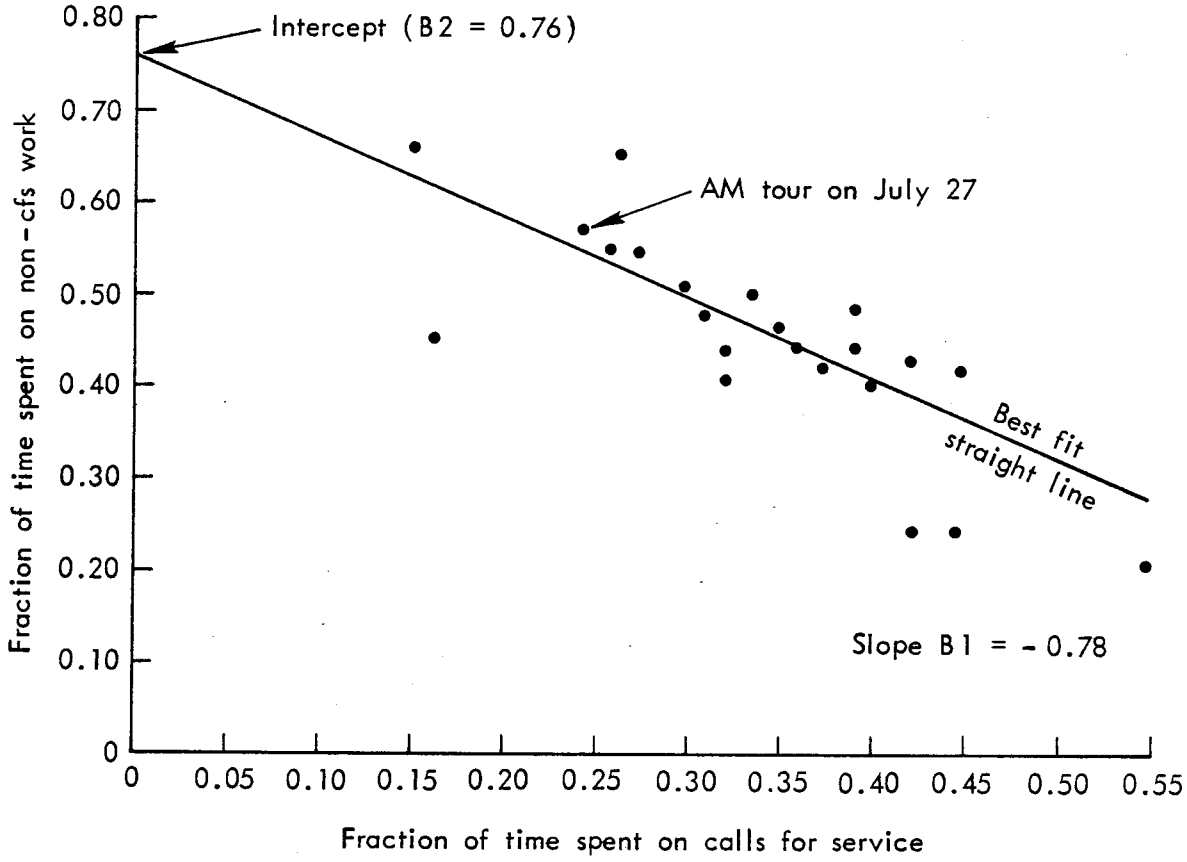


Fig. 3 — Example showing calculation of unavailability parameters

In the Los Angeles Police Department, the available data on non-cfs work were found to be unreliable, and therefore the amount of non-cfs work was estimated from data showing the actual fraction of calls that were delayed in each tour in each precinct. A computer program that calculates these estimates and then calculates B1 and B2 for each precinct (by a method called *regression analysis*) is shown in Appendix B of the companion *Program Description* (R-1786/3). *This is not a part of the PCAM program*, because many departments will collect data in a different form from that collected in Los Angeles. However, if a department is able to use this program to calculate the unavailability parameters B1 and B2, it will obtain a very good match between PCAM's estimates of calls delayed (see next chapter) and the actual experience of the department in regard to the fraction of calls delayed.

Calculating B1 and B2 may be the most difficult and controversial part of data preparation for PCAM. However, a variety of shortcuts may be taken. First, as mentioned above, the user may set B1 and B2 to zero and count non-cfs work in the call rate parameter and hourly call rate factors. This is not recommended but is preferable to ignoring non-cfs work entirely.

Second, the user can set B1 to zero, and enter for B2 the average fraction of time spent on non-cfs work (averaged over all tours in the precinct). This will cause PCAM to assume that the same fraction of a patrol car's time (namely, B2) is spent on non-cfs work in every tour. If the department has no data from which to estimate B2, we recommend trying several values of B2 in the range from 0.3 to 0.6 and having PCAM estimate queuing delays for the current allocation of patrol cars.* Whatever value of B2 results in a match between PCAM's estimates and the actual experience of the department will be a "good guess" for B2.

The user should note that if some precincts have especially high non-cfs unavailabilities, PCAM's prescriptive calculations will suggest allocating a large number of patrol cars to those precincts. Commanders from other precincts may then complain that the precincts in question are being "rewarded" for wasting much of the time of their patrol cars. For this reason, when operating PCAM in prescriptive mode, the department may prefer to enter values of B1 and B2 that reflect departmental standards for the maximum amount of non-cfs work permitted, rather than the values of B1 and B2 that fit the data for the precincts with large amounts of non-cfs work.

CALCULATION OF PERFORMANCE MEASURES

Queuing Delays

By making certain technical assumptions** it is possible to estimate the *fraction of calls* that will have to be placed in queue to await

---

*The SET command, described in Chapter III, will permit the user to try several values of B2.

**These are: (a) incidents occur according to a Poisson process, (b) all incidents have the same exponential distribution of service time, and (c) the system is in steady state in each hour. See

an available patrol car and the *average length of time* (in minutes) that calls in each priority level will have to wait in queue. The equations used in these calculations are shown in Appendix B.* Because the assumptions underlying the queuing model are least accurate for priority 1 calls, the average length of the queuing delay for priority 1 calls is not displayed. Despite the fact that the department may be most interested in the delays for priority 1 calls, we feel that the estimates provided by the model would not be accurate enough to be useful for planning purposes.

According to the queuing model used in PCAM, *all* calls, no matter what their priority, have the same probability of being placed in queue. However, high-priority calls wait in queue a shorter time than low-priority calls. PCAM calculates the average length of time spent in queue by all calls, *including* calls that do not wait at all. Thus, for example, if 70 percent of priority 2 calls are dispatched without delay, and the other 30 percent wait in queue for an average of 10 minutes, PCAM shows an average delay of 3 minutes for priority 2 calls.

PCAM calculates queuing statistics for each hour and then averages the hourly figures for each shift, weighting by the number of calls for service in each hour. This method has been found to be more accurate than calculations based on first averaging the call rates over a shift and then calculating queuing statistics.

Standard queuing models assume that the number of patrol cars is an integer, but, as we have seen, the effective number of cars may not be an integer but a fraction. PCAM handles this situation by calculating delays for the integers below and above the effective number of cars, and then interpolating between the two figures. This method is

---

Glossary for definitions. All patrol car allocation programs make the same assumptions.

However, it should be noted that the method mentioned above for estimating service times is based on the observation that different types of incidents have different service times, while the queuing model assumes all incidents are similar in this regard. This internal inconsistency is justified by the accuracy of the resulting calculations.

*Technically oriented readers may skip the rest of this chapter and substitute Appendix B.

only an approximation, but it yields accurate estimates of queuing delays.

Utilization

The *utilization* of a patrol car is defined to be the fraction of time it is busy on cfs work. Utilization is a measure of the workload of a car. As an example, if one patrol car has a utilization of .20, it is busy on cfs work 20 percent of the time. In an eight-hour tour this car will spend 1.6 hours on cfs work. Another patrol car with a utilization of .40 has twice as much cfs workload as the first car.

PCAM displays two different utilization statistics. The operationally interesting one is the *average utilization of an actual car*. This is defined by the equation:*

$$\begin{matrix}\text{AVG.}\\ \text{UTIL.}\\ \text{(ACT)}\end{matrix} = \frac{\text{expected number of car-hours of cfs work}}{\text{total number of car-hours fielded}} .$$

For example, consider an 8-hour shift in which 5 cars are fielded and 12 calls for service are expected. The total number of car-hours fielded is $5 \times 8 = 40$. If the average service time for a call for service is 45 minutes (= 3/4 hour), then there will be $12 \times 3/4 = 9$ car-hours of cfs work during the shift. The average utilization of an actual car is then $9/40 = 0.225$.

In this case the word "average" means that the utilization is averaged over all 5 cars. Some cars will spend more than 22.5 percent of their time on cfs work, others less; but the average is 22.5 percent. By comparing the averages for different tours in one precinct, it is possible to see what times of day have the most workload per car, and by comparing the averages for different precincts during a single tour, the imbalance in workload among different geographical areas can be determined. PCAM also displays averages over all the tours in a day for each precinct, or over all the precincts in the city for a single tour, depending on the commands entered by the user. The definition given above applies to all these averages.

---

*In this chapter, output measures, such as average utilization, appear on the left side of equations in the exact form in which they appear in the PCAM output, to facilitate recognition.

The second utilization figure displayed by PCAM is the *average utilization of an effective car*. This is defined by

$$\begin{matrix}\text{AVG.}\\ \text{UTIL.}\\ \text{(EFF)}\end{matrix} = \frac{\text{expected number of car-hours of cfs work}}{\text{total number of effective car-hours}} .$$

Continuing the 5-car example from above, if we suppose that patrol cars are busy on non-cfs work 20 percent of the time, then there are 4 effective cars. The number of effective car-hours during the shift is $4 \times 8 = 32$, and the average utilization of an effective car is $9/32 = 0.281$.

This utilization figure tells what fraction of *the time that could possibly be devoted to cfs work* is actually devoted to cfs work. It is displayed because PCAM's queuing calculations are based on the average utilization of an effective car, not on the average utilization of an actual car. If the average utilization of an effective car is near 1, which is the maximum permissible value, then queuing delays will be very long and little preventive patrol will be performed.* PCAM cannot perform its calculations at all if the user attempts to provide input data that result in the average utilization of an effective car being equal to 1 or larger in any hour. For this reason, the PCAM program includes error-checking capabilities that prevent this from happening.

In prescriptive mode, PCAM will permit the user to enter a constraint on the value of the average utilization of an effective car, so displaying these figures also serves the purpose of helping the user to identify sensible values for the constraint.

Another numerical example showing the difference between AVG. UTIL. (ACT) and AVG. UTIL. (EFF) will be discussed in the description of PCAM's output tables (p. 64).

Average Number of Cars Available

Sometimes during a tour the dispatcher may find that no cars are available for dispatch in the precinct when a call for service arrives;

---

*We cannot say what the maximum permissible value is for the average utilization of an *actual* car, as this will vary from shift to shift.

then the call must be placed in queue. At other times, there may be one car available, or two cars available, and so forth. If the dispatcher wrote down the number of cars available for dispatch each time a call arrived and averaged these numbers at the end of the shift, he would have the *average number of cars available.*[*]

By definition, a car is *available* if it is not doing cfs work or non-cfs work, so in any shift we have

$$\begin{pmatrix}\text{number of}\\ \text{car-hours}\\ \text{available}\end{pmatrix} = \begin{pmatrix}\text{number of}\\ \text{car-hours}\\ \text{fielded}\end{pmatrix} - \begin{pmatrix}\text{number of}\\ \text{car-hours}\\ \text{of cfs work}\end{pmatrix} - \begin{pmatrix}\text{number of}\\ \text{car-hours of}\\ \text{non-cfs work}\end{pmatrix}.$$

The average number of cars available can then be simply calculated as

$$\begin{matrix}\text{AVG. CARS}\\ \text{AVAIL.}\end{matrix} = \frac{\text{number of car-hours available}}{\text{number of hours in the shift}}.$$

This figure is of interest because it is equal to the *average number of car-hours of preventive patrol per hour* and can be used to compare the amount of preventive patrol at different times of the day or week in a single precinct.

Average Travel Time

A very simple relationship, called the *square-root law*, can be used to estimate how far, on the average, a patrol car will travel from its location at the moment of dispatch to the scene of the incident. If there are N patrol cars available, and the precinct has area A (in square miles), then the square-root law states that the average travel distance (in miles) is given by the equation

$$\begin{pmatrix}\text{average}\\ \text{travel}\\ \text{distance}\end{pmatrix} = (\text{constant}) \times \sqrt{\frac{A}{N}}.$$

[*] Alternatively, he could write down the number of cars available every minute, and average these at the end of the shift.

The constant is approximately equal to 2/3. This relationship was derived by mathematical modeling and has been validated using real and simulated data.

If the patrol cars respond at an average speed s (in miles per hour), then the average travel time (in hours) is

$$\begin{pmatrix}\text{average}\\ \text{travel}\\ \text{time}\end{pmatrix} = \frac{\text{average travel distance}}{s} .$$

The average travel time in minutes is then simply 60 times this figure.

Since the number of cars available (namely, the number N that appears in the equation) changes from time to time during a shift, the square-root law has to be modified slightly to take this into account. It has been found that a very good approximation is

$$\begin{matrix}\text{AVG.}\\ \text{TRAV.}\\ \text{TIME}\end{matrix} = 60 \times \frac{(\text{constant})}{(\text{response speed})} \times \sqrt{\frac{\text{area}}{\text{AVG. CARS AVAIL.}}} ,$$

if the average number of cars available is not too small. PCAM uses this relationship when the average number of cars available is at least 2.0, and makes an adjustment for smaller numbers of cars. Details are given in Appendix B.

Average Total Delay

The *average total delay* or *average response time* is simply the sum of the average queuing delay and the average travel time. It tells how long a caller will wait, on the average, from the time he contacts the police department until a patrol car arrives at the scene.*

Average Patrol Frequency

If patrol cars travel at speed s (miles per hour) while on preventive patrol, then in one hour the number of miles patrolled will equal

*The length of the telephone call and processing delays in the dispatch center are not included in the total delay.

s × (AVG. CARS AVAIL.). If this just happened to equal the number of street-miles in the precinct, then every point in the precinct could be passed by a patrol car once per hour. If the number of miles patrolled was half the number of street-miles in the precinct, a randomly selected point in the precinct would be passed once every two hours. In general, the average patrol frequency is

$$\begin{matrix}\text{AVG.}\\ \text{PATROL}\\ \text{FREQ.}\end{matrix} = \frac{s \times (\text{AVG. CARS AVAIL.})}{\text{number of street-miles}},$$

where s is the patrol speed.

The average patrol frequency tells how many times per hour a random point will be passed by a car on preventive patrol.* It is proportional to the probability that a crime will be intercepted by patrol officers and is useful for comparing the amount of preventive patrol among precincts having different sizes.

Preventive Patrol as Related to Crime Rates

PCAM displays two statistics that relate the amount of preventive patrol to crime rates. To utilize this capability of PCAM, the department must decide what categories of crimes it wants to compare with preventive patrol and enter the numbers into the data base, as mentioned above. The crimes selected by the department will be called "suppressible crimes" for the purpose of this discussion. In some departments, crime reports include an indication of whether the location of the crime could have been observed by a passing patrol car. If so, such crimes might reasonably constitute the "suppressible" category.

The first statistic, *patrol hours per suppressible crime*, is provided for departments that feel it is more important to compare preventive patrol to crime rates than to the size of a precinct. It is appropriate where departmental policy specifies that available patrol

*If the user enters "effective" street-miles into the data base instead of street-miles, the patrol frequency describes how many times per hour a patrol car passes the locations with the heaviest patrol coverage. See Chapter 4 of Ref. 20.

cars are not to engage in random patrol but are to undertake specific directed anticrime activities. It is defined by the equation

$$\text{PATROL HRS. PER SUPP. CR.} = \frac{\text{AVG. CARS AVAIL.}}{\text{avg. number of suppressible crimes per hour}} .$$

The second statistic is *average patrol frequency times suppressible crimes per hour*. It is proportional to the *number* of crimes that will be intercepted by patrol officers. In one figure it blends information about the amount of preventive patrol, the size of the precinct, and the crime rate of the precinct. For departments that view crime interception as the primary function of preventive patrol, this is the most appropriate statistic to use in comparing the level of preventive patrol across precincts or across times of day.

PRESCRIPTIVE CALCULATIONS

Meeting Constraints

For most of the performance measures described in the preceding section, PCAM will permit the user to specify a *constraint value*, i.e., to state that the department does not want the performance measure to be *higher* or *lower than* the specified value. A complete list of the measures subject to constraint appears in Chapter III in the description of the MEET command. In addition to specifying constraints on performance measures, the department can administratively decide that the number of cars on duty shall not be lower than a specified value.

All of the performance measures displayed by PCAM have the property that an increase in the number of cars on duty will lead to an improvement. (The improvement may be an increase or a decrease in the value of the measures; for example, patrol frequency increases and travel time decreases when more cars are added.) So PCAM can simply check whether the constraints are met for all the measures and, if not, increase the number of cars assigned by 1 and check again. The process stops when all constraints are met.

The user may specify that a constraint is to be met in all precincts during all tours of every day (i.e., in every shift), or just in certain precincts, or during certain tours, or in certain shifts. Whatever shifts are selected, PCAM assures that enough cars are on duty so that the average values of performance measures over each *time block* within the selected shifts meet the desired constraints. This is done because a time block is a period of time over which the number of patrol cars on duty does not change. Once PCAM has determined the number of patrol cars needed in each time block, it converts the results to shift allocations, using a calculation described in Chapter III of the companion *Program Description* (R-1786/3).

It may happen that a constraint is exceeded during particular hours, but PCAM will accept the allocation because the average over a time block is within the constraints. It may also happen (in the case of shifts containing two time blocks) that PCAM will allocate more cars than are absolutely needed to keep the average over the shift within the specified constraints. For example, suppose a shift is divided into two 4-hour blocks and the user specifies that the average number of cars available must be at least equal to 2. If one block currently has an average of 1.9 cars available and the other has 2.3, the user will see that the average for the whole shift is 2.1. Nonetheless, PCAM will add a car to this shift in order to meet the constraint. The user will not be surprised when this happens if he keeps in mind that PCAM is meeting constraints in every block.

If meeting constraints is the first prescriptive calculation requested by the user, PCAM begins by tentatively assigning to each specified time block the smallest whole number of patrol cars needed to make the utilization of an effective car under 1 in every hour. In other words, PCAM ignores the data indicating how many cars are currently assigned and instead determines an initial allocation that has just enough cars to handle the call-for-service workload. It then checks to see whether this allocation is adequate to meet all the constraints entered by the user. (It may happen that some constraints are met, but not all.) If one or more constraints are not met, it tentatively increases the number of patrol cars by 1 and checks again. This process continues until an allocation is achieved that meets all the constraints.

The last constraints to be met are identified as *limiting constraints*, meaning that all the other constraints were irrelevant in determining the number of cars needed. Limiting constraints are identified by asterisks in PCAM output for each shift. This permits the user to eliminate redundant constraints in future runs of the program or to identify which constraints must be relaxed if the resulting allocations turn out to be infeasible.

If some prescriptive calculation has been performed before the user asks PCAM to meet constraints, PCAM does not reinitialize the allocation to the minimum number of cars needed to handle the cfs workload. Instead, it checks to see whether the current allocation meets constraints and, if not, adds cars to only those shifts that need more. See Fig. 4.

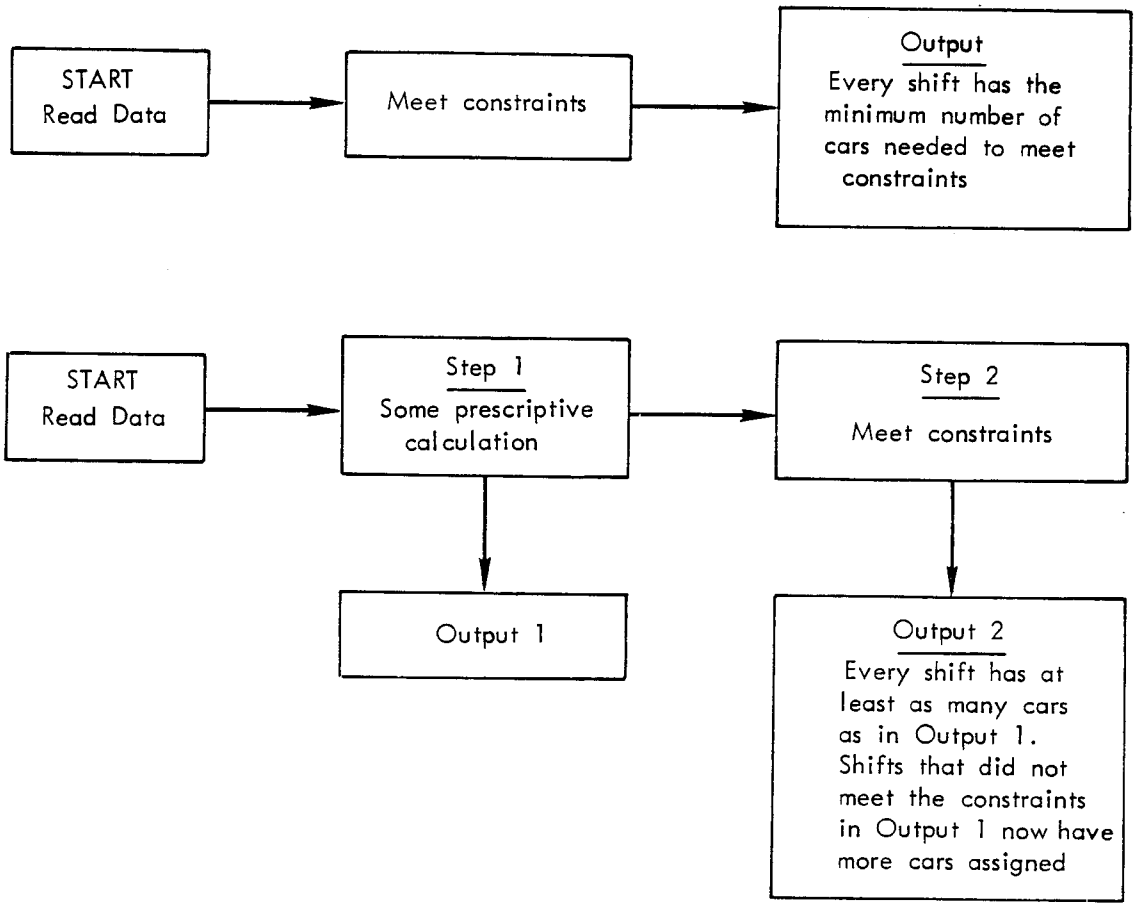


Fig. 4 — Two sequences for meeting constraints

Allocating a Specified Number of Car-Hours

The algorithm used by PCAM to allocate a specified number of car-hours over selected shifts is somewhat complicated in the case of overlay tours and is described in Appendix B. The algorithm is known to be optimal when there are no overlays and when there is an overlay tour having the same length as the two tours overlaid.* Here we shall only describe the situation where there are no overlays.

