# CRIMINAL JUSTICE MODELS: AN OVERVIEW

NATIONAL INSTITUTE OF LAW ENFORCEMENT AND CRIMINAL JUSTICE
LAW ENFORCEMENT ASSISTANCE ADMINISTRATION
UNITED STATES DEPARTMENT OF JUSTICE

, JME



# CRIMINAL JUSTICE MODELS:

# AN OVERVIEW

By

J. Chaiken
T. Crabill
L. Holliday
D. Jaquette
M. Lawless
E. Quade



This project was supported by Grant Number 75-NI-99-0012 awarded to the Rand Corporation by the National Institute of Law Enforcement and Criminal Justice, Law Enforcement Assistance Administration, U. S. Department of Justice, under the Omnibus Crime Control and Safe Streets Act of 1968, as amended. Points of view or opinions stated in this document are those of the authors and do not necessarily represent the official position or policies of the U. S. Department of Justice.

APRIL 1976

National Institute of Law Enforcement and Criminal Justice
Law Enforcement Assistance Administration
U. S. Department of Justice

For sale by the Superintendent of Documents, U.S. Government Printing Office
Washington, D.C. 20402 - Price $2.50
Stock No. 027-000-00429-6

## FOREWORD

Computer simulation models are widely recognized as valuable tools for planning and management. An effective model provides added insights by mathematically projecting the consequences of alternative solutions to exceedingly complex problems.

Originally designed for strategic military use, simulation models have since gained acceptance throughout the public sector. Criminal justice agencies have been among the last to adopt computer models to the planning and decisionmaking process—in part, because it is only recently that the criminal justice system has been treated *as a system,* rather than a series of unrelated parts.

This study searches out examples of the best existing criminal justice simulations, describes their characteristics, and discusses their value for criminal justice agencies. It is an excellent resource for criminal justice administrators considering the use of a simulation model.

Gerald M. Caplan
Director
National Institute of Law
Enforcement and Criminal Justice

# PREFACE

This review of criminal justice models was funded by the National Institute of Law Enforcement and Criminal Justice of the Law Enforcement Assistance Administration under grant 75-NI-99-0012. The report is written primarily for the guidance of

- Criminal justice planners who wish to locate a suitable model for a particular application
- Criminal justice funding agency personnel
- Model builders

The summary of this report is written in nontechnical language and is intended as an executive summary for administrators of criminal justice planning agencies. The body of the report contains some technical details, but these are explained in either the text or the glossary.

The authors have attempted to be impartial in reviewing individual models. However, it must be admitted that complete impartiality was impossible, because some of the models were developed by one of the authors or by colleagues at The New York City-Rand Institute. These models may have been treated more positively or more negatively than the others, due to greater familiarity with their design, documentation, or history of implementation. The reader must understand that this is a limitation of any review article where the authors have some past association with work in the field.

The contributions of each of the authors to this report are as follows: Leo Holliday, project leadership and Chapters 5 and 6; Edward Quade, Chapter 2; David Jaquette, Chapter 3; Jan Chaiken and Thomas Crabill, Chapter 4; Michael Lawless, Chapter 7. The remaining parts of the report were written by Jan Chaiken based on materials drafted by all the authors.

# SUMMARY [1]

The term "model" refers to a device or procedure for providing insight into the consequences of a decision. For this study we reviewed models related to the criminal justice system that (a) operate on a computer and (b) are intended to assist decisionmaking by criminal justice agencies. This study located 46 such models in 1974. Based on the adequacy of documentation and the availability of the computer programs to criminal justice agencies that might want to use them in the future, 20 of these 46 models were selected for detailed description in the text.[2] These descriptions are intended to be adequate for criminal justice planners and policymakers to determine whether an appropriate model already exists for handling a particular problem, and, if so, which one would best meet their needs.

In addition to describing the models, this study reviewed the circumstances under which criminal justice models are or are not implemented by operating and planning agencies. In general, models have failed to achieve the level of use for policy decisions that was intended by the model builders and those who funded them. Our findings concerning the causes of implementation successes and failures indicate how federal research administrators might improve the quality and usefulness of models in the future.

## APPRAISING MODELS

While, in principle, models can be designed to assist policymakers in nearly any kind of decision, in practice no one would take the effort to use a model unless the decision presents difficulties such as one of the following:

- So many alternatives are available that it is not practical to consider each one before selecting the best.
- The consequences of each alternative are too complex to be anticipated with assurance.
- Numerous tedious calculations must be performed to evaluate each alternative.
- The decision must be performed rapidly following specified rules (e.g., selecting a particular patrol car to dispatch to a reported crime).

In such situations a model can provide vital information that otherwise would not be available for making the decision. In addition, models produce clear documentation of the decision process that can help persuade others of the correctness of the policymaker's position.

No models can tell a policymaker exactly what decision he should make in a given situation. Instead, models must be used with common sense, good judgment,

[1] References to the literature describing the models and studies mentioned in this summary are given in the body of the report. See glossary for definitions of unfamiliar terms.

[2] The actual number of models discussed in the text is larger, because some submodels are reviewed separately, and some unselected models are described in an historical context.

and an understanding of political and budgetary constraints to make decisions. Some models, *descriptive* in nature, do not even pretend to suggest any policy recommendation; they simply provide a tool for anticipating the consequences of policy changes invented by the user. Other models *prescribe* a "best" solution to a specified problem, but even here the user often has a choice of how the term "best" is to be defined, and he always has to use his own judgment in weighing performance characteristics not encompassed by the model builder's definition of "best."

If it appears that a model could potentially be helpful, the policymaker must then know how to appraise particular models to determine whether they are suitable. Several factors must be considered. Most important is the *match between the model and the policy issue* to be addressed. A very accurate model that answers a question of no interest to the agency would not be of any value. Next is the *time until completion* of the model. If a decision must be made before the model can be installed and appropriate data can be collected, then the model will not be useful.

*Technical quality* of a model is often difficult for a policymaker to judge, but evidence that *verification* and *validation* of the model have been (or will be) conducted should serve as adequate assurances of quality. *Verification* refers to checking that the model does what the model builder intended it should do. This is accomplished by using test data for which the answer is known or by comparing one model with another previously verified model. *Validation* refers to checking that the output from the model agrees with reality. This important step is often omitted because it may be difficult or expensive, but a validated model is definitely to be preferred over an unvalidated one.

Another important characteristic for appraising models is the *amount and nature of data* required. If two models are equally satisfactory for answering the policy issue at hand, but one requires less data or more readily available data than the other, then it is to be preferred. The *cost* of a model is generally important only in terms of the types of personnel needed to use the model and the length of time they will have to work with it before decisions can be made from the output. Very rarely are the differences in the costs of computer processing large enough to be an important factor in choosing among models.

The *mode of operation* of the model is often considered to be important. Some models are *interactive*, meaning that the user sits at a terminal and enters information directly into the model via his keyboard; the output appears immediately at the same terminal. Others operate in *batch mode*, whereby instructions to the program are prepared on cards or a similar input medium and the output emerges later on a high-speed printer. Interactive programs are claimed to have advantages for facilitating training and maintaining user interest, but many computer systems cannot support interactive programs. Our study failed to identify either mode as more likely than the other to result in successful implementation.

Examples of *previous implementation and use* of a model are helpful in appraising it. However, examples of failure to implement are not necessarily to be taken too seriously, since we found that such failures were often unrelated to characteristics of the model itself.[3]

[3] This is discussed in the section "Implementation of Models."



## TYPES OF MODELS

The basic types of models of interest in the criminal justice field are as follows:

- *Analytic models*. These determine an outcome or solution from mathematical analysis, such as solving a set of equations. Generally, many features of the system to be studied are ignored or simplified in an analytic model, but the results may nonetheless be accurate enough for policy decisions. One type of analytic model is an *optimization* model; this tells how to obtain the *lowest* or *highest* possible value of some performance measure (for example, how to schedule a fixed number of court cases in a week so as to handle them with the lowest possible cost).

- *Computer simulations*. These imitate the operations of a system so as to produce the same statistical behavior as found in the real world. For example, a simulation model could follow a large number of imaginary court cases, keeping track of the dates of their appearances, the outcome of each case, and so forth. The output of the simulation model would describe statistical properties of all the imaginary cases, for example the average length of time from arraignment to final disposition for burglary defendants not released on bail. Simulation models can in general capture more details of actual operations than can analytic models, but they may be more expensive to use, and data collection may be more difficult. Simulation models are always descriptive; they tell the policymaker *what will happen if* he makes a certain decision but do not suggest any decisions to be considered.

- *Operational gaming*. This is a form of simulation in which human participants imitate some aspect of the real world. For example, the participants can pretend to be drug sellers who modify their operations in response to new legislation.

- *Group judgment*. Some models are structured procedures for obtaining forecasts or estimates from a group of people. An example, called Delphi, involves using anonymous feedback of statistical information about the previous estimates provided by the group, until a consensus or firm disagreement is reached. These techniques are commonly part of a larger modeling effort, perhaps serving to provide "good guesses" for the data needed by some other model. For example, the probability that a proposed legislative bill will actually become law may be required as input to a model, and a group judgment could determine an estimate of this probability.

For the most part, operational gaming and group judgment models have not been widely applied to criminal justice problems, but they nonetheless have potential for the future. Good examples of analytic and simulation models exist, and for several types of applications it should be possible to use an existing model rather than develop a new one. These will be described below according to the part of the criminal justice system addressed by the model: the entire system, police operations (primarily patrol), courts, or corrections.

## OVERALL MODELS OF THE CRIMINAL JUSTICE SYSTEM

Models of the entire criminal justice system (CJS) have been developed as part of, and as a consequence of, the work of the President's Crime Commission[4] in the mid-1960s. These have focused on the flow of offenders through the various components of the CJS: police, prosecution, courts, corrections, and parole. Although there is no organizational structure with control over the entire CJS, these models have been useful to planners for anticipating the effects of policy changes in one part of the system on later changes elsewhere.

A single model, called JUSSIM, has been the central development in this field. Constantly undergoing improvement and elaboration, it has spawned a number of variants with other names. While the differences among these models may be of some importance to potential users and are discussed in the text, here we shall give only a general outline of their common features.

Individuals, both recidivists and new offenders in society, perpetrate crimes. Some are detected, some not; some reported, and some not. Reported crimes are processed by the police, arrests are made, and some of the arrestees are charged with a crime. These arrests become cases to be processed by the courts, and those convicted may be sentenced to the corrections subsystem. Parole and eventual release return individuals to society. A fraction of these plus a fraction of those released from other parts of the CJS inevitably commit crimes again. These are the recidivists who return to the "front end" of the CJS.

The model considers groups of these individuals, distinguished perhaps by crime type, age, sex, or other characteristics relevant to how they will be processed by the system. The user specifies how many offenders there are (or will be) in each group and what fraction of each group will proceed from one stage of the CJS to another (e.g., from arrest to arraignment, or arrest to release by the prosecutor). In regard to recidivists, the user specifies what fraction of burglars (for example) will return as burglars, robbers, etc. Based on this kind of information, the model calculates projections of cost, workload, or resources needed at each stage of the CJS. The models differ as to whether these estimates are provided year by year into the future or only for a single period of time.

To use the models, the planner must consider a possible policy change (such as a diversion program that will reduce the number of drug offenders processed by the courts) and estimate how the change will affect the numbers that are provided as input to the computer program. The program then calculates and displays new measures of workload (such as the prison population) and other information that permits the decisionmaker to anticipate the consequences of the proposed policy change.

One overall CJS model described in this report (DOTSIM) is a case-by-case simulation that follows each individual offender through the system. It can calculate certain performance measures, such as how long defendants wait for trial, that are not available from the other models. However, DOTSIM has not been accepted and used to the same degree as JUSSIM and its descendants.

The primary value of overall CJS models to date has been to train planners to understand the interactions among different parts of the system and to focus their

[4] Officially titled the President's Commission on Law Enforcement and Administration of Justice.



data collection efforts on information having clear value for management purposes. One of the models discussed in the text was designed specifically as a training tool.

## POLICE MODELS

Nearly all models for police applications have been directed at patrol forces. There are several types, which will be discussed separately.

### Patrol Car Allocation Models

These analytic models specify the number of patrol cars that should be on duty in each geographical command of a city at various times of day on each day of the week. They can be used to analyze policy issues of the following types:

- Determining the total number of patrol officers a department should have (e.g., during budget preparation)
- Allocating a fixed total number of officers among geographical commands
- Determining how many officers in a command should work each tour or shift
- Determining the hours at which tours or shifts should begin.

They cannot be used to design patrol beats (areas covered by a single patrol car or a small number of cars).

At the start of this project, there were several models of this type, each of them in use in a small number of departments or not in use at all. After reviewing the features of each of them, the Rand staff designed a new model incorporating by user option nearly all the capabilities of the previous models, after which we observed directly the obstacles to user acceptance of new computer models.

One of the earlier models (LEMRAS), now withdrawn, provided the user with the capability to predict how many calls for service would be received at different times of day from various locations. This feature is not present in Rand's patrol car allocation model (PCAM). In other respects, by describing PCAM's capabilities we can describe a composite of all the previous models.

PCAM has both descriptive and prescriptive capabilities. In descriptive mode it calculates performance measures for any allocation proposed by either the user or the program itself. These include the workloads of the cars, the amount of preventive patrol provided by the cars, and average travel times and response times to incidents. (In this model, response time is defined as the sum of travel time and a queuing delay incurred if the call has to wait until a car is available to be dispatched; any delays before the dispatcher handles the call are not included.)

In prescriptive mode, PCAM can specify the minimum number of patrol cars that must be on duty to meet standards of performance established by the department. (A typical standard would be that no more than 15 percent of calls should experience a queuing delay.) Or, it can prescribe how a specified total number of patrol man-hours should be distributed geographically or by time of day so as to minimize some measure of performance such as average response time.

Such a model requires very little data, is easy to use, and can be clearly shown to be preferable to traditional patrol allocation methods, such as hazard formulas.

But, as already mentioned, acceptance of computer models for patrol allocation has not been very widespread to date.

### Simulation Models for Patrol Systems

Four simulation models are described in the text, and many others have been mentioned in the literature. These imitate step by step the operations of patrol cars. (For example, the model imagines that a call for service arrives; then a patrol car is selected for dispatch according to current or proposed dispatching rules; then another call arrives and a second car is dispatched; then the first car arrives at the scene after a travel time that is calculated from its location when dispatched and the location of its destination; and so forth.) These models can capture many more details of patrol operations than an allocation model, and they provide more accurate and voluminous performance statistics. But, correspondingly, they require substantially more data and a higher level of expertise in the user.

Typical policy issues that can be addressed with simulation models are:

- The effects of changing dispatching rules
- The potential value of a car locator system
- Whether to assign different functions to different cars (e.g., some handle primarily traffic accidents, others respond only to serious crimes in progress, others respond to minor incidents in a small neighborhood, etc.)
- Whether to move cars from one part of the city to another as unavailabilities develop.

Although the models are well designed and at least one has been validated, no examples of sustained use of simulation models was found, nor did we find a single instance of important decisions made based on the output from such a model.

#### Beat Design Models

Two models have been used to design patrol beats and analyze other questions related to geographical details of patrol operations within a command. These are analytical models intermediate in detail between patrol car allocation models and simulation models. Their value arises from the fact that most police patrol operations involve sufficient complications that it is nearly impossible for a planner to look at a map and make accurate "guesses" regarding the workloads of the cars or the locations where travel times may be high.

The models calculate a variety of performance measures for each beat design proposed by the user, permitting him to develop better designs step by step. One of the models recommends a beat design that minimizes (or comes close to miminizing) average travel time in the study region, but there is no agreement among researchers that such a design is necessarily better than others (which may, for example, have a more even balance of workload among the units).

Since police departments do not redesign their beats frequently, instances of one-shot uses of these models can be considered successes (unlike patrol car allocation models, which, if not used from time to time, can be considered implementation failures). A substantial number of successful and useful implementations of beat design models were encountered in this study.



### Dynamic Queuing Model

This is an analytic model that calculates queuing delays under the assumption that the number of patrol cars on duty changes from hour to hour. This can also be accomplished by patrol car allocation models, but less accurately. The model has been used to evaluate changes in the starting times of tours and meal hours.

### Linear Programming Model for Scheduling Patrol Cars

This optimization model recommends the hours at which tours and meals should start so as to achieve specified numbers of cars on duty while consuming the smallest possible number of car-hours. It has been used to schedule tour starting times.

### Manpower Scheduling Models

The report describes two models for determining which days each officer should work, which days he should be off duty, and when he should rotate from one tour to another. These models provide a much better match between manpower on duty and manpower required than traditional scheduling methods. Moreover, the schedules can be designed to be completely equitable, meaning that in the long run all officers experience the same work patterns. The models are well documented but have not achieved any noticeable level of acceptance to date.

## COURT AND CORRECTIONS MODELS

Most court and corrections models are similar to the models of the overall CJS described above; that is, they estimate characteristics of cases or offenders moving through various stages of processing, or they calculate data needed to estimate offender flow characteristics. However, court and corrections models would ordinarily have a greater level of detail for the relevant subsystem than would an overall CJS model. For example, arraignment might be considered a single stage in an overall CJS model, whereas in a court model it might be represented as a number of courtrooms handling possibly different types of crimes. This permits the model to answer various questions about court management for the guidance of administrators.

Modeling work in the field of corrections has been very limited. Only one such model is discussed in the text, and it is being designed in Canada. While it is in an earlier stage of development than the other models reviewed in this report, the design work appears especially competent and includes careful validation efforts.

One of the court models addresses an entirely different type of policy issue, namely, the process by which jurors are assigned to trials. The purpose of the model is to minimize the number of persons who must be called for jury duty in order to provide the needed number of jurors for each trial, and to reduce the idle time of members of the panel and the length of time required to select a jury for the average trial.

## IMPLEMENTATION OF MODELS

Through a series of interviews with model builders and personnel in agencies that attempted to implement models, a picture of the implementation process was obtained. In general, criminal justice models have failed to achieve any notable level of use for policy decisions. This finding conforms to the observations of other researchers who examined all federally supported mathematical models (not primarily criminal justice models) using a mailed survey technique.

The explanation for this discouraging history lies only partly with the characteristics of the models themselves; primarily it rests with characteristics of user agencies and the interactions between model builders and user agencies.

### Obstacles to Implementation

Although examples were found of failures to implement because the user agency was unable to understand the programming language or the conceptual foundations of the model, the main *model attribute* that proved to be an obstacle to implementation was a requirement for data that was unavailable to the agency. However, the same models that posed insuperable data-collection problems in some agencies were nonetheless successfully operated by others. Thus, we cannot identify any type of model as being "too complex" for use by criminal justice agencies.

The *agency characteristics* found to be obstacles to implementation were as follows. First, the introduction of a model is generally not undertaken in response to some pressing need or problem to be solved. Instead, the model is intended to replace or improve a process that is currently considered adequate by the agency. For this reason, the introduction of the model may not be planned for in the agency's budget, and other matters considered of greater importance can divert resources or personnel from development and use of the model.

Second, it was very often the case that a single advocate in the user agency saw the need for a model, conducted a search for the appropriate one, sponsored his choice before agency administrators, and pursued implementation with little support from others. The progress of implementation then depended on the advocate's judgment, continued attention, and political skills. If he became discouraged or transferred to another position, the implementation would not be pursued. In addition, a change in personnel at management levels above the advocate could result in rapid suspension of interest in the model. A corollary to these points is that vulnerability to changes in personnel increases as time elapsed on a project increases. These causes of implementation failure, observed in about one-fourth of the cases, are clearly unrelated to the attributes of the model itself.

The third agency characteristic found to be an obstacle to implementation was the lack of professionalization among the planners, meaning that the agency's personnel did not have advanced training, a tradition of using any kind of analytical techniques, or a world view that extended beyond the immediate organization. This problem is a far-reaching one, extending beyond modeling per se, and touching on the current capabilities of criminal justice agencies to support a competent planning process.



### Requirements for Successful Implementation

Indicators of successful implementation were found to be:

1. A clear and realistic understanding at the start of the project of the policy issues to be addressed and the time frame over which results would be obtained from the model
2. The availability of suitable written documentation of the model oriented to the user
3. A direct personal contact between agency personnel and the model builder or one of his associates.

The last point suggests one of the key difficulties in sponsoring widespread implementation of models in the future, since it is impossible for a small number of individuals to assist a large number of agencies directly, and in any event many model builders are ill-suited by inclination and temperament for this task. By analogy with the physical sciences, there are few *engineers* in the field of criminal justice modeling to translate theoretical concepts into practical applications.

## POLICY IMPLICATIONS

Criminal justice modeling is a young field that has demonstrated value for training planners to understand their agency's operations and interactions with other agencies, but has had little impact on policy decisions to date. Optimism about the future course of the field rests primarily on the successes that have been achieved by models in such other applications as business planning, architectural design, and military studies.

We believe that a reasonably sustained effort to encourage implementation of existing high-quality models over a several-year period should give a clear indication of whether models can serve a useful function in the criminal justice planning process. At the same time, the development of new models should not be discouraged, under the assumption that a reasonable period of time is needed between the design of models and a demonstration of their value.

We believe that an effort should be made to institute some form of peer review in the model-funding process, because no one individual can be sufficiently familiar with the details of existing models to know whether a proposed new model duplicates available capabilities. In addition, funding agencies should concentrate some efforts on testing models in a variety of jurisdictions and developing clear documentation in the form of user's manuals and case studies of implementations that *failed*. These will provide guidance as to the pitfalls to be avoided in the future. Since the production of such a document may be a painful experience for the model builder and the funding agency, independent chroniclers could be assigned this task.

While we have noted that the absence of engineers in the modeling field presents dissemination problems, no easy solutions are apparent. Whoever plays the role of the engineer will necessarily have a less adequate understanding of the model than its designer and, to stay in business, may be forced to behave in ways that are contrary to the interests of agencies with which he deals. For example, it may be necessary to make overly optimistic or vague promises about the capabilities of a model in order to win a contract.

Nonetheless, even modest efforts to improve the dissemination process might be fruitful. In the past, descriptions of models rarely appeared in publications and at conferences of general interest to criminal justice agency personnel. Grants and contracts related to models could easily require dissemination through more appropriate media in the future.

In addition, a federal center could be established for the purpose of making documentation and computer programs readily available. The personnel of such a center would have to be capable of identifying which models (if any) meet requesters' needs, but they would not necessarily have to know how to install the models or collect data for them. Instead, a list of organizations that have already used each model could be maintained by the center to provide a starting point for further inquiries.

We also believe that training programs providing for students to operate models themselves, using illustrative data, have already proved their value as dissemination devices and should be actively encouraged. The students in such programs have included criminal justice agency planners, analysts, and administrators, and also high-ranking government officials. The benefits of such courses are many:

- Potential users can come to understand that a model is quite easy to use, even though its documentation may appear forbidding.
- Students make a personal contact with the instructor, a circumstance which we found to be important for successful implementations.
- Students who have used a model are unlikely to be confused about the policy questions that it can and cannot answer.
- Planning personnel who attend such courses may come to view model-related activities as important and worthy of a personal commitment, thereby reducing the phenomenon of the "vanishing advocate."
- Communications gaps between model builders and criminal justice agency personnel can be reduced by informal social contact.
- If some of the students are administrators, and they become convinced of the value of a model, they can instill a sense of purpose in the planners who will operate the model.
- Even if students do not implement models in their own agencies, the function of models to inform and enlighten planners will already have been accomplished by the training course.

Finally, the potential value of models to indicate the types of information and data that are needed for management purposes is not being fulfilled, because the implications of models for management information systems has not been summarized in a form accessible to the designers of such systems. We believe a project should be funded specifically for the purpose of addressing this problem and developing suitable manuals and other publications for specialists in information systems.

# ACKNOWLEDGMENTS

We are deeply indebted to the many model designers and personnel in criminal justice agencies who responded to our requests for documentation, answered questions by mail or telephone or during site visits, or demonstrated computer programs for us. In addition, early drafts of sections of this report were provided to the vast majority of designers of models that are described in any detail. Many of them responded with lengthy and thoughtful critiques or recent documentation that greatly assisted us in improving the accuracy of this report. In this connection, we particularly wish to thank Deepak Bammi, Alfred Blumstein, Robert Hann, Peter Kolesar, Richard Larson, Spencer Smith, and Warren Walker, without suggesting that they necessarily agree with our descriptions or conclusions.

# GLOSSARY

**Algorithm:** A finite set of ordered procedures, steps, or rules, usually mathematical in nature, for determining a number or other outcome.

**Analytic:** Refers to a model in which the outcome or solution is determined from mathematical analysis, such as solving a set of equations.

**Backlog:** Persons or cases awaiting processing.

**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.**

**Beat:** Subarea of a precinct to which a patrol car can be assigned. Also called **sector**.

**Block Diagram:** A chart setting forth the particular sequence of operations to be performed for handling a particular problem; a tool of programming.

**Branching Ratio:** The proportion of cases in one stage of a system that move into a specified succeeding stage.

**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.

**CJS:** Criminal justice system.

**Crime-Switch Matrix:** A collection of numbers describing the probability that an offender who commits one type of crime will commit another (or the same) type of crime when he recidivates.

**Cumulative Distribution Function (CDF):** A function indicating the probability that a certain random variable assumes an experimental value less than or equal to any specified number.

**Debugging:** Eliminating programming errors.

**Decision Variable:** Quantity over which the policymaker has some control.

**Delphi:** A procedure for arriving at a forecast or estimate by refining the estimates of individuals using anonymous feedback of information about previous estimates.

**Deterministic:** Having no elements of chance. Contrasted with **stochastic**.

**Dispatcher Delay:** Interval of time between the moment a caller to the police finishes his telephone conversation and the moment the dispatcher is ready to consider the call. Not to be confused with **queuing delay**.

**Distribution:** If a variable can take a set of values with a certain relative frequency or probability, the distribution describes how often any specified set of values occurs.

**Dynamic:** Changing over time.

**Effective Car:** The equivalent of a patrol car that does not engage in any non-cfs work.

**Event-Paced:** A type of computer simulation in which the simulation clock, when requiring update, is advanced to the time of the next simulation event. By contrast, another type of simulation advances the simulated time by fixed amounts, such as a year.

**Exponentially Distributed:** A random variable T is exponentially distributed if there is a parameter $\mu$ such that

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

**Exponential Smoothing:** A mathematical procedure for predicting the number of events to occur by averaging past data, placing greater weight on recent events than on past ones.

**Feedback.** Feedback is present in a model or process if it is able to adjust future conduct or operation on the basis of past performance or outcome.

**Flowchart:** A chart to represent, for a problem, the flow of data, procedures, growth, equipment, methods, documents, machine instructions, etc.

**Fourier Transform:** A mathematical technique that converts the cumulative distribution function of a random variable into another function that is easier to use in certain types of calculations.

**Gaming:** A simulation involving human participants.

**Heuristic:** Refers to an optimization model that produces "good" values of the objective function, but not necessarily the best possible.

**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**.

**Intuition:** An informed guess.

**Iteration:** The process of repeating several times.

**LEAA:** Law Enforcement Assistance Administration.

**Linear:** Refers to a functional relationship that can be graphed as a straight line, plane, etc.



**Linear Programming:** A mathematical process used to determine the best or optimum use of resources when the limitations on the available resources can be expressed by simultaneous linear equations. A mathematical model which assumes linear relationships and in which an optimal solution is sought (maximizing or minimizing) subject to one or more limiting constraints is used to represent the problem.

**Markov Transition Model:** A model in which it is assumed that the probability of changing from one state or stage of processing to the next one is independent of the previous history of the system. (For example, it is assumed that the probability of being released on parole does not depend on whether the offender pled guilty or was convicted by a jury trial. This is an approximation to the actual operation of the system.)

**Median:** The median of a set of numbers is that value above which 50 percent of the numbers fall.

**Model:** A device or procedure for providing insight into the consequences of a decision.

**Module:** Part of a computer model.

**Monte Carlo Method:** Any procedure that involves statistical sampling techniques in order to obtain a probabilistic approximation to the solution of a mathematical or physical problem.

**Multiserver Queue:** A system in which there are several possible servers for each customer.

**Nolle Prosequi:** An entry on the record of a legal action denoting that the prosecutor or plaintiff will proceed no further in his action or suit either as a whole or as to some count or as to one or more of several defendants.

**Nolo Contendere:** A plea by the defendant in a criminal prosecution that, without admitting guilt, subjects him to conviction but does not preclude him from denying the truth of the charges in a collateral proceeding.

**Objective Function:** A performance measure to be maximized or minimized by an optimization procedure.

**One-Shot:** Refers to a decision made once, or infrequently.

**Operational Gaming:** A simulation involving human participants.

**Optimization:** A procedure for finding the values of decision variables that make some performance measure as high or as low as possible.

**Order of Magnitude:** A ratio of about 10.

**Parameter:** A variable essential to characterizing some aspect of a model or input thereto—for instance, the environment or an alternative to be evaluated—that is held constant during a particular calculation but may vary from calculation to calculation.

**Poisson Process:** A sequence of events constitutes a Poisson process if there is a parameter $\lambda$ such that

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

**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.

**Preemption:** Interruption of service on one job to handle another job.

**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.

**Probability Density Function (PDF):** A nonnegative function for which the probability that the corresponding random variable lies between x and x + $\Delta$x ($\Delta$x small) is approximately equal to the function evaluated at x multiplied by $\Delta$x.

**Probability Distribution:** See Distribution.

**Quantitative:** Represented in terms of numbers, mathematical equations, or computer programs.

**Queue:** A waiting line, as of customers before a checkout counter or incident reports before a dispatcher.

**Queuing Delay:** Length of time spent in queue.

**Recidivism:** The return of criminal offenders to criminal activity or to involvement with the criminal justice system.

**Recreation:** Days on which a person does not work.

**Sector:** Subarea of a precinct to which a single patrol car is assigned. Also called **beat.**

**Sensitivity Analysis:** A method of investigating the effect of uncertainty on the output of a model by varying the values of parameters which characterize some aspect of the model or input to the model.

**Simulation:** A method of replicating the operations of a system with a computer model that incorporates the same statistical behaviors as found in the actual system. Parts of the system may be simulated by human participants.

**SPA:** State Planning Agency.

**Standard Deviation:** The most common measure of the dispersion of a distribution about its mean or average value.

**Steady State:** A situation in which the characteristics of a system do not change over time.

**Stochastic:** A variable is stochastic if the value it assumes is governed by chance and the values it may assume can be described by a probability distribution.



**Suboptimization:** A method of approximating the optimal solution to a problem by taking as given some aspects that should in principle be determined as part of the analysis and thus simplifying the process of optimization.

**Time-Sharing System:** A computer system that can interact with several users simultaneously.

**Tour:** A period of time during which a patrol officer is on duty. Also called *shift* or *watch*.

**Validation:** Checking that the outputs of a model agree with reality.

**Verification:** The process of determining that a computer program does what it is intended to do.

**Virgin:** A criminal offender who has not previously entered the criminal justice system.

**Voir dire:** A preliminary examination to determine the competency of a witness or juror.

## CONTENTS

# Chapter 1

# INTRODUCTION

While only a decade has passed since the first applications of computer-based models to criminal justice policy issues, there is already a great diversity of such models, and more of them are being designed or proposed every year. Many criminal justice planners and operating agencies are uncertain about the circumstances under which models can be useful, whether an appropriate model already exists for handling a particular problem, and, if so, which one would be best suited to their needs.

This report is intended to serve as a guide for answering such questions. It discusses the roles that models can play in decisionmaking, their advantages, and limitations. It describes selected models in sufficient detail that a potential user should be able to identify the ones of possible interest to him and obtain copies of source documentation for more careful evaluation. Additional models are reviewed briefly in an appendix.

As is the case with most models designed for governmental planning purposes, criminal justice models have not been used to as great an extent as the model builder might have hoped. While the descriptions given in this report may in themselves assist in the future dissemination of useful models, we also present a review of the obstacles to implementation that have been present in the past and suggest possible remedies.

The criteria established for including a model in this study were as follows:

- The model operates on a computer.
- It is intended to assist decisionmaking by criminal justice agencies. This criterion excludes statistical packages, information systems, and models designed to advance theory or knowledge (for example, to illuminate the relationship between demographic variables and crime rates).

The project began in August 1974, with a brief survey to find models meeting these criteria. While we cannot claim to have located all of them, or even necessarily a majority, the search was sufficiently comprehensive that models with readily accessible documentation were unlikely to be overlooked. Sources included papers and reports that have been published in technical journals, were referenced in previous reviews of models (e.g., Gass [65] and Fromm [64]), or were listed by the National Criminal Justice Reference Service or the National Technical Information Service; responses to an item in the *Criminal Justice Newsletter* or to letters we mailed to state and regional criminal justice planning agencies; and personal contacts of the authors.

This resulted in the identification of more than 60 models for review and screening. Out of these, 46 were considered suitable for brief descriptions (see appendix); out of those, approximately 20 were selected for detailed description. (It is difficult to be more precise in this case because the models described in detail tend to run in families, and more than 20 are actually mentioned.) For the original 46, an effort was made to locate and interview users of the models, as described in Chapter 7.

The criteria used to select models for detailed consideration included the adequacy of descriptive material and the availability of the computer program to other criminal justice agencies that might want to use it in the future. In addition, the number of models of different types played a role in selection, so that some models have been selected for the purpose of illustrating the diversity of applications, while others have been omitted because prior or better-known examples of a similar nature had already been included. Degree of implementation was also considered, but this was not used as an eliminating factor because it was found that adequate models sometimes failed to reach implementation due to factors beyond the control of the model builders. Judgments of quality did not enter into the selection process, since the models were not reviewed in detail until a later stage.

For the selected models, additional information was obtained from the model builder and users, in many cases by site visit. Thus, more current information than may have been available in August 1974 is presented for some of the models, and related references that were initially provided to us in draft form have been updated. However, with one exception, models that were not documented until after August 1974 have not been included. The exception is a patrol car allocation model that was designed specifically to alleviate the inadequacies of existing models of this type that we found in the initial stages of the present study. The process of attempting to achieve user acceptance for this model was monitored closely to give us a direct observation of the obstacles to implementation discussed in this report.

The report is organized as follows: Chapter 2 contains a general discussion of what models are and how they can be used. Overall criminal justice models and models of police, courts, and corrections are described in Chapters 3 through 6, followed by a separate discussion of implementation problems in Chapter 7. Some general impressions gained and conclusions drawn during the study are discussed in Chapter 8.

The appendix contains brief descriptions of 46 models considered and lists documentation related to each model. A complete list of references appears separately at the end of the report.

# Chapter 2

# MODELS AND THEIR USE

## THE CONCEPT

The heart of any attempt to analyze a situation or issue and make a rational decision is the existence or creation of a device or procedure to provide insight into the consequences of any decision that might be contemplated. That device or procedure is termed a "model." Most commonly, it is a simplified representation of whatever part of the real world is important to the issue under study, one that can be manipulated to forecast or at least give some clue as to the outcome that is likely to follow a particular action.

The models examined in later chapters of this report are represented quantitatively (i.e., using numbers) and are expressed in terms of mathematical equations and computer programs. However, the word "model" can be interpreted much more broadly. A model might be purely verbal (discussing how a labor leader is likely to respond to a management initiative is a verbal model), or a simple diagram to guide one's thinking about a complex process, as in Fig. 1, or physical, say a shaped piece of wood used in conjunction with a wind tunnel to predict the performance of an airfoil. No matter how it is represented, however, a model in the sense the word is used in this report is designed to help a decisionmaker make a better decision than in its absence he might otherwise make. We do not use the word "model" in the sense of "exemplary case," as in the expression "model corrections facility."

Decisions are often made intuitively without the use of an explicit model. Nevertheless, if a decisionmaker weighs the consequences of his alternatives, he has a model in mind even though it may consist of no more than a few hazy assumptions about the factors that operate. Thus, a judge who sentences equally guilty partners in crime to differing sentences has a model in mind that relates justice and the factors in the case to the background and characteristics of the personalities involved. Considerable effort, however, would be required to design a computer program that takes into account the same information used by the judge and arrives at the same conclusions.

The adequacy of a quantitative model (that is, the confidence we have that the inferences drawn from it are accurate) depends on how well it captures the essence of the issues and how well the numerical values it requires can be estimated. For example, suppose a model is constructed to compare proposed legislative measures for combatting juvenile delinquency by estimating how the total annual cost to the nation of juvenile delinquency would decrease if each measure were presented to the legislature.[1] The model could be a simple equation that takes into account only the current cost of juvenile delinquency, the cost of implementing the proposed legislation, and the user's subjective estimates of the chances that the legislation could be enacted and the effectiveness of the proposal.

Such a model has conceptual difficulties, because the impact of juvenile delin-

[1] This model has been described in more detail by Helmer [86].

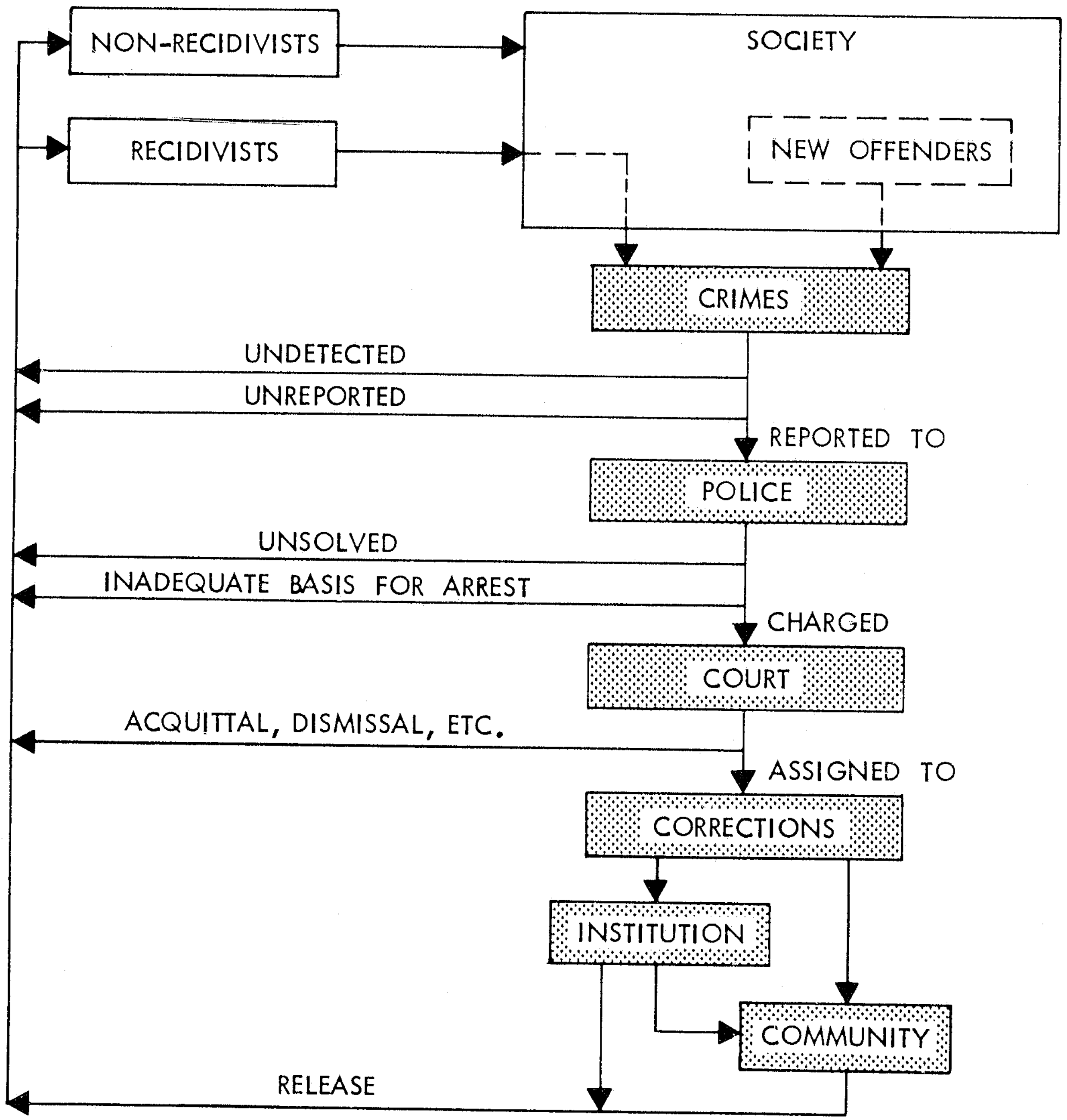


Fig. 1—Flow diagram of the criminal justice system



quency involves nonmonetary and quite intangible factors—human happiness, for instance—and the proposed measures might also differ in nonmonetary ways, such as the extent to which they deprive juveniles of their civil rights. In addition, the model requires the user to estimate the current total cost to the nation of juvenile delinquency, a figure that is not readily available, to say the least. Nonetheless, the model could be useful if even rough estimates of the required input information lead to identifying one of the measures as clearly best in terms of cost reduction.

As the example illustrates, a model is a simplified representation of the real world and of phenomena in which we are interested. The representation is incomplete. Some elements of the situation are omitted through ignorance; others, usually many, are omitted deliberately because they appear insignificant or irrelevant to the model builder. The hope is to make the approximation adequate for the problem at hand so that the answers obtained from questions put to the model will give clues or insights adequate to guide the user in dealing with that part of the real world to which the model corresponds.

It is clear that if we simplify a model too much, we cannot depend on it to tell us what will be likely to happen. On the other hand, if we make it too realistic, and thus too complicated, we may no longer be able to obtain results from it. The dilemma is that a model must be simple enough to allow the user to think with its aid, but at the same time faithful enough to reality to produce reasonably valid predictions. What constitutes reasonable validity depends on the questions to be answered and on the context. For questions that seek to increase efficiency in situations where it's clear what "efficiency" means, the results can be very good. For questions of what is best, where the criteria of "best" are multiple and conflicting and dependent on politics and mores—involving, say, juvenile delinquency, recidivism, or rehabilitation—we must sometimes count ourselves lucky if the model points our actions in the right direction.

## ROLE IN DECISIONMAKING

A decisionmaker faced with a problem (that is, a situation in which he must decide whether or not to take some action and, if so, what) may seek analytic help from his staff, or by contract from outsiders, or may attempt to provide it himself. It any case, if that help is to be effective, he needs to develop a good idea of his objective or what it is he wants to accomplish and, if others are involved, communicate it correctly. Once this is done, it is possible to seek out various alternatives or options, actions that appear to offer some posibility for attaining the objective. Assuming the decisionmaker wants to do the best that can be done under the constraints he faces (for instance, those on his budgetary resources and/or the need to maintain "due process"), each alternative should be investigated to determine its consequences or impacts. Chief among these are how well it accomplishes what is wanted and what must be given up to obtain his goal. To forecast or estimate these consequences, models are used. If the problem is at all complicated, explicit models are usually required to estimate the consequences of action with any sort of reasonable confidence. Additional models may also be used to compare and rank alternatives, although this is often done intuitively.

## WHAT MODELS PROVIDE

One hopes that his model can be made to describe the problem under investigation so faithfully that the results obtained from it can be accepted as completely valid for all practical purposes. In the physical sciences such models exist and are called theories. Elsewhere there are problems for which models that approach this ideal can be developed, but they require a situation in which the underlying relationships are well understood, in which data are abundant (and accessible), and in which the results of preliminary versions can be tested on a number of interesting cases. In more general problems, where behavioral, political, and social factors play a large role, we have to base our calculations on, and supplement our model results with, a great deal of judgment. Models and model building provide guidance for that judgment.

Reliance on judgment and intuition is crucial to every decision. This reliance permeates every aspect of analysis in isolating the question to be analyzed, in limiting the extent of the inquiry, in deciding which hypotheses are likely to be more fruitful, in selecting what factors to include, in determining what the "facts" are, and in interpreting the results. A great virtue of models and model *building* is that they provide a systematic, explicit, and efficient way to focus the required judgment and intuition, particularly that of experts and specialists on whom analysts must usually depend for practical knowledge and experience.

An explicit model, quantitative or not, introduces structure into a problem, enabling involved decisions to be broken into constituent parts that can often be considered one at a time. In using and building models, analysts and the experts on whom they call are compelled to use a precise terminology and to develop their ideas and exercise their judgment and intuition in a well-defined context, one that puts their judgments in proper relation to those of the others. Moreover, if they initially disagree, they must reach an acceptable compromise. The model thus provides an effective means of *communication*.

In addition, a model provides *feedback* to guide the participants in refining their earlier judgments. This point is important; by "exercising" the model and testing for sensitivity, information can be generated that may lead the users to alter their original judgment, and even to intuit a solution in spite of deficiencies in the calculations.

Even in well-established scientific fields, model building is a highly creative activity—an art, not a cut-and-dried process. In an area such as criminal justice, the model builder is likely to find himself in a situation where the relationships between the elements are very imprecisely known and little data exist for determining them. His approach is to select certain elements as being relevant to the problem under consideration (and to set aside at least temporarily all the others); to make explicit, where known, the relationships between the elements selected; and to conjecture the nature of other relationships that he judges significant. His model is thus likely to be *ad hoc* and tentative, subject to modification and improvement as new information and insight become available. He improves his model by working with it, trying it out for cases in which the results are known or can be determined, and relying on the judgment of experienced people who can recognize when a result predicted by the model "seems reasonable." The model frequently rewards this effort by suggesting new alternatives and guiding the builder as to what data to collect and what to analyze it for.



In brief, we should not look at a model merely as a "black box," a device to provide a route from a set of hypotheses to a prediction about the real world. So narrow a view ignores a most important product of the modeling process: the insight into the problem it can provide.

## TYPES OF MODELS

For public policy problems (as well as those found in business and industry) the models most used, on the whole the most useful, and most often the only sort even considered by analysts, are quantitative models that resemble the "scientific" models developed in the physical sciences. Such models consist of a system of logical relationships that attempt to express the processes that determine the outcome of alternative actions by means of a set of mathematical equations and/or computer programs. Quantitative models divide into two categories. In one, the *analytic models*, the outcome or solution is extracted from the model by mathematical analysis. In the other, the *simulations*, the outcome is estimated by means of a series of imaginary experiments on the model. Both of these types of models will be described below.

A model would be strictly quantitative if the situation or activity under investigation was represented by that model so faithfully that a decision could be made solely on the basis of the results obtained from the model. Few real-world issues are susceptible to resolution by such a completely quantitative treatment; almost always, judgment will be needed at the end as well as earlier. Hence the term *quantitative* is used somewhat loosely to refer to any model where most of the factors considered are encompassed by a mathematical or computer representation.

Unfortunately, many criminal justice problems cannot be handled satisfactorily or even approached sensibly by means of quantitative models. Of these, many are problems that depend heavily on the social sciences which, because of the nature of their subject matter, have developed few models of predictive quality comparable to the models found in the physical sciences. For these problems, other model types have been developed, depending more on the direct use of judgment and intuition and far less on quantitative relationships to provide insight. These models, discussed below under the headings of Operational Gaming and Group Judgment, are as yet not much used to tackle criminal justice problems, but an understanding of their characteristics may encourage future applications.

### Analytic Models

At its simplest, an analytic quantitative model, once set up, may involve no more effort than the substitution of numerical values in a mathematical expression or formula and a little arithmetic. For example, a simple equation[110] has been developed to give a good estimate of the average length of time required for a police patrol car to travel to the scene of an emergency. To use the equation, one needs to know the number of patrol cars available to respond in a region, the area (square miles) of the region, and the travel speed of the patrol cars. A computer program that permits the user to calculate the average travel time from this equation is an analytic model. In applications, travel times would be calculated for many different regions to permit comparisons.

If this equation were being used to help a police department decide how many patrol cars to have on duty, then the number of patrol cars would be called the *decision variable.* However, the model could also be used to assist in designing patrol regions, in which case the area of the region is the decision variable. In general, a model could have more than one decision variable.