PCAM begins with either the current allocation or the minimal allocation needed to keep the utilization of an effective car under 1 in every hour, depending on instructions from the user. It then calculates the value of the objective function** for the chosen allocation.

Next, PCAM imagines that a single car is added to the first selected shift and remembers what the value of the objective function *would be* if this car were added. Then it imagines that a single car is added to the second selected shift, and so on for all the selected shifts.

Now PCAM compares the potential improvement in the objective function *per car-hour added*** among all the selected shifts and adds one car to the shift that is best in this regard. This changed allocation then becomes the new "current" allocation, and PCAM starts at the beginning to imagine one car being added to each selected shift. This process continues until all the car-hours specified by the user have been assigned to some shift. (A small number of car-hours may remain unassigned if they are not enough to equal one car working for one shift.)

This simple iterative procedure guarantees that the resulting allocation has the smallest value of the objective function that can possibly be achieved with the specified number of car-hours.

---

*"Optimal" means that the allocation found by PCAM has the smallest possible value of the performance measures that the user chooses to minimize. Although we do not know of any departments with overlay tours that are shorter or longer than the overlaid tours, PCAM permits such an arrangement. However, the algorithm for allocating car-hours has not been proved optimal in this case. Consult Appendix B for details.

**This is the performance measure that the user selects to minimize to get the "best" allocation.

***Some shifts may be longer than others.

Because we have adopted a special meaning for the word *shift*, this discussion may have obscured the fact that PCAM can allocate car-hours across precincts. Therefore, we specifically point out that the selected shifts could refer to a single tour in several precincts, in which case car-hours are allocated among the precincts for that tour. Or, the user could select *all* shifts, in which case the allocation is performed across tours and precincts simultaneously.

Allocating with Constraints

To allocate a specified number of car-hours so as to minimize an objective function *subject to* constraints on other performance measures, the user must enter two commands to PCAM. One of these meets constraints, and the other allocates whatever car-hours are left after the constraints are met. See Fig. 5. The calculations performed by PCAM in this two-step procedure have already been described above.

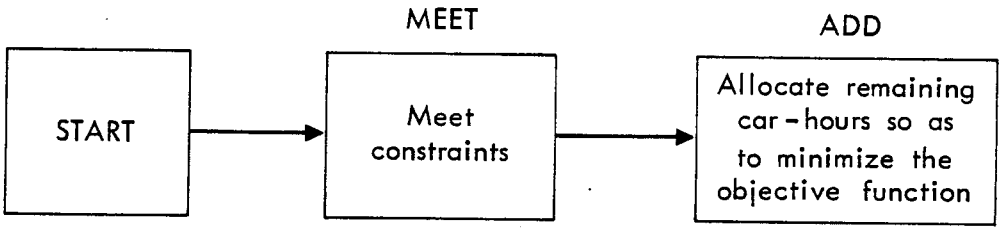


Fig. 5 — Two-step procedure for minimizing an objective function subject to constraints

III. USER'S GUIDE TO PCAM

This chapter provides instructions for entering PCAM commands and gives detailed descriptions of the format for each command and the actions taken by the program in response to each command. Appendix C summarizes command formats and lists valid specifications for objective functions, data items, and constraints. It provides, in a few pages, all the information the user needs to operate the program, once he is familiar with the contents of this chapter. If desired, the user can begin to operate the program after reading Appendix C, and then consult this chapter as questions arise.

OVERVIEW OF PROGRAM OPERATION

The Patrol Car Allocation Model (PCAM) is designed to run in either batch mode (where user input is on cards and program output is on a line printer) or interactive mode (where the device for both user input and program output is a teletype or similar slow-speed terminal). In either mode, the user controls the operation of the program with a sequence of commands, each of which instructs the program to perform a particular operation on the data.

When PCAM encounters a user command, it interprets the command and immediately carries out its implied action (we shall refer to this process as "command execution"). When PCAM completes the action for a command, it accepts the next command presented to it. This process continues until the program encounters a special command (END) which tells it that the user is finished with the program.

The kinds of actions that PCAM can perform fall into three general categories: data selection and modification, descriptive mode, and prescriptive mode. Once an initial selection of data is made, the user is free to enter commands in any order he chooses. He may have the data printed out if he wishes. If he wants to change some of the data, he may do so, after which he can have the data listed again or not as he chooses. He can ask PCAM to prescribe an allocation, and then he has a choice of displaying the resulting allocation or not. If he does not

like the allocation, he can start all over. Thus, in a typical session with PCAM, the user will proceed back and forth through the different categories of commands.

Data Selection and Modification

For the purposes of this discussion, we will use the name DATABASE to refer to the input file that the user must prepare as described in Chapter II. When operating the PCAM program, some or all of the data in DATABASE are read into the computer's memory, where they are available for the calculations that are performed by PCAM. This function is performed by the READ command. The version of the data base that resides in the computer's memory will be called CURRENT-DATA. See Fig. 6.

Many of the commands that the user enters into PCAM result in modifying CURRENT-DATA. For example, the user might add cars to certain shifts, or he might change the unavailability parameters to see what happens to performance measures. PCAM "remembers" all these modifications by updating CURRENT-DATA in accordance with the user's instructions. If, at any time, the user is dissatisfied with what he has done, or if he is finished with the calculation, he can start again by reading part of all of DATABASE into CURRENT-DATA. This causes PCAM to "forget" whatever was in CURRENT-DATA before and to begin with a fresh copy of CURRENT-DATA.

There are several reasons why the user might want to read only part of DATABASE into CURRENT-DATA. First, he may plan to do several different calculations, all of which refer to a single precinct, or all of which refer to the same tour. If he reads the entire DATABASE into CURRENT-DATA, then each time he types a command into PCAM he will have to specify the particular precinct or tour of interest to him. On the other hand, if he reads only the relevant part of DATABASE, then his typing is simplified because he does not have to make the same specifications in his later commands.

Second, the cost of operating PCAM increases according to the amount of core space set aside for CURRENT-DATA. If none of the users will want to use the entire data base at one time (for example, the users might be precinct commanders) the data processing personnel can install the program

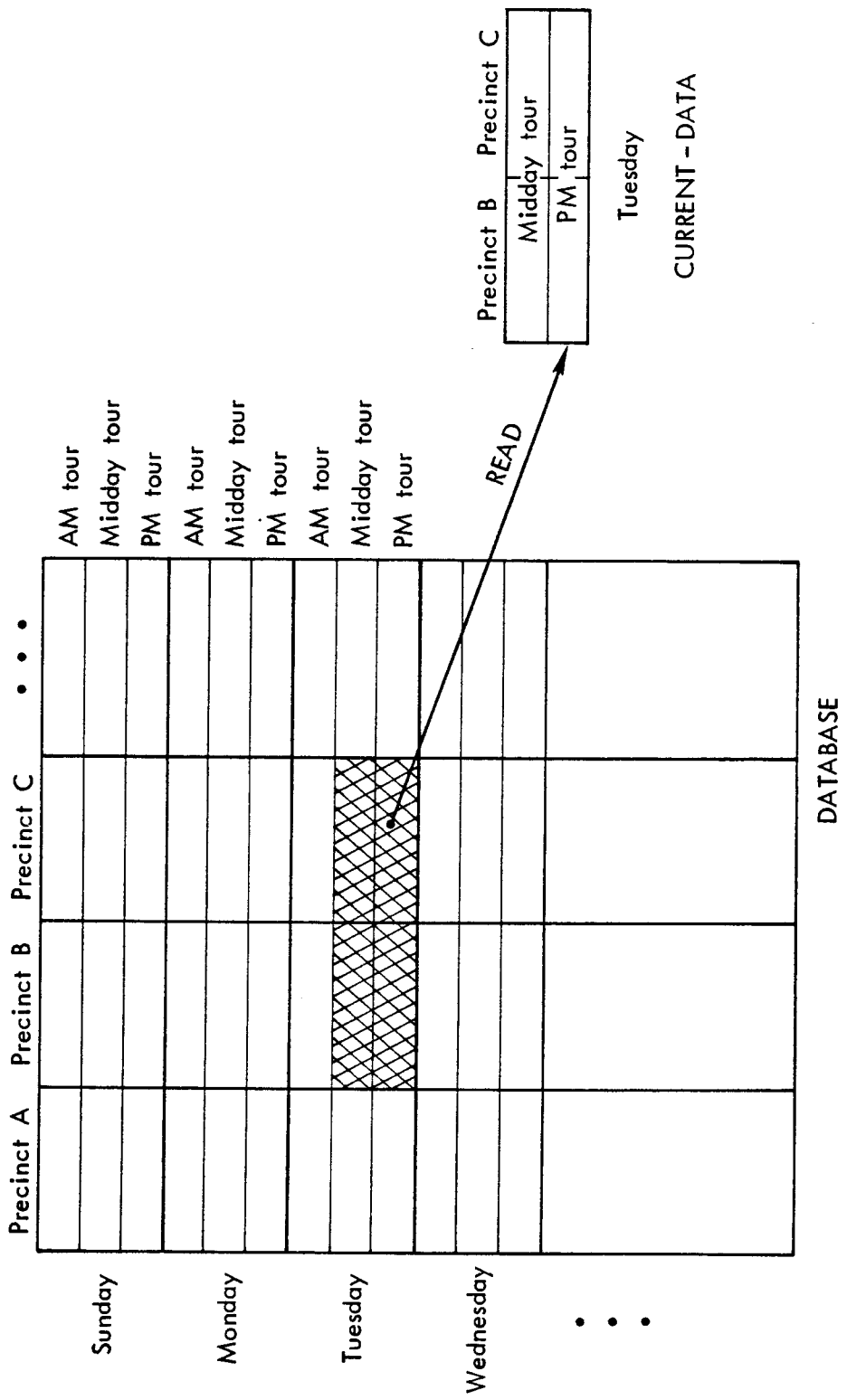


Fig. 6 — Conceptual example of a READ command

READ DATA FOR PRECINCT=(B,C),DAY=TUESDAY,TOUR=(MIDDAY,PM)

in such a way that the core space requested by PCAM is just enough for each user's needs. This will minimize computer costs while permitting each user to read the part of the data base he requires.

An important concept in PCAM is the *scope* of a command, which is the collection of precincts, days, and tours to which the action of the command applies. The scope of the READ command specifies what part of DATABASE is to be read into CURRENT-DATA. For example, the READ command shown in Fig. 6 would be entered into PCAM as follows:

READ DATA FOR PRECINCT=(B,C),DAY=TUESDAY,TOUR=(MIDDAY,PM) .

The part of the command that specifies its scope is called the *qualifier*. In the above example the qualifier is PRECINCT=(B,C),DAY =TUESDAY,TOUR=(MIDDAY,PM).* If PRECINCT or DAY or TOUR is not included in the qualifier of a READ command, then the missing categories are assumed to be unlimited. For example, the command

READ DATA FOR PRECINCT=B

will read the data related to Precinct B for all tours of all days included in DATABASE. The unqualified command "READ" will read all of DATABASE into CURRENT-DATA.

After a READ command has been issued, all the commands that follow until the next READ or END command can refer only to the data in CURRENT-DATA. Thus, any command that follows the command READ DATA FOR PRECINCT=B can only refer to Precinct B, and therefore it is unnecessary to mention Precinct B in its qualifier. Such a command could, however, have a qualifier referring to DAY and/or TOUR.

Once PCAM has completed execution of a READ command, the user may wish to modify the values of some of the data items in CURRENT-DATA.

---

*The words which may appear on the left side of an equal sign in a qualifier are DIVISION (or a substitute provided by the user), PRECINCT (or a substitute), TOUR (or a substitute), and DAY. For example, if a department calls its precincts AREAs, then AREA=(EAST,NORTH) would be a valid qualifier, but PRECINCT=(EAST,NORTH) would not be permitted.

This is done by using the SET command. With a SET command it is possible, for instance, to change the average response speed in a precinct for one tour of a day. If the SET command has no qualifier, it will change the values of data items of a particular type for all precincts, days, and tours included in CURRENT-DATA. Alternatively, by using qualifiers, a SET command causes a change in the values of data items for all tours of all days for a particular precinct or for all tours of a particular day for all precincts, etc.

If a SET command "overqualifies" the data elements for a data item, extraneous qualifications are ignored. For example, if a SET command indicates that the value of the call rate parameter for a particular precinct, day, and tour is to be changed, the tour specification is ignored, since call rate parameters apply only to days. This feature facilitates using one SET command to change several data item values that require different levels of qualification.

After the user has completed all his data modifications and prescriptive calculations (to be described below), he may wish to save a new data base that reflects all the changes that have been made. This is accomplished by means of the WRITE command. The WRITE command copies all or part of CURRENT-DATA into permanent storage in a file called NEW-DATA. See Fig. 7. Whereas the information in CURRENT-DATA is lost

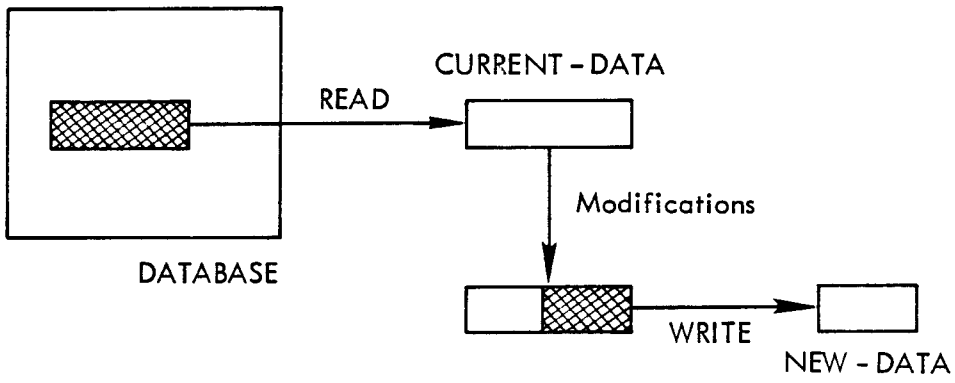


Fig. 7— Data selection capabilities of PCAM

whenever a READ or END command is issued, NEW-DATA is available for use as input in later runs of the PCAM program. However, the WRITE command has no effect on DATABASE, and all subsequent READ commands will refer to DATABASE until the session is terminated and data processing personnel change the file specifications for the PCAM program.

Descriptive Mode

The PCAM program provides facilities for displaying data items and various performance measures derived from data items. The execution of a LIST command causes printing of the values of the data items associated with all precincts, days, and tours within its scope. As with all other commands except READ, the scope of a LIST command can extend only to data included in CURRENT-DATA. The LIST command prints the data items for each tour, within each day, within each precinct in its scope. Some of the values shown for tours are derived by averaging values that can vary within a tour.

The execution of a DISP (display) command causes printing of various performance measures that PCAM calculates from the data provided for the precincts, days, and tours within its scope. These measures include average utilization of cars, average travel time to incidents, average patrol frequency, average number of cars available, fraction of calls delayed, and average delay for calls of different priorities.

At the user's option, DISP command output can be produced in order of tour within day within precinct or in order of precinct within tour within day. The difference between these two modes of output is explained in the detailed description of the DISP command, below. Except for summary statistics, both modes produce exactly the same information; the only difference is in the appearance of the output. Once the user has seen illustrations of both modes of output, he will not have any difficulty in deciding which he prefers for a particular application.

If the number of cars assigned to a precinct for a particular day and tour results from meeting a constraint on an output measure (see next page), then the constraining measure for the precinct, day, and tour is flagged with an asterisk in the output listing.

Prescriptive Mode

PCAM's capabilities to allocate car-hours among tours and precincts are implemented by three commands: MEET, ALOC, and ADD. The MEET command causes PCAM to calculate the minimum number of car-hours needed in every shift within its scope to meet constraints established by the user. The ALOC command allocates a specified total number of car-hours among the shifts within its scope so as to minimize a measure of performance selected by the user. The ADD command permits the user to allocate car-hours in addition to those already assigned in CURRENT-DATA.

The MEET command operates in either of two modes, depending upon the commands which have preceded it. If a MEET command is the first prescriptive command after a READ command, then command execution begins by assigning exactly enough cars to keep the effective utilization under 1 in each precinct for each hour of each tour of each day within the scope of the command. Then additional cars are assigned, as necessary, to meet constraints. In other words, a MEET command that follows a READ command ignores all information contained in CURRENT-DATA about the number of cars assigned to shifts. However, if some other prescriptive command (ALOC, ADD, or MEET) has been executed between the last READ command and the current MEET command, then command execution consists only of adding cars to shifts within the command scope where the number already assigned in CURRENT-DATA are not adequate to meet the constraints. This was shown in Fig. 4 (p. 38).

Execution of the ALOC command always begins by assigning just enough cars to all shifts within the command scope to keep effective utilization under 1. Any remaining car-hours are then allocated among shifts in such a way as to minimize the selected objective function. Execution of an ALOC command causes PCAM to ignore the assignments included in CURRENT-DATA, and all the assignments for shifts in the scope of the ALOC command are replaced by new values in CURRENT-DATA.

The ADD command will allocate a specified number of car-hours to the shifts within its scope over and above the number already assigned in CURRENT-DATA. Cars are allocated to shifts so that the average value of a selected objective function is minimized. Any constraints that have been set previously will still be met after execution of an ADD

command. Thus, ADD differs from ALOC in that the allocation begins with the number of cars asigned to each shift in CURRENT-DATA just prior to execution of the ADD command.

If the user wishes to allocate a certain total number of car-hours so that his selected objective function is minimized, *subject to certain constraints*, he must execute two commands (as shown in Fig. 5 on p. 40). First, a MEET command is entered, and PCAM indicates the total number of car-hours needed to meet the constraints. Next, an ADD command is entered to allocate the remaining car-hours. The user can subtract the number already allocated from his desired total and enter the difference in the ADD command, or he can have PCAM perform the subtraction for him.

ENTERING COMMANDS

When PCAM is run in the interactive mode, commands can be entered only in response to the message "COMMAND?" typed by the program. In batch mode, PCAM reads cards containing commands, one command at a time. In either mode, the action specified by a command is carried out as soon as the text of the command has been read and interpreted.

All commands consist of an *identifier* (three, four, or five letters), followed, in most cases, by required or optional additional information, called *parameters*. When entering commands, the identifier must be entered first, and at least one space must be entered between the identifier and any parameters. The allowed identifiers (together with their meanings) are as follows:

READ	(read data)
LIST	(list input data)
DISP	(display performance measures)
ALOC	(allocate a specified number of car-hours)
ADD	(add a specified number of car-hours)
MEET	(meet constraints)
SET	(set data values to new values)
WRITE	(write a new data file)
END	(terminate operation of program)

Commands can consist of as many lines of input as are required to specify them completely. Each line of a command, except the last, must

end with an ampersand ("&"), which indicates that the command is continued on the following line. Ampersands cannot appear within numbers or words (see below).

Except for letters, digits, parentheses, asterisks, hyphens, periods, and ampersands, all characters and blanks are treated as *delimiters*. At least one delimiter must be entered whenever blanks appear in the command formats below. This means that blanks, commas, colons, semicolons, equal signs, etc., can be used freely to improve readability when entering commands.

COMMAND FORMAT CONVENTIONS

In describing the format of a PCAM command, we will use the following conventions:

- Any text in uppercase letters that is not contained in square brackets must be entered exactly as shown. For example, one must enter "LIST" (without quotation marks) to tell PCAM to list data item values.
- Any text in uppercase letters set off by square brackets ([ ]) is optional. That is, it may be entered as shown *without the brackets* or may be completely omitted. For example, [BY] means that one may enter the characters "BY" if one wishes, but they can also be omitted.
- Text in angle brackets (⟨ ⟩) indicates that one must replace the angle brackets and the text with a valid member of the class of items indicated by the text. For example, wherever ⟨NUMBER⟩ appears in a command format description, the user must substitute a number, such as 1.3. The members of the specified classes make up the PCAM program vocabulary and are completely defined below.
- A *character string* is defined to be a sequence of characters made up of any combination of letters, numbers, hyphens, or periods with no imbedded blanks. Of course, certain character strings have specific meanings to the PCAM program, and these are explained below.

Filler Words

To improve the readability of commands, especially for those who are not familiar with the details of using the program, PCAM allows certain character strings, called *filler words*, to be interspersed among command parameters. These words are completely ignored by PCAM and may be used in any combination desired. Suggested uses of these words are indicated by their appearance in square brackets [ ] in the command definitions below. For example, the definition of the PCAM command that causes the program to list the user's input data is: LIST [DATA] [FOR] ⟨QUALIFIER⟩. When the LIST command is entered, the filler words DATA and FOR may be included or not, as the user wishes. A complete list of filler words is provided in the next section.

PROGRAM VOCABULARY

The PCAM program vocabulary consists of all of the character strings that can be used to enter PCAM commands. These character strings fall into various classes, which are identified by angle brackets in the definitions that follow.

1. ⟨FILLER⟩ is any one of the following character strings: BY, CAR, CARS, DATA, FOR, HOUR, HOURS, ON, TO. A ⟨FILLER⟩ can appear in any position in any command and is always ignored.
2. ⟨KEYWORD⟩ is any one of the following character strings:
   a. DAY, P, C, T, and F. DAY is used to refer to days of the week, P is a label for data elements that can be changed by the user, C is a label for constraints, T is a label for tables to be displayed, and F is a label for objective functions.
   b. The command identifiers listed above: ADD, ALOC, DISP, END, LIST, MEET, READ, SET, WRITE.
   c. All ⟨FILLER⟩s.
   d. Three names that are provided by the user and will be referred to in this report as TOUR, DIVISION, and PRECINCT. PRECINCT refers to the smallest geographical

region to which allocations may be made. A DIVISION is an aggregation of precincts, and may be defined to suit the needs of the user. (If the department has no geographical commands that encompass several precincts, then it can define all precincts to belong to a single division.) TOUR refers to a span of hours that is the same for every day of the week. For example, the PM tour could be the hours from 1200 to 2000 every day of the week.

Except for the ⟨FILLER⟩s, all ⟨KEYWORD⟩s have specific meanings in the PCAM command language and may not be used as ⟨NAME⟩s (see below) except when they appear in a ⟨NAMELIST⟩ enclosed in parentheses. For example, the user cannot have a "day" tour unless the tour name DAY always appears in parentheses as in TOUR=(DAY). The construction TOUR=DAY is invalid.

3. ⟨NUMBER⟩ is any decimal number, positive or negative, with or without a decimal point; e.g., 23 or -5.21.

4. ⟨NUMBERLIST⟩ is a ⟨NUMBER⟩ enclosed in parentheses or preceded by a delimiter, or a list of ⟨NUMBER⟩s enclosed in parentheses and separated by commas; e.g., (1121) or (.023,1.1,50).

5. ⟨NAME⟩ is a string of one to eight characters that is not a ⟨KEYWORD⟩. The first character of a ⟨NAME⟩ must be a letter. The remaining characters can be letters, numbers, hyphens, or periods with no imbedded blanks. ⟨NAME⟩s are used to identify days, tours, precincts, and divisions.

6. ⟨NAMELIST⟩ is a ⟨NAME⟩, or a list of ⟨NAME⟩s enclosed in parentheses and separated by commas; e.g., PCT1 or (MORNING, AFTRNOON, NIGHT, OVERLAY). It is possible for a ⟨NAMELIST⟩ to consist of only left and right parentheses. This construction can be useful in a DISP command ⟨QUALIFIER⟩ (see below).

7. ⟨QUALIFIER⟩ consists of any subset of the following phrase types, separated by delimiters: "TOUR=⟨NAMELIST⟩", "DAY=⟨NAMELIST⟩", "DIVISION=⟨NAMELIST⟩", "PRECINCT=⟨NAMELIST⟩". A ⟨QUALIFIER⟩ defines the scope of the command with which it is

associated (see "Overview of Program Operation," above). Examples of valid ⟨QUALIFIER⟩s are TOUR=FIRST and PRECINCT =(P1,P2),DAY=SUNDAY. The uses of qualifiers are explained in the command definitions that follow.

COMMAND DEFINITIONS

READ Command

READ [DATA] [FOR] ⟨QUALIFIER⟩

Action: The READ command causes PCAM to read all or part of the information contained in DATABASE, making this information available for use in CURRENT-DATA. The ⟨QUALIFIER⟩ specifies the precincts, days, and tours for which data are to be read. Any previously read data are replaced.

Effect of using qualifiers: The PRECINCT=⟨NAMELIST⟩ phrase of the ⟨QUALIFIER⟩ provides a list of precincts for which data will be read. Each element of ⟨NAMELIST⟩ must be the name of some precinct that the user has entered into the data base. The DIVISION=⟨NAMELIST⟩ phrase tells PCAM to read data for all precincts of each division named in ⟨NAMELIST⟩. If both the PRECINCT and DIVISION phrases are included in the ⟨QUALIFIER⟩, then data will be read for all precincts in all specified divisions and for all other explicitly requested precincts.

As an example, consider two divisions named D1 and D2. D1 consists of precincts P1 and P2, while D2 consists of precincts P3 and P4. The qualifier DIVISION=(D1,D2) refers to precincts P1, P2, P3, and P4; the qualifier DIVISION=D2 refers only to precincts P3 and P4; the qualifier PRECINCT=(P1,P3) refers to precincts P1 and P3; and the qualifier DIVISION=D1,PRECINCT=P4 refers to precincts P1, P2, and P4. If both the PRECINCT and DIVISION phrases are omitted from the qualifier, then data are read for all precincts in DATABASE.

The DAY=⟨NAMELIST⟩ phrase of the qualifier provides PCAM with a list of days for which it will read data. Each element of the ⟨NAMELIST⟩ must be the name of a day entered into the data base by the user. Data are read for the same days for all precincts specified by the PRECINCT and DIVISION phrases. If the DAY phrase is omitted from the qualifier, PCAM will read data for all days in the data base for the specified precincts.

Continuing the above example with 4 precincts and 2 divisions, assume that the data base contains data for days named MONDAY, TUESDAY, WEDNSDAY,* THURSDAY, FRIDAY, SATURDAY, and SUNDAY. Then the qualifier PRECINCT=P1,DAY=SUNDAY refers to the data for SUNDAY in precinct P1; the qualifier PRECINCT=P2 refers to the data for all days of the week in precinct P2; the qualifier DAY=THURSDAY refers to the data for THURSDAY for precincts P1, P2, P3, and P4; and the qualifier DIVISION=D2,DAY=(TUESDAY,WEDNSDAY) refers to the data for TUESDAY and WEDNSDAY for precincts P3 and P4.

The TOUR=⟨NAMELIST⟩ phrase of the qualifier specifies the tours for which PCAM will read data. Data are read for the same tours for all days specified in the DAY phrase. If the TOUR phrase is omitted from the qualifier, then data are read for all tours in the data base for the days and precincts specified in the qualifier.

Continuing with the same example, assume each day is divided into three tours, named FIRST, SECOND, and THIRD. Then the qualifier TOUR=SECOND,DAY=FRIDAY refers to the SECOND tour of FRIDAY for precincts P1, P2, P3, and P4; the qualifier PRECINCT=P3,DAY=TUESDAY refers to the FIRST, SECOND, and THIRD tours of TUESDAY for precinct P3; and the qualifier TOUR=(FIRST,THIRD), DIVISION=D2,DAY =(SATURDAY,SUNDAY) refers to FIRST and THIRD tours of SATURDAY and SUNDAY for precincts P3 and P4.

READ command qualifiers also have an effect on the order in which output measures are printed by the DISP command. This is fully discussed below in the section on the DISP command.

---

*Recall that the names of days are limited to eight characters.

Data check: When the READ command is entered, PCAM does more than simply read the data from the input file; it also checks to determine whether enough cars have been assigned to each tour so that the number of cars on duty in each hour is enough to make the effective utilization under 1 in every hour. If this condition is not met, PCAM will not be able to perform its calculations correctly, so the READ command increases the number of cars assigned.

When PCAM increases the number of cars assigned to a shift, it prints a message giving the precinct, day, and tour that specify the shift, and it indicates the number of cars it has assigned. When the user receives such a message, it is a warning for him to check the accuracy of his data.

Reading overlay tours: If the data base contains data for overlay tours, the user has the option of reading the overlay tour data or not. If the user chooses not to read overlay tour data, then all tours that are read are treated as though no overlay tour existed. However, if the user selects the overlay tour in a READ command, then he must also select both tours that are overlaid.

Example: We now give an example showing the use of the READ command. Both the user's input and the program's output are reproduced from actual program runs. The demonstration data used in the examples are realistic, representing a section of a large city. However, some numerical values and all names of precincts and divisions have been changed in the demonstration data base. A listing of this data base is given in Appendix A of the companion *Program Description* (R-1786/3).

Our sample city, which will be used to illustrate all of the commands, has two divisions: HIGHLAND and LOWLAND. HIGHLAND is made up of two precincts: NORTH and EAST. Lowland has three precincts: SOUTH, WEST, and CENTRAL. There are data for seven days for each precinct. The day names are: SUN-MON, MON-TUE, TUE-WED, WED-THU, THU-FRI, FRI-SAT, and SAT-SUN. The data for the first day begins with the hour 0800-0900 on Sunday and ends with the hour 0700-0800 on Monday. The other days are defined similarly. The

selection of days that do not begin at midnight was done to accommodate an overlay tour, which PCAM requires to start and end in the same day (see Chapter II). Each day has four tours: MIDDAY, PM, AM, and FOURTH. Each tour is eight hours in length, and the FOURTH is an overlay tour, beginning with the fourth hour of the PM tour and ending with the third hour of the AM tour.

A sample READ command and response are shown in Fig. 8.

```
COMMAND? READ DATA FOR PRECINCT=(WEST,CENTRAL),&
    DAY=(TUE-WED,SAT-SUN)

*** 15. CARS NEEDED IN PRECINCT WEST     FOR TOUR     MIDDAY    ON DAY TUE-WED
*** 13. CARS NEEDED IN PRECINCT WEST     FOR TOUR     PM        ON DAY TUE-WED
*** 14. CARS NEEDED IN PRECINCT WEST     FOR TOUR     MIDDAY    ON DAY SAT-SUN
*** 13. CARS NEEDED IN PRECINCT WEST     FOR TOUR     PM        ON DAY SAT-SUN
*** 11. CARS NEEDED IN PRECINCT WEST     FOR TOUR     FOURTH    ON DAY SAT-SUN
***  8. CARS NEEDED IN PRECINCT CENTRAL  FOR TOUR     PM        ON DAY SAT-SUN
***  6. CARS NEEDED IN PRECINCT CENTRAL  FOR TOUR     AM        ON DAY SAT-SUN
```

Fig.8 — Sample READ command and response

This command caused PCAM to read data for all tours of TUE-WED and SAT-SUN in precincts WEST and CENTRAL. The messages preceded by three asterisks indicate that the number of cars listed in DATABASE for seven of the shifts was not sufficient to keep the effective utilization under 1 in every hour. The program computed the minimum number of cars needed for each shift and assigned that number to the shift. The modified car assignments (which are rounded before being displayed) are entered into CURRENT-DATA and will be reflected in subsequent calculations and output.

LIST Command

LIST [DATA] [FOR] ⟨QUALIFIER⟩

Action: Causes PCAM to list most of the data in CURRENT-DATA for the shifts specified by the qualifier. Data are displayed for each tour, within each day, within each precinct. The values of time-invariant characteristics of each precinct are displayed first, followed by the values that apply to each day as a whole. For each day, values are displayed for each tour selected.

A sample LIST command and response are shown in Fig. 9.

```
COMMAND? LIST DATA FOR DAY=TUE-WED

PRECINCT: WEST     ; AREA= 51.9; STREET MILES=678.5; B2=0.637; B1=-.476

  DAY: TUE-WED ; CALL RATE PARM= 6.65; SERVICE TIME PARM=45.13

                AVG.                                 AVG.  AVG.  FRAC.  FRAC.  FRAC.
              ACT.  EFF.  RSP.  PTL.  AVG.  AVG.  FRAC.  FRAC.  FRAC.
TOUR          CARS  CARS  VEL.  VEL.  SERV  CALL  OF P1  OF P2  OF P3
                                       TIME  RATE  CALLS  CALLS  CALLS
MIDDAY        15.0   7.3  15.0   7.5  36.1   6.4  0.061  0.810  0.129
PM            13.0   9.4  15.0   7.5  49.6   9.1  0.061  0.810  0.129
AM             9.1   5.3  25.0   7.5  49.6   3.4  0.061  0.810  0.129
FOURTH         4.9

PRECINCT: CENTRAL ; AREA= 47.3; STREET MILES=498.3; B2=0.658; B1=-.746

  DAY: TUE-WED ; CALL RATE PARM= 3.66; SERVICE TIME PARM=40.20

                AVG.                                 AVG.  AVG.  FRAC.  FRAC.  FRAC.
              ACT.  EFF.  RSP.  PTL.  AVG.  AVG.  FRAC.  FRAC.  FRAC.
TOUR          CARS  CARS  VEL.  VEL.  SERV  CALL  OF P1  OF P2  OF P3
                                       TIME  RATE  CALLS  CALLS  CALLS
MIDDAY         7.6   4.2  15.0   7.5  44.2   2.9  0.067  0.810  0.123
PM             5.8   4.8  15.0   7.5  36.2   4.6  0.067  0.810  0.123
AM             4.3   2.8  25.0   7.5  36.2   2.0  0.067  0.810  0.123
FOURTH         3.5
```

Fig. 9 — Sample LIST command and response

This command requests PCAM to display data for all TUE-WED tours in CURRENT-DATA. Since the command was issued after the READ command in Fig. 8, there are only two precincts in CURRENT-DATA, and these are the only ones shown in the output.

Explanation of output: The first line of output gives the precinct name (WEST), its area in square miles (51.9), and the number of miles of streets in the precinct (678.5). It also shows the values of the unavailability parameters B1 and B2 (see Chapter II). These items are identical to whatever appears in CURRENT-DATA for the West precinct.

The second line of output gives the name of the first day for which data are listed (TUE-WED) and also the call rate (6.65) and service time (45.13) parameters for that day. These parameters characterize the call rate and service time for the day as a whole, although PCAM does its computations using hourly data for call rates and service times. These parameters can be changed to reflect overall changes in call rates and service times, and proportional changes will be made in the hourly data that PCAM uses (see Chapter II and the SET command description below). The items on the second line of printout are also identical to whatever appears in CURRENT-DATA.

The next three lines of output produce column headings to identify the data to be displayed for each of the tours. From left to right, the first column heading identifies the tour name (MIDDAY for the first tour). The second column heading identifies the number of actual cars* assigned to start at the beginning of the tour (15.0 for the MIDDAY tour--this is the value set by the last READ command). The third column gives the average number of effective cars (see Chapter II) on duty during the tour (7.3 for the MIDDAY tour and 9.4 for the PM tour). This figure refers to the entire eight-hour period covered by the tour and takes into account any change in the number of cars on duty during a tour due to an overlay tour. For example, during the first three hours of the PM tour there are 13 cars on duty, and during the next five hours there are 17.9 cars on duty (13.0 + 4.9 from the fourth tour). When converted to effective cars and averaged over eight hours, this results in 9.4 effective cars during the PM tour.

The fourth column gives the average speed** of cars when responding to calls, in miles per hour (e.g., 15.0 for the MIDDAY tour). The fifth column gives the average speed of cars while on preventive patrol (e.g., 7.5 for the MIDDAY tour). The response and patrol speeds displayed are taken directly from CURRENT-DATA.

The sixth and seventh columns show the averages of hourly service times and call rates over the tours (these are 36.1 minutes and

---

*Input data do not have to be integers. They can be constructed by averaging the number of actual cars over several weeks.

**VEL is an abbreviation for *velocity*, which means *speed* here.

6.4 calls per hour, respectively, for the MIDDAY tour). These averages will change in direct proportion to any changes the user might make in the call rate and service time parameters for the day.

The next three columns show the fractions of all calls in a tour that are of priority 1, priority 2, and priority 3, respectively (these are 0.061, 0.810, and 0.129, for the MIDDAY tour). The fractions of priority 1 and priority 2 calls were provided in DATABASE and read into CURRENT-DATA. PCAM computes the fraction of priority 3 calls by subtracting the sum of the priority 1 and priority 2 fractions from 1.0.

Note that only the number of actual cars on duty is shown for the FOURTH tour. That is because it is an overlay tour, and the information already listed for the other three tours completely describes the entire day.

The types of output described above are repeated for all days selected and for all selected precincts.

DISP Command

DISP T⟨NUMBERLIST⟩ [FOR] ⟨QUALIFIER⟩

Action: The DISP command instructs PCAM to print the groups of output measures specified by ⟨NUMBERLIST⟩ for the shifts specified by the qualifier. The permitted choices are T(1) (or T 1), T(2), T(1,2) T(2,1).

Table 1: If the number 1 (one) appears in the ⟨NUMBERLIST⟩, the following output measures will be produced for each shift:

- Average utilization of an effective car
- Average utilization of an actual car
- Average travel time to calls for service
- Patrol hours per suppressible crime
- Average patrol frequency
- Average patrol frequency times suppressible crimes per hour, and
- Average number of cars available for dispatch.

Table 2: If the number 2 (two) appears in the ⟨NUMBERLIST⟩, then the output measures produced for each shift will be:

- The number of actual cars assigned to start the shift
- The number of car-hours in the shift
- Call rates and service times, averaged over the hours of the shift
- The probability that a call will be placed in queue (i.e., not dispatched immediately)
- The expected queuing delays (in minutes) for priority 2 and priority 3 calls, and
- The expected total delay for any call (this is queuing delay plus travel time).

The interpretations of these measures are explained in Chapter II.

We will refer to the two groups of output measures as output tables. Either one or both tables may be specified in a DISP command. If both are specified, output is printed for all shifts selected by the qualifier for the first table in the list, and then for all shifts for the second table in the list.

Order of output: The output tables can be printed in either of two ways. One way will begin with the first precinct selected and show output measures for each tour in turn, producing summaries for each day and then for the precinct as a whole. After all output is given for the first precinct, the results will be shown for the next precinct (if any). This output mode is called "tour within day within precinct." It is appropriate for looking at output measures as they vary from tour to tour within a particular day for a particular precinct. An example of this output order is given in Fig. 10. (The interpretation of this table will be given below.)

The second output order will begin with the first tour of the first day selected and show output measures for all the precincts, producing summaries for the tour. Then it proceeds to the next tour. This mode is called "precinct within tour within day." It is more appropriate for seeing how output measures vary from

COMMAND? DISP T 1

PRECINCT: WEST ; DAY: TUE-WED

TOUR	AVG. UTIL. (EFF)	AVG. UTIL. (ACT)	AVG. TRAV. TIME	PATROL HRS PER SUPP CR	AVG. PATROL FREQ.	AV PTL FREQ TIMES SUPP CR PER HR	AVG CARS AVAIL.
MIDDAY	.530	.257	14.2	6.08	0.04	0.021	3.42
*PM	.798	.467	16.9	2.62	0.02	0.015	1.90
*AM	.533	.259	10.6	6.03	0.03	0.011	2.49
+FOURTH	.707	.387	14.0	4.91	0.03	0.016	2.64
AVERAGE	.645	.339	14.9	4.59	0.03	0.017	2.60

PRECINCT: WEST ; DAY: SAT-SUN

TOUR	AVG. UTIL. (EFF)	AVG. UTIL. (ACT)	AVG. TRAV. TIME	PATROL HRS PER SUPP CR	AVG. PATROL FREQ.	AV PTL FREQ TIMES SUPP CR PER HR	AVG CARS AVAIL.
MIDDAY	.644	.337	15.0	8.35	0.03	0.009	2.61
*PM	.615	.316	12.8	4.68	0.04	0.040	3.86
*AM	.661	.350	9.9	4.98	0.02	0.009	2.05
+FOURTH	.630	.327	10.2	4.98	0.05	0.041	4.11
AVERAGE	.636	.331	12.6	5.50	0.03	0.017	2.84

PRECINCT: WEST

AVERAGE	.640	.334	13.6	5.03	0.03	0.017	2.72

PRECINCT: CENTRAL ; DAY: TUE-WED

TOUR	AVG. UTIL. (EFF)	AVG. UTIL. (ACT)	AVG. TRAV. TIME	PATROL HRS PER SUPP CR	AVG. PATROL FREQ.	AV PTL FREQ TIMES SUPP CR PER HR	AVG CARS AVAIL.
MIDDAY	.510	.281	16.8	7.15	0.03	0.009	2.06
*PM	.577	.346	16.2	5.80	0.03	0.010	2.03
*AM	.432	.218	10.7	21.44	0.02	0.002	1.61
+FOURTH	.453	.234	13.1	10.97	0.04	0.009	2.47
AVERAGE	.518	.289	15.2	7.99	0.03	0.007	1.90

PRECINCT: CENTRAL ; DAY: SAT-SUN

TOUR	AVG. UTIL. (EFF)	AVG. UTIL. (ACT)	AVG. TRAV. TIME	PATROL HRS PER SUPP CR	AVG. PATROL FREQ.	AV PTL FREQ TIMES SUPP CR PER HR	AVG CARS AVAIL.
MIDDAY	.595	.366	18.4	7.31	0.02	0.006	1.64
*PM	.601	.372	14.9	6.34	0.04	0.016	2.61
*AM	.553	.322	10.0	4.48	0.03	0.012	1.96
+FOURTH	.612	.385	12.4	5.83	0.04	0.018	2.77
AVERAGE	.585	.355	14.4	5.79	0.03	0.010	2.07

PRECINCT: CENTRAL

AVERAGE	.556	.325	14.8	6.66	0.03	0.009	1.99

GRAND

AVERAGE	.609	.331	14.0	5.61	0.03	0.013	2.35

Fig. 10 — Sample DISP command for Table 1 ordered by tour within day within precinct

precinct to precinct over the city during a particular time of a particular day. An example of output ordered by precinct within tour within day is given in Fig. 11.

```
COMMAND? READ DATA FOR DAY=TUE-WED, TOUR=(MIDDAY,PM)
*** 14. CARS NEEDED IN PRECINCT EAST     FOR TOUR   PM       ON DAY TUE-WED
*** 15. CARS NEEDED IN PRECINCT WEST     FOR TOUR   MIDDAY   ON DAY TUE-WED
*** 15. CARS NEEDED IN PRECINCT WEST     FOR TOUR   PM       ON DAY TUE-WED
*** 11. CARS NEEDED IN PRECINCT SOUTH    FOR TOUR   PM       ON DAY TUE-WED
***  7. CARS NEEDED IN PRECINCT CENTRAL  FOR TOUR   PM       ON DAY TUE-WED
COMMAND? DISP T 1
DAY TUE-WED ; TOUR     MIDDAY

            AVG.   AVG.   AVG.   PATROL    AVG.    AV PTL FREQ
            UTIL.  UTIL.  TRAV.  HRS PER   PATROL  TIMES SUPP   AVG CARS
 PRECINCT   (EFF)  (ACT)  TIME   SUPP CR   FREQ.   CR PER HR    AVAIL.
 EAST       .620   .409   10.6     7.69    0.05      0.022       3.36
 WEST       .530   .257   14.2     6.08    0.04      0.021       3.42
 NORTH      .598   .287    9.8     7.35    0.04      0.013       2.30
 SOUTH      .608   .312   17.2     5.06    0.03      0.017       2.53
 CENTRAL    .510   .281   16.8     7.15    0.03      0.009       2.06

 AVERAGE    .579   .311   13.3     6.51    0.04      0.017       2.73

DAY TUE-WED ; TOUR     PM

            AVG.   AVG.   AVG.   PATROL    AVG.    AV PTL FREQ
            UTIL.  UTIL.  TRAV.  HRS PER   PATROL  TIMES SUPP   AVG CARS
 PRECINCT   (EFF)  (ACT)  TIME   SUPP CR   FREQ.   CR PER HR    AVAIL.
 EAST       .686   .510   11.7     3.84    0.05      0.041       3.26
 WEST       .832   .500   18.3     2.09    0.02      0.012       1.52
 NORTH      .696   .442   12.3     2.83    0.03      0.016       1.50
 SOUTH      .769   .590   19.5     3.12    0.03      0.016       1.95
 CENTRAL    .620   .395   17.3     4.83    0.03      0.009       1.69

 AVERAGE    .734   .499   15.7     3.22    0.03      0.018       1.99

DAY: TUE-WED

 AVERAGE    .662   .400   14.7     4.55    0.03      0.017       2.36
```

Fig. 11 — Sample DISP command for Table 1 ordered by precinct within tour within day

PCAM determines the ordering within output tables from the ordering of qualifier phrases, as follows. The program initially assumes that the user wants his output in the first order described above, that is, by tour within day within precinct. However, if any phrases appear in a READ command qualifier, then this default order is overridden. The rule for establishing a new default output order is that if either a TOUR or a DAY phrase is the first phrase of a READ command qualifier (as is the case in Fig. 11), then the default output order for subsequent DISP commands is by precinct within tour. If a PRECINCT or DIVISION phrase appears first in a READ command (as is the case in Fig. 8), then the default output order is the same as if no qualifier were entered, namely, tour within day within precinct. The default output order established by a READ command remains in effect only until another READ command is entered. If a subsequent READ command has no qualifier phrases, then the original default is reestablished.