Another example of a simple analytic model would be an equation that permits calculating the average number of days a newly arraigned defendant will have to wait until his first hearing (based on the number of judges, and other information provided by the user). If the model is used to consider whether to hire more judges, the number of judges is the decision variable.

In somewhat more complicated analytic models, the *form* of the desired equation is established by the model builder, but the equation includes some constants whose value is not known in advance. These constants, called *parameters,* may vary from city to city or from time to time and are determined from appropriate data. An example of such a model would be an equation stating that the number of emergency calls received by a police department during the hour beginning at 9 p.m. on a Friday night can be approximately calculated as the number of calls received during the hous beginning at 8 p.m., times some constant. In one city the constant might be 1.18, so that the model is

$$N_9 = 1.18 \times N_8,$$

where $N_8$ is the number of calls in the hour beginning at 8 p.m., and $N_9$ the number beginning at 9 p.m. The number 1.18 is a parameter, and is calculated from data for many previous Friday nights.

Another type of analytic model, called an *optimization model,* is still more complicated. In an optimization model, the user does not have to try every possible value of the decision variables to see which results look best to him. Instead, an equation in the model relates the decision variables to some measure of performance (e.g., the cost of operating the system being modeled), and the model includes a procedure for finding the values of the decision variables that make the performance measure as *high* or as *low* as possible. (In the case of cost, as low as possible.) These models answer questions of the form "What is the *best* way to accomplish such-and-so?" However, the definition of *best* is whatever the model builder puts into the equations in the model, and it may not incorporate all the factors the user has in mind.

### Computer Simulation

Simulation is the term applied to the process of modeling the essential features of a situation, and then predicting what is likely to happen by operating with the model case-by-case—i.e., by estimating the results of proposed actions from a series of imaginary experiments (imaginary because they are performed on the representation of the situation, the model, rather than on the situation itself). Most often, and most usefully, the simulation is a computer simulation in which the representation is carried out numerically on a digital computer, frequently without using formal analytic techniques. In a fair number of cases the computer simulation forms only part of the model, other aspects being simulated by human participants who interact



with the computer and each other and often represent their real-life counterparts or some sector of the problem that does not lend itself to numerical representation.

A great advantage of computer simulation for investigating complex problems is that a digital computer can be used to represent, with precision, processes for which satisfactory analytic approximations do not exist. The description of an intricate process, say traffic control, can be set up out of elementary activities. Traffic flow, for instance, can be expressed in terms of simple events (such as a car turning left at an intersection or a vehicle preparing to park) and simple rules (such as that when turning left, a car waits until oncoming traffic has gone by, or a vehicle attempting to park forces following cars to stop until it has completed its parking maneuver). Typically, a real system is subject to chance elements; these can be taken into account in the computer program by the use of random numbers. The computation is carried out with relationships that imitate the manner in which real activities might take place in real time. A large measure of realism can thus be attained. A high-speed digital computer is an ideal device for performing the massive bookkeeping required to deal with the large number and variety of elementary events.

To give an idea of how a police activity might be simulated on a computer, consider the following hypothetical problem. A single patrol car is assigned to an area where calls for service occur randomly but on the average of one per hour. Assuming it takes 36 minutes on the average to service each call, what sort of delays might a caller in this area expect before the car is free to respond? While it is possible, by making certain assumptions, to develop an analytic model to answer this question, a typical analytic model would ignore complications such as rest breaks, actions that might be initiated by the patrol officer, and mechanical breakdown of the vehicle.

A simulation model, on the other hand, could imitate step by step all the events related to the patrol car, and incorporate as many complexities as were deemed important. When the simulation is operated, the first event might be a call for service occurring 53 minutes after the patrol car starts on duty. The model would then imagine a time at which the car is finished handling the call. The next event might be an on-view incident at which the patrol officer makes an arrest. While the officer is proccessing the arrestee, a second call for service is imagined to occur, and the caller must wait for the patrol car to become available. This process continues, with the simulation model keeping track of the length of time each imaginary caller has to wait and then providing appropriate averages and other statistics at the end of the run.

Simulation with a high-speed digital computer is a powerful technique. It has some drawbacks, however. The ease with which a simulation can be put together makes it tempting to employ the technique where insufficient data exist to justify such a model. Because of the apparent concreteness and detail, a misleading air of realism can be imparted to the model, which masks the incomplete information on which it is based. Construction of the program is time-consuming and usually requires many revisions. This, together with the very large number of cases that may have to be run to attain an accurate result, can make a simulation very expensive.

### Operational Gaming

An operational game is a simulation involving human participants acting as simulators for at least some aspect of the problem—that is to say, an exercise in which an attempt is made to learn something about what is likely to happen by having the participants simulate the actions of individuals (highjackers, for instance), or factions in a society, or even such things as sectors in an economy. Operational gaming is an outgrowth of military war gaming, a procedure that has had a long history of usefulness for training and for testing war plans, and, more recently, has become a tool to study future weapons and potential conflict. Military and business gaming is now widespread, but the extension to the investigation of public policy problems is in its infancy.

Gaming was originally developed to investigate the problems of a decisionmaker whose actions might be countered by those of one or more intelligent opponents. It thus offers a way to investigate such issues as organized crime or youth gangs. Since the activity of the participants in such encounters usually bears some resemblance to playing a game, to term it gaming may be reasonable even though "play" may be against a computer program.

A game, say one to investigate policy options in the field of organized crime, might be formulated as follows:

1. A player team, Blue, to simulate in some sense a National Council on Organized Crime plus local authorities;
2. A player team, Red, simulating the activities of organized crime in city X;
3. A control or umpire team, Green, to structure the game, provide a start-up situation, rule on moves, etc.

The game would start from an initial situation (prepared by Green) with a move by Red—e.g., various actions involving gambling, loan-sharking, dishonest businesses, and the like. This would be followed by Blue's move, involving mainly actions by the local authorities. The results would then be evaluated by the control team, taking into account both the local moves and the legislative and operational components of an overall strategy to combat organized crime previously formulated by Blue in its role as a National Council; the activity of preparing this latter is probably the most important aspect of the game.

After the results are communicated (in part) to the player teams, another move follows. The control team determines the number of moves and the timing, updates the scenario, and provides information about such factors as the state of the economy and the political situation. Conclusions are drawn at the end based on the experience of all concerned.

Few question the valuable role of games for the education and training of participants, for improving communication among players with diverse backgrounds, for generating hypotheses, and for providing insight. Since predictive quality is so clearly dependent on the intuitive insight provided by the participants and controllers, the extent to which the results of games can be used to support policy recommendations is still the subject of controversy.

Gaming is an approach one can use to tackle problems of that wide class for which no satisfactory quantitative model can be constructed. A game can furnish the players with a very realistic and concrete evironment in which they can jointly and simultaneously experiment, acquiring a kind of experience to guide their judgment. The participant is forced, no matter how narrow his expertise, to consider aspects of his actions that might not weigh heavily on his mind were he working in



isolation. By allowing for the introduction of judgment at every step, a game provides the opportunity to take into account intangible factors often considered completely beyond the scope of analysis—courage, cooperation, commitment, and morale, for instance. In an analytic formulation, decisions about such things along with others must be made in advance; in a game they can be made one at a time, in context and as the need arises.

## Group Judgment

The use of a committee or panel to provide advice on a decision or policy is a time-honored, well established, and much used procedure. The common way for such a group to arrive at its recommendations is by unstructured, around-the-table discussion with face-to-face confrontation by the group members. This procedure is open to a number of well-known objections and often leads to very biased and ill-considered recommendations. A number of ways to improve the procedure by structuring the discussion have been suggested, the most promising of which, other than gaming, appears to be the Delphi approach. (See, for example, Dalkey [52,53].)

Delphi is a procedure for arriving at a forecast or estimate by eliciting and refining the opinions of a group of people by means of a series of individual interrogations. Since it can serve the same roles as a model, providing insights into or predictions about a contemplated action, a Delphi procedure can be considered an extended form or at least a replacement for the standard representative model. While, in cases where the results can be checked, the accuracy of Delphi estimates and predictions is in general greater than that obtained from unstructured committee discussion, Delphi is not a substitute for an analytic model or simulation unless one feels so little confidence in their validity that he is willing to depend on committee judgment instead.

The Delphi approach is characterized by three simple ideas: anonymity, iteration and controlled feedback, and statistical group response.

1. **Anonymity.** The participants are queried and they respond by means of a formal mode of communication. Originally, this was by a written questionnaire but recently, with increasing frequency, by online computer console. In determining an estimate or prediction, the responses are not matched with the respondents, and even the identity of the participants may be concealed from each other until the end of the exercise.

2. **Iteration and Controlled Feedback.** Discussion is replaced by an exchange of information controlled by a steering group or exercise manager. After each round of questionnaire, all or part of the information generated in the previous stages is fed back to the participants in order that they may use it to revise their earlier answers. In this way, "noise"—irrelevant or redundant material—can be reduced.

3. **Statistical Group Response.** Although the group opinion tends to converge with feedback, the normal outcome is a spread of opinion even after several iterations. Rather than making an attempt to force unanimity, some form of *summary statistic*, usually the median, is used to represent the group response. This way of defining the group judgment reduces pressure for conformity and insures that the opinion of every member plays a role in determining the final response. To illustrate

the potentialities of Delphi for problems of the criminal justice system, we outline how it might be applied to allocating a budget for crime prevention.[2]

To begin with, one might ask a panel drawn from the policymakers, their advisors, and experts familiar with the area, to list measures that they feel should be included in any program. There will always be alternatives competing for funds: more police, better training, changes in court and parole procedure, new laws, and so forth. In the usual circumstance, not all promising measures can be financed and only part of the budget can be used for new measures; invariably, some portions are already firmly committed for previously contracted obligations such as pension payments. The problem is to devise a scheme to suggest and compare alternatives, and to select a preferred allocation of the freely disposable residue of the budget.

Now, a measure is rarely, if ever, of an all-or-nothing kind; that is, there is a degree to which it can be executed. Salary raises, support of research, retraining programs, subsidies to youth centers, are all of this kind. Even in the case of one-time actions, such as building a new correctional institution, clearly there are aspects under budgetary control—such as the expected time to completion or the size and quality of effort—which can be reflected in the degree of its acceptance. Hence, in addition to the measure itself, its degree of adoption should be suggested. (This could be measured in dollars, numbers of police officers to be trained, amount of equipment to be purchased, or whatever is appropriate.)

The directors of the study would refine the original list. Measures strictly complementary, in the sense that neither can be adopted meaningfully in the absence of the other, should be combined. (If the directors feel they are sufficiently knowledgeable in the area, they might even want to add or eliminate measures.)

In order to reduce the number of alternatives and to provide a basis for costing, each panel member might next be asked to estimate (for each alternative or measure) two numbers, the lowest amount of adoption that would be sensible in his opinion, and a highest amount of adoption, above which the marginal benefits are so small as to make additional adoption wasteful, or where the cost would exceed the available budget. (One would expect many of the highest values to be zero, indicating total rejection of the measure.) The estimates of the panel can then be combined to establish two approximate bounds, representing a consensus as to an amount of adoption below which the adoption of the measure would be pointless, and a value above which the marginal benefits are so small as to make a higher degree of adoption wasteful.

After obtaining these numbers for each alternative, the directors would be ready to obtain cost estimates. Because the costs are future costs, they cannot, in principle, be fixed with any great accuracy. The next step would be to ask a team consisting of people with costing experience to work out an estimate of the amount required to implement each measure at adoption levels in the range of interest. (Of course, the expected cost of a measure depends to some extent on other measures that might be adopted, but at this stage we must largely ignore such interactions.) Depending on the state of our knowledge about costing in this context, a Delphi procedure might or might not be used here.

Next, estimates of effectiveness, or benefits, associated with each alternative must be obtained. For this we return to the original panel. Here no ready-made unit

[2] The procedures outlined here were originally suggested in a more general form by Olaf Helmer.



of measurement, comparable to the dollar in the case of costs, is available. Whereas certain consequences may have objectively measurable effects (better police training may result in an increased number of convictions following arrests, for instance), any proposal to fight crime, such as giving police greater freedom to gather evidence, is likely to have a multitude of effects that are incommensurable because their evaluation will depend on individual subjective preferences. They are also likely to have the inherent vagueness characteristic of social attitudes. Hence, to communicate the values to be assigned to the various alternatives among members of the panel, some unit of measurement, however vague, must be established.

One way to do this is as follows. Take the initial situation (that is, with only the precommitted measures in existence) as having zero value. Imagine the unknown budget allocation that the appraiser would regard as optimal to have a value of 100. Each panelist would be asked to assess the contribution of specified amounts of each measure as the percentage by which it, considered in isolation, would raise the initial situation toward the "ideal" situation.

By combining the panel's value estimates with the cost estimates, the directors can now construct, for each alternative measure, a curve of effectiveness versus cost, as in Fig. 2. (The dotted lines represent the previously agreed-upon lowest and highest sensible amounts of each measure.)

A first approximation to the desired budget allocation can now be obtained. One way to do this is to use Delphi again, asking the members each to make what they

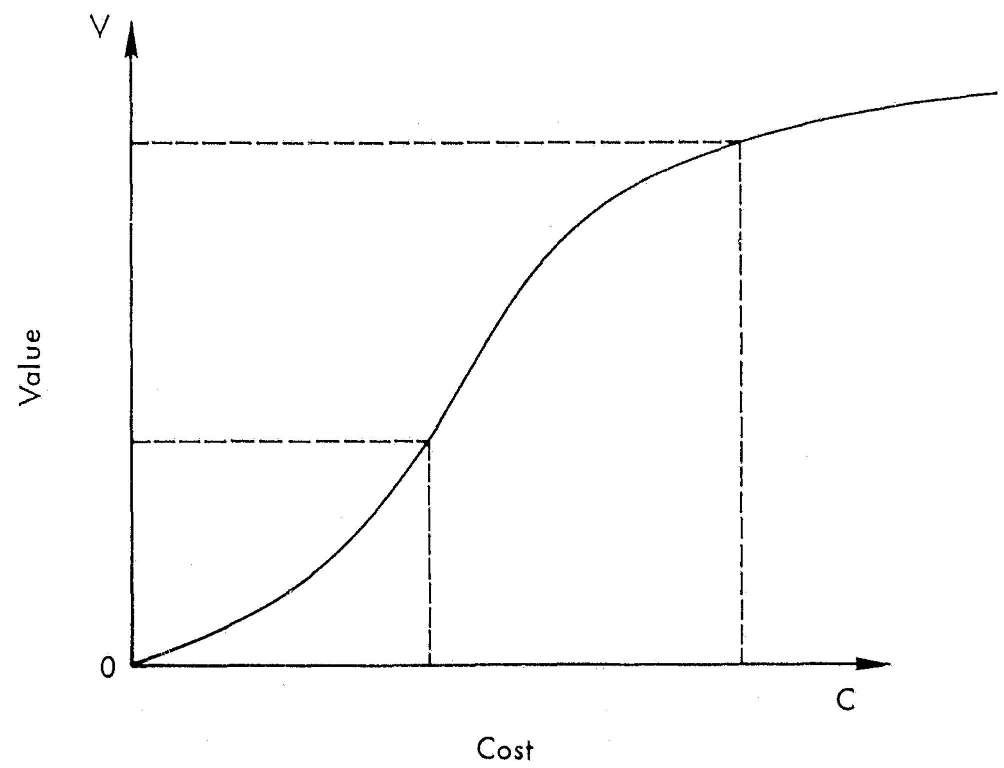


Fig. 2—Value as a function of cost

consider to be the optimum allocation, basing their judgments on the relations between effectiveness and cost as given by the curves of the type (Fig. 2) just derived.

This approach has many deficiencies. For one, were the budget to be implemented and the various recommended projects carried out, it is unlikely that the actual effectiveness obtained from a given measure would be identical with that projected earlier when the measure was considered in isolation. Also, both the costs and benefits are preliminary estimates, based (partly in the case of costs and wholly in the case of benefits) upon judgment. The budgetary process is likely to generate among the participants an increasing understanding of the implications of each decision they consider, and thus lead them to revise their earlier estimates. Consequently, more than one iteration of the entire process may be required.

Delphi is not an opinion polling technique. Its purpose is not to furnish the investigator with data about the respondents but, rather, to estimate the answers to questions for which there is no well-defined way to find a definitive answer at the time of the exercise. In comparison with the customary, informal types of individual and group utilization of experts that are prevalent in the advisory community today, Delphi techniques offer a way to introduce a systematic approach to problems where conventional models cannot be formulated.

## APPRAISING MODELS

Models may be strong in some aspects, weak in others, useful for one policy question but totally irrelevant for a closely related one. They can be appraised or compared only in the context of a particular policy decision and a host of other considerations. In this section, we give some guidelines for appraising models. The descriptions of individual models that appear in the next four chapters are designed to help the reader follow these guidelines in determining whether any of the models is suitable for the purposes he has in mind.

The guidelines are intended to be followed by an administrator who is considering whether to fund the adoption or design of a model. No hard and fast rules can be given for appraising models, only questions that should be raised and judgments that should be made. In the case of models that *already exist,* following the guidelines should not be difficult, although some careful reading of documentation may be required, and possibly some telephone calls and site visits. In the case of *proposed* models, the administrator will have to make uncertain judgments, since the ultimate characteristics of the model cannot be known in advance, and the possibility must be considered that the proposed model will never be completed in a form suitable for application.

### Match Between the Policy Issue and the Model

The first step in acquiring a model is to identify the policy issue to be addressed by using the model. The issue may arise in the administrator's mind, either in vague form (e.g., improving allocation of resources) or in specific form (e.g., designing new patrol beats for police cars); or the capabilities of an existing model may suggest the possibility of addressing an issue not considered previously.

Once the policy issue has been identified, the following types of questions should be asked. Is this problem of interest to my agency or to some group that is benefitted by my agency? Why is it of interest? Is the problem formulated properly, or is it just a symptom of a much larger and deeper problem? Why has the problem not been tackled or solved before? (In particular, has an absence of the kind of information provided by the model played any role in preventing previous solutions?) Is there any reason not to tackle the problem?



If the analysis can be carried out successfully and advice provided, what will be done with the results? Will anybody be able to act on its recommendations? Is the inquiry politically sensitive? Is it likely to commit the agency to continuing support?

Essentially, one wants to find out, before the tedious task of working through an elaborate model to determine if its predictions are correct, whether the results are likely to be worth the effort, and, if they are, whether anyone is willing and in a position to do anything with them.

The specificity of the problem definition has much to do with the desired scope of the model. If the problem is vaguely stated, then the model should be flexible and have multiple capabilities. If the problem is specific, then a model designed for the particular application will probably prove most satisfactory, unless an existing general-purpose model is easily adapted to current needs.

The nature of the decisions to be made with the model should also be considered. In particular, it is important to distinguish between *one-shot* and *recurring* decisions. Even though decisions to be made once (or infrequently) may well be the most important ones, justifying the time, effort, and cost of using a model for such decisions may be much more difficult than in the case of decisions made annually, monthly, weekly, or even continually during the course of each day.

In the case of *recurring* decisions, it is important to consider *how often* the model is to be used. If the model will be operated at least every month, then there is unlikely to be any problem of users forgetting how to access the computer program and make it work properly. But if the model is to lie dormant for six months or a year at a time, even those personnel who were previously most knowledgeable about the program will have to refresh their memories in order to operate it successfully again. A changeover in personnel during the intervening time can even leave the user agency with no one who knows how to run the model when the occasion arises.

In some cases the decisionmaker knows not only the problem to be addressed but also the solution he plans to propose. He anticipates that a model will confirm his decision and assist him in persuading others to adopt his plan. This is a legitimate use of models, but one should be prepared for the possiblity that the outcome will not be as expected, and the model's results will persuade the decisionmaker to adopt some alternative solution.

### Time until Completion

Designing a new model can easily require a year or more elapsed time before it is ready for its first use. Even with existing models, months or more may pass before the program works properly and appropriate data have been collected. The administrator must ask himself whether the timing of the desired decision is compatible with the timing of model implementation. Will current interest in this problem fade before the model is ready for use? Can a solution to the problem wait, or would it be better to take some action now, even if it must be based on inadequate information? Will budgetary or legislative schedules force a decision before the model is operating? Is the proposal to use a model simply a delaying tactic to forestall an inevitable decision? These questions are likely to be least troublesome in the case of recurring decisions, since current practices can be either continued or modified before model completion without precluding the possibility of later changes based on analysis.

### Technical Quality

Model builders are likely to be concerned with issues of technical quality that are of little interest to an administrator. Chief among these are that the model should be nontrivial (its output could not be easily guessed in advance), powerful (it offers a large number of nontrivial inferences), elegant (it uses a minimum number of carefully selected analytical tools), and efficient (it uses no more equations or computer space than necessary for its purpose). For optimization models, there is also a technical distinction between algorithms (procedures within the computer) that are *guaranteed* to find the optimal (highest or lowest) value of the objective function[3] and *heuristic* algorithms, which yield good, but not necessarily optimal, solutions. Most administrators would not have the technical expertise to make judgments about such matters.

While an administrator should attempt to assure himself of the technical competence of the model builder, through references, examples of previous work, and so on, it is more important for him to determine whether the model builder has policy relevance uppermost in his mind. Otherwise, the designer has no guide to tell him what to include in the model and what to leave out. In the pursuit of technical quality, the size and complexity of a model tend to increase up to the limits of the computer's capabilities or the availability of funds. In particular, an administrator should beware of a "technique in search of a problem." The person who raises the possibility of designing a model in the form "Do you have a problem I can solve by nonlinear programming?" is unlikely to produce a satisfactory product for the user agency.

Every model goes through a period in its development when programming bugs or other errors cause the model to malfunction. These are ordinarily not fatal and should not unnecessarily discourage the administrator. Even the best-tested programs occasionally present unexpected problems. As long as there is some means for correcting errors when they do occur, the user should not judge the overall technical quality of the model by such isolated occurrences, saying "The model does not work."

### Verification and Validation

Verification and validation are two aspects of quality that a potential user can check for himself, and should. A model is said to be *verified* if it does what the model builder intended for it to do. This means that its equations are correct and have been properly programmed. Typically, a model is verified by testing it with sample data that correspond to known output, by setting some of the data input to extreme values (say zero), or by holding some of the variables constant to determine whether

[3] This is an expression for the quantity to be minimized or maximized.



the output changes in anticipated ways as the other variables are changed. In many cases a model is verified by checking its output against results provided by previously verified models. An administrator can determine whether an existing model has been verified and can require that a proposed model go through this process.

*Validation* refers to checking that the outputs of the model conform to reality. In some cases the model's output can be compared to data from historical sources or from an experiment conducted for validation purposes. However, easy validation is the exception rather than the rule. Models that predict what will happen in the future are particularly difficult to validate, since a model may fit past data well without having good predictive qualities. Even if a model has been previously validated in another jurisdiction, there is no guarantee that it will be correct in a new application. Therefore, administrators should require that validation be undertaken wherever feasible in connection with new applications.

Validation by the judgments of experienced agency personnel should not be underestimated. Before a model can be fully accepted, its outputs must "feel right" to those who will make decisions with it. If a model fails to pass this crucial but very subjective test, the chances that it will be used are slim indeed, even though it may be correct. If the model builder is convinced that the outputs are more accurate than agency personnel are willing to believe, a formal validation experiment will probably be required.

### Data Requirements

Models differ greatly in the amount and level of detail of data required. An important question to ask is whether most or all of the required data are available. If not, is any way known to obtain them, or must they be estimated? Is invasion of privacy an issue in collecting the data? Are there any legislative restrictions on the nature or form of the data?

If two models are equally satisfactory for answering the policy issue at hand, and one requires less data or more readily available data than the other, then it is to be preferred. However, a need to collect substantial amounts of data, if understood in advance and planned for, is not necessarily a negative feature of a model. In many instances, one of the most useful functions of a model is to focus an agency's attention on the types of data that will best serve subsequent management purposes.

### Cost, Personnel, and Computer Requirements

Questions of cost arise at several levels:

- Designing or acquiring the model
- Collecting data
- Operating the model on a computer system
- Analyzing the output of the model
- Implementing the decisions arising out of the analysis.

While information concerning computer costs is more likely to be available than information about the other costs, in most cases the computer costs are a very small fraction of the total. Only fairly complex simulation models entail computer costs large enough to be a factor in whether or not to use the model. In the case of existing models, information from prior users is likely to be the best guide. However, one must understand that the first use of a model is substantially more costly than subsequent uses.

Some models can be operated by persons having little or no technical training. Others require the assistance of specialists in a particular programming language or statistical technique, who will ordinarily not be found working for a governmental agency. An important question, then, is whether the model can be operated by the intended user agency. If outside assistance will be required, both the costs and the chances of ultimate acceptance are affected. In most cases, assistance by local consulting firms or university professors and students is likely to be most satisfactory, because frequent interaction is needed between the implementers and the potential users.

Computer requirements become an issue if the model is "too big" to fit on any computer system accessible to the user agency (i.e., the computer does not have enough storage space), or the model is written in a programming language that cannot be compiled by the potential user. Special features of the computer system, such as the availability of interactive terminals, may also be relevant for some models.

### Documentation

For most models that might be considered by an administrator (including all of the models listed in later chapters of this report), written materials are available describing the *concepts* by which the models operate. However, unless detailed user's manuals are available, an agency will have to contact someone who already knows how to operate the program if it wants to use the model. In the case of existing models, the nature of available documentation can be readily determined. For proposed models, the administrator should require in advance that appropriate documentation be prepared, unless there is no plan for anyone other than the model builder to operate the program.

### Implementation and Impact

Perhaps no better means is available to an administrator for appraising a model than to find a case example of successful implementation, especially if the application had a favorable impact on some agency's operations. An example of dismal failure, however, is not necessarily to be taken too seriously, for reasons that will be discussed in Chapter 7.

## OVERVIEW

For describing models in Chapters 3 through 6, we have adopted a structure that corresponds closely to the topics just discussed, with a few self-explanatory additions:

- Historical Background
- Policy Issues Addressed
- Structure of Model



- Data Base Required
- Output
- Cost and Computer Requirements
- Validation and Verification
- Implementation and Impact
- Limitations of Model
- Transferability
- Documentation.

## Chapter 3

# OVERALL MODELS OF THE CRIMINAL JUSTICE SYSTEM

## INTRODUCTION

Some criminal justice models apply to only one of the major components of the criminal justice system (CJS): police, courts, or corrections. These will be described in later chapters. A model of the overall CJS must integrate the three components, not only modeling the behavior of each component itself but also including the interaction among them. If overall system policy changes are contemplated, affecting for example the composition and character of the flow of offenders between police and courts, such an overall CJS model will be required.

Interactions among the elements of the CJS are really of many types, but models have focused on offender flow, and the primary interactions here consist of downstream effects (e.g., some arrests lead to work by trial judges) and feedback of recidivists. Since recidivist arrests constitute approximately 85 percent of all arrests, the treatment of arrestees by prosecution, courts, and corrections presumably has a major influence on the number and type of crimes committed and on the population of corrections facilities. For example, a change in parole policy can affect the makeup of arrests, thereby affecting the workload of the courts, which ultimately influences corrections, the initiator of the change. The term "feedback," in this example the feedback of recidivists, is used to indicate a form of interaction in which downstream events affect later upstream events.

Overall CJS models are useful tools for planners even though there is currently no organizational structure, other than perhaps within the military, with single management control over a total CJS. The importance of the broader approach, however, has not been neglected by CJS planners. The President's Commission on Law Enforcement and Administration of Justice recommended that a closer relationship be developed among the elements of the CJS [36]. Funds to stimulate this were provided by the Omnibus Safe Streets and Crime Control Act of 1968 [147], which directed federal funds toward State Planning Agencies (SPAs) for developing comprehensive plans to improve law enforcement.

The State Planning Agencies, using their federal and state funds, have added to direct funding to researchers from such federal agencies as the National Science Foundation or the Department of Justice to aid development and implementation of overall CJS models. Although these planning agencies are not managing the CJS, but can only recommend and stimulate programs believed to be valuable through funding, these models are beginning to show their long-promised value.

Several overall CJS models have been selected for detailed examination in this chapter. The history of their development, the policy issues they can and cannot address, the theoretical underpinnings of the models, and the manner and extent of their current use will be described briefly. Essentially similar descendants or predecessors of reviewed models are mentioned and their internal and operational differences are described. A discussion of their successes and failures and of problems experienced in implementation will be given later (Chapter 7). However, seri-





ous restrictions and/or limitations that should be considered in any implementation will be dealt with here.

## THE JUSSIM MODEL

### Historical Background

The best-known overall CJS model is JUSSIM, designed by Belkin, Blumstein, and Glass in the Urban Systems Institute at the School of Urban and Public Affairs, Carnegie-Mellon University [11]. The basic concepts of the approach were expressed in 1964 [164] and 1965 [172], and expanded and applied to overall U.S. data in 1967 [17]. The major testing and validation for this approach was based on data from California [18,19]. Detailed flow, cost, and workload estimates were made for the state and the model was run, using the distribution of reported crimes as input. Output measures from the model were judged as reasonably good predictors of real-life observations. This work included recidivism feedback, which was to become one of the important features of the second-generation model called JUSSIM II. The technical aspects were presented in a theoretical journal in 1969 [18].

The JUSSIM modelers have taken two major approaches, each represented ultimately by a different model. JUSSIM, first completed in 1970 [10] and undergoing continued improvement since then [11], was funded by LEAA first at the Institute for Defense Analyses, where many of the design concepts were formulated. Then at Carnegie-Mellon, LEAA monies supported the original interactive computer model and implementation trials with a State Planning Agency for the State of Connecticut.

JUSSIM II, an interactive feedback model for criminal justice planning [12], represents the other approach, which takes account of the feedback effect of recividists on the types and numbers of crimes. Its concepts grew out of limitations of the orginal JUSSIM. The second-generation JUSSIM II was designed in 1973, based on earlier research [18,172]. JUSSIM II will be discussed in some detail later in this chapter.

### Policy Issues Addressed

JUSSIM can be used to address policy issues that propose changes in the flow and processing of crimes, offenders, and prisoners in the overall criminal justice system. It provides the user with estimates of the first-order effects on the workload and costs at each of the system processing stages under each of a number of such proposed changes. The model forces the user to quantify his intuition about the interactions between one part of the criminal justice system and another. It is just one part of the total planning process in which the policy planner uses the model to describe the possible impact of a proposed change on the total CJS.

Since the user must design the proposed change, that is must postulate the structural changes in the CJS and estimate or hypothesize the parametric changes characterizing the new program, and then make value judgments selecting among the proposals, the results are only as good as the sophistication of the user. The model becomes a tool to help select the better programs based on feasibility of implementation and on JUSSIM's predictions of the performance.

## Structure of the Model

The JUSSIM model is an interactive computer program that operates on a data file representing the user's criminal justice system. The first step in using JUSSIM is to prepare the base case data file which will represent the model of the user's criminal justice system. The CJS must be modeled by the user as *a linear steady-state production process* (these terms will be explained shortly) where crimes and associated offenders are the basic unit of flow, and the processing stations are the different stages through which the arrested offender passes. These types of models are similar to traffic flow models in which the movement of automobiles traveling through a network of streets is predicted, the model maintaining aggregate records on performance measures such as flows and resources applied in processing the flows. A simple chart of this flow process has already been shown in Fig. 1.

A *steady-state* model is one in which the parameters of the system do not change with time, and the long-run characteristics are all that are judged relevant. A *linear* model is one where parameters are independent of one another and independent of flow rates through the system.

The CJS model is constructed by the user and is often graphically displayed as a flow chart. Figure 3 shows a simplified version of the diagram representing a model of the Allegheny County CJS. Individuals, both recidivists and new offenders in society, perpetrate crimes. Some crimes are detected, some not; some reported, and some not. Reported crimes are processed by the police, arrests are made, and a fraction of arrestees are charged with a crime. These arrests become cases to be processed by the courts, and those convicted are assigned to the corrections subsystem. Parole and eventual release return convicted individuals to society.

The emphasis of the model is on the units of flow, usually offenders, criminal acts, prisoners, etc., which advance through the system by completing processing in various stages. At each stage, the units of flow consume resources, such as the time of police officers, prosecutors, and judges, and the model calculates the rate of consumption of the resources. The output of each stage goes to alternative stages in proportions called *branching ratios*.

The actual flow system in the JUSSIM model is substantially more complicated than indicated here, and in turn the real-life flows are more complicated than those that can be modeled. JUSSIM is not a case-by-case simulation in which each offender is followed through the system, but rather considers offenders in aggregate groups whose behavior can be described by the branching ratios.

## Data Base Required

In building the data file for use in JUSSIM, a set of crime types must first be established. This is done so that differences in flow can be taken into account. While typical legal categories of crimes can be used for this purpose, the "crime types" can also be distinguished by sex or other characteristics of offenders that may be related to their processing through the system. Resources, in terms of police, judges, attorneys, correctional officers, probation officers, etc., that are required to process flow, are defined as located at the appropriate stages. Next, the stages in the CJS are described, representing at an aggregate level the processing of the flow. With this structure, then, data are required on workload (measured by the time it takes each resource at each stage to process a unit of flow for each crime type) and on the unit



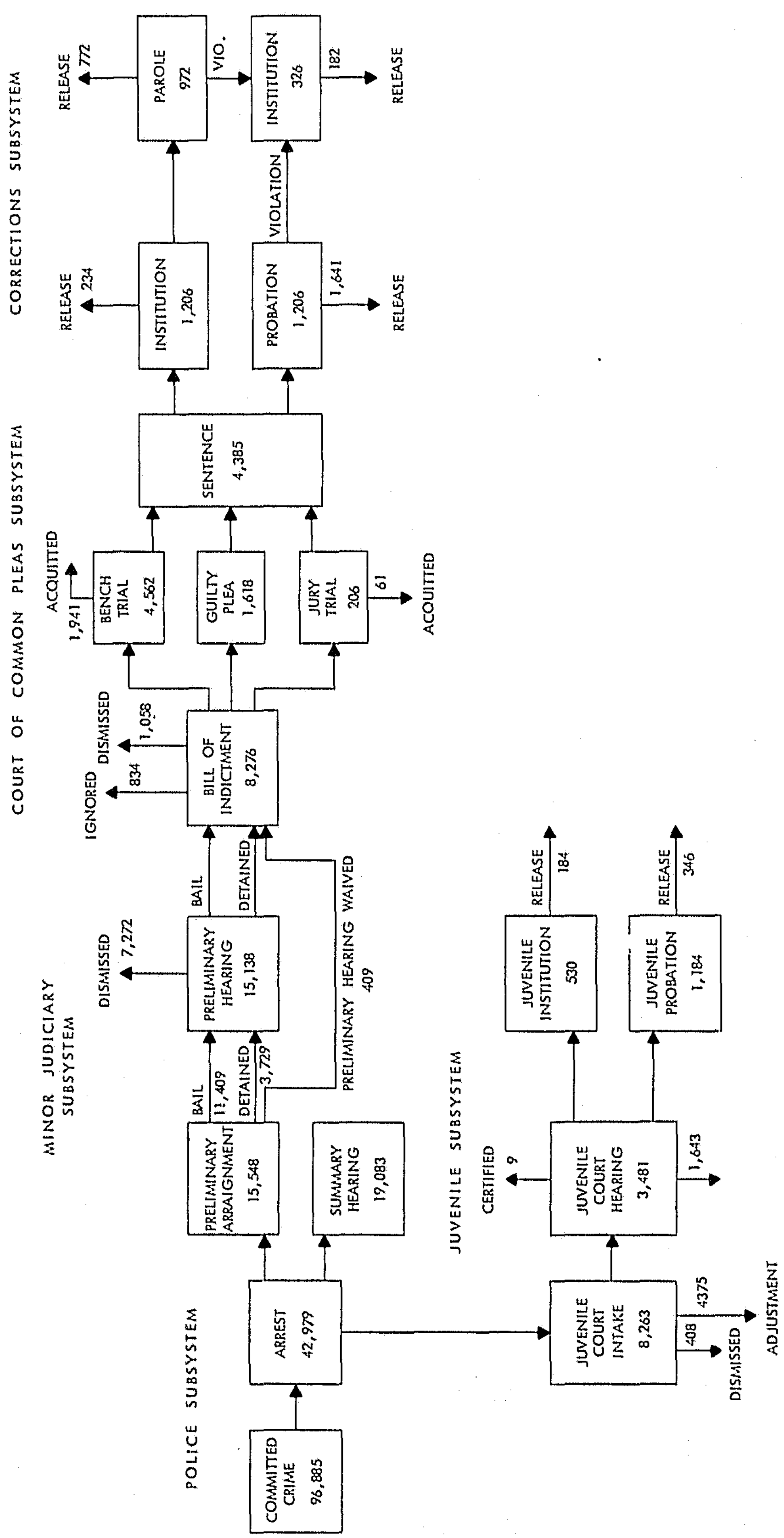


Fig. 3—Flow diagram of the Allegheny County criminal justice system, redrawn from Ref. 42, p. 120. Numbers represent estimated annual flows of individuals.

costs for each resource measured in dollars per unit time. These data for the user's system must be supplied as input for JUSSIM. This data base represents what is called the "base case," and a run of the model will produce base case output characterizing the system.

Data for a typical CJS are developed by first developing a chart or diagram of a model of the flow (as in Fig. 3), indicating how all crimes committed flow through the system.

### Output and Use

The model calculates the downstream flows, the total costs, resource requirements, and workloads. These system performance measures are available in a disaggregated form to provide the user with cost, resources, or workload for each stage, crime type, or subsystem. The user sits at a computer console and creates a "test case" by making changes and additions to his "base case" data file. This is accomplished by answering a series of prompting questions directed from the JUSSIM program. JUSSIM was designed to operate in this interactive mode so as to make it accessible to the user who does not know computer programming, and to bring his judgment into the analysis process.

The user imagines a possible change in operation of his CJS. Such changes may be potential improvements associated with planned programs or they may be legal or budgetary changes. These changes must be translated to structural or parametric changes in the input file. The values of those parameters may be derived from tests or experiments; they may be reached through a consensus of experts through a Delphi process; or they may reflect only the judgment of the user. As with any model, such parameters must be externally generated, and cannot be derived from operation of the model. Once satisfied that the new data file, called the "test case," appropriately reflects the system change, the user runs the model, and the output is displayed at the terminal. Continued iteration of this cycle can be made at the user's option to test a number of different estimates of a particular change or of different proposed changes.

The process of preparing test case data for JUSSIM helps the user recognize the importance of indirect consequences of policy changes. Changes in one stage not only affect other stages' flows and workloads, but might also affect their parameters. Such effects must be considered by the user when he prepares his test case data file. For example, a change in the police subsystem that the user thinks will result in increased burglary arrests (e.g., new hardware, such as radio access to a computer file of arrest records) might have an indirect effect of reducing the chance of indictment (as a result of the larger number of burglary cases in court). Such an indirect change must be entered by the user in the form of a reduced indictment branching ratio in his test case input to JUSSIM.

These relationships are not modeled within JUSSIM itself, because the designers recognized that any assumptions made by them might not reflect actual operations in a particular CJS. Incorporating assumed relationships in the model would simply hide from the user the complex judgments he has to make to understand his system. While estimating the test case data may be conceptually difficult for the user, JUSSIM's designers felt this was necessary until empirical knowledge identifies the relationships better.



The output of the JUSSIM run on each test case data base is used to evaluate the effect that the postulated test case changes might have on the flow, costs, workloads, and resources in various parts of the CJS. Comparison among test cases and with the base case can be made to assess each proposal on these grounds.

One important contribution of models of the overall criminal justice system is to the development of statistics and to the collection, aggregation, and comparison of data that are consistent and compatible across the entire CJS. These would replace the data that are more normally gathered independently in the CJS at different levels and places, and which often cannot be aggregated or presented in comprehensive form for coherent analysis. Thus, the data collection and analysis forced upon the user is one of the beneficial aspects of using the JUSSIM model, but at the same time current data incompatibilities represent one of the most important obstacles to its use. The interactive mode, allowing users to work from remote consoles, trains the user to think of the CJS in terms of system ramifications and gives the model the characteristics of a "management game."

JUSSIM, or models of similar type, can be viewed as a catalyst for establishing the data collection and decisionmaking organizations necessary for improvement in any CJS. Furthermore, the input and output requirements of the model are easily understood, and comparison of proposed changes with the output base case allows a user to see quickly where in the CJS the impact of policy changes will occur. Simple hand calculations could have been made to evaluate the same changes, but the JUSSIM model does them rapidly and frees the user to think about the policy changes and the interrelations among parts of the system.

### Cost and Computer Requirements

The computer program is written in FORTRAN IV and can be used on any machine with a FORTRAN IV capability and memory storage capacity of 32K (K = 1024) words of storage. Since two-thirds of this storage is for data, this requirement can be reduced in some applications by permitting fewer crime types, stages, workloads, etc. While JUSSIM is intended to be run on time-sharing computers in an interactive mode, it could easily be operated in a batch-processing mode for running a large number of test cases.

The mathematical operations required for JUSSIM are simple, and computer time is dominated by user input speed and the amount of output requested. A copy of the computer program and a detailed manual are available from the Urban Systems Institute for a nominal charge.

As with all CJS models, constructing the data base is the most difficult and expensive part of implementing JUSSIM. The program designer Blumstein estimates that it will cost $50,000 for the average urban CJS to provide a minimum of detail for analysis and some preliminary meaningful experimentation with JUSSIM. The Los Angeles area State Planning Agency (SPA), in evaluating the possibility of implementing JUSSIM, estimated roughly that data gathering in more complete detail and implementation of the model would run to almost ten times this amount.

It should be noted, however, that JUSSIM is a comparatively simple model and requires substantially *less* input information than more complex models.

### Validation and Verification

*Verification* refers to an examination of the internal workings of the model to make sure the model does what the model builder intended. This JUSSIM model is quite simple mathematically and has been verified.

*Validation*, which means examining and testing models to see if their predictive and descriptive capabilities are accurate, has not, to our knowledge, been conducted on the JUSSIM model. In the validation procedure, model parameters would be estimated on a set of data from, for example, 1970-1972, and the model's predictions for 1973 would be compared with actual 1973 results. An alternative method of validation would be to compare the predictions for cost and performance with the actual cost and performance of some special program that was instituted in accordance with JUSSIM predictions. While neither of these has been done, repeated application and testing does constitute informal validation; see for instance the application of JUSSIM concepts to California [18,19] or Allegheny County [173].

Validation of any model is often omitted. Even validation in one jurisdiction may not be adequate, because the model may perform very differently at a new site. Because of this, some checks validating model use ought to be made at the intended implementation site before placing confidence in the predictive capability of any model.

### Implementation and Impact

The natural testing ground for JUSSIM, given that its developers were located at Carnegie-Mellon University and worked with the Allegheny Regional Planning Council established by the Pennsylvania Governor's Justice Commission, was Allegheny County, which includes the city of Pittsburgh. Much of the data on offender flows and on resource costs and workloads, as described in flow charts such as Fig. 3, were collected by graduate students and the Allegheny Regional Planning Council (ARPC) staff during the development of JUSSIM. Currently, the ARPC updates the data base and uses the JUSSIM model regularly to explore various proposals for improving the CJS [173]. While initial data collection may be quite expensive, the establishment of a regular reporting system through a planning agency substantially reduces the continuing cost.

While the first and perhaps most successful implementation of JUSSIM to date has been by the Allegheny Regional Planning Commission, other agencies and companies have used JUSSIM (see Table 1). The impact that JUSSIM has had on the CJS systems is well documented in the Allegheny County implementation [43, 63,173]. The Maryland SPA has a pilot JUSSIM implementation in Prince Georges County. The California Council on Criminal Justice in Sacramento and the Denver Regional Planning Agency are reported to have implemented JUSSIM on at least a test basis.

Modifications of JUSSIM have been installed elsewhere in the United States and Canada. These modifications have seen implementation in Philadelphia [21,158], Alaska [158], and Canada [28,29,92]. The descendants of JUSSIM developed during implementation in Philadelphia and Canada are described later in this report. Discussion of implementation problems and organizational aspects of these models will appear in Chapter 7.



Table 1

PURCHASERS OF JUSSIM AND JUSSIM II

U.S. PURCHASERS OF JUSSIM

*Governmental*

Department of Health and Social Services, Juneau, Alaska
California Council on Criminal Justice, Sacramento, California
Denver Regional Council of Governments, Denver, Colorado
Washington Council of Governments, Washington, D.C.
Governor's Crime Commission, St. Paul, Minnesota
Robert J. Bradley, State Highway Patrol, Missouri
Division of Criminal Justice Services, Albany, New York
Law Enforcement Planning Agency, Salt Lake City, Utah
Governor's Commission on Crime Control and Prevention, Montpelier, Vermont
Southwest Wisconsin Criminal Justice Planning Council, Wisconsin

*Universities*

University of Arkansas
University of California, Los Angeles
Duquesne School of Business Administration (Pittsburgh, Pennsylvania)
L. R. McPheters, Department of Economics, Florida Atlantic University
National Clearinghouse for Criminal Justice Planning and Architecture, University of Illinois (Urbana)
Center for Criminal Justice Training, University of Indiana
Criminal Justice Systems Center, Michigan State University

*Other*

Arthur Andersen and Co., Detroit, Michigan
L. Charles Miller, Battelle Memorial Institute, Columbus, Ohio
Crime Prevention Systems Corporation, Charlottesville, Virginia
Family Health Foundation, New Orleans, Louisiana
Government Studies and Systems, Philadelphia, Pennsylvania
National Planning Association, Washington, D.C.
Peat, Marwick, Mitchell & Co., Los Angeles, California
Public Systems, Inc., Sunnyvale, California
Science Applications, Inc., La Jolla, California
Thomas Shaugnessy, Newport News, Virginia
Sperry-Univac, King of Prussia, Pennsylvania
Stanford Research Institute, Menlo Park, California

FOREIGN PURCHASERS OF JUSSIM

Ludwig Boltzmann Institute, Vienna, Austria
Ministry of Urban Affairs, Canada
Treasury Board, Canada
John Kuss, Vancouver, Canada
Sune Arkeus, Rikspolisstyrelsen, Stockholm, Sweden

PURCHASERS OF JUSSIM II

Michigan State University
Secretariat of Ministry of the Solicitor General, Canada

## Limitations and Benefits

Many advantages and disadvantages of JUSSIM have already been described, but all are summarized here.

The original JUSSIM model cannot provide answers or guidance in the following areas:

1. Congestion and associated delays due to saturation in stages of processing or paperwork requirements
2. Tracking of individual offenders or cases; rather it deals with aggregate flows
3. Effects of feedback and interaction phenomena through recidivism (addressed in JUSSIM II)
4. Flow times through the system, e.g., average time from arrest to final disposition, or number of trial continuances
5. Priority questions of handling cases, offenders, etc.
6. The size of typical variations in flows and workloads that arise from random events
7. Guidance to estimate the ramifications that the test case changes have on the system branching ratios.

Each of the proposals that can be tested by using JUSSIM has an impact on offender flow processing. The user is required to supply the estimates of parameters and structural features as test case input to the JUSSIM model. While JUSSIM is not designed to answer directly what will happen if more judges are added at a particular stage, it can be used to explore this proposal by estimating the number of judges required. If the number available is much less, the user must then estimate the branching ratios and flow rates that would occur with these fewer judges or with some additional judges. The workloads computed by JUSSIM would then be seen to change in the courts and subsequent stages, reflecting the effects of varying the number of judges.

Any policy proposal that can be expressed as direct changes in flow processing through parameters or even in the flow diagram can be tested directly. For example, the following types of proposals might be tested:

- The effects of drug offender diversion programs
- The costs and savings of changes in the bail release program
- The impact of a police crackdown on burglary
- The impact of an increase in psychological counseling during incarceration.

One of JUSSIM's benefits is its simplicity, which facilitates understanding and implementation, as compared to more complex models requiring substantially more input information. This has helped to centralize additional data gathering for use by decisionmakers or policy recommending groups.