If any phrases appear in a DISP command qualifier, then the default output order is overridden for that DISP command only. If a PRECINCT or DIVISION phrase is the first in a DISP command qualifier, then output is ordered as tour within day within precinct. If the first qualifier phrase is a DAY or TOUR phrase, then output will be ordered as precinct within tour within day.

While the above explanation may seem somewhat complex, in practice the determination of output order works out quite naturally. For instance, the command READ DATA FOR PRECINCT=NORTH causes PCAM to read data for all days and tours in the data base for precinct NORTH, and establishes the default output order of tour within day for this precinct. Then a subsequent DISP command with no phrases in its qualifier will display the selected tables in this order. Conversely, the command READ DATA FOR DAY=TUE-WED,TOUR=MIDDAY causes the program to read data for all precincts in the data base for the MIDDAY tour of Tuesday, and establishes the default output order of precinct within tour within day. In this case, a subsequent DISP command without phrases in its qualifier will display the selected tables in this order.

The user may sometimes want to have all shifts in CURRENT-DATA printed, but in the order opposite to that implied by qualifiers in the READ command. This is possible by entering null qualifiers in the DISP command. For instance, the command READ DATA causes PCAM to read data for all precincts, days, and tours in the data base and establishes the default output order of tour within day within precinct. If the user wishes printout ordered by precinct within tour within day, he may enter a command such as DISP T(1) FOR DAY=( ). This works because a ⟨NAMELIST⟩ in a qualifier phrase of the form ( ) has the same effect on data selection as omitting the phrase entirely, but still produces the output order described above.

Overlay indicators: Each line of output from the DISP command corresponds to one shift (i.e., a tour of one day in one precinct). This is true irrespective of the output order. To help clarify the relationships among shifts involved in overlays, each line of output values is prefixed by an indicator. If the indicator is blank, then no overlay was involved in computing the measures shown. An asterisk (*) at the beginning of an output line indicates that the measures displayed reflect the presence of an overlay shift that started or ended during the shift represented by that line of output. A plus sign (+) at the beginning of an output line indicates that the corresponding shift is an overlay shift.

Summary statistics: The DISP command output contains averages of measures for days, tours, or precincts, depending upon the output order. Whatever order is chosen, overall averages will be displayed. If an overlay tour is included in the output, all hours of the day that are of interest are included in *other tours besides the overlay tour*, so the figures displayed for the overlay tour are not used in calculating averages for the day.

In Table 2 the summary statistics include the total number of cars assigned and the total number of car-hours. These, of course, include the cars assigned to the overlay tour, if any.

Explanation of output: We return now to Fig. 10 and discuss the interpretation of Table 1 output. The first line of output identifies the precinct (WEST) and day (TUE-WED) to which the tour output applies. The next three output lines are column headings that identify the output measures for tours that appear on subsequent lines. The first output column identifies the tour to which subsequent columns on the line apply (MIDDAY for the first line of output measures). The second column gives the average utilization of an *effective* car averaged over the hours of the tour (.530 for the MIDDAY tour). The interpretation of this number is as follows. The PCAM program has estimated, from information provided in CURRENT-DATA, that each car assigned to WEST precinct during the MIDDAY tour will spend approximately half* of its eight-hour tour unavailable for reasons unrelated to calls for service. *Of the remaining four hours,* the car will spend, on the average, 53 percent of its time busy at calls for service. This amounts to 2.06 hours busy at calls for service.

The third column gives the average utilization of an *actual* car (.257). The interpretation of this figure is that, on the average, 25.7 percent of a car's *entire eight-hour MIDDAY tour* (i.e., the same 2.06 hours) will be spent busy on calls for service.

The fourth column of Table 1 gives the average travel time, in minutes, to incidents during the tour (14.2 for the MIDDAY tour). The figures shown in the fifth column are the total hours of patrol during the tour divided by the total number of suppressible crimes during the tour (6.08 for the MIDDAY tour). The sixth output column gives the average patrol frequency (0.04 for the MIDDAY tour). The seventh output column shows the average patrol frequency times the number of suppressible crimes per hour (0.021 for the MIDDAY tour). The eighth and last column of Table 1 gives the average number of cars on patrol and available for dispatch during the tour (3.42 for the MIDDAY tour).

One line of output is printed for each tour of TUE-WED. The lines for the PM and AM tours are flagged with asterisks to indicate

*This figure is not displayed.

that the measures displayed reflect a change in the number of cars on duty during the tour due to the beginning or ending of the overlay tour. The output line for the FOURTH tour is flagged with a plus sign to indicate that it is an overlay tour and that the measures displayed represent averages over hours of the day that are also included in other tours.

The next line of output, labeled AVERAGE, gives the values of the measures described above averaged over the hours of the tours for which output is displayed (the entire day in this case). The hours covered by the overlay tour are counted only once in the averages.

The sequence of output lines described above is repeated for day SAT-SUN. A final two lines of output for precinct WEST are then printed. The first of them identifies the next line as one displaying averages over all days selected for precinct WEST and the last line of the Table 1 output presents the averages of the measures for all tours of TUE-WED and SAT-SUN in precinct WEST. The above sequence is then repeated for precinct CENTRAL and a final line of output shows averages over both precincts.

If the reader compares the figures shown in Figs. 10 and 11 for the PM tour of TUE-WED in the WEST and CENTRAL precincts, he will see that they differ, while the figures for the MIDDAY tour are the same. This illustrates the fact that cars assigned to the overlay tour (FOURTH) in DATABASE are not included in CURRENT-DATA unless the overlay tour is read. The READ command at the top of Fig. 11 does not include the overlay tour, so, for example, the number of cars on duty in Central Precinct during Tuesday's PM tour is a constant 5.8 in Fig. 11. In Fig. 10, the number of cars increases from 5.8 to 9.3 when the FOURTH tour begins.

In Fig. 12, we show an example of Table 2 output. This is the table that the user will want to inspect most frequently, as it gives the allocation and queuing information needed to evaluate alternative plans.

The first output line identifies the day (TUE-WED) and the first tour (MIDDAY) for which Table 2 output is displayed. The next two lines are column headings. The first column of each of the next five

```
COMMAND? DISP T 2

DAY TUE-WED ; TOUR     MIDDAY

           ACT.   CAR    CALL   SERV   PROB CALL   AVG P2   AVG P3   AVG TOT
PRECINCT   CARS   HRS    RATE   TIME    DELAYED     DELAY    DELAY    DELAY
EAST       13.4  107.2   6.8   48.3      .208       3.46    12.61    14.77
WEST       15.0  120.0   6.4   36.1      .249       3.92    17.51    19.75
NORTH      11.9   95.2   5.0   41.1      .219       3.63     8.84    13.82
SOUTH      12.6  100.8   5.2   45.7      .297       8.83    70.27    31.30
CENTRAL     7.6   60.8   2.9   44.2      .319      10.29    45.84    31.01

AVERAGE    12.1   96.8   5.3   43.0      .250       5.40    27.82    20.85
TOTAL      60.5  484.0

DAY TUE-WED ; TOUR     PM

           ACT.   CAR    CALL   SERV   PROB CALL   AVG P2   AVG P3   AVG TOT
PRECINCT   CARS   HRS    RATE   TIME    DELAYED     DELAY    DELAY    DELAY
EAST       14.0  112.0  10.1   42.6      .312       5.93    46.89    21.18
WEST       15.0  120.0   9.1   49.6      .557      14.68   238.90    61.19
NORTH       7.8   62.4   6.8   30.5      .423       9.04    41.40    23.81
SOUTH      11.0   88.0   8.5   45.7      .458      12.38   112.09    40.53
CENTRAL     7.0   56.0   4.6   36.2      .373       9.50    44.53    30.74

AVERAGE    11.0   87.7   7.8   42.1      .427      10.30   113.35    36.28
TOTAL      54.8  438.4

DAY: TUE-WED

AVERAGE    11.5   92.2   6.5   42.4      .356       8.33    78.90    30.07
TOTAL     115.3  922.4
```

Fig. 12 — Sample DISP command for Table 2

lines identifies the precinct to which the line applies. Remember, the precinct, day, and tour identify the shift for which output is presented. The second column shows the number of actual cars that were assigned to start the shift. This number reflects the car assignments in CURRENT-DATA. In Fig. 12, the car assignments are the same as after the READ command in Fig. 11, but, in general, CURRENT-DATA may have been modified by MEET, ALOC, ADD, or SET commands. The third column shows the number of car-hours for each tour. This is the product of the number of cars assigned to start the tour and the length of the tour in hours and does not include overlay effects.

The fourth and fifth columns give the average number of calls per hour and the average service time (in minutes) for calls in the tour. Note that the service time is averaged over calls (and not over hours, as it is in output from the LIST command).

The sixth column of Table 2 gives the probability that the dispatch of a car to a call for service will be delayed because no car is available. This can be interpreted as the fraction of calls expected to experience a queuing delay.

The seventh and eighth columns give the average dispatch delay (in minutes) for priority 2 and priority 3 calls, respectively. The averages include calls of each priority that experience no delay, as well as those that are queued before dispatch. The ninth and last column of Table 2 shows the average total delay for all calls. This is the sum of the average queuing delay and the average travel time to incidents.

The output line labeled AVERAGE, which follows the lines for precincts NORTH, SOUTH, EAST, WEST, and CENTRAL, gives the averages of the Table 2 measures described above for all of the precincts for the MIDDAY tour of TUE-WED. The next output line, labeled TOTAL, gives the total number of actual cars and car-hours for the MIDDAY tour of TUE-WED in all precincts.

The pattern is then repeated for the PM tour on TUE-WED. Next appear lines giving averages and totals over whatever part of the day has been selected. Hours included in overlay tours are counted only once. If more than one day is selected, the overall averages and totals are printed.

ALOC Command

ALOC ⟨NUMBER⟩ [CAR] [HOURS] [TO] ⟨QUALIFIER⟩ [BY] F⟨NUMBERLIST⟩

or

ALOC * [CAR] [HOURS] [TO] ⟨QUALIFIER⟩ [BY] F⟨NUMBERLIST⟩

or

ALOC * -⟨NUMBER⟩ [CAR] [HOURS] [TO] ⟨QUALIFIER⟩ [BY] F⟨NUMBERLIST⟩

Action: All three ALOC commands allocate a specified number of car-hours among selected shifts in such a way as to minimize the objective function given by F⟨NUMBERLIST⟩. The valid entries for ⟨NUMBERLIST⟩ will be listed and explained on the next page.

In the first version of the ALOC command, the number of car-hours to be allocated is specified by ⟨NUMBER⟩, which must be positive. In the other two versions of the ALOC command, the asterisk (*) denotes the total number of car-hours assigned to the selected shifts in CURRENT-DATA. This total is determined from the data base or from preceding ALOC, MEET, SET, or ADD commands.

The second version of the ALOC command causes a reallocation of the current total number of car-hours among the selected shifts. The third version of ALOC will allocate the number currently assigned minus ⟨NUMBER⟩. For example, if 320 car-hours are currently assigned, ALOC * -16 ... will allocate 304 car-hours.

As usual, the qualifier specifies the precincts, days, and tours over which car-hours will be allocated and for which car-hours are currently allocated. The ALOC command qualifier serves no other function.

The ⟨NUMBERLIST⟩ specifies the objective function to be used for the allocation. Its first element is always a number that selects the function to be used. Other elements in the list specify parameters required for particular functions. The available objective functions are as follows:

F(1)	Average fraction of calls delayed in queue
F(2)	Average length of time calls are delayed in queue
F(2,1)	Average length of time priority 1 calls are delayed in queue
F(2,2)	Average length of time priority 2 calls are delayed in queue
F(2,3)	Average length of time priority 3 calls are delayed in queue
F(3)	Average total response time (queuing + travel time)

As an example, the command ALOC 480 TO DAY=TUE-WED, TOUR=MIDDAY BY F(2,2) will allocate 480 car-hours (or 60 cars) among the precincts on Tuesday's MIDDAY tour in such a way as to make the citywide average delay for priority 2 calls as small as possible.

Warning message: The program's first step in carrying out an ALOC command is to assign enough cars to each shift within the command scope to keep effective utilization under 1 in every hour. If the number of car-hours required for this step is greater than the number of car-hours specified in the ALOC command, the program prints a message giving the minimum number of car-hours required, and the allocation is performed as if the user had asked to allocate that number of car-hours.

Memory: The assignment of cars resulting from the execution of an ALOC command is stored in CURRENT-DATA, replacing whatever assignments were previously stored. Therefore, the new assignment will be reflected in the output resulting from subsequent DISP, LIST, ADD, or SET commands. However, DATABASE is not changed by the ALOC command, so the user can cause the program to "forget" the assignment by executing a new READ command.

Example: Figure 13 gives an example of the use of an ALOC command. This command causes the reallocation of the current number of car-hours assigned to precinct CENTRAL on TUE-WED so that the probability of a call being delayed is minimized for the day as a whole. The DISP command outputs show the old and new assignments and their effects on the output measures.

From this output, we can see that the probability of delay for precinct CENTRAL on TUE-WED is reduced from .320 to .289, so (as desired) the new allocation has improved the value of this performance measure. It happens that all the other performance measures on Table 2 have been improved for the day as a whole by this allocation, but the user may not be so lucky with his own data! While no other allocation of 168 car-hours can have a lower average probability of delay than the one shown at the bottom of Fig. 13, the use of objective functions other than F(1) will lead to lower values for the associated performance measures.

These two sets of Table 2 output also indicate that the total number of car-hours for TUESDAY in precinct CENTRAL has decreased from 169.6 to 168. This is because only whole numbers of cars are

```
COMMAND? READ DATA FOR PRECINCT=CENTRAL,DAY=TUE-WED

COMMAND? DISP T 2

PRECINCT: CENTRAL ; DAY: TUE-WED

            ACT.   CAR    CALL   SERV   PROB CALL   AVG P2   AVG P3   AVG TOT
 TOUR       CARS   HRS    RATE   TIME    DELAYED     DELAY    DELAY    DELAY
 MIDDAY      7.6   60.8    2.9   44.2      .319      10.29    45.84    31.01
*PM          5.8   46.4    4.6   36.2      .335      10.07    82.88    34.75
*AM          4.3   34.4    2.0   36.2      .290       9.20    23.50    21.35
+FOURTH      3.5   28.0    3.4   36.2      .164       2.71     6.99    16.24

 AVERAGE     5.3   42.4    3.2   38.6      .320       9.95    58.91    30.75
 TOTAL      21.2   169.6

COMMAND? ALOC * BY F(1)

COMMAND? DISP T 2

PRECINCT: CENTRAL ; DAY: TUE-WED

            ACT.   CAR    CALL   SERV   PROB CALL   AVG P2   AVG P3   AVG TOT
 TOUR       CARS   HRS    RATE   TIME    DELAYED     DELAY    DELAY    DELAY
 MIDDAY      8.0   64.0    2.9   44.2      .292       8.58    33.65    27.67
*PM          9.0   72.0    4.6   36.2      .221       3.84    11.61    19.54
*AM          4.0   32.0    2.0   36.2      .438      16.63    46.66    31.72
+FOURTH      0.0    0.0    3.4   36.2      .252       6.43    17.76    21.68

 AVERAGE     5.3   42.0    3.2   38.6      .289       8.01    25.81    24.62
 TOTAL      21.0   168.0
```

Fig. 13 — Sample ALOC command and response

assigned to shifts by the ALOC command. If another car had been assigned to any shift, it would have brought the total number of car-hours to 176, which is more than were in the original assignment.

ADD Command

ADD ⟨NUMBER⟩ [-*] [CAR] [HOURS] [TO] ⟨QUALIFIER⟩ [BY] F⟨NUMBERLIST⟩

Action: The ADD command allocates a specified number of car-hours to selected shifts in such a way as to minimize the average value of the objective function identified by F⟨NUMBERLIST⟩. The command works in a manner similar to the ALOC command, and the valid entries for F⟨NUMBERLIST⟩ are the same as given above in the description of the ALOC command. The only difference between the two

commands is that while the ALOC command assigns the specified total number of car-hours, the ADD command assigns cars to the selected shifts in addition to those already assigned in CURRENT-DATA. The total number of car-hours assigned to the selected shifts in CURRENT-DATA is denoted by an asterisk (*). If the optional expression [-*] is *not* used, the ADD command increases the total number of car-hours assigned to the selected shifts by ⟨NUMBER⟩. If the form ADD ⟨NUMBER⟩ -* ... is used, the command adds the *difference* between ⟨NUMBER⟩ and the current total; in other words, the final total will be ⟨NUMBER⟩.

The ADD command cannot be used to *subtract* cars. If the user attempts to ADD a negative number of car-hours (for example, ADD 280 -* BY F(1), when the current allocation is 310 car-hours), the program will take no action and will simply await the next command. To allocate a smaller number of car-hours, the user must start over, using the ALOC command, or the sequence of commands READ, MEET, and ADD, or the sequence ALOC Ø (zero), MEET, and ADD. (The command ALOC Ø will reinitialize the number of cars in each shift to the minimum number required to keep effective utilization under 1 in each hour.)

Example: The ADD command shown in Fig. 14 was given to PCAM after the commands in Fig. 13. Therefore, the ADD command increases by 40 the total number of car-hours assigned to precinct CENTRAL on TUE-WED. Exactly the same effect would be achieved by entering the

```
COMMAND? ADD 40 CAR HOURS BY F(3)

COMMAND? DISP T 2

PRECINCT: CENTRAL ; DAY: TUE-WED

            ACT.   CAR    CALL   SERV   PROB CALL   AVG P2   AVG P3   AVG TOT
 TOUR       CARS   HRS    RATE   TIME   DELAYED     DELAY    DELAY    DELAY
 MIDDAY     10.0   80.0    2.9   44.2     .161       3.31     8.97    17.96
*PM         10.0   80.0    4.6   36.2     .172       2.58     6.86    16.86
*AM          6.0   48.0    2.0   36.2     .201       4.49     8.75    15.04
+FOURTH      0.0    0.0    3.4   36.2     .164       2.80     6.45    16.04

 AVERAGE     6.5   52.0    3.2   38.6     .175       3.21     7.91    16.80
 TOTAL      26.0  208.0
```

Fig. 14 — Sample ADD command and response

command ADD 208 -* CAR HOURS BY F(3). Cars are assigned to shifts so as to achieve the greatest improvement in the average total delay for calls. Since all shifts are eight hours in length, the total number of cars assigned to CENTRAL precinct on TUE-WED has been increased by five. By comparing Table 2 in Fig. 14 with the second table in Fig. 13, the user determines the improvement that could be achieved by adding 5 cars.

MEET Command

MEET C⟨NUMBERLIST⟩$_1$=⟨NUMBERLIST⟩$_2$ [FOR] ⟨QUALIFIER⟩

Action: The MEET command causes PCAM to assign cars to all the shifts within its scope so that a specified set of output measures meets certain constraints. The element of ⟨NUMBERLIST⟩$_1$ are codes for the measures that are to be constrained. The elements of ⟨NUMBERLIST⟩$_2$ specify the limiting values on each of the output measures listed in ⟨NUMBERLIST⟩$_1$. Therefore, ⟨NUMBERLIST⟩$_2$ must have the same number of elements as ⟨NUMBERLIST⟩$_1$, and corresponding elements of the two lists produce a pair giving (1) the measure to be constrained and (2) the value of the constraint.

The following is a list of valid entries in ⟨NUMBERLIST⟩$_1$ and their meanings.