JUSSIM succeeds in its stated purpose, subject to the limitations of its assumptions: to provide a means of testing large-scale overall CJS proposals. Its limitations relate to its simplicity, but every model builder faces a choice between a simple, usable model and a more complex model that may be difficult to understand. The naive user may not be aware of, nor have included as parametric changes to test case



data, the effect of interrelationships in system branching parameters that may result from the test case changes, and so he may ultimately be misled by the results of JUSSIM. In addition, JUSSIM is not suitable for use as an operating tool for daily microscopic criminal justice planning because of the aggregation of its flows and processing stages.

In summary, JUSSIM's primary value is as a catalyst for developing a data collecting and policy recommending organization for the entire CJS, and for focusing attention on the implications of changes in one part of the system on other parts. Implementation requires the development of a description of the overall CJS in terms of flows and stages, which even by itself is of value to decisionmakers. The model provides valuable training for users, teaching them to think about the large-scale ramifications of policy proposals. However, implementation of the JUSSIM model is limited by the data requirements and the organizational difficulties in sustaining the JUSSIM model, either in its role as a center for data gathering and report generation or in its role as a tool in criminal justice planning.

## Transferability

The model itself is and has been proven to be easily transferable—see examples under Implementation.

## Documentation

Documentation of JUSSIM is very complete and available from the Urban Systems Institute of Carnegie-Mellon University. Setting up the computer program and getting the interactive remote terminal working can be done in a straightforward way from information obtainable from the Institute.

## Relationship of Other Overall CJS Models to JUSSIM

JUSSIM has spawned several progeny. Two of the more successful, in terms of implementation, are discussed here, as well as the second edition of JUSSIM itself. PHILJIM [21,158], developed in Philadelphia, is basically the same model, modified somewhat with options designed for the special requirements of a new and different CJS. JUSSIM II [12], structurally quite different from JUSSIM, was designed by the originators of the earlier version to include the important recidivism feedback feature and the dynamic aging of the CJS from the initial condition over the next several years. CANJUS [28,29,92] is basically an application of JUSSIM in Canada.[1]

[1] Another JUSSIM type model, the Criminal Justice System Planning model developed by Fred McCoy at Ernst and Ernst in Washington, D.C., is another example of a successful application of large-scale CJS models. The study made for the Richmond, Virginia, Planning District is reported in three volumes, a technical manual, a user's manual, and a programmer's manual, May 1973; the documentation is well written and complete.

Deterministic aggregate flows of individuals are simulated through Police, Court, Corrections, and Rehabilitation modules, each of which contains a transition matrix based on collected historical flow data. Each module also estimates the resource requirements necessary to perform the functions in each simulated time period. The Forecasting module is based on socioeconomic inputs specified by the user and predicts the input load of offenses in 28 crime categories. Each module's input is the preceding module's output as the simulation continues forward year by year. The 28 crime types can be followed through each module year by year at user option.

The system is currently in operation for Richmond on a time-sharing Univac 1108 system maintained

## CANJUS

The CANJUS project was undertaken by the Ministry of the Solicitor General of Canada with the objective of developing a comprehensive simulation model of the Canadian justice system. The model was to provide a basic quantitative description, assist in planning and policy changes, and provide a basis for expanding research. Rather than develop a new model, the decision was made to employ the existing JUSSIM model. The name CANJUS refers to the project which designed the application to the Canadian system.

The details of the implementation are well documented in the project reports [28,29,92]. This series of reports contain the clearest descriptions available of implementation, from data preparation (with verbal description of each of the stages and parameters) to operating guidelines for the user. The CANJUS model uses the JUSSIM computer program for processing the data. This makes it an example of implementation of JUSSIM rather than a new model.

The CANJUS project staff have research underway to make significant changes in the JUSSIM model. Recidivism feedback has been given priority as one of the desirable additions to the basic model, an addition that is central to the new JUSSIM model called JUSSIM II. A crime generator to feed the flow of virgin arrests into the model as a function of Canadian social and economic characteristics is also under investigation.

The CANJUS system has the same beneficial aspects, can be criticized on identical grounds, and is useful to evaluate the same policy issues as the JUSSIM model. Any potential user of JUSSIM would be well advised to obtain and read the reports documenting this implementation in Canada.

## PHILJIM

### Historical Background

The Philadelphia Regional Planning Commission funded a project with Government Studies and Systems, Inc., to develop and introduce an annual planning system. Part of this project was a computer model and report generator called PHILJIM, the Philadelphia Justice Improvement Model [21,158]. The model is an adaptation of JUSSIM designed to fit the needs of the Philadelphia Planning Council. The Department of Corrections in Juneau, Alaska, paid for a portion of the development and had input on the form and extent of the output to be made available. The model, as always, can be thought of as just one part of the comprehensive criminal justice planning process. Here it is viewed as a management tool to aid in deciding where to direct LEAA and other available funds for improving the CJS.

### Policy Issues Addressed

The policy issues addressed are virtually identical to those of JUSSIM. A base

by Ernst and Ernst. Modifications updating data and the system output are made by running the model on the interactive computer system.

Use of the McCoy model has not spread beyond the initial installation. It has not been included in the body of this review because PHILJIM, a similar simulation model also based on a JUSSIM type aggregation and Markov flow, has had greater distribution and is discussed in some detail.



case model of the yearly flow of offenders and cases is constructed by the user. Any alternative plan which makes a structural change in the base case flow diagram, e.g., a diversion system for drug violations, or a change in the level of parameters such as workload, branching ratios, etc., can be tested by PHILJIM, which then calculates several performance measures relating to annual workload, by crime type, that have been judged relevant indicators of CJS performance. These include cost estimates and total annual resource units required.

The model helps predict outcomes under current programs and can test the design of new ones. In itself it gives no guidance on how to choose new programs nor does it have a single performance criterion upon which to judge a proposal. The summary list of JUSSIM capabilities applies to PHILJIM.

### Structure of the Model

PHILJIM is a linear model with a somewhat larger number of user options than JUSSIM. Because it predicts one year into the future at a time, it is not a steady-state model in the same sense as JUSSIM. This allows for a provision which considers that existing court backlogs and undercapacity will prevent some offenders from passing through the system during a year. JUSSIM estimates the resource requirements at each stage, while PHILJIM has an option that accumulates backlog cases when resources such as the courts cannot handle the input load.

Other options included within the model allow for considering both linear variable resource costs and fixed costs that might better indicate the impact of nonvariable components of the system's cost, such as payment for support staff and equipment. Flows may enter anywhere in PHILJIM; in fact, offenders may split into multiple entities (cases and people) for subsequent processing of each type through routes, accumulating different costs and using different resources.

### Data Base Required

There is no substantial difference between PHILJIM and JUSSIM in regard to data requirements, except that, as a batch rather than interactive computer programs, PHILJIM requires prior preparation of an input file describing proposed changes to the system.

### Output and Use

Output is voluminous and detailed but self-explanatory. It contains the workload estimates for all resources used at each of the modeled stages, available by crime type or aggregate groupings called "crime groups." In contrast to the steady-state calculations of JUSSIM, PHILJIM is more of a one-year flow prediction, often described as an *aggregate deterministic simulation*. Initial backlog can be supplied as input and the model will calculate final backlog if sufficient resources are not made available to service the demand at each stage.

A user of PHILJIM postulates some change in his existing CJS that might be implemented. This change, represented by a change in the data base, is the input to the model. Output consists of the workload and costs of the test case proposal at all of the processing stages by various groupings of crime types.

### Cost and Computer Requirements

The computer program is written is FORTRAN IV and is currently limited to running in a batch processing mode. The authors feel that with the options incorporated in PHILJIM the batch is preferable to the interactive mode. Under the default allocations for basic data storage, the program occupies 260K bytes of core storage. Modest jobs cost less than $5.00 per run on an IBM 370 machine. Full-scale typical Philadelphia runs with 28 stages and 29 crime types cost about $11.00 per run [158].

### Validation and Verification

Verification, again, was quite simply accomplished during testing and development of the program. Input, output, and data handling dominate the program, as the mathematics of a one-year simulation are simple addition or multiplication. Thus, verification simply checks to see that the proper categories were added or multiplied. Validation has not been done in a formal way. For discussion of the ways to validate, and the difficulties of doing so, as well as caveats to users of such models, see the earlier discussion of JUSSIM.

### Implementation and Impact

Distribution of PHILJIM has been wide. Implementation was first achieved in Philadelphia and then in Alaska. Other SPAs, Regional Planning Councils, or State Departments of Justice have picked it up and have experimented with it. Denver, Sacramento, and Washington, D.C., have the model.

In Philadelphia and Alaska PHILJIM is not in current use. The model was tested in both places, used during the course of the contracts as a report generator, but never used as a policy tool. Data collection problems and lack of institutional support have left these projects dormant. With proper funding, though, the model has a potentially valuable policy capability, just as JUSSIM has demonstrated in Allegheny County. The impact in criminal justice planning has been a one-shot increase in the awareness of the importance of system data and of attempts to overcome these collection and consistency problems. The need for development of a centralized organization for data and planning and the problems of ongoing support are now recognized.

### Limitations and Benefits

Limitations and benefits of PHILJIM from the technical side are the same as for JUSSIM except that PHILJIM was improved by the increase in the number of user options and the CJS cost analysis which is provided by the one-year run. From the user standpoint, the required batch mode operation eliminates the interactive capability and inhibits the gaming or learning experience feature of the predecessor model JUSSIM.

### Transferability

The transferability of PHILJIM is adequate, but not as good as JUSSIM's. The program deck can be purchased from Government Studies and Systems, Inc., Philadelphia.



### Documentation

Documentation is in the form of informal papers presented at various professional society meetings and is just barely adequate, particularly to those who do not already have an understanding of JUSSIM.

## JUSSIM II

### Historical Background

This second edition of JUSSIM is designed to include recidivism, the major source of feedback among CJS components, which was lacking in the original version. The historical development follows that of the original version until JUSSIM II was designed in 1973. Many of the ideas were examined earlier in an application to corrections in California in 1965 [172], where a mathematical method was developed to predict the flow of freed offenders upon their reintroduction into the general society. Recidivism has a time delay effect on the CJS so that the impact of any new program to change the CJS may not be felt for several years hence. The once-through calculations made in JUSSIM were sufficient to estimate its steady-state performance, but once the feedback of recidivists was included, a dynamic year-by-year model that accounted for these time lags became necessary.

The design of the model has been completed and was published in 1973 [12]; the computer program is available from the Urban Systems Institute, Carnegie-Mellon University.

### Policy Issues Addressed

The policy issues which can be addressed by the JUSSIM II model include all those discussed above for JUSSIM and its cousin PHILJIM. Parametric and structural changes in the flow diagram, constructed to represent a new bureaucracy or simply a new way of doing things, can be investigated. Performance measures of such a test case run can be compared with the base case, as well as with other previously tested alternative proposals, to help the user choose which should be implemented.

JUSSIM II has the added capability of being able to address policies that change recidivism, now an integral part of the model. Changes in the times between successive criminal acts can be investigated as well, since time-dependent (yearly) outputs of resource workload, costs, and flows are available from the model.

### Structure of the Model

Every year a number of crimes are committed which are differentiated by type within the model. The number of such crimes is determined by historical statistics; crimes are differentiated between those committed by first offenders ("virgins") and by recidivists. These units of flow advance through the CJS as in the original JUSSIM model.

All processing is assumed completed within a year's time, and the resource capacity at each stage is assumed sufficient to handle the load. The output from each of the stages, including the corrections facilities, is divided into the fraction who eventually become recidivists and those who return to the general society and behave as normal citizens. Recidivists may switch crimes (as determined by a matrix which represents a Markov transition model of crime switching) and are reintroduced into the CJS crime-committing stage at later times, representing the various time intervals between the commission of crimes.

### Data Base Required

The program itself contains no data on a modeler's CJS but rather operates on a data file that is constructed by the user, depicting a flow diagram of his own CJS. The input is based on this diagram, similar to that constructed for a JUSSIM implementation, and consists of stages representing processing of crimes or offenders within the CJS, of flow paths between the stages, of release or dropout points from which recidivists are taken, of an input process generating crimes by first offenders, and of a crime switch mechanism to generate crimes by recidivists.

User input to make the data file for the base case includes all of the information on the stages and links between them. Specific branching ratios and resource consumption per unit flow and cost, all by crime type, are the parameters needed. Necessary input for the recidivist component of the JUSSIM II model includes the crime switch matrix describing the proportional flow of recidivists (coming from a pool of arrestees, defendants, probationers, or released convicts) among various crime types, and the associated delay factors which indicate time between criminal acts.

### Output and Use

A JUSSIM II user sits at a remote terminal and interacts with the model. Modifications to the base case data file are made through a series of questions and answers. Once the test case data file is judged to represent the modification to the CJS being considered, the user runs the model, and output is produced and is used to compare performance measures of resource units, workloads, and cost for each resource type by crime type.

The JUSSIM II model extends JUSSIM in the sense that output is available for each year in the future, although the extrapolation may become unreliable after five or six years. The multiyear runs available are particularly useful in evaluating the dynamic effect resulting from time-lagged changes in recidivism, because the numbers of virgin arrests are supplemented with recidivists resulting from each earlier year's policies. The resultant output is identical in range and format (base case and test case resources, workloads, and costs) for any future years the user desires.

### Cost and Computer Requirements

The model was written in the FORTRAN IV language and operates in interactive mode on computers with a facility in FORTRAN IV and time-sharing capability. Storage capacity requirements are about 32K words of core. Some minor features that facilitate the running of JUSSIM II are not included in FORTRAN IV, strictly speaking, but there are ways to circumvent these limitations. Computer time and cost are minimal, as the mathematical calculations are elementary. Input-output



and data manipulations through the time-sharing mode contribute the major fraction of the overall cost of operation.

### Validation and Verification

Internal consistency is verified by checking that no units, offenders, cases, crimes, etc., are lost in the system. The mathematics of the model are very simple, and so verification presents no significant difficulty. However, no evidence is available concerning the validity of JUSSIM II. If other models of similar type were validated as providing reasonably accurate predictions, then one could assume that JUSSIM II might be valid as a predictive model. The family of overall CJS models does not contain any validated models. This omission should not be a significant factor for a potential user. The accuracy of all models is site- and user-dependent. Knowing a model was validated in Pittsburgh is not sufficient to conclude that it would be valid in Dallas. Certainly the chances of valid use in Dallas would be higher if it were validated in Pittsburgh, and if a model is proven invalid in Pittsburgh, the chances are lower that it would be valid in Dallas. All models should be used with some caution until such time as they have been validated by the potential user.

### Implementation and Impact

There has not been any implementation of JUSSIM II to date. Currently tests are being conducted in the city of Pittsburgh, but the impact and success of the model are still unknown. Because of its similarity to its predecessor, JUSSIM, one can expect that some experience will be gained in this regard shortly.

### Limitations and Benefits

JUSSIM II is limited in the same way as the original version, with two exceptions. Because of its dynamic structure, the new model provides "snapshots" of the system performance for several years into the future. Thus, questions such as the effects of increases in crime rates and of gradual implementation of selected CJS proposals can be answered. Second, of course, JUSSIM II does model recidivism effects. Release of offenders from the court subsystem, and early parole of those sentenced, return potential recidivists to society. Proposed programs that might change these flow rates and the performance of such changes over time can now be evaluated.

The model provides estimates of the consumption of resources in total units, yearly workload, and cost. The component of this workload processed by each of the various stages and crimes is also available. Issues addressable are the same as in the earlier version, but additional proposals affecting the time-dependent and recidivist factors of a CJS can be tested by the new model. JUSSIM could only predict downstream effects of changes. The new version will be able to estimate the eventual upstream impact on flow, cost, and resources of the various test cases.

The side benefits which are derived indirectly but are stimulated as a result of implementation of the original JUSSIM, including the organizational changes necessary to promote data collection and decisionmaking, are projected to apply to the new version as well. The interactive computer mode will provide the learning effect typical of management games. The user will begin to appreciate the broader implications of changing the existing system and so become a better user of JUSSIM II in the sense that he will increasingly include estimates of the dependent changes in branching ratios and other CJS parameters. Hopefully, this will ultimately be reflected in an improvement in his real-world CJ planning.

The chief potential limitation of JUSSIM II, as with the earlier version, is its simple, deterministic, and aggregated model of the CJS. Whether such a model will prove to be a valid predictor and thus a valuable tool on the basis of its predictive mode alone is still unclear.

### Transferability

Though no evidence exists, the JUSSIM II model should be readily transferable. The JUSSIM model has proved to be easily implemented, as indicated by a wide range of applications, and this new version ought to be just as mobile.

### Documentation

Adequate documentation of the JUSSIM II model exists [12]. Background exploration by the user prior to implementation should include discussions of the recidivism feedback process [18,19] and documentation of the implementation of the older version of JUSSIM [42] and similar models [21,28,29,92,158].

## DOTSIM

### Historical Background

A resource allocation and modeling effort, part of the Model Criminal Justice System Development Project, was undertaken in 1972 by Public Safety Systems of Santa Barbara, California, under a grant from LEAA [94,95,96]. The project attempted to improve the data collection and subsequent evaluation and planning capabilities of the CJS in Ventura County, California, in order to improve the allocation of CJ resources. As part of this effort, a mathematical modeling feasibility analysis was conducted, during which an analysis of CJ planners' needs and means for resource allocation was made. No existing model was found during the literature review that satisfied the set of requirements for the estimation of costs and effectiveness of alternative policies that was thought necessary. The existing models were each rejected for a variety of different reasons. JUSSIM, for example, is unable to model the effect of queuing or to track individual offenders as entities. A prototype Dynamic Offender Tracking Simulation (DOTSIM) model was developed in an attempt to accomplish the goals of the project.

Testing of the completed model was conducted on the Ventura County CJS using sample data. DOTSIM is still viewed as a prototype model developed not for on-line implementation but for experimental use to demonstrate the usefulness of such models. Upon implementation it would provide a means for discovering and testing alternative planning policies.



### Policy Issues Addressed

Planning policies that are addressable with DOTSIM include those of the JUSSIM model and its descendants as well as questions relating to queuing delays and the random nature of the processing of offenders. Thus, in addition to policy proposals affecting the processing of offenders, arrests, and cases, changes in the amounts of various resources are added. The impact that such changes might have on time delays and average system flow times is an added output measure of system performance.

DOTSIM cannot help in such policy questions as allocating police activity or choosing correctional disposition to reduce recidivism, nor can it provide guidance on the interdependent nature of any proposed change in the CJS.

### Structure of the Model

DOTSIM simulates the movement of offenders through the user's CJS. Each case or entity within the model is created with characteristics chosen from random number generators based on statistics on the frequency and type found in the real world. The descriptive flow model constructed from the graphs or diagrams, similar to those used in the JUSSIM type of models, can be thought of as a network through which the DOTSIM model simulates the processing of arrests and offenders, keeping track of their movement and accumulating statistics of all these entities as time is advanced. The randomness of real-world phenomena could then, in principle, be reproduced within the model.

The authors of DOTSIM stated their objectives for the model as follows [96]. It should:

- Reflect the actual procedural step-by-step processing of offenders through a CJS.
- Represent the correct utilization of the CJ resources at each procedural step.
- Determine the time required for each step.
- Determine queuing delays that result from unavailability of resources.
- Account for information transfer delays.
- Assign priorities to the processing of any crime type.
- Use historical or desired policies.
- Assign fully burdened direct and indirect costs based on utilization at each step.
- Handle recidivism and any type of offender feedback.
- Differentiate recidivists and virgin arrests. [96]

Cases interact in the formation of queues when the workload demand exceeds the ability of the resource to process cases. The model accumulates measures of both costs and times attributable to each case as it passes through the processing stages. Costs and average transit times for the offenders through each of the stages are available by crime type.

Use of the model enables planners to predict resource (personnel, judges, equipment, facilities, etc.) workload and cost, as well as the extent of delays occurring in the operation of the CJS.

### Data Base Required

To use DOTSIM, a system flow chart representing graphically the sequencing and interaction of offender flows and a historical data base on CJS operations must be constructed, and key parameters from it be provided as input for the program. In addition to these inputs, which are similar in many respects to the JUSSIM family of models, the simulation model requires the distribution of the lengths of time spent on each processing step in the form of maximum, minimum, and most likely times. The characteristics needed for the model to generate crimes—the rate or frequency by type—are also necessary. Probabilities of branching upon leaving each of the stages, the available total quantities of each resource, and the resource requirements at each stage by crime type are all needed parameters.

### Output

Cost breakdown by CJ agency area (prevention, apprehension, adjudication, and corrections) and crime type (felony, misdemeanor, juvenile, crimes against property, against persons, no victims, and miscellaneous) is available as output from any run of DOTSIM. Another form of output gives the numbers of processed offenders, queuing delays, costs, processing times, all averaged for those processed through each stage. A third output summarizes resource utilization and cost, and for a subset of the stages a breakdown of statistics by crime type is available.

A summary of output judged desirable and made available by the authors is:

A. *For each procedural step*
- Number of offenders processed
- Average processing time (distribution optional)
- Queuing statistics
- Processing costs (distribution optional)
- Queuing costs

B. *For each resource*
- Workload distribution
- Utilization/availability
- Average cost and time per case (distribution optional)

C. *For each crime type*
- Resource workload distribution
- Averages and totals for time and cost by resource type (distribution optional)

The computer run represents a one-year-long (or longer at user option) collection of statistics on processing of offenders. As such, the measures of output are viewed as random variables, since many of the processing times were random. Multiple repetitions of the run are needed to see how variable the performance measures aggregated for one year really are.

### Cost and Computer Requirements

The program is operational by batch processing on a CDC 6400 computer and requires 20K words of storage, for 60 crime types and 25 resource types. It is written



in FORTRAN but contains special techniques only possible on the CDC machine. Modifications would be required for use on the IBM 360 series, and the storage requirements would be substantially larger on an IBM machine. The authors report that one year of simulation time with about 20,000 arrests took approximately 7 CPU minutes of computer time. This program is therefore more expensive to operate than models like JUSSIM.

### Validation and Verification

Limited validation and verification are reported. The model was never fully implemented, but developmental computer runs were made on sample data. Aggregate measures of costs as determined by the simulation run were stated to be adequate estimates of the true costs, but this does not constitute genuine validation. Validation of simulation models such as DOTSIM may be even more difficult than validation of the models discussed earlier.

This model is of the same type as simulation models of industrial jobshops, which have found wide use in the fields of industrial engineering and operations research. Techniques for validation and verification of jobshop models are well established. However, each model must be tested for accuracy on an individual basis, and DOTSIM never progressed to a point where the developers could conduct the necessary tests.

### Implementation and Impact

DOTSIM has not been implemented, although it was tested and demonstrated to Ventura County officials. The design concept has been noticed by several other State Planning Agencies, but after consideration it was not adopted. At this time no successful implementation or testing has been made outside of the design work completed in California.

### Limitations and Benefits

With no significant operational experience available, it is difficult to judge the limitations and advantages of the DOTSIM model. One might conjecture, however, based on general evidence of the difficulty of using simulation models in jobshops, computer queuing, biological growth processes, and police allocation areas in particular, that the DOTSIM model will not find a broad use on a regular operational basis. However, if the same analogy holds, a model like DOTSIM might find use as a research tool for addressing large-scale, long-range policy questions on criminal justice systems.

One reason why stochastic entity simulation models have not been adopted widely is that, in general, they require significantly more information on the precise rules for the movement of each case or entity through the CJS. Data to support a statistical project to obtain estimates of these is difficult and expensive to obtain. Even aggregate flow information is gathered with considerable difficulty, as evidenced by the data problems experienced with the JUSSIM type models, and recidivism or other forms of feedback are very hard to quantify at a level of refinement necessary even for such relatively simple models as JUSSIM II. From these experiences, one can project that DOTSIM would prove difficult to implement because of much greater data problems.

Simulation models consume ten times as much computer resources as simple aggregate models. Usually they require an experienced simulation analyst or operations research analyst to run and interpret results. A statistical analysis must be made to establish the required length of the computer run and the true significance of output measures before any decisions can be made. The requirement for computer support and technical analysis for simulation models has limited their use in regular operations in other fields. In criminal justice these requirements will prove even more difficult to satisfy.

One limitation of the DOTSIM model, which is characteristic of all the models reviewed, is that parameters describing interaction between system components must be prescribed by the user. The authors of DOTSIM, recognizing this fact, describe a proposed later modification which would allow behavior probabilities (the branching ratios) to vary as downstream queues and associated delays develop. Even so, someone, in this case the DOTSIM programmer, would have to establish some estimate for the interaction among system elements, and there is no indication that available knowledge in the near future would permit this to be done accurately.

The DOTSIM model in its present form has the major advantage over the other models reviewed that it can be used to address questions relating to the queuing delays so common in the court subsystem. Revisions of scheduling and operating procedures, e.g., priority queuing systems, computer scheduling, etc., can each be evaluated. Aside from these features, a simulation model like DOTSIM is simply an expensive way to calculate output provided by JUSSIM type models.

Time-dependent results can be predicted using the DOTSIM model, although a repetition of runs necessary to achieve the desirable accuracy of such predictions may be costly. This use of DOTSIM has not been made, but where other types of simulation models have been used, this benefit has been successfully exploited.

### Transferability

DOTSIM is written in FORTRAN for the CDC 6400 and uses features, particularly the 60-bit word size, peculiar to this type of machine. Thus it is transferable to CDC machines with this feature, but its implementation generally would require some reprogramming, and its transfer to another type of machine would require the services of a professional computer programmer. Additionally, as with all overall CJS models, implementation would require the assembly of large quantities of data relative to the CJS under study.

### Documentation

Reports which provide an overview of the model [94,95,96] are available from Public Safety Systems, Inc., in Santa Barbara.

## CRIMINAL JUSTICE SYSTEM TRAINING MODEL

### Historical Background

The CJS Training Model was developed under a Georgia State Crime Commission grant of LEAA funds, which established feasibility and the design of such a model [61,62]. The initial work took place in the summer of 1972. A one-year direct grant from LEAA followed, and during 1973-1974 the model war programmed, tested, and documented. It is included here in brief because of its unique objective of training CJ planners and because it does contain an internal model for projecting flow of cases through a typical CJS.



### Policy Issues Addressed

Planning policy questions can be addressed only indirectly with the Training Model. The designers of the model intended that it be a training aid similar to management games used in business schools or to war games used in the military. The theory behind the use of models of this type is that a user's understanding and decisionmaking skills in the criminal justice area will be improved by using the training model. The model and the general effect of such a policy will be remembered. Thus the CJS Training model is used quite indirectly in actual criminal justice planning. In fact, the developers of the model warn [61, p.10] that "since the model is for training rather than decision making, the scope of the project is somewhat limited and the predicted time histories generated by the model should not be used for planning or policy making purposes."

Policy is input by the model's user in the form of system resource and utilization changes. Typical examples of such planning policies are the hiring rate (number per year) and firing rate of policemen, the expansion rate of the court system or corrections facilities, total budgetary limits placed on elements of the CJS, and budget allocation made for public education on concerns of the CJS. System parameters which are not normally thought of as policy can be changed to see what effect these system changes will have on the subsequent behavior of the CJS. Examples typical for this type of sensitivity testing are the rate of natural attrition of policemen, persons arrested per officer, recidivist rates, or various system delay times. They can all be changed at user option.

### Structure of the Model

The model is a *deterministic simulation model* of the flow of cases through the Atlanta CJS during the decade of the 1960s. Models of this type were developed by J. W. Forrester as part of a subject known as "industrial dynamics." A computer simulation language called DYNAMO, which was developed specifically for such models, could have been used for the CJS Training Model. Similarly, a dynamic numerical solution to the interrelated differential equations that form the basis of such models could have been obtained by using an IBM simulation language called CSMP (Continuous Systems Modeling Program). Both were rejected in favor of the interactive mode of operation that was deemed necessary in the training environment. The style of DYNAMO was maintained, but the program was written directly in an interactive version of FORTRAN IV.

Given the rates of change of system components such as resources and cases/offenders, the deterministic simulation advances time incrementally, adding to or subtracting from the various state variables according to the rates of change multiplied by the time jump. This approach is similar to the basic one-year-at-a-time simulation which is the basis for JUSSIM II and PHILJIM.

Six major sectors (criminal flow, police, courts, corrections, financial, and community) are modeled within the system. Key variables represent the level of criminal activity, cases, resource size, etc., within each sector, and may interact with similar variables in other sectors. Through time the people within a criminal justice jurisdiction move among the six states (noncriminal, free active criminal, person with an unresolved arrest case, in prison, on probation, or on parole). Criminals are not disaggregated by crime type. The rates of flow are influenced by the level of key variables describing the status, in terms of operating capability and costs, of the six major sectors of the CJS. The flow of criminals in the CJS in turn affects the response of the CJS as modeled by the key variables. The system is described by a large set of interrelated differential equations. The policy controls that manipulate the model are the various rate parameters or cost constraints that are typically used to regulate the CJS.

### Data Base Required

The data requirements of this deterministic simulation model are substantial. However, in its role as a training model, the importance placed on accuracy is not as severe as in predictive models. Most of the data needed for models such as DOTSIM are required (except for the random variable parameters), but the aggregation of the modeled activities in the CJS Training Model does mean that the sheer quantity and difficulty of obtaining the information is reduced. The CJS Training model is more of a concept than an off-the-shelf model, and as such it requires data in proportion to the detail desired by the user. It is hard to generalize about the type and difficulty of collecting the data for such a model other than to say that, as with all simulations, it presents one of the more significant roadblocks impeding implementation.

### Output

Output consists of yearly estimates projected over the length of the active simulation run for CJS budget, crime rate in the region, reported crimes per police officer, percent of the prison capacity in use, and open cases per prosecutor. Changes in any one or more parts of the CJS will produce changes in other parts of the system. By observing the simulated output of the model, the user will gain increased understanding of the likely impact in a real CJS. A full range of output information on each of the system state variables and parameter values throughout the run of the simulation could be made available to the user if programming changes were made.

### Cost and Computer Requirements

A computer with interactive FORTRAN capability is necessary in order to realize the benefits of the training aspects. Current implementation is on a Univac 1108, and the program would require modification before placing it on another type of computer.

### Validation and Verification

The time histories of the operating CJS in Atlanta were compared with the



modeled output, and the approximation "is rather close" [61,p.20]. However, the authors caution the user against using the Training Model as a predictive model.

As a training model, validation of the predictive aspect is not crucially important. However, it is important that policy inputs by the trainee vary the output of the model in a realistic way. Here relative changes are more important than the absolute value of the various system variables. For example, when the number of policemen is increased, it is important to see more arrests; when the number of prosecutors is decreased, the backlog of cases should rise. No formal validation has been made, but as with all CJS models so far reviewed, aggregate performance measures have compared well with real-world data.

The model has not been validated as a training model. Here it would be necessary to demonstrate that relative changes were correct and/or that the users actually gained beneficial experience which then improved real-life performance in some way.

Verification is again almost an insignificant requirement. Although it is not stated as having been verified, the modeling component of the CJS Training model is not complicated and, as with most models, one has to trust that the developers have executed the design correctly.

### Implementation

The model was developed and test implementation made at the Georgia Institute of Technology. Currently, only the members of the development research team are prepared to conduct training exercises using the model. Use would require a programmer or operations research specialist familiar with Forrester-type DYNAMO simulations. Minor alterations representing parametric changes can be made by the user, but major changes to flow structure may require starting from scratch with the design of the program as a skeleton.

### Limitations and Benefits

The current status of the model does not allow enough evidence to be gained as to its limitations or long-term benefits to the nation's CJ planners. The usefulness of such a training model to improve the capability of CJ planners can be conjectured but has not yet been scientifically validated. If one can extrapolate from other training games and models, such a training model in criminal justice should prove extremely valuable in this regard. The predictive capability of overall CJS models is just one of the important benefits of the modeling efforts. Significant data collection and validation problems exist in implementation of all such models; the CJS Training Model, with its primary objective of training, bypasses these significant problems to some extent.

Models of this type are further limited by the deterministic nature of the flow equations. Random elements, such as queuing which is a serious aspect of the performance of criminal justice systems, are not included. In the CJS Training Model crime is aggregated to one type. Differentiation into a number of crime types would provide the user with additional realism.

### Transferability

The model now consists of data and system elements taken from Atlanta in 1960-1970. Structural changes will be necessary before transferring the model to another city. Training exercises can take place in any city once the Atlanta CJS is understood by the trainee, but it would seem desirable that the training be conducted on a familiar CJS. Transferring the model would require a modeler and programmer experienced in simulation of continuous systems, who could use Forrester's concepts.

### Documentation

Limited documentation exists in the form of a final project report on the LEAA grant [61,62]. The second part consists of a trainee's user guide. Whether these reports are sufficient to provide a basis for understanding and provide for transferability of the design or even the concept has not yet been established.

## CONCLUSIONS

In the selection of overall CJS models, we have discussed a wide variety, as indicated by the stochastic entity simulation approach of DOTSIM, the deterministic steady-state flow of the JUSSIM family, the deterministic simulation of JUSSIM II, and the industrial dynamics type CJS Training Model. Modelers have indicated different objectives in developing models. Some hope to reproduce the real operating system, as in DOTSIM; another has training as its central purpose; others wish to establish a system for decisionmaking and data collection with the installation of a simple planning model such as JUSSIM.

Because of the diversity of purposes and model structure, it is hard to lump them all together for criticism and praise. Each individually has its failings and strong points, as has been indicated in the preceding discussion, model by model.

Overall CJS models have already had some impact on the synthesis and analysis of planning policy in criminal justice systems. The organizations that have implemented these models have become focal points for data collection and training of CJ analysts. These benefits have unfortunately been transient because the existence of centralized decisionmaking organizations and the use of these models have not generally endured. (See Chapter 7.)

In any event, the concept of comprehensive CJS planning at many levels has been tested in practice, and many of the early deficiencies will be corrected in the future. One of the major benefits has been the indirect training of CJ planners that takes place while using models such as JUSSIM, CANJUS, PHILJIM, or DOTSIM; and in the Georgia Tech model, training is the objective, as we have seen. Here the learning is in two major areas. The user manipulates the model with his proposed changes in policy to see how the primary impact on system characteristics will change. This predictive aspect of models is extremely useful for choosing among policies; the user begins to get a feel for what are good policies, why they are good, what segments of the CJS are affected by given policies, and which choices are most efficient in reaching complex goals.



Secondly, the user of these models is constantly asked to think through the implications of his policy. These form the basis of the changes made to the input of the model. Interactions and dependencies that arise from behavioral factors and from changes in the composition of an aggregated variable must be prescribed by the user in all of the CJS models. This is a required part of all modeling. In the CJ area the model builder is not usually as intimately familiar with these relationships as is the user, but it often takes the stimulus of working with the model for the user to develop and express his knowledge quantitatively.

In summary, then, the experience gained from overall CJS models has been beneficial primarily from the learning that has taken place on the part of both users and model builders. Implementation has been successful on a continuing basis only in isolated cases. It seems clear that these models, in spite of their limitations and in spite of implementation problems such as data collection and organizational politics, have been partially sucessful. During the next few years, with additional implementation attempts and operating experience gained, and with prospects for model improvement, we should begin to see more concrete evidence of these various benefits in the short as well as the longer term.

# Chapter 4

# POLICE MODELS

## INTRODUCTION

Nearly all models designed for police applications have been directed at the patrol subsystem. The first group to be discussed consists of patrol allocation models, which are used to specify the number of patrol cars to have on duty in various geographical commands at different times of day or days of the week. Our initial review of such models for the purposes of this study revealed that none of them was entirely satisfactory in terms of capabilities or availability. Rather than simply discussing these problems, we undertook to design a new model, incorporating many of the best features of the ones reviewed. For this reason, the description of patrol car allocation models differs somewhat in format from the other model reviews in this report.

The next group, simulation models, is the most numerous after allocation models. We describe four such simulation models, all of which deal with the police patrol force system, including in certain instances the communications system for handling calls for service.

The next group, beat design models, differ from allocation models in that they provide information about individual patrol unit activity as related to the geography of the patrol area. They are analytic models, and in this sense are more akin to the allocation models than to the simulations.

Then we describe the Dynamic Queuing Model, also an analytic model. It differs from the allocation models and the beat design models because the latter are "steady-state" models, i.e., they compute system characteristics assuming that system parameters are constant over time and that the system has been operating for a very long time. The Dynamic Queuing Model, by contrast, analyzes the characteristics of a system that has parameters changing over time. For example, in police patrol systems both the rate of calls for service and the number of patrol units are likely to change from hour to hour. The Dynamic Queuing Model can compute the changing characteristics of the system at all points in time.

Next, we discuss a linear programming model that determines for a patrol system the best times to start tours of duty, the number of patrol units to be fielded on each tour, and the best times for patrol units to take meal breaks. It is an optimization model that minimizes the total number of patrol tours required in a day (or week), subject to the constraint that the number of available patrol units meets or exceeds a user-specified car requirement for each hour of the planning period.

The final group of models, manpower scheduling models, is the only group that does not deal exclusively with the police patrol system. The purpose of such models is to determine work charts or schedules of working and off-duty days for individual policemen. Two models are discussed. The first is for scheduling only patrol personnel and is rather restrictive in the possible vacation day patterns that are allowed. It is an optimization model designed to minimize average response time over all





tours occurring in a week. The second manpower scheduling model is more general. It can be used to schedule any type of personnel. Rather than supplying a single schedule that is "optimal," it calculates a variety of schedules, each matching the user-supplied tour-by-tour workload requirements. Schedules are ranked according to their vacation day patterns, but these rankings are only a guide for the decision-maker who makes the final schedule selection. The use of these models, especially the latter, may take on added importance as police departments become unionized and demands are made for better working conditions. There are examples where use of the second model can improve the vacation pattern for police personnel without adversely affecting the manpower coverage in the various tours of the week.

## PATROL CAR ALLOCATION MODELS [1]

During the last decade, a considerable amount of effort has been devoted by operations research analysts to methods for allocating police patrol cars. Out of this work have evolved several computer programs for specifying the number of patrol cars that should be on duty in each geographical command of a city at various times of day on each day of the week. Most of these are based on, or are similar to, either the resource allocation system of the St. Louis Police Department [50,136,137,167] or a program designed by Richard Larson [119]. The most widely known program based on the St. Louis system is the Law Enforcement Manpower Resource Allocation System (LEMRAS), a proprietary IBM package [93]. Among the programs that are based on Larson's, we found the following ones at the start of this study:

- The Police Resource Allocation Program (RAP), a proprietary program of Urban Sciences, Inc. [182].
- The New York City Police Department's patrol car allocation program, written by Richard Mudge at The New York City-Rand Institute [144].
- A program designed for the Los Angeles Police Department by a UCLA class, "Public Systems Analysis" [2].

All of these programs are analytic models, and they have many elements in common. However, each of them has one or more minor features that are considered either especially desirable or particularly inadequate by some analysts or police departments. These features may relate to the program's mode of operation (batch or interactive), input requirements, assumptions underlying its calculations, or capabilities to take certain performance measures into account. As a result of these distinctions among the programs, none of them has achieved general acceptance.

In this section we shall describe briefly the patrol car allocation models that existed at the start of this study, and then give details for a model called PCAM (Patrol Car Allocation Model) that was designed by Chaiken and Dormont [32,33,34] after the review of earlier models had been completed.

### General Principles

In describing the programs, we shall use the term "precinct" to refer to an

[1] This section also appears, slightly modified, as an appendix in Ref. 33.

independent geographical command that is commonly called a precinct, division, district, or area (and sometimes, but rarely, a 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 (a) that its commander has the capability or authority to decide how many patrol cars will be fielded at various times, and (b) that the dispatchers of patrol cars treat the precinct as an independent command by sending only precinct cars to incidents in the precinct, except under unusual circumstances.

Some police departments are small enough that they do not have separate geographical commands. For them, a patrol allocation program 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.

All patrol car allocation models operate 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, the programs assume 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, most of the programs assume that the order in which the waiting calls will be assigned to cars depends on their relative "importance."

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 plain-clothes unit will be sent to the scene, or some other way will be found to respond to the call.

If these variations from the assumptions in the programs occur infrequently, then they may be ignored for all practical purposes. However, if the variations are extreme, for example if a department *regularly* dispatches two cars to every incident, then either the input to the program must be adjusted to account for departmental practices or the output must be interpreted differently. In the example of two-car dispatch, if both cars remain at each incident for the same length of time, the department could 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.

The most important input data for all the programs is an estimate of the *call-for-service (cfs) workload* that will occur in each precinct; this may be broken down by hour or by "tour" (also called "shift" or "watch"). The cfs workload is the total number of car-hours that will be spent handling calls for service. One way to estimate the workload is to predict how many calls will occur (the call rate) and the average length of time it will take to handle each one (service time); multiplying call rate by average service time gives cfs workload. However, it is not necessary to



separate workload into these two components; it can be estimated directly from data showing total car-hours spent on calls for service in the past.

There are two arguments in favor of considering the call rate and the service time separately. First, if the program calculates the *length of time* callers will have to wait in queue, as opposed to calculating only the *number of calls* that will be delayed, it requires as input both the average service time and the workload. Second, it may be assumed that differences among tours or precincts in average service time are due to differences in the mix of calls they receive, e.g., that some precincts just happen to get a larger proportion of calls taking more than an hour to handle than other precincts do. Under this assumption, average service times for various types of calls can be estimated from citywide data rather than from precinct data; presumably this will yield more accurate estimates.

The computer programs for patrol car allocation can be distinguished according to whether they do or do not assist the user in predicting cfs workload. Those that make no predictions have sometimes been operated simply by using averages of past data, and sometimes a separate program has been used to make the required predictions. The details of prediction capabilities will be described below in the discussion of individual programs.

Another important distinction is how the programs handle unavailabilities of patrol cars that occur for reasons other than calls for service (meals, auto repairs, on-view incidents requiring police intervention, special assignments by a commanding officer, and the like). One approach that has been taken (but is not recommended) is to ignore these unavailabilities altogether. If this is done, the output of the program will bear no relationship to reality and is therefore virtually useless as an aid to planning.

A second approach is to consider these "non-cfs" unavailabilities as if they were calls for service and include the time spent on them in cfs workload. If the estimates of non-cfs workload are accurate, this method will work well. However, it is not appropriate to make such estimates by projecting data from the past, because non-cfs workload will change if the number of cars on duty is changed. Particularly in departments where patrol cars are unavailable for dispatch during mealtimes, it is apparent that increasing the number of cars on duty will increase the non-cfs workload, quite independent of how much non-cfs workload there was in the past. Whether this effect is large enough to be of importance will vary from department to department.

A third approach to handling non-cfs workload in a patrol car allocation program is to assume that cars busy on non-cfs work are not "effectively" present. As an example, if six cars are fielded, and each one spends one-third of its time on non-cfs work, then the "effective" number of cars is four. Using this method, the computer program first calculates how many "effective" cars are needed in each tour in each precinct, and then it applies a correction factor to determine how many "actual" cars must be fielded in order to achieve the desired number of "effective" cars. The advantage of this method is that the calculations can be performed so as to take into account automatically the change in non-cfs work that will occur as the number of fielded cars is changed. How this is done by the various programs will be described below.

## St. Louis Police Department

The computer programs for the St. Louis Police Department were initially proposed and documented by Richard F. Crowther [50] in 1964. (See also Shumate and Crowther [167].) During the four years that followed, these methods were refined, programmed for the department's computer, and applied in one precinct (called "district" in St. Louis) by a project team at the police department [136,137]. While the total resource allocation project covered many topics, we shall describe only those that are related to determining the number of patrol cars needed in each precinct. The programs for this purpose have two components, prediction and queuing, which we shall discuss separately.

**Prediction.** 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.

**Queuing.** By making certain technical assumptions,[2] it is possible to calculate from the predicted workload the percentage of callers who will experience a queuing delay, given how many cars are in the field. A program was written to generate tables showing the percentage of calls in each tour that would experience a delay supposing, for example, that three cars were fielded, four cars were fielded, and so on. Department policy was established that at least enough cars should be fielded to keep the number of calls placed in queue under 15 percent of the total number of calls. By consulting the tables it is possible to see how many cars are needed to accomplish this objective.

These programs perform certain basic functions needed for any patrol car allocation system. They were operated by the department in batch mode on a regular basis for at least five years, although it is our understanding that they are no longer used.

For purposes of comparison with programs to be described below, 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 can be used with any of the patrol car allocation programs.

[2] These are: (a) incidents occur according to a Poisson process, (b) all incidents have the same exponentially distributed service-time distribution, and (c) the system is in steady state.
All the programs discussed in this section make the same assumptions. However, it should be noted that the method used to estimate workloads is based on the observation that different types of incidents have different service times, while the queuing model assumes that all incidents are similar in this regard. This internal inconsistency is justified by the accuracy of the resulting calculations.



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 IBM software package (now withdrawn) was based on the St. Louis system and includes all of its features, together with a number of improvements [93]. 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. 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 improvements in the LEMRAS system were not conceptual in nature but were for the purposes of assisting the user in preparing data for input, providing flexible output formats, etc. Like its St. Louis predecessor, LEMRAS is a batch program. 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 on-line interactive program, while LEMRAS operated in batch mode.

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

Some LEMRAS users entered all patrol car work, whether for calls for service or not, into the data input and were satisfied with both the predictions and the recommendations for the number of cars to be fielded. Other departments, such as the Los Angeles Police Department (LAPD) [2], 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 earlier, it is conceptually erroneous to try to make predictions from past data. Departments that found their non-cfs predictions satisfactory presumably did not vary the number of cars on duty in a given precinct and tour to any great extent, or for some other reason they were lucky to have a slowly varying pattern of non-cfs work. The LAPD happened not to fall into this group.

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

### Larson's Program

In 1968 and 1969, Richard Larson designed a program for patrol car allocation and applied it, as a test case, to data from New York City [118]. Later, he described the program, together with potential improvements that could be made, in his book, 



*Urban Police Patrol Analysis*[119]. 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 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:

- Average travel time to incidents
- Average patrol frequency (how often a car passes a random point in the precinct while on preventive patrol)
- Patrol hours per outside crime.

In one method of using the program, called the descriptive mode, the user could try various numbers of patrol cars in each precinct, and the program would calculate these three performance measures, together with the percentage of calls that would have to wait in queue. If the user had in mind a desired maximum or minimum for some of the measures (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 above-mentioned measures in each precinct. In addition, he could establish administratively a minimum permissible number of patrol cars for some or all precincts. The program would then calculate how many patrol cars were needed in each precinct to meet all the specified constraints, without the user having to inspect a large number of descriptive tables.

**Allocation of a Fixed Number of Cars.** Larson 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 must 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.[3] 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.

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).

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 NYCPD never used this particular version for any planning purposes. The MAD language was unpopular and was eventually abandoned by MIT, at which time the program "died." The Rotterdam Police Department [138] wrote their own version of this program and implemented it.

### Urban Sciences, Incorporated

Urban Sciences, Inc., rewrote Larson's program in FORTRAN and greatly enhanced its interactive capabilities [182]. Under the name of RAP, this program was made available to police departments by contract, and Urban Sciences also assisted departments in preparing the required data base. In all conceptual aspects it is identical to the program just described above. We understand this program will be withdrawn and replaced by PCAM.

### New York City Police Department (Mudge's Program)

This program was written in 1972 by Richard Mudge at the New York City-Rand Institute [144]. While based on Larson's program, Mudge's program is not exactly the same. The two primary differences are:

[3] 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.



- 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.
- Mudge's program distinguishes between "effective" cars and "actual" cars.

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:

* Proportion of time spent serving calls for service
* 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

It will be noted that this program also permits only three priority levels and that the wait for priority 1 calls is not displayed. It was thought that priority 1 calls would be handled in a special way if all the precinct cars were busy, and thus the program's estimates for the delay of such calls would be inaccurate.

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. Substracting 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 was available in two versions, batch and interactive. The NYCPD used this program from time to time over a two-year period for long-term planning purposes. It has been replaced by PCAM, which will be described below.