1	Average utilization of an effective car	↓
2	Average travel time (minutes)	↓
3	Average number of cars available	↑
4	Patrol hours per suppressible crime	↑
5	Patrol frequency (passings per hour)	↑
6	Minimum number of cars	↑
7	Fraction of calls delayed	↓
8	Average delay for priority 2 calls (minutes)	↓
9	Average delay for priority 3 calls (minutes)	↓
10	Total wait (queuing + travel) (minutes)	↓

The arrows indicate that the constraint is met if the measure is *lower* than the specified value for downward arrows (↓) or *higher* than the specified value for upward arrows (↑).

For example, the command MEET C(3,9)=(4,15) will assign enough cars to each shift so that the average number of cars available is 4 *or larger* and the average wait for priority 2 calls is 15 minutes *or less*.

Operation: The MEET command operates on all shifts within its scope independently but in the same manner. If a shift has not been selected in an allocation-type command (MEET, ALOC, or ADD) since the last READ command, then just enough cars are assigned to the shift to keep the effective utilization under 1 in each hour. Otherwise, the MEET command starts with the current assignment (based on the previous MEET, ALOC, or ADD command). Then the program adds enough cars to the initial assignment to meet all specified constraints. (See Fig. 4.) The program also "remembers" which constraint required the use of most cars and prints an asterisk next to the corresponding measure for the shift in the output from subsequent DISP commands. If a subsequent MEET command imposes a more stringent constraint on the shift, then the measure corresponding to that constraint will supersede the previous measure and will be identified with an asterisk in DISP command output. No output measures will be flagged if all constraints are met by the minimum allocation of cars calculated by the MEET command or by the allocation from a previous ALOC or ADD command.

If a shift is selected in the qualifier of a subsequent ADD or ALOC command, then any indication of a constraint having been met is removed and no measures will be flagged in subsequent output (unless and until another MEET command selects the tour). A subsequent ALOC command completely negates the effect of a MEET command (except that the total number of car-hours assigned by a MEET command can be reallocated).

Output: When cars have been assigned to meet all constraints for all shifts, the program prints a message giving the total number of car-hours allocated to all shifts within the MEET command scope.

Example: Figure 15 illustrates the use of the MEET command for TUE-WED in precinct CENTRAL so that the probability of a call being delayed does not exceed .3 and the average wait for priority 2 calls does not exceed 8 minutes. 192 car-hours were required to meet the specified constraints.

```
COMMAND? READ DATA FOR PRECINCT=CENTRAL, DAY=TUE-WED

COMMAND? MEET C(7,8)=(.3,8)

  192 CAR HOURS ALLOCATED.

COMMAND? DISP T 2

PRECINCT: CENTRAL ; DAY: TUE-WED

            ACT.   CAR    CALL   SERV   PROB CALL    AVG P2   AVG P3   AVG TOT
 TOUR       CARS   HRS    RATE   TIME    DELAYED      DELAY    DELAY    DELAY
 MIDDAY      9.0   72.0    2.9   44.2      .226     *  5.43    17.05    21.87
*PM          9.0   72.0    4.6   36.2     *.221        3.84    11.61    19.54
*AM          6.0   48.0    2.0   36.2     *.201        4.49     8.75    15.04
+FOURTH      0.0    0.0    3.4   36.2      .196        3.63     9.42    17.88

 AVERAGE     6.0   48.0    3.2   38.6      .218        4.46    12.65    19.29
 TOTAL      24.0  192.0
```

Fig. 15 — Sample MEET command and response

Here the READ command reinitializes CURRENT-DATA (negating the effects of past commands). The allocation contained in CURRENT-DATA after the READ command for precinct CENTRAL on TUE-WED can be seen in the output of the DISP command in Fig. 13. Then, from Fig. 15, we can see that 22 additional car-hours were required to meet constraints. The output of the DISP command in Fig. 15 reveals that the most constraining measure was average priority 2 delay for the MIDDAY tour and probability of delay for the others. Since the overlay tour required no cars, no output measures were flagged.

SET Command

SET P⟨NUMBERLIST⟩$_1$=⟨NUMBERLIST⟩$_2$ [FOR] ⟨QUALIFIER⟩

Action: The SET command can be used to alter certain data values. It does not change values in DATABASE, but like all other commands except READ, operates only on CURRENT-DATA. Thus, changes effected by a SET command last only until the next READ command is executed. If the user wishes to save the changes for later runs of the PCAM program, he must use the WRITE command. Each element of $\langle \text{NUMBERLIST} \rangle_1$ specifies a type of data to be altered. Corresponding elements of $\langle \text{NUMBERLIST} \rangle_2$ specify the value to be assigned to each type of data for all precincts, days, and tours within the command scope.

The permitted values for $\langle \text{NUMBERLIST} \rangle_1$, together with their meanings, are as follows.

1	Unavailability parameter B1	(precinct)
2	Unavailability parameter B2	(precinct)
3	Call rate parameter	(day)
4	Service time parameter	(day)
5	Actual cars assigned	(shift)
6	Response speed	(shift)
7	Patrol speed	(shift)
8	Fraction of calls that are priority 1	(shift)
9	Fraction of calls that are priority 2	(shift)
10	Total number of suppressible crimes	(shift)

As indicated by the descriptor in parentheses, some types of data automatically refer to all days for a precinct (like unavailability parameters) or all tours of a day (like call rate parameters). Therefore, if a SET command qualifier contains a phrase which would be an "overqualification" for a type of data, then the phrase is ignored for that type of data when the command is executed. This feature is useful when altering a series of data values that require different levels of qualification. For example, the command

```
SET P(5,3)=(10,6.3) FOR PRECINCT=NORTH,DAY=TUE-WED,TOUR=MIDDAY
```

causes the number of cars assigned to precinct NORTH for the MIDDAY tour of Tuesday to be changed to 10 and the call rate parameter for

TUE-WED in precinct NORTH to be changed to 6.3. The TOUR=MIDDAY phrase is ignored when the call rate parameter is set.

With the above exception, all the usual rules for command scope apply.

Warning message: Processing of the SET command always insures that all shifts within its scope end up with enough cars to keep effective utilization under 1 in every hour. If a tour needs additional cars, the minimum number required is computed, printed, and assigned to the tour. Failure to meet the minimum car requirement can result not only from changes in the assignment of cars, but also from changes in other factors such as unavailability parameters, call rates, and service times.

Figure 16 gives a simple example of a SET command. Here the call rate parameter in CENTRAL precinct is changed from 3.66 to 4.10. By comparing the output with Fig. 9, we see that not only have the call rates been affected by the SET command, but also the minimum number of cars acceptable to PCAM has changed in the PM tour, and the average number of effective cars has changed in the other two tours (because the relationship between actual and effective cars depends on the call rate).

```
COMMAND? READ PRECINCT=CENTRAL, DAY=TUE-WED

COMMAND? SET P(3)=4.1

***  7. CARS NEEDED IN PRECINCT CENTRAL  FOR TOUR      PM        ON DAY TUE-WED

COMMAND? LIST DATA

PRECINCT: CENTRAL ; AREA= 47.3; STREET MILES=498.3; B2=0.658; B1=-.746
  DAY: TUE-WED ; CALL RATE PARM= 4.10; SERVICE TIME PARM=40.20

                AVG.                          AVG.   AVG.   FRAC.   FRAC.   FRAC.
          ACT.  EFF.   RSP.   PTL.   AVG.     SERV   CALL   OF P1   OF P2   OF P3
TOUR      CARS  CARS   VEL.   VEL.   TIME    RATE   CALLS   CALLS   CALLS
MIDDAY     7.6   4.4   15.0    7.5   44.2     3.2   0.067   0.810   0.123
PM         7.0   5.5   15.0    7.5   36.2     5.1   0.067   0.810   0.123
AM         4.3   2.9   25.0    7.5   36.2     2.3   0.067   0.810   0.123
FOURTH     3.5
```

Fig. 16 — First sample SET command and response

A second SET command is shown in Fig. 17. This changes the response speed to 30 mph and unavailability parameters B2 and B1 to .3 and 0, respectively, for both precincts (EAST and NORTH) of the HIGHLAND division on SAT-SUN. This change in the unavailability parameters results in a constant three-tenths of all cars fielded being unavailable on non-cfs work. The DAY phrase is ignored when the unavailability parameters are set, and the response speed is changed for all tours of SAT-SUN in both precincts. The message preceded by three asterisks indicates that at least 12 cars were needed in precinct EAST for the PM tour of SAT-SUN to keep effective utilization under 1 in every hour, as a result of the change in unavailability parameters.

```
COMMAND? READ DATA FOR DIVISION=HIGHLAND,DAY=SAT-SUN
***  9. CARS NEEDED IN PRECINCT NORTH    FOR TOUR    MIDDAY   ON DAY SAT-SUN
*** 10. CARS NEEDED IN PRECINCT NORTH    FOR TOUR    PM       ON DAY SAT-SUN
***  8. CARS NEEDED IN PRECINCT NORTH    FOR TOUR    PM       ON DAY SAT-SUN
COMMAND? SET P(6,2,1)=(30,.3,0)
*** 12. CARS NEEDED IN PRECINCT EAST     FOR TOUR    PM       ON DAY SAT-SUN
COMMAND? LIST DATA
PPECINCT: EAST     ; AREA= 34.9; STREET MILES=503.4; B2=0.300; B1=0.0
  DAY: SAT-SUN ; CALL RATE PARM= 7.71; SERVICE TIME PARM=37.91

                 AVG.                          AVG.  AVG.  FRAC.  FRAC.  FRAC.
           ACT.  EFF.  RSP.  PTL.              SERV  CALL  OF P1  OF P2  OF P3
   TOUR    CARS  CARS  VEL.  VEL.              TIME  RATE  CALLS  CALLS  CALLS
   MIDDAY  11.8   8.3  30.0   7.5              37.9   9.0  0.065  0.840  0.095
   PM      12.0  10.9  30.0   7.5              37.9  12.1  0.065  0.840  0.095
   AM       8.8   7.7  30.0   7.5              37.9   6.9  0.065  0.840  0.095
   FOURTH   5.8

PRECINCT: NORTH    ; AREA= 24.1; STREET MILES=409.1; B2=0.300; B1=0.0
  DAY: SAT-SUN ; CALL RATE PARM= 5.32; SERVICE TIME PARM=35.28

                 AVG.                          AVG.  AVG.  FRAC.  FRAC.  FRAC.
           ACT.  EFF.  RSP.  PTL.              SERV  CALL  OF P1  OF P2  OF P3
   TOUR    CARS  CARS  VEL.  VEL.              TIME  RATE  CALLS  CALLS  CALLS
   MIDDAY   9.0   6.3  30.0   7.5              42.3   4.7  0.061  0.850  0.089
   PM      10.0   9.1  30.0   7.5              31.4   7.8  0.061  0.850  0.089
   AM       8.0   6.9  30.0   7.5              34.2   5.1  0.061  0.850  0.089
   FOURTH   4.8
```

Fig. 17 — Second sample SET command and response

WRITE Command

WRITE [DATA] [ON] ⟨NUMBER⟩ [FOR] ⟨QUALIFIER⟩

Action: Writes a NEW-DATA file on FORTRAN unit ⟨NUMBER⟩. This command cannot be used unless data processing personnel determine what ⟨NUMBER⟩ is to be used and prepare appropriate file specifications in accordance with instructions in Chapter I of the companion *Program Description* (R-1786/3). There is no displayed output from a WRITE command.

The NEW-DATA file contains information from CURRENT-DATA for those precincts, days, and tours specified by ⟨QUALIFIER⟩. The file NEW-DATA is suitable as input for subsequent runs of the PCAM program, but any READ commands issued after the WRITE command in the current session with PCAM will read data from DATABASE.

Figure 18 gives an example of a WRITE command:

```
COMMAND? WRITE DATA ON 18
```

Fig. 18 — Sample WRITE command

Since this command was issued after the READ and SET commands shown in Fig. 17, the NEW-DATA file differs from DATABASE in the following ways:

- NEW-DATA contains data only for HIGHLAND division and only for SAT-SUN.
- The response speed in NEW-DATA is 30 mph, and the unavailability parameters are B2=0.3, B1=0.
- The number of cars assigned to precinct NORTH are 9 for the MIDDAY tour, 10 for the PM tour, and 8 for the AM tour. The number of cars assigned to precinct EAST is 12 for the PM tour.

END Command

END

Action: The END command causes the PCAM program to terminate execution and return control to the operating system. This *must* be the last command for every run of the PCAM program. After the END command is issued, PCAM prints a message telling the largest amount of core storage used for CURRENT-DATA during the run. This message is provided to assist data processing personnel in minimizing the cost of operating the program.

ERROR CONDITIONS

When an error condition arises during the interpretation or execution of a command, the program indicates this by printing a message preceded by three asterisks. The error message will indicate the nature of the error and what action the program is taking and/or what action the user should take.

The three possibilities for further action when an error condition is encountered are:

- The program will terminate execution,
- The program will take corrective action and continue execution,
- The program will ignore the command just entered and accept the next command.

The error messages indicate which course of action applies to particular errors.

SELECTING AN OBJECTIVE FUNCTION

Users have a choice of six different objective functions for the ALOC and ADD commands. We encourage experimentation with all six of them to determine which one seems most appropriate for a particular department. As a first step, try allocating a fixed total number of

car-hours, using each objective function in turn. After each ALOC command, enter a DISP command to see the results.

In many instances it will happen that the allocations suggested by PCAM are identical, no matter what objective function is used, or they differ only by one car in a few shifts. In this case, the particular objective function that one chooses is immaterial, since the data are never accurate enough to permit the user to decide that one allocation is better than another one when they are nearly identical.

It is especially likely that the objective functions F(2), F(2,1), F(2,2), and F(2,3) will lead to nearly identical allocations, because they all refer to the average length of time that calls are delayed in queue. Exceptions will occur only where the fraction of high priority calls varies substantially by time of day or by precinct.

Supposing now that the allocations obtained using different objective functions are not similar, one should examine the grand averages of performance measures for the different allocations. (These are obtained from the DISP command.) It often happens that some of the averages are approximately the same, no matter what objective function is used, while others vary by large amounts. For example, the smallest possible average fraction of calls delayed might be .084, which is found by using F(1) as an objective function, but the average fraction of calls delayed using other objective functions might be very close, say, .089 or .092. On the other hand, the average delay for priority 2 calls might vary from 2.2 minutes, when F(2,2) is the objective function, to 12.3 minutes when F(1) is the objective function. In this example, it is clearly better to use one of the F(2)'s as an objective function rather than F(1), since one is trading off a small increase in the fraction of calls delayed for a large improvement in the length of time calls are delayed. *In general, choose the objective function that produces a big improvement in its associated performance measure while not seriously impairing other performance measures.*

If these comparisons are not decisive, then we recommend using F(3) (average response time) as the objective function. This is the only objective function that blends geographical with queuing characteristics. If a department has some large precincts with few calls for service and

other small precincts with many calls for service, the user will find that this objective function produces allocations that appear to strike a reasonable balance among their differing requirements. In addition, total response time is the only performance measure calculated by PCAM that has been shown to be related to the chance that an offender will be apprehended at the scene of a crime [5,9].

Appendix A

HISTORY OF PATROL CAR ALLOCATION PROGRAMS

Analysis of police patrol car operations using the techniques of operations research has been under way for more than a decade. 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 at various times of day on each day of the week. PCAM was designed to incorporate many of the best features of these programs in a single package.

This appendix describes the precursors of PCAM and indicates how they differ from PCAM. The reader is assumed to be familiar with PCAM's capabilities and with the terminology used in this report (see Glossary).

ST. LOUIS POLICE DEPARTMENT

The first patrol car allocation program was proposed and documented in 1964 by Richard F. Crowther [7]. (See also Shumate and Crowther [25].) During the four years that followed, a project team at the St. Louis Police Department refined the methodology, programmed it for the department's computer, and applied it in one precinct (called "district" in St. Louis) [21,22]. The project covered many allocation topics, but we shall describe here only the computer program that was designed to determine the number of patrol cars needed in each precinct.

The program had two components, *prediction of cfs workload* and *queuing analysis*. In the prediction component, the city was divided into small areas about the size of several blocks. These were named Pauly Areas after the officer who did the work. Dispatchers' records were coded according to the Pauly Area in which the incident occurred, and a program was written to count how many incidents in each of eight different categories occurred in each Area. These counts were projected into the future by means of a statistical technique called "exponential smoothing" that takes into account variations in the call rate by time of day, day of week, and week of year and also adjusts for overall trends

in the call rate (general increase or decrease, compared to last year). The result is a predicted estimate of the number of calls to occur in each Pauly Area in each hour of the week. The service times of incidents were smoothed in similar fashion.

Since a precinct can in principle be any collection of Pauly Areas, to estimate the hourly workload in a precinct all that has to be done is multiply the expected number of incidents of a particular type in an hour in a Pauly Area by the corresponding service time and add these over all incident types and Pauly Areas in the precinct.

The prediction component of the St. Louis PD's computer program performed these calculations, thereby providing a capability not present in PCAM.

In the queuing component, the program used the same equation as now used in PCAM to estimate the percentage of calls in each shift that would experience a queuing delay, supposing that three cars were fielded, four cars were fielded, and so on. The primary differences as compared to PCAM were (1) the distinction between *actual* cars and *effective* cars had not yet been discovered, and (2) the St. Louis program simply printed out all the queuing probabilities and left it to the reader of the output to establish a constraint on queuing and consult the tables to see how many cars were needed to meet the constraint. The department's policy was 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.

These programs were operated by the department in batch mode on a regular basis for at least five years, although it is our understanding that they are now used only sporadically.

For purposes of comparison with PCAM and the other programs, we shall point out certain details of the St. Louis system. 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 appears satisfactory and accurate, and it is still recommended for applications of PCAM.

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 (because all calls have the same chance of experiencing a queue). 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 cfs workload with the predictions. As part of any application of exponential smoothing, it is necessary for the user to select one or more parameters that specify how much weight will be given to recent data as compared to old data. If patterns in cfs workload are changing rapidly, the quality of the predictions may be sensitive to the exact choices of these parameters. Apparently the St. Louis Police Department did not have a problem in this regard.

LEMRAS (Law Enforcement Manpower Resource Allocation System)

This recently withdrawn IBM software package was based on the St. Louis system and includes all of its features, together with a number of improvements [8]. It, too, was a batch program. Once again, the city is divided into small areas (which are called "reporting areas" instead of Pauly Areas), and the number of incidents and their service times are predicted by exponential smoothing. Incidents may be divided into a large number of event codes that correspond to the names given to incidents by dispatchers, and these are aggregated into, at most, 20 "event classes" for purposes of statistical analysis. Each event class can be assigned to one of three "priority levels."

In an advancement over the St. Louis system, the LEMRAS program operates on the assumption that when calls are queued all queued calls of priority 1 (i.e., the most urgent calls) will be assigned to patrol cars before any queued calls of priority 2, which in turn will be

assigned prior to any queued calls of priority 3. Thus, while all calls in a given precinct during a particular hour have the same chance of being placed in queue, priority 1 calls will be calculated to be less likely to remain in queue for any specified period of time (say, five minutes) than will calls of lower priority.

For each specified number of patrol cars on duty, the LEMRAS program will estimate what percentage of calls in each priority level will be delayed for five minutes, for ten minutes, and so forth. By taking into account how many calls in each event class are expected to occur each hour, this information is then summarized for each event class on a weekly basis, or whatever is desired by the user. Thus a department using the LEMRAS system can, if it wishes, allocate enough cars so that at most 10 percent of callers experience a queuing delay and at most 2 percent of priority 1 calls wait in queue more than five minutes. PCAM does not calculate the distributions of delays by priority level which are provided by LEMRAS. Instead, PCAM calculates the *average* length of time a call of a given priority will wait in queue. This choice was made simply to reduce the volume of output provided by the program.

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. LEMRAS was withdrawn by IBM at the end of 1974, because the program is not compatible with the latest generation of operating systems being marketed by the corporation, and most customers were interested in an online 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 cfs workload, 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 in the case of cfs work 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 in Chapter II, 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 [19]. Later, he described the program, together with potential improvements that could be made, in his book *Urban Police Patrol Analysis* [20]. 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 the probability that the call will wait more than five minutes, more than ten minutes, etc. The mathematical formulas used by Larson, which were derived by Cobham [6], serve as the basis for PCAM's calculations of queuing delays by priority level.

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 the actual ones included in his program were:

a. Average travel time to incidents,
b. Average patrol frequency, and
c. Patrol hours per outside crime.

The latter two are identical to measures calculated by PCAM, while Larson's equation for average travel time has been improved in designing PCAM.

In descriptive mode, the user of Larson's program 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 (e.g., not more than 10 percent of calls delayed, travel time under eight minutes, patrol frequency at least twice per eight hours) he could inspect the tables and see how many cars were needed to accomplish the objectives. Thus, the descriptive mode represents 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 is 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 abovementioned measures a, b, and c 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. This notion serves as the basis for PCAM's capabilities to meet constraints.

Allocation of a Fixed Number of Cars

Larson believed that the total number of patrol cars available for fielding in the city was an important consideration in allocating cars to precincts. What good does it do to find out how many cars are needed in each precinct to keep less than 10 percent of calls from being queued if the sum for all precincts is twice as many cars as the department can field? Conversely, if a department had more cars to field than were indicated as needed by the program, would it really be confident that the 10 percent figure was "good enough?"

So in the prescriptive mode of Larson's program, the user may input 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 allocates cars to precincts in such a way that, first, all the constraints discussed above are met, and, second, the additional cars (if any) are allocated so as to minimize the citywide average time a call would wait in queue.* The user does not state in advance what level of queuing delay he wants to achieve, but instead gets an allocation which is as good as possible in this regard, given fixed resources and the specified constraints.

The principle underlying this calculation has been incorporated directly into PCAM, with two important differences. First, PCAM will allocate across tours as well as across precincts. Second, PCAM permits minimization of other objective functions. A third, minor, difference is that PCAM requires two steps to perform an allocation which constitutes one step in Larson's program.

For each precinct, Larson's program requires the following input information: its area (square miles), number of street miles, and, for each tour, the precinct's

- Expected call rate
- Average service time
- Patrol car response speed
- Patrol car patrol speed
- Number of outside (i.e., "visible" or "suppressible") crimes
- Maximum average travel time desired
- Minimum patrol frequency desired
- Minimum patrol hours per outside crime desired
- Minimum number of cars permitted (administrative).