### UCLA Program

As mentioned above, for several years the LAPD used the LEMRAS program, as modified by its own input and output routines, and was having some difficulty with it. In 1974, a class in public systems analysis at the University of California, Los Angeles, prepared a patrol car allocation program for consideration by the LAPD.[2] It was based on the Mudge and Larson programs. In common with the Mudge 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 operates on the assumption that the amount of non-cfs work performed by a car will vary according to the amount of cfs work.[4] The relationship between non-cfs work and cfs work is modeled as a linear equation, separately for each precinct, using data from the precinct [2]. The conversion between "effective" cars and "actual" cars is then calculated from the linear equation.

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 will not use this particular program for operational purposes. Instead, it will consider adopting PCAM.

### Interim Version of PCAM

During the process of programming PCAM, which will be described next, an interim version of the program was provided to the New York City Police Department and the Seattle Police Department [145]. 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 is available only as an interactive program. This model was used for over a year in Seattle, where it was validated against actual data for travel times and the probability that a call enters queue.

### PCAM (Patrol Car Allocation Model)

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 will operate in batch or interactive mode, depending on the choice of a small number of program

[4] This was found to be true in Los Angeles, by analysis of available data.



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 certain input parameters.

**Policy Issues Addressed.** PCAM can be used to analyze any policy question related to the *number* of patrol cars or patrol officers a department should have on duty, or the *times of day* at which patrol cars begin work. Examples of typical applications would be:

- Determining the total number of patrol officers a department should have (e.g., during budget preparation).
- Allocating a fixed total number of patrol officers among geographical commands.
- Assigning patrol cars to geographical commands at the start of each tour (i.e., some patrol cars may not have a permanent geographic assignment).
- Allocating a fixed total number of officers by time of day.
- Analyzing the possibility of an *overlay* tour. (This is a tour that begins during one regular tour and ends during the next one. For example, there could be an eight-hour tour beginning at 4 p.m. and another one at midnight, plus an overlay tour running from 7 p.m. to 3 a.m. PCAM has the capability to recommend allocations to tours when one tour in each day is an overlay.)
- Studying possible changes in tour starting times.
- Analyzing the adoption or modification of call priority structure (i.e., which calls are classified by dispatchers at different levels of importance).

PCAM cannot be used for designing patrol beats or for studying changes in dispatching practices or equipment to be used by dispatchers (e.g., automatic vehicle locator systems).

**Structure and Output of the Model.** PCAM is a simple analytic model. It calculates performance measures from approximate equations which are calculated hour-by-hour and take non-cfs work into account. The 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 includes 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 (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 ten minutes? What is the smallest number needed so that both of these conditions are met? This capability is implemented by a simple iterative procedure that increases the number of patrol cars one at a time until all the 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":

- 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.

This capability is provided by an optimization program that also operates iteratively but is somewhat complicated. The optimization algorithm has been proved optimal when there are no overlay tours or when there is an overlay tour and it has the same length (in hours) as the tours overlaid.

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.

The user can consider a single tour and specify the total number of patrol cars on duty in the entire city. PCAM will then prescribe how many of them should be assigned to each geographical command. Or, the user can consider a single geographical command and specify the total number of car-hours that can be fielded on one day, say Monday. PCAM will then prescribe how many cars should be on duty during each tour on Monday in such a way that the allocated car-hours add up to the specified total. The user can also consider a single command for an entire week, or the entire city for a day or a week.

**Data Requirements.** The model accepts user input for the *names* to be given to various features being modeled. For example, the user can choose to call precincts "districts," in which case the word "district" is provided as input and appears in all output table headings, etc. Similarly, tours can be called "watches," or whatever the department chooses. Each precinct and tour is also given a name, such as Midtown or Third. A precinct is described geographically by its area (square miles) and number of street-miles to be patrolled, which are input data.



For each hour of each day in each precinct, PCAM requires data telling the expected number of calls for service and the expected service time. Calls may be broken down into three priority levels, if desired. If the user wants to relate the amount of preventive patrol to crime rates, he must select some category of crimes to be called "suppressible" (meaning that they are presumably affected by the amount of preventive patrol) and provide as input the expected number of such crimes at different times of day for each precinct.

Characteristics of patrol cars that must be entered as input are their response speed, their patrol speed, and two *unavailability parameters* that describe the linear equation used to calculate the amount of non-cfs work.

**Cost and Computer Requirements.** The PCAM program is written in FORTRAN IV and obeys all ANSI standards,[5] except that extended subscripts are used, and quoted literals appear in format statements. As the program is distributed, it requires 160K bytes of core storage on an IBM System/370 computer, but this can be reduced (to no lower than 120K) in many applications by redimensioning two arrays in the program. Large departments may require more than 160K bytes of core storage for elaborate calculations, but the existing program accommodates one day's data for all the precincts in New York City.

The program costs about $10 to compile and then costs about $2 to $10 for typical runs. It is provided by The Rand Corporation at modest cost in either batch form or interactive form (which differs from batch on only a few lines of the program).

**Validation and Verification.** This program has been verified using test data and comparing its output to that of previous patrol car allocation models. The part of the program that estimates travel times has been validated against real and simulated travel-time data. The calculation of queuing statistics when there is no non-cfs work has also been validated against simulated data. However, the calculations of queuing statistics using the conversion of actual to "effective" cars have only been validated in Seattle, so new users would be wise to do the same for their localities. (The unavailability parameters, which are provided as input, permit adjusting the model's estimates of queuing delays so that they fit real data.)

**Implementation and Impact.** PCAM has been implemented by police agencies in Atlanta, Edmonton (Alberta), Minneapolis, The Netherlands, Seattle, and Toledo, and it has been used by a private consulting firm for work in Wilmington. It has also been acquired and operated with test data by a dozen other departments. The model is too young for a discussion of impact, but the first use of PCAM (for Wilmington) resulted in allocations that were put into practice.

**Limitations.** The model contains no geographical structure and is insensitive to the locations of patrol cars within a precinct and to differences in call rates, crime rates, or patrol densities in various parts of a precinct. It cannot handle more than three levels of priority for calls. Its model of dispatching practices is quite simplified and cannot take into account the following types of practices:

[5] American National Standards Institute, *FORTRAN*, X-3, 9-1966.

- Dispatching patrol cars across command boundaries for high-priority calls
- Placing low-priority calls in queue to await the availability of the local beat car, even when other patrol cars in the command are available to be dispatched
- Holding some cars in reserve for high-priority calls
- Preempting service on low-priority calls in order to dispatch the busy car to a high-priority call.

In addition, the dispatch of more than one patrol car to an incident can be handled only approximately in this model.

PCAM does not assist the user in calculating certain vital input data: call rates and service times by hour and precinct, and the unavailability parameters. Most departments would have to write subsidiary computer programs for this purpose. The documentation gives some guidance.

**Transferability.** The program was designed specifically to be transferable, and no installation problems have been encountered to date by potential users who have acquired the program.

**Documentation.** The model is completely documented by an executive summary [32], which contains about the same kinds of information as given in this report, a user's manual [33], and a program description [34] that gives installation instructions as well as an annotated program listing.

## SIMULATION MODELS FOR PATROL SYSTEMS

The four models discussed here are:

- A model developed at Brooklyn Polytechnic Institute for the New York City Police Department [77].
- A model developed by a research team at the Illinois Institute of Technology for the Aurora, Illinois, Police Department [170].
- A model developed by R. Larson, and later modified by Urban Sciences, Inc., for use by the Boston Police Department [181].
- A model developed by The New York City-Rand Institute for the New York City Police Department [114,115].

### Background

A simulation model is a representation of a real system that can be manipulated to examine the effects of various changes in the manner in which the system is operated. It can be used to examine the effects of changes that are too expensive, too disruptive, or too time-consuming to make on the real system. For example, one can examine the effects of a reduction in the number of patrol cars assigned to a given area without actually subjecting the residents of that area to the loss of service that would be incurred if the reduction were actually carried out. A number of different car assigment rules can be examined without the administrative problems caused by changing the rules of the real system.

A simulation model, while general in the sense that it can be constructed to model a real system in as much detail and comprehensiveness as is desired, is, in its final form, also restrictive. It can only model systems that match the operating rules and structure that are built into the simulation. Thus a simulation model of the New York City patrol system has a specific structure, and it may or may not be possible to use this model in Mansfield, Ohio. It depends on the way the patrol system of Mansfield is structured in relation to that of New York City. One who expects to transfer a simulation model developed for one city to another city must have a good understanding of the model structure and its potential value in the new situation. This is not to say that simulation models are useful only for the city in which they were developed; in fact, most of them are general enough to apply in many cities.



Here we introduce some additional terminology. The term *precinct* is used in the same sense as in the preceding section (also see the Glossary). By a *sector* we mean a subarea of a precinct that defines a patrol region. A patrol car is assigned to one or more sectors in which the car performs preventive patrol when not busy with a job or out of service. A sector is sometimes termed a beat, area, watch, or post. Sometimes a sector has more than one car assigned to it, but unless no patrol is desired, each sector must be assigned at least one car.

By *job priority* we mean a number that signifies the importance of a call for service. Priority 1 jobs, such as a crime in progress, are the most important. The higher the priority number, the lower the importance of a job. Different simulations have different numbers of priority classes. Sometimes, as for The New York City-Rand Institute model, job priority indicates not only the importance of the job, but some special condition, such as the fact that the job was generated by the patrol officer rather than a call from the public. Priority numbers are used primarily to determine the number and identity of patrol cars to be assigned to a call for service.

Although we have been referring only to patrol cars, some simulations have the capability of including other patrol units, such as scooters, footmen, patrol wagons, and even horses (if one thinks of a horse as a large, slow scooter).

### Policy Issues Addressed by Simulation Models

We list here some of the questions that can be examined with the aid of simulation models. These issues are not the exclusive province of simulation models. Some of them can be considered using other models, such as the allocation models discussed in the preceding section. However, many of them can be addressed in a more accurate and direct manner by simulation models.

1. What are the effects of increasing or decreasing the number of patrol units? This is the major issue addressed by allocation models; however, simulation models can give much more accurate estimates of the effects and can disaggregate the performance measures by patrol car or by geographical subarea.

2. What are the effects of different dispatching rules? For example, one rule that may be investigated is the assignment of a certain number of patrol units to answer only a certain set of calls for service. Another is the holding of certain low-priority calls for the sector car even when there are available cars elsewhere in the precinct.

3. What are the best sector boundaries and how should cars be assigned to different sectors? Is it better to have two cars in separate sectors or should they be assigned to patrol both sectors simultaneously? How should sectors be designed to equalize workloads or to minimize response time? These issues are addressed less accurately by beat design models (see next section).

4. How do workloads of cars vary with different dispatching rules and patrol assignments? Because of the way cars are dispatched, car workloads are not necessarily equalized by assigning them to patrol areas of equal workload. Centrally located cars tend to be busier than other cars.

5. What is the effect of enlarging a precinct or of combining two or more precincts into a single dispatch command region? Pooling more cars has the effect of decreasing the waiting time before job assignment, but larger areas mean increased travel times. What is the effect on total response time, queuing delay, and travel time?

6. What is the benefit of knowing the exact location of each patrol car at each point of time? This would allow closest-car dispatching. A simulation can provide estimates of reduction in response time that can be used to determine whether the cost of a car locator system is justified.

7. What is the effect of adding scooter patrol units to the patrol car system? To what extent can scooters relieve the workload of patrol cars?

**General Properties of Patrol System Simulation Models.** All of the simulation models we describe have certain elements in common. Models differ primarily with respect to the manner in which the following items are handled:

1. *Geographical Structure*—Most of the simulation models have a method of recording job and car location information. Usually an x-y coordinate system is defined. Elementary reporting areas are the smallest geographical units in the simulation, sometimes called "atoms" or "nodes." Usually these are groups of city blocks, with the centroid of the area used as the location in the simulation.

2. *Definition of Sectors*—Sectors are usually defined as a group of reporting areas. One method is to list all reporting areas in each sector. Another is to define sectors as polygons and then input only the corner areas into the simulation. The simulation program then determines internally which reporting areas belongs to which sector.

3. *Patrol Units*—Some models can simulate only patrol cars. Others can also handle scooters, patrol wagons, or footmen.

4. *Assignment of Patrol Units to Sectors and Precincts*—This is simply a list of patrol units with their associated sector and precinct assignments. It is usually an input to the simulation.

5. *Assignment of Precincts to Zones*—If a model can simulate more than one precinct, the precincts can be grouped into zones. This is primarily for statistical analysis purposes.

6. *Job Input Stream*—This is the record of job information that is necessary for the simulation. It can contain all or part of the following: reporting time, job location, crime type, radio code, job duration, job priority, and the number of men or cars required to service the job. Job records may be actual historical records or a randomly generated job stream that has approximate statistical properties for the area being simulated. The generation of the latter may be external or internal to the simulation itself. In either case, use of generated job streams requires an historical statistical analysis of job characteristics.

7. *Dispatcher Delay*—This refers to the interval between the time a caller finishes his telephone conversation and the time when the dispatcher is ready to assign a patrol car. This delay may be caused by mechanical processing, or the dispatcher may be dispatching other jobs or other precincts and cannot process the job immediately. It is not the same as *queue delay*, the delay caused when there are no available cars to take the job. Some simulations ignore dispatcher delay; others use randomly generated delays based on an analysis of past delays.



8. *Unit Assignment Rules*—Once a call for service is ready for assignment, the simulation must determine the type, number, and identity of patrol units to be assigned. These dispatch rules are the heart of simulations and are usually specifically designed to match the rules used in the city being modeled.

9. *Job Travel Times*—Simulation models usually calculate job travel time based on the distance from the patrol unit to the job and a response speed. One model accepts as input a matrix of distances between locations, eliminating the need for x-y coordinates. Usually, distances are measured as right-angle, or rectangular, distances rather than straight-line distances, so as to correspond more closely with city street patterns.

10. *Preemption Rules*—Some simulations allow certain jobs to preempt other jobs already in progress. Usually this is on the basis of the job priority. If a job is preempted it must be assigned to another car or returned to the queue of jobs awaiting assignment.

11. *Patrol Activity*—This refers to the manner in which the model accounts for the movement of patrol units while they are performing patrol.

12. *Out-of-Service Times*—This refers to non-cfs work, when a patrol unit is unavailable to answer calls for service due to such activities as getting gas, mechanical failure, meal breaks, etc. Different models handle this in different ways; one ignores it entirely.

13. *Output Statistics*—Models differ with respect to the amount and form of the simulation results. *All of them summarize basic average queuing information*, but some have special features, such as snapshot records, that allow the user to examine the state of the system at any selected point in time. One model can create a record of each job's experience as it passes through the system.

We now discuss the four models in terms of these and other relevant factors.

### Model 1: Polytechnic Institute of Brooklyn, 1969

**History.** This model was developed under a grant from the Law Enforcement Assistance Administration, contract number LEAA030 (OLEA), to the Polytechnic Institute of Brooklyn, New York. With the assistance of the New York City Police Department, the research team, headed initially by Daniel Duffy and later by Norbert Hauser, developed three simulation models. One was a model of the Manhattan communication center that was disbanded in 1968. Another dealt solely with the receipt and processing of calls for service up to the point of transmittal of information to the dispatcher. Neither of these models is discussed in this report. The reader is referred to Ref. 77 for their details. The third, the "dispatching and field response" model, is the only one that concerns the patrol system.

**Structure of the Model.** The simulation is written in GPSS, a readily available IBM language especially designed for the construction of simulation models. The major features of the model are as follows:

1. *Geographical Structure*—The model does not maintain the exact location of calls for service. It simulates the operation of more than one precinct and the only location information is that a call for service is in a particular precinct. The only input is the number of precincts.

2. *Definition of Sectors*—No sectors are defined. Patrol units are assumed to be always in their respective precincts, but the location of patrol units is not considered.

3. *Patrol Units*—The simulation considers both cars and scooters as patrol vehicles. The input is the number of each for each precinct.

4. *Assignment of Patrol Units to Sectors*—Does not apply.

5. *Assignment of Precincts to Zones*—There are no zones, just a single set of precincts.

6. *Job Input Stream*—All job information is generated internally. The time between jobs is exponentially distributed with a user-specified mean. There is no provision for changing the rate of arrival of calls for service over time. The precinct of occurrence for a job is chosen according to an historically derived probability distribution (a user input). Other attributes are ascribed to the job by user-supplied probability distributions. They are: the type of crime (8 types are allowed), whether the crime is inside or outside, and the probability that the crime will still be in progress at the time of the call for service. No priority structure as such is given, but under dispatching rules it will be seen that the type of call determines whether a car or scooter is assigned to the job.

7. *Dispatcher Delay*—This is handled by assuming that dispatcher delay time has an exponential distribution with a mean that increases as a function of patrol unit utilization. The mean function must be supplied by the user.

8. *Unit Assignment Rules*—Each of the eight types of calls for service is assigned to either a car or scooter. The user specifies this matching. The car or scooter assigned to a job is the first available. Only one unit is assigned to each call. Cars and scooters do not interchange jobs, so the model is simply two separate service systems being run simultaneously.

9. *Job Travel Times*—The user must specify two functions, giving the mean travel time as a function of patrol unit utilization for both cars and scooters. Travel times are assumed to be exponentially distributed with the specified mean function.

10. *Preemption Rules*—No preemption of jobs is allowed.

11. *Patrol Activity*—The model has no provision for recording patrol locations.

12. *Out-of-Service Times*—No out-of-service times are considered, not even meal breaks.

13. *Special Features*—This model contains a provision for calculating the number of arrests made during the simulation. To do this, the model requires that the user supply a function relating the probability of arrest (for a crime in progress) to response time.

14. *Output Statistics*—The output gives the distributions of the time until a call is dispatched, the total response time (travel time plus total delay of dispatcher), and the time until completion of the jobs. Also, a summary of the number of job requests, by type, and the average number of busy cars and scooters is provided.

**Data Base Requirements.** The simulation itself requires little data input, but before the model can be used a thorough analysis of call-for-service characteristics must be made to determine mean job service times, distribution of calls between precincts and among types of calls, job interarrival times, and a verification of the assumption of exponential distributions. Also, an analysis of dispatcher delay, arrest probabilities, and travel-time characteristics must be made. The latter is especially difficult as it must be carried out at different car utilization levels in each precinct to be simulated. This is discussed below under limitations.



**Cost and Computer Requirements.** No cost figures are available. Any computer that has the GPSS program can run this model. Although storage requirements are not stated, it is apparent that they are minimal.

**Validation and Verification.** No validation of this model was attempted. Internal verification was performed as part of the model building process. Sample runs based on the analysis of limited data were made and results are presented in Ref. 77.

**Implementation and Impact.** The model was never implemented by the New York City Police Department. In fact, it was dropped soon after the research project was completed. Interestingly, one of the most useful results of the simulation was to demonstrate that New York City did not need a computerized dispatch system as much as it needed a revamping of the method for handling incoming calls. However, at the time of the study the New York City Police Department had already committed itself to a computerized system, the SPRINT system installed after 1969, and this finding is not discussed in the final project report.

One of the authors of the study has stated in private communication that he would be reluctant to become involved with a project of this nature again unless he were satisfied that the result of the work would have a good chance of being implemented.

**Limitations and Benefits.** The major limitation of this model from the police planner's viewpoint is that it does not record the location of jobs and does not record the identity of cars or scooters that are assigned to jobs. This means that it is basically modeling a standard multiple-server queuing system with indistinguishable servers, which is exactly the system that the allocation models discussed in the previous section are designed to solve, using exact queuing formulae, at a fraction of the time and cost of a simulation. Because of these limitations the model cannot be used to investigate the effects of different dispatching rules; it cannot measure individual patrol unit workloads; and it cannot be used to examine the design of patrol sectors.

Additionally, the average travel time is not calculated by measuring travel distances and applying a response speed. Travel time is an input to the simulation, a function depending only on utilization of the system. Thus, this model cannot be used to calculate accurate average travel times. In fact, if one knew enough about the patrol system to construct the response time function required by the model, then the model would almost be superfluous.

Another limitation of the model is that, in holding out the promise of computing arrest characteristics, it misleads the user. At the present time no one has constructed a reliable arrest probability curve as a function of response time. Any model predictions based on this input would probably be inaccurate.

Another limitation is that the dispatch rule calls for only one unit to be assigned to each job.

A good feature of the model is that it provides for a dispatcher delay that is a function of car utilization. This is true in most patrol systems and has been neglected by other model builders.

**Transferability of the Model.** This model can be easily transferred. The data requirements are not extensive, although the analysis required to prepare the input is, and the model does not require a great amount of user modification. A GPSS language capability is required.

**Documentation.** The model is well documented and the complete computer program is listed in Ref. 77, the final LEAA project report.

### Model 2: Superbeat—Illinois Institute of Technology, 1973

**History.** This simulation model was developed as one part of a 1972-1973 research project supported by the Illinois Law Enforcement Commission (grant number 2-10-04-0372-01). The study, directed by Spencer B. Smith, produced a number of other models: a forecasting model, a beat design model, and a patrol manpower scheduling model. The simulation model is designed to reflect the patrol system of Aurora, Illinois, the prototype city for the research project.

**Structure of the Model.** The program is written in FORTRAN V and is run in batch mode. The major features of the model and the input requirements are:

1. *Geographical Structure*—The user must group city blocks into a set of "nodes." Nodes are the smallest geographical units used in the simulation. No x-y coordinate information is required in the program. The simulation input requires a matrix that defines the (shortest) distance between nodes. These matrices must be computed before using the simulation. The Superbeat report [170] discusses the method for doing this. It requires a separate computer program, and x-y coordinate information must be gathered for this program.
2. *Definition of Sectors*—A collection of contiguous nodes forms a sector. They are defined in the input data by simply listing the nodes in each sector.
3. *Definition of Precincts*—This model defines "sections," which are collections of sectors similar to precincts except that dispatches can be made across section boundaries.
4. *Patrol Units*—The simulation allows for patrol cars containing a variable number of patrolmen. Basically, one- or two-man cars are considered.
5. *Assignment of Patrol Cars to Sectors*—These assignments are specified by the user by listing them as part of the input.
6. *Job Input Stream*—The model has a rudimentary capability to generate a job stream internally. Generally, it is operated with either real job data or externally generated imaginary jobs. The following information is required for each call for service: arrival time, the type of call, the location (node) of the call, the service time, and optionally, one of two possible job priorities, and job manpower requirements (number of men that must be sent to the job). The job stream must be sorted by arrival time. Preparation of this job stream is the task of the user.
7. *Dispatcher Delay*—There is no dispatcher delay built into the model. It is assumed tht when a call arrives, the dispatcher is available to dispatch it immediately. (As mentioned above, this is to be distinguished from queuing delay, which is calculated by this model as well as the others.)
8. *Unit Assignment Rules*—The assignment of cars to jobs is based on the manpower required by the job and the distance of the closest free car (or cars). There are rules for deciding to send two 1-man cars or one 2-man car to a job requiring 2 men. Low-priority jobs are not assigned to cars outside of the precinct of occurrence.
9. *Travel Times*—As mentioned above, a matrix provides the distance from any node to any other node. Three speed factors are supplied by the user: a base speed, t, and two correction factors. One modifies the base speed t according to time of day, the other modifies t by the type of call. The travel time between nodes is then distance divided by speed. A special formula gives travel times for travel within a node.
10. *Preemption Rules*—Low-priority jobs are preempted by high-priority jobs up to a maximum number of preemptions. This number is supplied by the user.
11. *Patrol Activity*—The model simulates patrol by assuming that cars move from node to node in their sectors in a random fashion when they are on patrol.
12. *Out-of-Service Times*—These times are entered as standard job information. Downtime is assigned to cars at random and is taken when the car first becomes free.
13. *Output Statistics*—The following information is presented by the simulation:

   a. By sector and for all sectors:
      (1) Average travel time
      (2) Average waiting time
      (3) Average service time
   b. A preemption profile
   c. The number of dispatches per sector
   d. The number and percentages of dispatches to each sector car
   e. Car activity profile: patrol, travel, and service times
   f. An optional job file may be created to contain all pertinent information about each job's passage through the system.