---

*Actually, the user could specify weights for each priority level, and the program would minimize the weighted average waiting time. For example, by giving the highest priority weight 1, and all other priorities weight zero, the allocation would minimize the average time that priority 1 calls wait in queue.

It does not utilize hourly data, as do the two programs described above, but works from averages for a tour (usually eight hours). It also has no special capabilities for handling non-cfs work, other than by including such work in the call rate and the service time.

Larson's program was written in a language called Michigan Algorithm Decoder (MAD), 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 New York City Police Department 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 [26]. This program is made available to police departments by contract, and Urban Sciences also assists departments in preparing the required data base. In all conceptual aspects it is 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 [23]. While based on Larson's program, Mudge's program is not exactly the same. The two primary differences are:

- Mudge's program distinguishes between "effective" cars and "actual" cars.
- Mudge's program will not allocate a specified total number of patrol cars. In prescriptive mode, this program simply calculates the number of patrol cars needed in each precinct to meet constraints entered by the user.

In addition, 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 returns 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.

The performance measures that can be displayed on output are listed below, with the ones subject to constraint by the user indicated by an asterisk:

* Average utilization
* Average travel time
* Patrol hours per tour
* Patrol hours per outside crime
* Average patrol frequency

  Patrol frequency times number of outside crimes
* Average waiting time in queue for priority 2 calls
* Average waiting time in queue for priority 3 calls
* Fraction of calls that will be placed in queue

  Average waiting time in queue for those calls that wait

  Average number of cars available.

Thus, Mudge had adopted nearly all the performance measures now incorporated in PCAM, although some refinements were made in designing PCAM.

It will be noted that this program also permits only three priority levels and that the wait for priority 1 calls is not displayed. Mudge found 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 were inaccurate.

The program handles the conversion of "actual" cars to "effective" cars as follows. The user inputs a fraction (the same for all precincts) representing the fraction of time that cars are busy on non-cfs work. Subtracting this fraction from 1 and multiplying the difference by the number of "actual" cars yields an "effective" number of cars, which is then rounded to an integer.

Mudge's program is similar to Larson's in that it does not assist the user in predicting cfs workload or service times and it uses average workload data for a tour, rather than hourly data. It was written in

FORTRAN and is 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. It has now been replaced by PCAM.

UCLA PROGRAM

As mentioned above, the LAPD has 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 UCLA Engineering School class prepared a patrol car allocation program for consideration by the LAPD [1]. It was based on the NYCPD and Larson programs. In common with the NYCPD program, it permits the user to specify constraints on queuing delays as well as other performance measures. In common with the Larson program, it permits the user to allocate specified total resources. The primary differences between this program and the other two are:

1. The UCLA program allocates cars across tours instead of across precincts. This means that the user specifies the total number of car-hours available in a precinct during a day, and the program prescribes how many cars should be on duty during each tour. Or, alternatively, the user specifies constraints on performance measures and the program prescribes how many cars are needed in each tour, adding these to show total car-hours in a day for the precinct in question.
2. The UCLA program introduced the notion of modeling the relationship between non-cfs work and cfs work as a linear equation, separately for each precinct, using data from the precinct. The conversion between "effective" cars and "actual" cars is then calculated from the linear equation, using the unavailability parameters B1 and B2.

This program was written in PL/I and operates in batch mode. It does not make predictions of workload and service time, which were available from LEMRAS in any event. However, it accepts as input hourly data rather than averages for a tour. It does not have descriptive capabilities, although the output displays the performance measures

for the recommended allocations. It was acquired by the LAPD and run on the city's computer system, but the department did not choose to use this particular program for operational purposes. Instead, it plans to adopt PCAM.

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 [24]. This program was an improvement over the original NYCPD program described above 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 the linear relationship between non-cfs workload and cfs workload.

However, it was limited to allocations across precincts (i.e., it would not allocate car-hours across tours), and it used average workload data for a tour rather than hourly data. The interim version was available only as an interactive program. A data base for the Los Angeles Police Department was constructed and used in a demonstration of the capabilities of this model to adapt to conditions and terminology other than the NYCPD's.

PCAM

PCAM was designed to incorporate, by user option, nearly all the features of the programs described above, except that it will not predict cfs workload. It can be operated in batch or interactive mode by changing a small number of program statements. Its descriptive capabilities are based on Larson's program as improved by Mudge, and its capabilities to meet constraints are derived from the same sources. Its capability to allocate a specified number of car-hours is based on Larson's algorithm, with substantial improvements. Its model of non-cfs workload is based on analyses by the UCLA class and is easily converted to the method used in any of the other programs by appropriate choice of the unavailability parameters.

The unique features of PCAM, not available in any of its predecessors, are the following:

1. Capability to estimate queuing delays for a noninteger number of effective cars.
2. Improved estimates of travel times.
3. Ease of substitution of user terminology for the words "precinct," "division," and "tour."
4. Ease of manipulating the data base so as to focus on the time period(s) or precinct(s) desired.
5. Allocation of a specified number of car-hours across tours, or precincts, or both.
6. Choice permitted for the objective function to be minimized by the allocation.
7. Automatic calculation of feasible allocations in the case of an overlay tour. In previous programs, the period covered by an overlay tour could be artificially divided into shorter "tours," which are the same as time blocks in PCAM. However, the allocations recommended by the program for these artificial "tours" might be infeasible. As an example, consider three 8-hour tours beginning at 1600, 1900, and 2400. The artificial "tours" would be 1600-1900, 1900-2400, 2400-0300, and 0300-0800. If a program recommends 9 cars on duty from 1600-1900, 11 cars on duty from 1900-2400, 7 cars on duty from 2400-0300, and 3 cars on duty from 0300-0800, there is no way to achieve this allocation by starting cars on 8-hour tours at 1600, 1900, and 2400. The allocation is therefore *infeasible*. PCAM will not recommend such an allocation if the tour from 1900 to 0300 is treated as an overlay tour.

Appendix B

MATHEMATICAL FORMULATION OF PCAM'S CALCULATIONS

In this appendix we present the mathematical details of the calculations performed by PCAM. The reader is assumed to have some familiarity with queuing theory.*

QUEUING DELAYS

Single Hour

Consider a single hour in a single precinct. Calls for service are assumed to arrive according to a Poisson process at rate λ, and the service times of all calls are assumed to be independently exponentially distributed with mean $1/\mu$. Calls belong to one of three priority levels which are served out of queue in a first-come-first-served manner within priority level. Priority 1 calls are served before priority 2 calls, and priority 2 before priority 3. The probability that a call belongs to priority p is α_p, p = 1, 2, 3 ($\sum \alpha_p = 1$).

In a standard steady-state queuing formulation with a fixed number n of servers, the average number of servers busy is $\rho = \lambda/\mu$, which must be less than n; the probability that a call encounters a queue is given by Erlang's formula

$$P_Q(n) = P_o(n)\rho^n/(n!(1 - \rho/n))$$

where

$$P_o(n) = \left[\sum_{k=0}^{n-1} \frac{\rho^k}{k!} + \frac{\rho^n}{n!(1 - \rho/n)}\right]^{-1} ;$$

*There are many excellent textbooks on queuing theory, varying according to the assumed mathematical background of the reader. An example of a good introductory text is Kleinrock, Ref. 12.

the average time spent waiting in queue by a randomly selected call (including calls that do not wait) is

$$\bar{W}(n) = P_Q(n)/(n\mu(1 - \rho/n)) \ ;$$

and the average time spent waiting in queue by a priority p call is given by Cobham's formula [6]:

$$\bar{W}_p(n) = P_Q(n) \Bigg/ \left(n\mu \left(1 - \frac{\sum_{k=1}^{p} \alpha_k \lambda}{n\mu} \right) \left(1 - \frac{\sum_{k=1}^{p-1} \alpha_k \lambda}{n\mu} \right) \right) .$$

(The sum $\sum_{k=1}^{p-1}$ is zero if p = 1.)

However, in the context of patrol cars, the number of servers (n) is not fixed but varies stochastically due to the occurrence of non-cfs work. We model this phenomenon by calculating an effective number of patrol cars, N. If N is an integer, then we treat the system as if the number of servers is constant and equals N. If N is not an integer, say $N = N_o + \nu$ (where N_o is an integer and $0 < \nu < 1$), then we treat the system as a probabilistic mixture of N_o servers and $N_o + 1$ servers.

The number of effective servers is calculated as follows. If N_a is the actual number of cars fielded, then the fraction of time they spend on non-cfs work is assumed to equal

$$U = B1 \cdot \bar{\rho}/N_a + B2 \ ,$$

where B1 and B2 are unavailability parameters specific to the precinct in question, and $\bar{\rho}$ is the average number of servers busy in the time block to which the hour in question belongs.* (N_a need not be an integer, but may represent an average of the number of cars actually

*Suppose the block is H hours long, and λ_i and μ_i are the call rate and service rate, respectively, in hour i. Then $\bar{\rho} = (\sum_i \lambda_i/\mu_i)/H$.

fielded over a period of weeks or months.) The number of effective cars is then

$$N = (1 - U)N_a \ .$$

Writing, as above, $N = N_o + \nu$, where N_o is an integer and $0 \leq \nu < 1$, we require that $\rho/N_o < 1$. (In the text we have loosely referred to this condition as the requirement that the average utilization of an effective car must be less than 1. However, the average utilization of an effective car during the hour is really ρ/N.)

Then, we estimate the *probability that a call will encounter a queue* by linear interpolation:

$$P'_Q(N) = (1 - \nu)P_Q(N_o) + \nu\, P_Q(N_o + 1) \ .$$

The *average waiting time* is estimated as

$$W'(N) = P'_Q(N)/(N\mu(1 - \rho/N)) \ ,$$

and the *average waiting time for a priority p call* is estimated as

$$W'_p(N) = P'_Q(N) \Bigg/ \left(N\mu \left(1 - \frac{\sum_{k=1}^{p} \alpha_k \lambda}{N\mu}\right)\left(1 - \frac{\sum_{k=1}^{p-1} \alpha_k \lambda}{N\mu}\right)\right) .$$

These two equations are approximations to linear interpolation and are used instead of linear interpolation so as to reduce the amount of computer time needed to perform the calculation.

Additional queuing measures that have not been included in PCAM but could be programmed by the user, if desired, are the probability that a call waits longer than time T:

$$P(W > T) = P'_Q(N) \exp\ (- N\mu T(1 - \rho/N)) \ ,$$

and the probability that a call of priority p waits longer than T:

$$P(W_p > T) = \frac{P_Q'(N)}{\alpha_p} \exp(-N\mu T)$$

$$\cdot \left[\left(\sum_{k=1}^{p} \alpha_k \right) \exp\left(\sum_{k=1}^{p} \alpha_k \lambda T \right) - \left(\sum_{k=1}^{p-1} \alpha_k \right) \exp\left(\sum_{k=1}^{p-1} \alpha_k \lambda T \right) \right] .$$

Averages

PCAM averages queuing statistics over time blocks, shifts, tours, precincts, days, and divisions. To describe these averages, we shall imagine that each hour-precinct combination that enters into the average is indexed by j. Thus, $\lambda(j)$ is the call rate in the hour and precinct labeled j, $\mu(j)$ is the service rate, $\alpha_p(j)$ is the fraction of calls of priority p, $N_a(j)$ is the actual number of cars on duty, and $N(j)$ is the effective number of cars. ($\alpha_p(j)$, $N_a(j)$, and $N(j)$ do not change as j varies over hours belonging to a single time block in a single precinct.)

The *average probability that a call encounters a queue* and the *average waiting time* are calculated by weighting the hourly statistics by the call rate $\lambda(j)$. The *average waiting time for a priority p call* is calculated by weighting the hourly statistic by $\alpha_p(j)\lambda(j)$.

The *average utilization of an actual car* is

$$\left(\sum_j \lambda(j)/\mu(j) \right) \Big/ \sum_j N_a(j)$$

and the *average utilization of an effective car* is

$$\left(\sum_j \lambda(j)/\mu(j) \right) \Big/ \sum_j N(j) .$$

TRAVEL TIME

Consider a single hour in a single precinct, with call rate λ, service rate μ, N effective cars, area A (square miles), and response

speed v (miles per hour). The average number of cars available is then $M = N - \lambda/\mu$. If $M \geq 2$, the *average travel time* (in minutes) is estimated according to the square-root law

$$\bar{T} = 60 \cdot \frac{0.711}{v} \sqrt{\frac{A}{M}} .$$

If $M \leq 1$,

$$\bar{T} = 60 \cdot \frac{0.678}{v} \sqrt{A} .$$

If $1 < M < 2$, the estimate is a linear interpolation:

$$\bar{T} = \frac{60}{v} \sqrt{A} \left(0.080 + \frac{0.598}{\sqrt{M}} \right) .$$

(The number 60, which converts hours into minutes, is adjusted slightly in the model to account for street density.)

These equations were developed by Peter Kolesar [14] and incorporate findings from mathematical models [15,20], analysis of experimental data [13,16], and analysis of data constructed by using a patrol car simulation model [17].

Average over time blocks, precincts, etc., are constructed by weighting according to the call rate in each hour.

TOTAL DELAY

The *total delay* (or *total response time*) is the sum of waiting time and travel time.

PATROL FREQUENCY

Let $K(x,\tau)$ be the number of times a patrol car on preventive patrol passes point x in time period τ. The average patrol frequency in a precinct is defined to be

$$\bar{F} = \frac{1}{\tau} \text{ Avg } \{K(x,\tau) : x \in \text{precinct}\} .$$

If the number of street miles in the precinct is L, and patrol cars travel at speed s (miles per hour) while on preventive patrol, then the *average patrol frequency* (passings per hour) can be shown [20] to equal

$$\bar{F} = \frac{sM}{L} ,$$

where M is the average number of cars available during the period τ.

When averaging patrol frequency over precincts, PCAM calculates the patrol frequency in the geographical union of the precincts. (This is done by weighting each precinct's patrol frequency by the number of street miles in the precinct.)

PATROL HOURS PER SUPPRESSIBLE CRIME

Let C be the expected number of suppressible crimes in a precinct in an hour and M the average number of cars available $(N - \lambda/\mu)$. Then the average number of patrol hours per suppressible crime is M/C. When averaging over shifts, precincts, etc., the number of suppressible crimes is used as the weight.

PATROL FREQUENCY TIMES SUPPRESSIBLE CRIMES PER HOUR

As indicated by the name, this is simply $\bar{F}C$. It is averaged in the same manner as $\bar{F}$.

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. See Fig. B-1. The initial assignment is either the current allocation or the minimum number of cars needed to keep effective utilization under 1 in each hour, depending on previous calculations performed by the program. The constraints that are met on the last iteration are called *limiting constraints*.

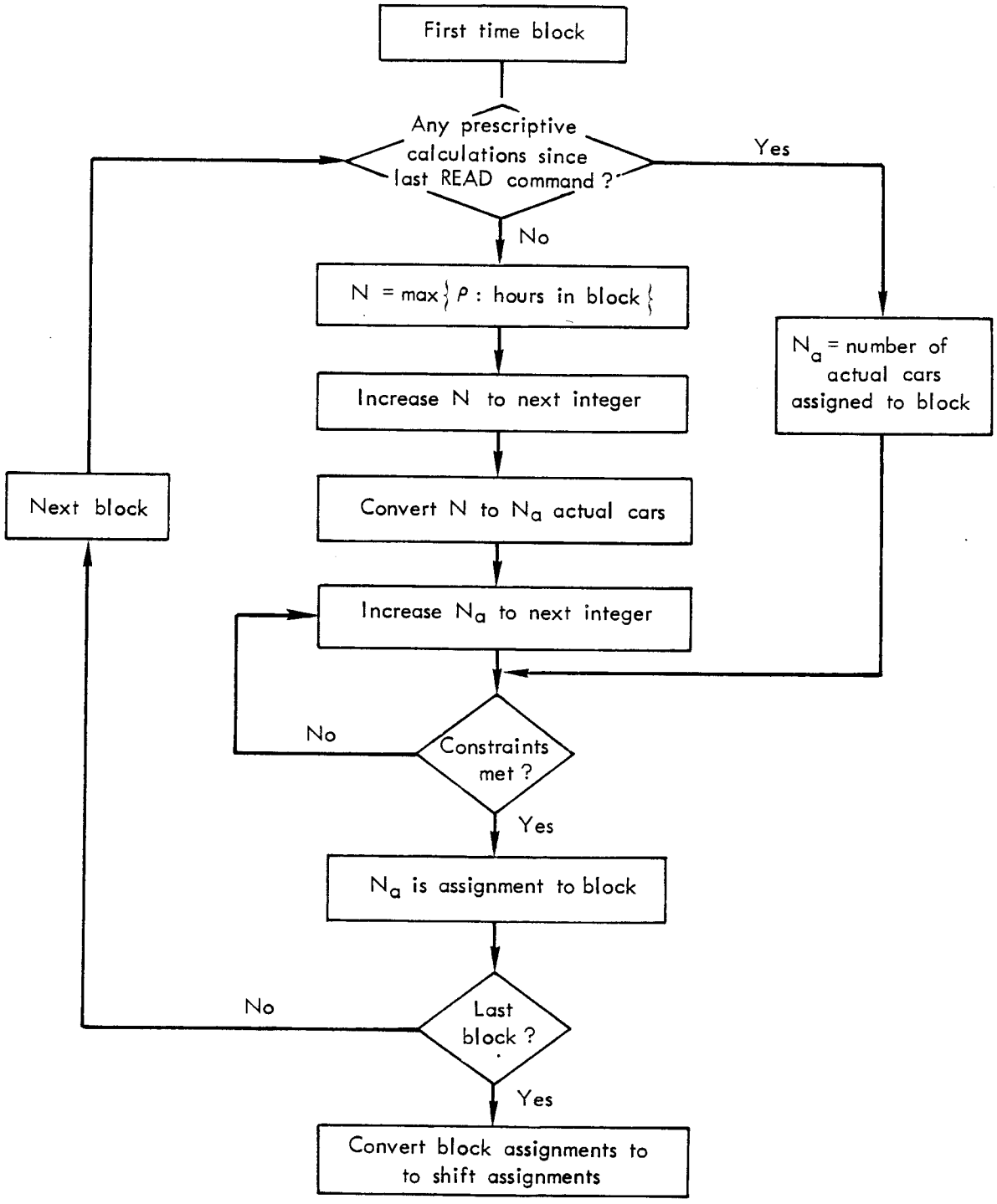


Fig. B-1 — Flow chart for meeting constraints

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

1. At least as many cars are assigned to each block as are required.
2. The tour 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 PCAM 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 be minimized by the allocation, which 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 (ADD command), or whether the allocation is to begin as if no cars were currently allocated (ALOC command).

PCAM then allocates an integer number of cars to each shift in such a way as to consume all the car-hours specified* and to minimize F. If we denote by B_1, B_2, ..., B_K the time blocks over which the allocation is to take place, then an *allocation* is defined to be a specification of the actual number of cars to be assigned to each block: n_1, n_2, ..., n_K. When there are no overlay tours, a time block is the same as a shift, and any combination of n_i's is feasible. When overlay tours

*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 in DATABASE has noninteger allocations, the user can, if he wishes, ask PCAM to add integers to the existing allocation, resulting in a noninteger allocation.

are present, the n_i's must satisfy certain compatibility conditions so that they correspond to cars beginning work only at the start of shifts.

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 has the following properties:

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

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

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 $B_1, \ldots, B_K$. PCAM's allocation algorithm begins with an initial allocation $n_1, n_2, \ldots, n_K$ 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. See Fig. B-2.

Then PCAM 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. PCAM 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. See Fig. B-3.

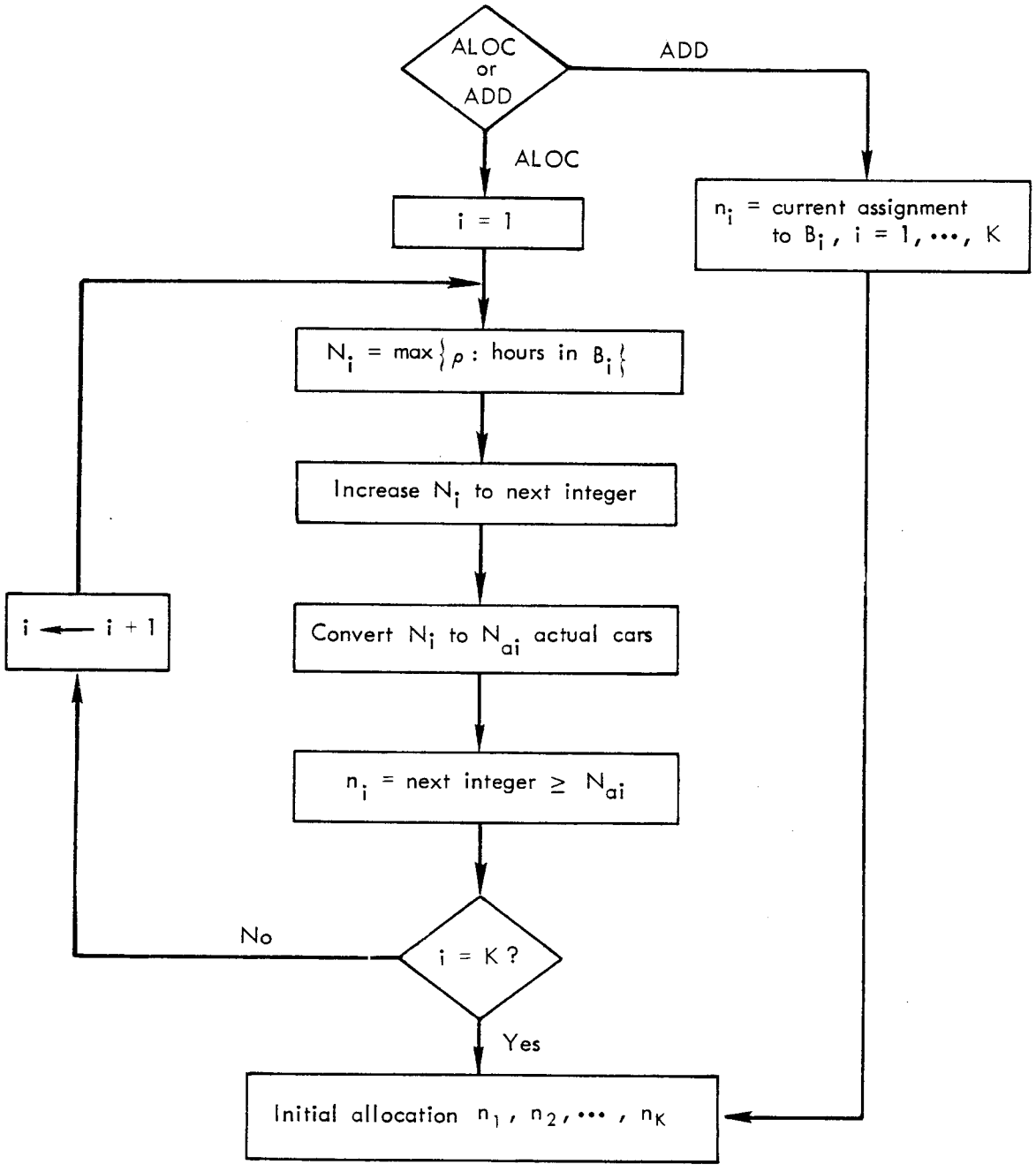


Fig. B-2 — Initial allocation: no overlay tours

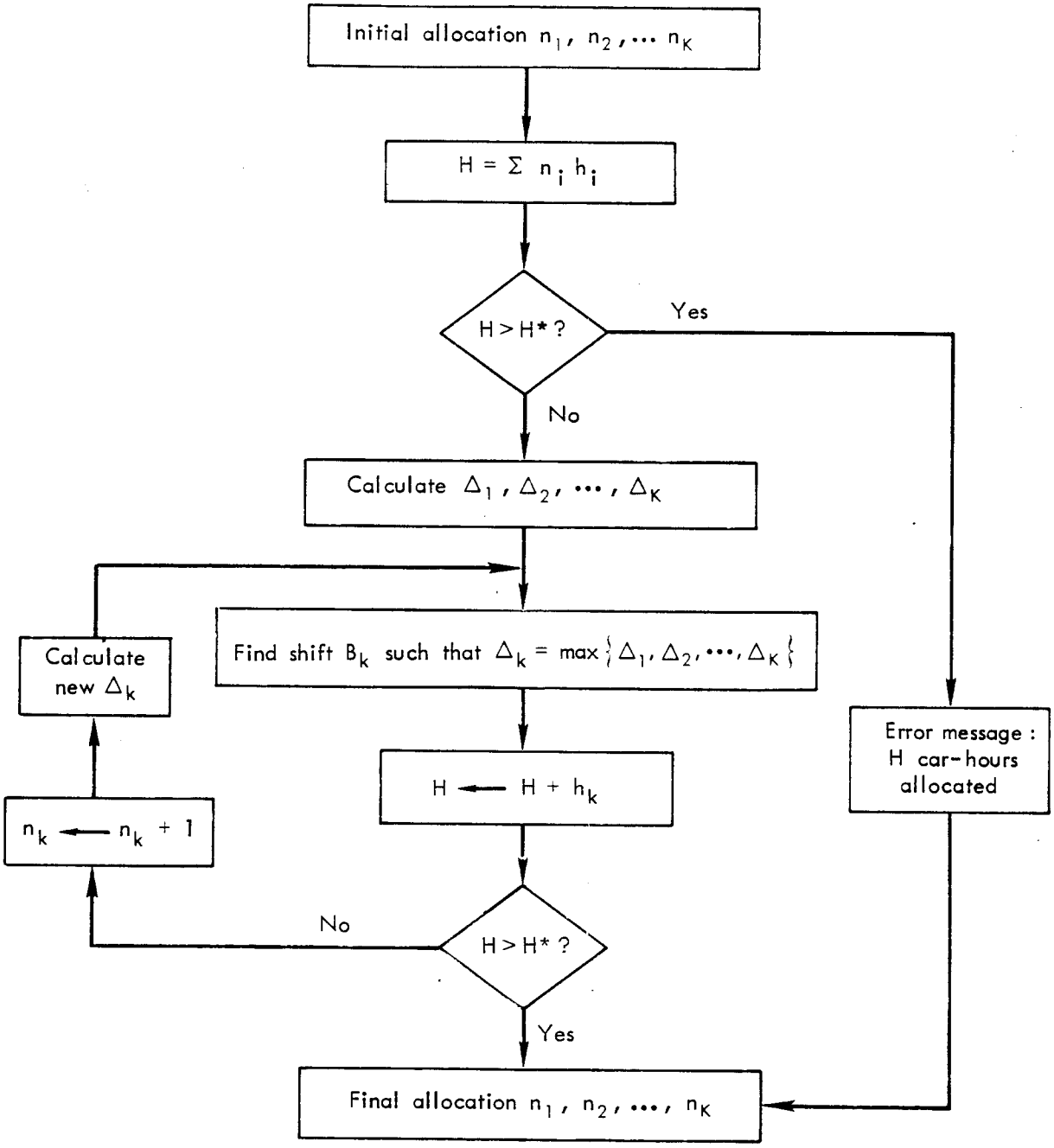


Fig. B-3 — Algorithm for allocating H* car-hours: no overlay tours

It is well known from the theory of optimization that this iterative process (marginal allocation) leads to an optimal solution because the objective function is *separable*. (That is, it is a sum of terms, each of which is a function of only one of the decision variables n_i. See Property 1.) However, for completeness we shall give a simple proof of optimality here. It is convenient to take into account the fact that the algorithm begins with an initial allocation and the fact that we are allocating car-hours rather than cars by defining new variable m_i = the number of car-hours added to shift B_i above and beyond the initial allocation. Thus, if n_i^o is the number of cars initially assigned to shift B_i, and n_i is the variable representing the number of cars assigned to shift B_i, we have

$$m_i = (n_i - n_i^o)h_i \ .$$

In terms of the variables $m_1, \ldots, m_K$, the objective function is $G(m_1, \ldots, m_K) = F(n_1^o + m_1/h_1, n_2^o + m_2/h_2, \ldots, n_K^o + m_K/h_K)$, and the number of additional car-hours allocated is $M = \sum m_i$.

We now modify the objective function in two ways that have no effect on the optimization problem but simplify the argument. First, since we are not allowing any of the m_i's to be negative, we extrapolate the functions f_i linearly for values of n_i less than n_i^o in such a way that $G(m_1, \ldots, m_K)$ is very large if any of the m_i's are negative integers. (For example, make $G(-1, 0, 0, \ldots, 0) > 2G(0, 0, 0, \ldots, 0)$ and extend linearly. Do the same with $G(0, -1, 0, \ldots, 0)$, etc.) Then the initial allocation ($m_1 = m_2 = \ldots = m_K = 0$) is the *optimal* allocation having $\sum m_i = 0$.

Second, we adjust the f_i's so that G varies linearly as m_i varies between integer multiples of h_i. (This has no substantive effect, since we are only interested in allocations where every m_i is an integer multiple of h_i.)

Now we are ready to prove the following:

Proposition 1. Suppose $\underline{m}^* = (m_1^*, m_2^*, \ldots, m_K^*)$ is an optimal allocation having M added car-hours and all the m_i's integers. (I.e., $\sum m_i^* = M$,

and $G(m_1^*, m_2^*, \ldots, m_K^*) \leq G(m_1, m_2, \ldots, m_K)$ for any integers $m_1, \ldots, m_K$ such that $\sum m_i = M$.) Then an optimal allocation having $M + 1$ added car-hours can be found by adding 1 car-hour to $\underline{m}^*$ in the shift i with the largest value of $\Delta_i = w_i(f_i(n_i^*) - f_i(n_i^* + 1))/h_i$, where $n_i^* = n_i^o + [m_i^*/h_i]$. (The square brackets denote "the integer part of.")

Proof:

Allocations that differ from $\underline{m}^*$ by 1 car-hour in one shift have the form $\underline{m}^* + \underline{\delta}_i$, where $\underline{\delta}_i = (0, 0, \ldots, 1, \ldots 0)$, with the "1" in the i*th* position. First we show that for any allocation $\underline{m}$ with $\sum m_i = M + 1$, there is an allocation of the form $\underline{m}^* + \underline{\delta}_i$ that is at least as good.

Write $\underline{m}$ in the form $\underline{m} = \underline{m}^* + \underline{\alpha}$, where $\sum \alpha_i = 1$. Then one of the α_i's, say α_k, must be positive. We claim that

$$G(\underline{m}^* + \underline{\delta}_k) \leq G(\underline{m}) \ .$$

To see this, let $\underline{\beta} = \underline{\alpha} - \underline{\delta}_k$. Then $\sum \beta_i = 0$, and so

$$G(\underline{m}^* + \underline{\beta}) \geq G(\underline{m}^*)$$

because $\underline{m}^*$ is optimal. But

$$G(\underline{m}) = G(\underline{m}^* + \underline{\beta} + \underline{\delta}_k) = G(\underline{m}^* + \underline{\beta}) + g_k(m_k^* + \beta_k + 1) - g_k(m_k^* + \beta_k) \ ,$$

where $g_k(x) = w_k f_k(n_k^o + x/h_k)/\sum w_i$. Therefore,

$$\begin{aligned} 0 &\leq G(\underline{m}^* + \underline{\beta}) - G(\underline{m}^*) \\ &= G(\underline{m}) - g_k(m_k^* + \beta_k + 1) + g_k(m_k^* + \beta_k) - G(\underline{m}^*) \\ &= G(\underline{m}) - G(\underline{m}^* + \underline{\delta}_k) \\ &\quad - g_k(m_k^* + \beta_k + 1) + g_k(m_k^* + \beta_k) \\ &\quad + g_k(m_k^* + 1) - g_k(m_k^*) \ . \end{aligned}$$

Because $\beta_k \geq 0$, Property 2 of the f_i's tells us that

$$g_k(m_k^*) - g_k(m_k^* + 1) \geq g_k(m_k^* + \beta_k) - g_k(m_k^* + \beta_k + 1) \ .$$

Therefore,

$$0 \leq G(\underline{m}) - G(\underline{m}^* + \underline{\delta}_k) \ ,$$

which is what we wanted to show.

Now we know that one of the following is an optimal allocation: $\underline{m}^* + \underline{\delta}_1$, $\underline{m}^* + \underline{\delta}_2$, ..., $\underline{m}^* + \underline{\delta}_K$. The lowest value of the objective function for these K possibilities is

$$\begin{aligned}
\min_k G(\underline{m}^* + \underline{\delta}_k) &= \min_k \{G(\underline{m}^* + \underline{\delta}_k) - G(\underline{m}^*)\} + G(\underline{m}^*) \\
&= G(\underline{m}^*) - \max_k \{G(\underline{m}^*) - G(\underline{m}^* + \underline{\delta}_k)\} \\
&= G(\underline{m}^*) - \max_k \{g_k(m_k^*) - g_k(m_k^* + 1)\} \\
&= G(\underline{m}^*) - \max_k \left\{ w_k \left(f_k(n_k^o + m_k^*/h_k) - f_k(n_k^o + \frac{m_k^* + 1}{h_k}) \right) \right\} \Big/ \sum w_i
\end{aligned}$$

Because of the assumed linear form of the f_k's, the quantity in brackets to be maximized is the same as Δ_k, and therefore the proposition is proved.

** END OF PROOF **

The result of the proposition is that, starting with the initial allocation, which is optimal, an optimal allocation with one more car-hour is found by adding one car-hour to the shift with the largest value of Δ_i. If this shift has more than one hour in it, the next car-hour should be added to the same shift, because Δ_i will not have changed, and so on until a whole car has been added to the shift. We then arrive at an optimal allocation having h_i additional car-hours.

At this point, the procedure we used to get this allocation (namely, adding one car-hour at a time) is irrelevant to the fact that

it is optimal. Moreover, if we restrict attention to those allocations that have all m_i's equal to integer multiples of h_i (i.e., integer numbers of added cars in each shift), this allocation is also clearly optimal. So the PCAM algorithm simply arrives at this allocation in one step, instead of hour by hour.

Beginning with this new optimal allocation, the proposition states that one car-hour should be added to whatever shift now has the largest value of Δ_i, and the process begins all over. This is exactly what the PCAM algorithm does.

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 shown in Fig. B-2, 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 must be exercised in selecting this 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 *shifts* instead of time blocks, it turns out that the procedure based on Proposition 1 *does not work*. To be more specific, it is *not* 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 proposition 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, it is not even clear how one should define the word "optimal." Figure B-4 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 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 *smaller* number of car-hours (namely 188) can be allocated to give a *lower* value of the objective function F_1. In fact, the police department should only be interested in the allocations which have been joined together by a line in Fig. B-4.

Since we do not know of any police department having an overlay tour which differs in length from the tours it overlays, we have not attempted to resolve all these problems in full generality. Instead, we have developed an algorithm which is optimal when the overlay tour is the same length as the overlaid tours and behaves "sensibly" in other cases. For example, the allocation algorithm only recommends allocations that lie on the line drawn in Fig. B-4, and it follows the

---

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

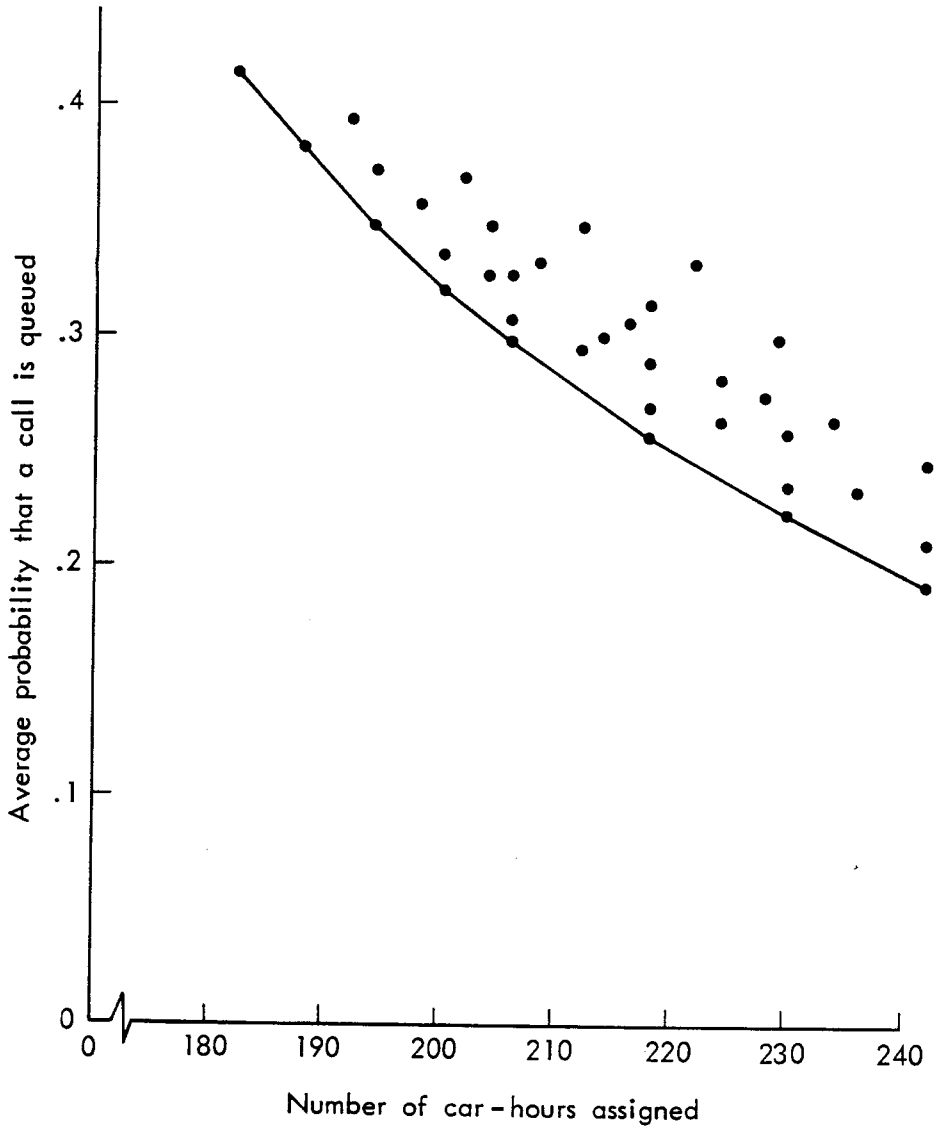


Fig. B-4 — Example of an overlay segment in which the tours do not have the same length

analogous line* in all other examples that we have tested. But we have not *proved* that it will always behave in this fashion when the tours have different lengths.

The algorithm works in the following way. The *initial allocation* for each time block is found in the manner shown in Fig. B-2. Then, in the case of the ALOC command, 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 meet condition 1.

Then the algorithm iteratively adds car-hours in a manner similar to the one shown in Fig. B-3, but more elaborate. The algorithm in Fig. B-3 checks the change in the weighted objective function per car-hour added, assuming that one car is added to each shift in turn. The full algorithm does this also, and for each overlay segment it also checks the possibility that 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

---

*This line is the graph of the maximal convex function φ having the property that $\varphi(x)$ is less than or equal to the value of the objective function for every feasible allocation having x car-hours.

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

(The algorithm shown in Fig. B-3 would only check the first three of these possibilities.)

We shall now show that this algorithm is optimal when all the tours in an overlay segment have the same length h. In this case the total number of car-hours assigned to an overlay segment is some integral multiple of h, and every integral multiple of h car-hours (starting with some minimum) is achievable. For this reason, the ambiguity about the meaning of the word "optimal" that we illustrated in Fig. B-4 no longer arises. In fact, if $f_i(n_{i1}, n_{i2}, n_{i3})$ is the average value of the objective function over the *ith* overlay segment when n_{i1} cars are assigned to the first overlaid tour, n_{i2} to the second overlaid tour, and n_{i3} to the overlay tour, then the optimal value of the objective function for a total of $n_i h$ car-hours in the segment is

$$f_i^*(n_i) = \min \{f_i(n_{i1}, n_{i2}, n_{i3}): \ n_{i1} + n_{i2} + n_{i3} = n_i\} .$$

The function f_i^* is convex decreasing and permits identifying an allocation n_{i1}, n_{i2}, n_{i3} as optimal if $f_i(n_{i1}, n_{i2}, n_{i3}) = f_i^*(n_{i1} + n_{i2} + n_{i3})$.

Suppose we are allocating H^* car-hours over shifts $B_1, B_2, \ldots, B_K$ that are not involved in overlay segments, plus overlay segments $S_1, S_2, \ldots, S_L$. (See Fig. B-5.) Then the objective function for the allocation is a weighted average of the objective functions for $B_1, \ldots, B_K$ and $S_1, \ldots, S_L$. Now it is apparent that the overall objective function cannot be minimized unless the assignment to S_i achieves the value $f_i^*(n_i)$ for some n_i, $i = 1, \ldots, L$. Hence, minimizing the

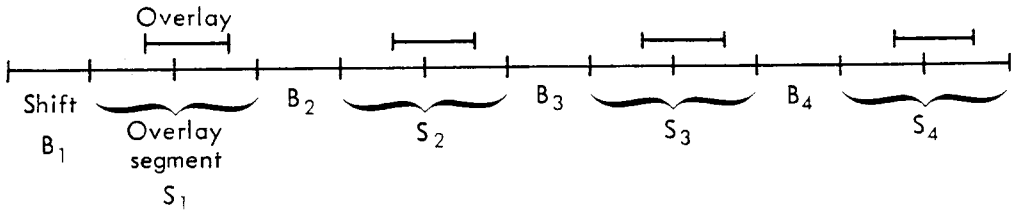


Fig. B-5 — Example of shifts and overlay segments

overall objective function is equivalent to minimizing

$$\left[\sum_{j=1}^{K} w_{B_j} f_{B_j}(n_{B_j}) + \sum_{i=1}^{L} w_i f_i^*(n_i)\right] \Bigg/ \left(\sum_{j=1}^{K} w_{B_j} + \sum_{i=1}^{L} w_i\right)$$

subject to the condition $\sum n_{B_j} h_j + \sum n_i h = H^*$. (Here f_{B_j} is the average of the objective function over shift B_j, and w_{B_j} is the corresponding weight.)

This function has the same form and properties as the objective function which we discussed in the previous section ("With No Overlay Tours"), and therefore the algorithm that iteratively calculates the improvement in the weighted objective function per car-hour added will also work here. We have therefore reduced the optimization problem to one of determining the allocation that achieves the value $f_i^*(n_i)$ for every n_i. In other words, once we show how to calculate the optimal allocation of n_i cars to a single overlay segment S_i, the problem of optimally allocating car-hours among various shifts and overlay segments is solved by the algorithm already described in the previous section.

Consider, then, an overlay segment S having three equal-length shifts. (We drop the subscript i.) An allocation to S is a three-tuple

$$\underline{n} = (n_1, n_2, n_3) ,$$

and the average value of the objective function over S has the form

$$f(\underline{n}) = g_1(n_1) + g_2(n_1 + n_3) + g_3(n_2 + n_3) + g_4(n_3) \ ,$$

where $g_j(x)$ is the weighted average objective function for the j*th* time block in segment S when x cars are assigned. Each g_j is convex decreasing. As in the previous section, $g_j(x)$ is assumed to be defined for values of x smaller than the minimum number of cars permitted in the j*th* time block, by linear extension of g_j. The extended values of g_j are larger than $2f(\underline{n})$, when $\underline{n}$ is the minimal permitted allocation.

Suppose $\underline{n}^o = (n_1^o, n_2^o, n_3^o)$ is an optimal assignment of N cars ($n_1^o + n_2^o + n_3^o = N$). Then the collection of all possible assignments of N cars forms a flat 2-dimensional lattice through $\underline{n}^o$. Selecting a basis for this lattice

$$\underline{M}_1 = (-1, 0, 1)$$

$$\underline{M}_2 = (0, 1, -1) \ ,$$

we can express every possible assignment of N cars in the form

$$\underline{n} = \underline{n}^o + \alpha_1 \underline{M}_1 + \alpha_2 \underline{M}_2 \ ,$$

where α_1 and α_2 are integers. To say $\underline{n}^o$ is optimal means that $f(\underline{n}) \geq f(\underline{n}^o)$ for all integer values of α_1 and α_2.

Proposition 2. If $\underline{n}^o$ is an optimal assignment of N cars to overlay segment S then one of the following is an optimal assignment of N + 1 cars:

$$\underline{n}^o + (1, 0, 0)$$

$$\underline{n}^o + (0, 1, 0)$$

$$\underline{n}^o + (0, 0, 1)$$

$$\underline{n}^o + (1, 1, -1) \ .$$

Proof:

Let $\underline{n}^1 = \underline{n}^o + (1, 0, 0) = (n_1^o + 1, n_2^o, n_3^o)$. Then a restatement of the proposition is that for any $\underline{n}$ of the form

$$\underline{n} = \underline{n}^1 + \alpha_1 \underline{M}_1 + \alpha_2 \underline{M}_2$$

there is an $\underline{n}' = \underline{n}^1 + \beta_1 \underline{M}_1 + \beta_2 \underline{M}_2$ with $\beta_1 = 0$ or 1, $\beta_2 = 0$ or 1, and $f(\underline{n}') \leq f(\underline{n})$. To see that this is a restatement, simply note that

$$\underline{n}^1 + 0 \cdot \underline{M}_1 + 0 \cdot \underline{M}_2 = \underline{n}^o + (1, 0, 0)$$

$$\underline{n}^1 + 1 \cdot \underline{M}_1 + 0 \cdot \underline{M}_2 = \underline{n}^o + (0, 0, 1)$$

$$\underline{n}^1 + 0 \cdot \underline{M}_1 + 1 \cdot \underline{M}_2 = \underline{n}^o + (1, 1, -1)$$

$$\underline{n}^1 + 1 \cdot \underline{M}_1 + 1 \cdot \underline{M}_2 = \underline{n}^o + (0, 1, 0) .$$

To prove the restated proposition, we shall show that if

$$\underline{n} = \underline{n}^1 + \alpha_1 \underline{M}_1 + \alpha_2 \underline{M}_2$$

with $\alpha_1 \geq 2$ or $\alpha_1 \leq -1$ or $\alpha_2 \geq 2$ or $\alpha_2 \leq -1$, then there is an

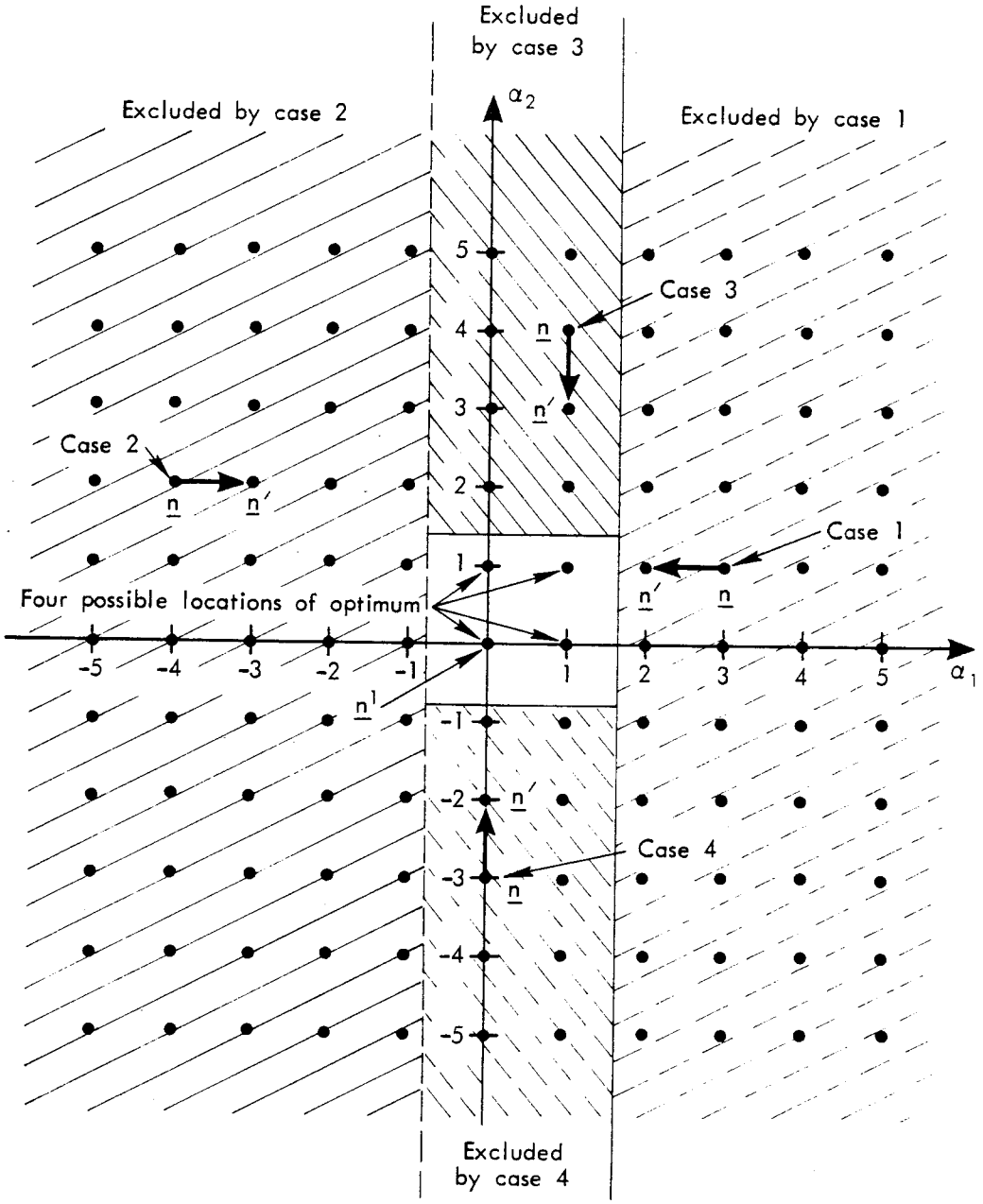
$$\underline{n}' = \underline{n}^1 + \beta_1 \underline{M}_1 + \beta_2 \underline{M}_2$$

with $\beta_1^2 + \beta_2^2 < \alpha_1^2 + \alpha_2^2$ and $f(\underline{n}') \leq f(\underline{n})$. By iteration this proves the desired result. See Fig. B-6. (We shall refer to the condition $\beta_1^2 + \beta_2^2 < \alpha_1^2 + \alpha_2^2$ by saying "$\underline{n}'$ is closer to $\underline{n}^1$ than $\underline{n}$ is.")

Case 1: $\underline{n} = \underline{n}^1 + \alpha_1 \underline{M}_1 + \alpha_2 \underline{M}_2$ with $\alpha_1 \geq 2$. Then

$$f(\underline{n}) = g_1(n_1^o + 1 - \alpha_1) + g_2(n_1^o + n_3^o + 1 - \alpha_2)$$

$$+ g_3(n_2^o + n_3^o + \alpha_1) + g_4(n_2^o + \alpha_2) .$$


Excluded
by case 3
Excluded by case 2
Excluded by case 1
α_2
α_1
Case 3
Case 2
Case 1
Case 4
Four possible locations of optimum
n
n′
n′
n
n
n′
n′
n
n′
n
n′
n

Consider $\underline{n}' = \underline{n} - \underline{M}_1$. Then $\underline{n}'$ is closer to $\underline{n}^1$ than $\underline{n}$ is. Also,

$$f(\underline{n}) - f(\underline{n}') = g_1(n_1^o + 1 - \alpha_1) - g_1(n_1^o + 2 - \alpha_1)$$
$$+ g_3(n_2^o + n_3^o + \alpha_1) - g_3(n_2^o + n_3^o + \alpha_1 - 1) .$$

Using the convexity of the g_i's, we have

$$f(\underline{n}) - f(\underline{n}') \geq g_1(n_1^o - 1) - g_1(n_1^o)$$
$$- [g_3(n_2^o + n_3^o) - g_3(n_2^o + n_3^o + 1)]$$
$$= f(\underline{n}^o + \underline{M}_1) - f(\underline{n}^o) .$$

Using the optimality of $\underline{n}^o$, we know $f(\underline{n}^o + \underline{M}_1) \geq f(\underline{n}^o)$, so

$$f(\underline{n}) - f(\underline{n}') \geq 0 .$$

Case 2: $\underline{n} = \underline{n}^1 + \alpha_1 \underline{M}_1 + \alpha_2 \underline{M}_2$ with $\alpha_1 \leq -1$.

Consider $\underline{n}' = \underline{n} + \underline{M}_1$, which is closer to $\underline{n}^1$ than $\underline{n}$ is.

$$f(\underline{n}) - f(\underline{n}') = g_1(n_1^o + 1 - \alpha_1) - g_1(n_1^o - \alpha_1)$$
$$+ g_3(n_2^o + n_3^o + \alpha_1) - g_3(n_2^o + n_3^o + \alpha_1 + 1)$$
$$\geq - [g_1(n_1^o) - g_1(n_1^o + 1)]$$
$$+ g_3(n_2^o + n_3^o - 1) - g_3(n_2^o + n_2^o)$$
$$= f(\underline{n}^o - \underline{M}_1) - f(\underline{n}^o) \geq 0 .$$

Case 3: $\underline{n} = \underline{n}^1 + \alpha_1 \underline{M}_1 + \alpha_2 \underline{M}_2$ with $\alpha_2 \geq 2$.

Consider $\underline{n}' = \underline{n} - \underline{M}_2$, which is closer to $\underline{n}^1$ than $\underline{n}$ is.

$$f(\underline{n}) - f(\underline{n}') = g_2(n_1^o + n_3^o + 1 - \alpha_2) - g_2(n_1^o + n_3^o + 2 - \alpha_2)$$

$$+ g_4(n_2^o + \alpha_2) - g_4(n_2^o + \alpha_2 - 1)$$

$$\geq g_2(n_1^o + n_3^o - 1) - g_2(n_1^o + n_3^o)$$

$$- [g_4(n_2^o) - g_4(n_2^o + 1)]$$

$$= f(\underline{n}^o + \underline{M}_1) - f(\underline{n}^o) \geq 0 \ .$$

Case 4: $\underline{n} = \underline{n}^1 + \alpha_1 \underline{M}_1 + \alpha_2 \underline{M}_2$ with $\alpha_2 \leq -1$.

Consider $\underline{n}' = \underline{n} + \underline{M}_2$, which is closer to n^1 than n is.

$$f(\underline{n}) - f(\underline{n}') = g_2(n_1^o + n_3^o + 1 - \alpha_2) - g_2(n_1^o + n_3^o - \alpha_2)$$

$$+ g_4(n_2^o + \alpha_2) - g_4(n_2^o + \alpha_2 + 1)$$

$$\geq - [g_2(n_1^o + n_3^o) - g_2(n_1^o + n_3^o + 1)]$$

$$+ g_4(n_2^o - 1) - g_4(n_2^o)$$

$$= f(\underline{n}^o - \underline{M}_2) - f(\underline{n}^o) \geq 0 \ .$$

** END OF PROOF **

By virtue of the way in which we have extended $g_j(x)$ for values of x smaller than the number of cars permitted in time block j, the initial allocation selected by PCAM is optimal in the overlay segment S. Since S is arbitrary, we shall restore the subscript and call it S_i. If $n_i h$ is the number of car-hours in the initial allocation, we

now know $f_i^*(n_i)$. Proposition 2 states that only four possibilities must be checked to find $f_i^*(n_i + 1)$:

- Add one car to the first overlaid tour.
- Add one car to the second overlaid tour.
- Add one car to the overlay tour.
- Subtract one car from the overlay tour and add one to each of the others.*

Since the PCAM algorithm checks all of these, it finds $f_i^*(n_i + 1)$. By iteration, it finds the entire function f_i^*. As we noted earlier, if the algorithm can calculate all the functions f_i^*, then it is optimal. Hence we have proved that the PCAM algorithm is optimal when overlay tours have the same length as the tours they overlay.

---

*If no cars are assigned to the overlay tour, this possibility is excluded.

Appendix C

PCAM REFERENCE SHEETS

GENERAL

AMPERSAND (&): At end of line, signifies command continues on the following line.

DELIMITER: Any character other than a letter, digit, parenthesis, hyphen, period, asterisk, or ampersand. Can be used freely to improve readability. Examples: blank, comma, colon, semicolon, equal sign.

FILLER WORD: FOR, CAR, CARS, HOUR, HOURS, TO, ON, BY, DATA. Ignored by program.

QUALIFIER: Any combination of

TOUR=<NAMELIST>
DIVISION=<NAMELIST>
PRECINCT=<NAMELIST>
DAY=<NAMELIST>

The words on the left of the equal sign are supplied by the user, except for DAY. Qualifier may be omitted in any command.

COMMANDS

1. READ [DATA] [FOR] <QUALIFIER>

   Reads data from DATABASE into CURRENT-DATA, establishes default output order for DISP, and increases number of cars assigned (if necessary) to assure that the utilization of an effective car is under 1 in every hour. The first command in any run of PCAM must be READ.

2. LIST [DATA] [FOR] <QUALIFIER>

   Lists data from CURRENT-DATA. Averages some data.

3. DISP T<NUMBERLIST> [FOR] <QUALIFIER>

   Displays Table 1 or Table 2 or both, depending on <NUMBERLIST>. <QUALIFIER> establishes the output order as well as the scope.

4. ALOC <NUMBER> [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   ALOC * [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   ALOC * -<NUMBER> [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   Allocates the specified number of car-hours so as to minimize F<NUMBERLIST>. Asterisk (*) represents the number currently assigned. At a minimum, allocates enough cars so that the utilization of an effective car is under 1 in every hour.

5. ADD <NUMBER> [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   ADD <NUMBER> -* [CAR] [HOURS] [TO] <QUALIFIER> [BY] F<NUMBERLIST>

   Adds the specified number of car-hours to the number currently assigned so as to minimize F<NUMBERLIST>. In the second version, execution of the command will result in the total number of car-hours assigned equaling <NUMBER>.

6. MEET C<NUMBERLIST>$_1$=<NUMBERLIST>$_2$ [FOR] <QUALIFIER>

   Assigns enough car-hours to each specified shift to assure that the measures indicated in <NUMBERLIST>$_1$ meet the constraints in <NUMBERLIST>$_2$ for every time block. One constraint value must be specified for each measure.

   If no ALOC, ADD, or MEET commands have been entered since the last READ command, MEET assigns the minimum number of car-hours needed to meet constraints and keep utilization of an effective car under 1 in every hour. Otherwise, car-hours are added to those already allocated, if needed to meet the constraints.

7. SET P<NUMBERLIST>$_1$=<NUMBERLIST>$_2$ [FOR] <QUALIFIER>

   Changes specified data items. There must be a one-to-one correspondence between data items in <NUMBERLIST>$_1$ and values in <NUMBERLIST>$_2$. SET also checks that enough car-hours are assigned to each shift so as to keep the utilization of an effective car under 1 in each hour.

8. WRITE [DATA] [ON] <NUMBER> [FOR] <QUALIFIER>

   Writes a NEW-DATA file on Fortran unit <NUMBER>. NEW-DATA contains the part of CURRENT-DATA specified by <QUALIFIER>.

9. END

   Terminates program. Must be last command.

OBJECTIVE FUNCTIONS FOR 'ALOC' AND 'ADD'

F(1)	Average fraction of calls delayed in queue
F(2)	Average length of time calls are delayed in queue
F(2,N)	Average length of time priority N calls are delayed in queue
F(3)	Average total response time (queuing + travel time)

CONSTRAINT SPECIFICATIONS FOR 'MEET'

C(1)	Average utilization of an effective car	↓
C(2)	Average travel time (minutes)	↓
C(3)	Average number of cars available (same as average patrol hours per hour)	↑
C(4)	Patrol hours per suppressible crime	↑
C(5)	Patrol frequency (passings per hour)	↑
C(6)	Minimum number of cars	↑
C(7)	Fraction of calls delayed	↓
C(8)	Average delay, priority 2 (minutes)	↓
C(9)	Average delay, priority 3 (minutes)	↓
C(10)	Average response time (queuing + travel) (minutes)	↓

DATA ITEMS FOR 'SET'

P(1)	Unavailability parameter B1	(precinct)
P(2)	Unavailability parameter B2	(precinct)
P(3)	Call rate parameter	(day, in precinct)
P(4)	Service time parameter	(day, in precinct)
P(5)	Actual cars assigned	(shift)
P(6)	Response speed	(shift)
P(7)	Patrol speed	(shift)
P(8)	Fraction of calls priority 1	(shift)
P(9)	Fraction of calls priority 2	(shift)
P(10)	Total number of suppressible crimes	(shift)

SAMPLE SEQUENCE OF COMMANDS

Command	Explanation
READ DATA FOR DIVISION=HIGHLAND	Data covering an entire week in Highland Division are read into CURRENT-DATA.
SET P(3)=7.85 FOR PRECINCT=EAST SET P(3)=5.45 FOR PRECINCT=NORTH	Call rates have increased slightly since the last time PCAM was used. These commands adjust the call rates.
DISP T(1,2)	User wants to see what has happened to performance measures with the new call rates.
ALOC 4000 CAR HOURS BY F(2) DISP T 2	User wants to see how to allocate the number of car-hours now planned for this Division (4000) so as to minimize average queuing delay.
ALOC * BY F(3) DISP T 2	Morning tours (with fast travel speed) appear to have unnecessarily low response times in the previous allocation, while most tours are too high. User attempts to minimize average response time, but finds he gets nearly the same allocation.
ADD 128 BY F(3) DISP T 2	User wants to see how much improvement can be obtained by allocating 4 more patrol officers (performing patrol car duty 32 hours a week) to this Division.
MEET C(10)=21	Response times are still too high. User wants to know how many additional car-hours are needed to keep response time under 21 minutes in every shift.
READ DIVISION=HIGHLAND SET P(3)=7.85 PRECINCT=EAST SET P(3)=5.45 PRECINCT=NORTH MEET C(10)=21	Number of car-hours needed in the previous allocation is too large. User starts over, trying to meet constraints *before* allocating. He uses the short form of the commands this time.
ADD 4000-* CAR HOURS BY F(3)	The user had some car-hours left over after meeting the constraint. Now he allocates the remainder so as to minimize average travel time.
DISP T(1,2) WRITE DATA ON 18	The user thinks this is a good allocation and writes out a NEW-DATA file.
END	User terminates this session with PCAM.

Appendix D

ADDRESSES FOR FURTHER INFORMATION

1. For copies of the PCAM program on card or tape, answers to questions about the program, and information about related emergency service deployment models:

Dr. Jan M. Chaiken
The Rand Corporation
1700 Main Street
Santa Monica, California 90406
(213) 393-0411

2. Research sponsors:

U.S. Department of Housing and Urban Development

Alan Siegel, Director
Hartley Campbell Fitts, Program Manager

Office of Policy Development and Research
Community Development and Management
Research Division
451 Seventh Street, S.W.
Washington, D.C. 20410
(202) 755-6970

National Institute of Law Enforcement and Criminal Justice

Dr. Richard Linster
Law Enforcement Assistance Administration
National Institute of Law Enforcement and
Criminal Justice
Office of Evaluation
633 Indiana Avenue, N.W.
Washington, D.C. 20530
(202) 376-3933

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., *Hypercube Queuing Model: Executive Summary*, The Rand Corporation, R-1688/1-HUD, 1975.

3. -----, *Patrol Allocation Methodology for Police Departments*, The Rand Corporation, R-1852-HUD, 1975.

4. Chaiken, Jan M., and Richard Larson, "Methods for Allocating Urban Emergency Units: A Survey," *Management Science*, Vol. 19, 1972, pp. P-110-130.

5. Clawson, Calvin, and Samson Chang, *Impact of Response Delays on Arrest Rates*, Seattle Police Department, 1975. (Submitted to *Management Science*.)

6. Cobham, A., "Priority Assignment in Waiting Line Problems," *Operations Research*, Vol. 2, 1954, p. 70.

7. Crowther, Richard F., *The Use of a Computer System for Police Manpower Allocation in St. Louis, Missouri*, Department of Police Administration, Indiana University, 1964.

8. 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.

9. 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.

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

11. Kelling, George L., Tony Pate, Duane Dieckman, and Charles Brown, *The Kansas City Preventive Patrol Experiment*, Police Foundation, 1909 K Street, N.W., Washington, D.C. 20006, 1975.

12. Kleinrock, Leonard, *Queuing Systems, Volume I: Theory*, John Wiley & Sons, New York, 1975.

13. Kolesar, Peter, *A Model for Predicting Average Fire Company Travel Times*, The Rand Corporation, R-1624-NYC, 1975.

14. Kolesar, Peter, private communication.

15. Kolesar, Peter, and Edward Blum, "Square Root Laws for Fire Engine Response Distances," *Management Science*, Vol. 19, 1973, pp. 1368-1378.

16. Kolesar, Peter, and Warren Walker, *Measuring the Travel Characteristics of New York City's Fire Companies*, The Rand Corporation, R-1449-NYC, April 1974.

17. Kolesar, Peter, and Warren Walker, *A Simulation Model of Police Patrol Operations*, The Rand Corporation, R-1625/1-HUD (*Executive Summary*), R-1625/2-HUD/NYC (*Program Description*), 1975.

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

19. -----, *Models for the Allocation of Urban Police Patrol Forces*, MIT Operations Research Center Technical Report No. 44, 1969.

20. -----, *Urban Police Patrol Analysis*, MIT Press, Cambridge, Massachusetts, 1972.

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

22. -----, 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.

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

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

25. 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.

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