**Data Base Requirements.** The information required is the node set, the two associated shortest-distance-between-nodes matrices, and the job stream file, all of which are described above. If a computer record of actual calls is available, it would not be difficult to create the job stream file. The distance matrices require a separate shortest-distance computer algorithm.

**Cost and Computer Requirements.** It is stated in Ref. 170 that using a Univac 1198, simulation of Aurora, Illinois (180 nodes) requires 70K of 36-byte word storage, 28 seconds CPU for initialization and output, and about 15 seconds CPU per day simulated. One day in Aurora has approximately 150 calls for service and so the cost per call is approximately .1 seconds CPU per call. If the call event record is requested, the additional time is about .02 seconds CPU per call. A FORTRAN V compiler capability is required.

**Validation and Verification.** The model was verified by running sample job streams and comparing the simulation outputs with hand computations. No validation of the model's accuracy in representing the real system was performed.

**Implementation and Impact.** This model was not implemented. A few test runs were made but the model was never integrated into the regular police planning functions. The authors of the study believe that the model has value as an evaluative tool for occasionally examining particular changes in patrol policies. However, the frequency of use for a city like Aurora, Illinois, does not justify maintaining the model and keeping personnel trained to use it. The designers believe that a police analysis center, staffed by trained systems analysts, maintaining such models for many different police departments, would be a good way to remove the burden of separate department maintenance, which usually leads to the eventual abandonment of such models. This will be discussed further in Chapter 8.

**Limitations of the Model.** A primary limitation of the model is that for cities with a large number of reporting areas (nodes), the determination and simulation

CONTINUED

1 OF 2

storage of the distance matrices may be a problem. Storage requirements increase with the square of the number of nodes. The program is currently limited to 200 nodes.

There is no easy provision for increasing or decreasing the number of patrol cars through time. This can be accomplished on an ad hoc basis using the out-of-service input structure, but it would be difficult if many changes were desired.

Another drawback is that the model assumes that the node location of all cars is known at the time of dispatch. This is not true in many cities.

**Transferability of the Model.** This model can be transferred to other cities easily, provided the number of reporting areas required is not over 200. Actually, for even the largest of cities it would be possible to simulate a portion of the city using this model. The developers of the model are willing to discuss applications in new cities. A FORTRAN V compiler must be available to run the model, and a systems analyst must be available to supervise the installation and interpret the results.

**Documentation.** The model is described in Ref. 170 and is fully documented in the *Superbeat Program Manual* [171].

### Model 3: Urban Sciences, Inc., 1971

**History.** In the late 1960s, Richard Larson of the Massachusetts Institute of Technology developed a general police patrol simulation model. The general properties of the program are described in his book [119]. The original model was programmed in the MAD computer language. In 1971, under contract with the Boston, Massachusetts, Police Department, the management consulting firm Urban Sciences, Inc., reprogrammed the model in the PL/I language and modified the model to fit the structure of Boston department operations.

**Structure of the Model.** The programming language is PL/I. The input requirements are noted in the various categories below.

1. *Geographic Structure*—The basic reporting area in the simulation model is called an atom. The boundaries of atoms are defined by the simulation input, which specifies the x-y coordinates of the corner points of each atom. An atom can be any size, but for accurate model results atoms should be small enough so that the spatial distribution of calls for service is approximately uniform over the atom.

2. *Definition of Sectors*—A sector is a collection of atoms. Program input requires a list of the atoms making up each sector. An atom may belong to more than one sector, allowing for overlapping sectors.

3. *Definition of Precincts*—A collection of sectors defines a precinct, called *districts* in Boston. The program can simulate as many as 11 precincts. Cross-precinct dispatches are not allowed, so the simulation of each precinct is separate.

4. *Patrol Units*—The program allows a single type of patrol unit.

5. *Assignment of Cars to Sectors*—This is done with a simple listing procedure as part of the simulation input. A car can be assigned to only one sector.

6. *Job Input Stream*—Jobs are generated internally. The user inputs the mean time between calls for service for each precinct to be simulated. Time between arrivals is assumed to have an exponential distribution. The mean time between arrivals is assumed constant over the entire simulation run. Four job priorities are allowed, with random selection of job priority, based on a user input distribution.



Job service time depends on job priority. The mean service time for each priority class is a program input, and service times are drawn internally from an exponential distribution. The location of jobs is randomly assigned; the simulation first randomly determines a node of occurrence, based on user input distributions, and then the x-y coordinates within the node are chosen according to a uniform distribution.

7. *Dispatcher Delay*—No dispatcher delay is accounted for by the model, only the queuing delay found in all the models.

8. *Unit Assignment Rules*—When a call for service is ready for assignment, its priority indicates the set of cars that can be considered for assignment. A special user input determines for each job priority class the set of lower-priority classes that can be preempted by that class. Assignment of a car to a job is based on the distance to the closest eligible car; eligible cars are those on patrol (an exception to this is noted below) and those servicing calls that can be preempted. A job is assigned to the closest eligible car in the precinct of its occurrence. However, a set of user inputs determines how accurately the distance calculation is made. Three user-selected situations are possible with respect to distance calculations:

a. The dispatcher assumes that the job is at the center of its sector and eligible patrol cars are at the centers of their sectors.
b. The job location is known exactly and cars are assumed to be at the centers of their sectors.
c. The job location is known exactly and approximate car location is determined using a bivariate normal distribution with mean at the exact location of the car and user-specified variance. (This imitates the performance of an automatic vehicle locator system.)

These options reflect a spectrum of location information ranging from practically nothing (option a) to exact knowledge (option c with zero variance).

By means of another user input, the assignment of cars to jobs can be restricted to assignment of only the car (or cars) assigned to the sector of the job. Usually all precinct cars are considered for assignment.

Another feature of this model is that individual patrol cars are assigned a priority for preventive patrol. By means of this device some cars can be assigned to only high-priority jobs. For instance, a car whose patrol priority is assigned a value of 3 can answer only Priority 1 and Priority 2 jobs. If a Priority 3 or Priority 4 job is received when this car is on patrol, it will not be eligible for assignment.

If a job is preempted, it returns to the waiting queue of jobs if no car is eligible for immediate reassignment. Reassignment policy for preempted jobs is chosen by the user. There are two options:

a. A first-come, first-serve rule
b. A closest-car, closest-job rule.

Under the first option, jobs are selected for reassignment in the order of their arrival in the system. The second option also requires the specification of a distance weighting factor for each job priority class. Distances are divided by this weighting factor and the job that is closest to a car using this modified distance-weight factor is assigned first. This has some subtle implications, for the weights can be chosen such that the first-come, first-serve policy is effectively used within each priority class.

In addition, the user must also specify maximum travel distances for job assignment. These can vary with job priority and the state of the patrol car (on patrol or being reassigned to another job).

9. *Travel Times*—Travel times are calculated by dividing response distance by an "effective" speed. The user must input an effective speed for each of the four priority classes.

10. *Preemption Rules*—Preemption is allowed. This was discussed above under *Unit Assignment Rules.*

11. *Patrol Activity*—Cars are presumed to patrol their sectors according to a probability distribution over the atoms of the sector. The user supplies this distribution.

12. *Out-of-Service Times*—No special routine is allowed for these times. They can only be accounted for by generating them internally in the job input stream. This means they will actually appear as regular jobs.

13. *Output Statistics*—Output statistics can be gathered by histogram on the following variables by specifying histogram spacings and truncation points:

a. Service time (includes travel time)
b. Workload (number of calls dispatched)
c. Preemptions (number)
d. Travel times (minutes)
e. Number in queue (by priority)
f. Queuing delay (by priority)
g. The fraction of times the dispatcher had to estimate the location of a car
h. The fraction of in- and out-of-sector dispatches
i. The fraction of times the closest car was actually assigned
j. Average extra response distance caused by not assigning the closest car.

Summary statistics (mean and variance) of the above are also presented. Both histograms and summary statistics can be grouped by the following aggregation levels:

a. Precinct
b. Precinct, by priority
c. Patrol unit
d. Patrol unit, by priority
e. Over all precincts (citywide).

Another user option is a system status dump at specified time intervals. Also, if desired, a trace of the state of the system at the time of selected events can be printed.

**Data Base Requirements.** The user must first group the area to be simulated into a set of polygonal reporting areas, sectors, and precincts, and then by historical analysis determine:

a. The mean time between all calls by precinct
b. The fraction of calls that occur in each reporting area of the precinct
c. The distribution of job priority classes for each reporting area
d. The x-y coordinates of the corner points of each reporting area
e. The mean service time for each priority class



Since interarrival and service times are assumed exponential by the model, it would be wise to check the accuracy of this assumption. The user must also specify a distribution of patrol effort over the reporting areas of each sector.

**Cost and Computer Requirements.** The simulation was split into an overlay program of five phases so it could fit into a 96K-byte storage area available in the computer of the Boston Police Department. This does not mean that the full program would take five times 96K bytes, because the program keeps one phase in memory throughout the entire execution. An estimate of the storage requirements if the entire program is put in memory is 200K bytes.

In the report on this program [181], no time or cost estimates are given. The program requires PL/I language capability.

**Verification and Validation.** During the development of this model, a few sample runs were made using data collected in Boston. Internal verification was performed and the program was completely debugged. A validation was not performed. The difficulty is that data on such things as travel time, intersector dispatches, and queuing delays were not available for the actual Boston system by shift by precinct. Some citywide 24-hour data were available, and comparisons of queuing delays, travel times, intersector assignments, and service plus travel times were made between an eight-hour simulation run and an actual three-day period. This showed that the simulation results were approximately correct, but the authors caution that this is not an adequate comparison for validation.

**Implementation and Impact.** The program was physically implemented in that it was installed on the Boston department's computer and turned over to the department. However, it was not implemented procedurally. The Boston department is not currently using the model.

**Limitations of the Model.** A major limitation of the model is that the rate of arrival of calls for service and of service times cannot be varied by time. Therefore, the results are not applicable over a period when the actual call rate varies substantially.

There is also no provision for meal breaks or changes in the number of patrol units in the simulation. This further restricts the use of the model to short periods during which car levels are relatively constant.

The model does not allow for interprecinct dispatches, which is not a major limitation if this is not a frequent occurrence.

Only one type of patrol unit is permitted by the model, which makes it impossible to study the effects of cars plus scooters or other patrol vehicles.

It is not easy to change the geometry of the patrol system (sector assignments). This is a limitation for the investigation of sector boundary designs.

There is no provision for dispatcher delay.

Another important limitation is that only one patrol unit is assigned to a call, and the effects of multiple-car dispatches cannot be simulated. Many departments currently dispatch two or more cars to high-priority jobs.

**Transferability of the Model.** To transfer this model to another city successfully, the potential user must be able to construct the data base requirements listed above. He must also contact Urban Sciences, Inc., since certain parts of the program software are proprietary. The program may be accessed interactively by subscribing to a commercial time-sharing service. The major changes to Larson's program made by Urban Sciences were for the purpose of facilitating interactive use. This caused the program to be more specifically tailored to Boston than Larson's program. Potential new users may find it is easier and more convenient to write their own program following the design principles in Larson's book [119].

**Documentation.** The principles of the program are described in Ref. 181. The documentation is partial. The program is not listed, and many parts of the data base are not explicitly displayed. All rules for running the program as it exists in Boston are given but, since many of the data files are permanently stored in Boston, it is not possible to use this description to implement the model elsewhere.

### Model 4: The New York City-Rand Institute, 1973

**History.** As part of a 1973 research contract with the New York City Police Department, Peter Kolesar and Warren Walker of The New York City-Rand Institute developed a patrol car simulation model. A validation of the model was carried out under the 1974 contract with the department.

**Structure of the Model.** The model is written in SIMSCRIPT II.5. The features and input requirements are noted below:

1. *Geographic Structure*—The basic geographical unit is a block. The x-y coordinates of the center of each block must be input by the user. The coordinate axes must first be oriented to coincide with street directions. A simulation block need not correspond to an actual city block.

2. *Definition of Sectors*—The blocks are first aggregated into a set of nonoverlapping neighborhoods. Each block belongs to exactly one neighborhood. A sector is defined as a collection of neighborhoods. Sectors may overlap, i.e., a neighborhood may belong to more than one sector. The user specifies these properties with input lists of each neighborhood and sector elements.

3. *Definition of Precincts*—This model simulates only one precinct (i.e., one dispatching region). All the blocks of the simulation constitute the precinct.

4. *Patrol Units*—One type of patrol unit is provided for, but there may also be supervisors' cars, report cars, etc.

5. *Assignment of Patrol Cars to Sectors*—Two types of cars are defined: sector cars that are assigned, by user input, to one or more sectors, and supervisory or special cars that are assigned to no sector. The latter answer only high-priority jobs and certain other jobs when no other car is available.

6. *Job Input Stream*—The simulation does not generate jobs internally. It will accept either a real historical job stream or an imaginary stream generated externally. The user-supplied job input contains the entry time, priority, job location (block), and job duration for each job. The input must be sorted by entry time. There are five priority classes.

7. *Dispatcher Delay*—There is no provision for dispatcher delay, only the queuing delay found in all the models.

8. *Unit Assignment Rules*—For each neighborhood the user must input a "nomination list" of cars. This is an ordered list of cars in the order of preference for dispatch to a job in that neighborhood. First on the list are the sector car(s) assigned to the neighborhood; next, one or more cars are designated as adjacent cars; and finally, all the remaining cars are listed in the order preferred for assignment to jobs in the neighborhood. Supervisory cars come last.

There are five dispatch rules corresponding to each of the five job priority classes. In the following rules, available cars are all cars on patrol or, in the case of Priority 1 calls, responding to a lower-priority job (not working on it, just traveling to it). Jobs in progress are never preempted in this model.

a. *Priority 1 Dispatch Rule*—Assign the first MAX.SENT of available cars. MAX.SENT is a maximum Priority 1 response specified by the user. If no cars are available, the job is discarded and this fact is recorded. Priority 1 jobs are never queued.
b. *Priority 2 Dispatch Rule*—Either one or two cars are assigned. If no cars are available, the job is queued in the Priority 2 queue and it will receive one car when a car is available. If a sector car *and* one of the adjacent cars are both available, they are both assigned to the job; otherwise, only one car is sent, the first available car on the nomination list.
c. *Priority 3 Dispatch Rule*—Exactly one car is assigned, the first available car on the nomination list. If no car is available, the job is queued in the Priority 3 queue.
d. *Priority 4 Dispatch Rule*—Only a sector car is assigned to a Priority 4 job. If a sector car is not available, the job is placed in the Priority 4 queue, from which it is assigned to the first available sector car. This priority class is intended to represent non-cfs work (e.g., an activity resulting from an officer observing some incident requiring his attention). Therefore, zero travel time is assumed for the response to a Priority 4 call.
e. *Priority 5 Dispatch Rule*—Same as Priority 4 jobs, except that travel time is calculated normally (see travel time section below).

Jobs that have been queued are assigned on a first-come, first-serve basis by priority; that is, first Priority 2, then Priority 3, 4, and 5 jobs are dispatched.

A special version of the program has been written that always dispatches the closest available car.

9. *Travel Times*—The distance from the assigned car to the job is calculated using rectangular distance. Each priority class has a user-supplied response speed. Distance is divided by the appropriate speed to get travel time. One exception, as noted above, is Priority 4 jobs, which have no travel time.

10. *Service Times*—The job input record contains a job duration value. For jobs that receive more than one car, the model assumption is that the first arriving car works for the entire job duration, the second arriving car works a fraction P2 of the job duration, and all other cars work a fraction P3 of job duration. The user must specify P2 and P3.

11. *Preemption Rules*—Once a car arrives at a job, that job will not be preempted. Preemption by Priority 1 calls is possible if a car is traveling to a lower-priority call.

12. *Patrol Activity*—If a car finishes work on a job in its sector, it does not move until called to another job. If a car finishes a job outside of its sector, it returns to a (user-supplied) centroid block in its sector, where it waits for another assignment. At the beginning of the program all cars are assumed to be at their sector's centroid block.

13. *Out-of-Service Times*—The simulation has a user-defined tour length (eight hours in New York). For each car, the user must define a meal break duration and the hour during a tour at which that car's meal is to be taken. Then, in multiple-tour

runs, a meal break is automatically scheduled for each car at the specified hour of each tour. In addition, the job entry record can contain special out-of-service times. The user specifies the car, the duration, and the time at which the out-of-service period begins. If a car is busy when it is supposed to go out of service or on meal break, the period does not begin until the car finishes the job it is working on.

14. *Output Statistics*—The output record is a summary of the entire simulation period. There are no trace or snapshot options in the program. There are six reports:

a. *Car Activity*—Number of jobs handled (within and outside of sector); the proportion of time spent working, responding, on patrol, and out of service. The averages over all cars are also given.
b. *Queuing Statistics*—For each priority class except Priority 1, there is a histogram of queue size and waiting time. Also, the average and variance of waiting times for all jobs and for those jobs that were delayed.
c. *Response Times*—For each priority class, the histogram of travel times and the total number of responses.
d. *Sector Summary*—For each priority class, the workload (number of jobs) and average and variance of response times.
e. *Car Availability*—A histogram of the number of cars on patrol.
f. *Number of Cars Sent to Calls*—For Priority 1 and 2 calls, the number of times they received one car, two cars, etc. This indicates how well the system responds to high-priority jobs.

**Data Base Requirements.** The major requirements are the x-y coordinate set for all blocks and the job input file. The job input file can be constructed from historical records or can be generated by a presimulation load module which duplicates historical experience. The latter program has not been written but would be relatively easy to construct.

**Cost and Computer Requirements.** This program is written in SIMSCRIPT II.5, and a SIMSCRIPT II.5 compiler must be available. The program requires 160K bytes core storage for compilation and 110K bytes core storage to run. A run of 2,500 jobs costs about $10. The largest problem run contained over 600 blocks, 10 sectors, and 21 cars. With about 10,000 jobs, the run took 50 seconds CPU time on an IBM 360 Model 85 computer.

**Validation and Verification.** This model has probably been run more often than other simulation models, and it is the only one to have had a formal validation effort. It was initially used on data from the 71 Precinct in New York to investigate changes in car levels [113]. Later, it was used to investigate the combining of the 40 and 42 Precincts into a single precinct [48]. In this effort, five runs were made over 10,000 jobs each and 10 runs with about 5,000 jobs each.

Another effort was the use of the special closest-car dispatching version to investigate the benefits of an automatic vehicle locator system in three New York precincts. This is reported in Ref. 109.

In 1974, a validation effort was carried out by The New York City-Rand Institute. Two complete 8-hour tours were observed in the 26 Precinct. The results of the simulation were compared with actual observations and the correspondence was found satisfactory. Some discrepancies were found in the correspondence of out-of-service times, but this was because many actual out-of-service times were not reported to the New York City central SPRINT record. Job input information for the



simulation, which is taken from the SPRINT record, thus lacked the proper amount of out-of-service time. When observed out-of-service times were added to the job input record, the correspondence of the model with the actual system was excellent. The validation effort is reported in Ref. 49.

**Implementation and Impact.** The model has been turned over to the New York Police Department and the Seattle Police Department, both of which have operated it successfully. It has not been used in New York for continuing analysis of deployment options. As mentioned in the preceding section, The New York City-Rand Institute had been carrying out most of this work under their contracts with the department. The Institute was closed in 1975, and it remains to be seen whether the New York Police Department will continue using the model. For this to happen, they must have a staff member who understands the model and can modify it to analyze specific options. Currently, although one or two department members can run the model, they are not assigned to do so.

**Limitations of the Model.** One limitation of the model is that it has no provision for dispatcher delay.

It can only simulate one precinct at a time, which is a limitation as compared with the Urban Sciences model (Model 3). Another drawback of the model is that whenever a Priority 1 job cannot be assigned immediately, it is discarded by the simulation. This rule was created because in New York when a job can't be handled by the precinct cars, cars from another precinct or special cars answer the job. However, when simulating situations with small numbers of cars, the proportion of calls discarded becomes high, and the model does not present a true picture of the workload and patrol availability.

It may be a limitation that the x-y coordinates of each block in the simulation must be given by the user. This depends on how many blocks are to be used and if the user has access to a coordinate file. In New York a planning department file exists, but over 10 percent of the coordinates are incorrect or missing. It requires a lot of effort to correct this. However, the alternative of creating areas with uniform spatial job distributions (as in the Boston model) may be equally difficult.

For departments without a computer file of past jobs, the fact that jobs are not generated internally is a drawback.

There is no direct method for changing the number of cars on patrol over time. This must be done with the out-of-service job input. For instance, if after the first tour of 8 hours two cars are to be removed for the entire next tour, then an out-of-service interval of 8 hours must be inserted for two cars. This technique does not imitate whatever changes in patrol sectors would occur in the real world if the number of cars changed.

The model can simulate only one type of patrol unit and cannot be used to evaluate cars plus scooters or other vehicles.

**Transferability of the Model.** The model can be transferred easily. It requires that the adopting department have a systems analyst familiar with the SIMSCRIPT language and a SIMSCRIPT II.5 compiler. The program is written as a set of modular subroutines, making it easy to change the structure of one aspect of the model, such as the dispatching routine, without disturbing the rest of the program.

**Documentation.** The program is well documented. A brief description of the model is given in an executive summary [114], and a complete program description and documentation is available in Ref. 115. The latter report includes a complete program listing and examples of input and output records.

## BEAT DESIGN MODELS

### Background

Patrol beats are geographical areas to which patrol cars are assigned. Ordinarily, the assigned patrol car is the dispatcher's first choice to respond to calls for service in its beat, and it carries out preventive patrol there when not otherwise occupied. Two models have been built to help police departments design patrol beats. One, called the Hypercube Queuing Model [30,121,123,126,127], was developed by Richard Larson at the Massachusetts Institute of Technology. The other is part of the Superbeat models and was developed by Deepak Bammi [5,6,7,8,170]. We discuss them together for the purpose of clarifying the differences between them.

### Policy Issues Addressed

As indicated by the section heading, the primary use of these models is for designing beats, that is, drawing on a map the boundaries of each patrol car's assigned area. However, they can also be used to analyze other geographical details of patrol car operations, such as the relative amount of time cars spend on preventive patrol in various parts of their beat, and aspects of dispatching strategy, such as methods for choosing an alternate car to dispatch when the car assigned to an incident's beat is unavailable.

For beat design purposes, these models help planners identify designs that accomplish one or more of the following objectives:

- Balancing workloads among units
- Equalizing response times among different parts of a command
- Minimizing average response time for the entire command
- Minimizing the extent to which patrol units are dispatched outside their assigned beats.

In general, it is impossible to achieve all of these objectives simultaneously, so the models also assist in finding acceptable compromises.

The models will permit analysis of designs in which beats overlap, as well as traditional nonoverlapping designs. This capability is particularly important to departments that wish to minimize or reduce the extent of out-of-beat dispatching. With nonoverlapping beats, it is inevitable that a substantial fraction of dispatches will take units outside their assigned beats. In fact, if patrol units are assigned to nonoverlapping beats and are busy (on cfs work or non-cfs work) about 60 percent of the time, then typically somewhere between 50 and 70 percent of calls for service will be handled by a unit other than the local beat unit. In these circumstances, it is extremely difficult for the patrol officer to establish a "neighborhood identity."

However, with overlapping beats the extent of out-of-beat dispatching can be substantially reduced. Many departments have recently introduced "team policing" or other allocation plans in which several units share responsibility for an area that is larger than a traditional patrol beat. These plans constitute various forms of overlapping beats, and the areas of responsibility for each team can be designed using beat design models.

In the absence of a mathematical model, most departments design patrol beats in such a way that they are "reasonably" shaped, lie wholly on one side of any



natural barriers to travel that may exist (limited-access highways, railroads, rivers, and the like), and correspond as closely as possible to recognizable "neighborhoods" (in the sense of commonality of spoken language, demographic characteristics, and land use). In addition, planners usually attempt to equalize the numbers of calls for service that can be expected in each beat. However, most other measures of performance are too difficult to estimate by looking at a map, so they are not considered in beat design.

When using a mathematical model, the planner must still be familiar with barriers to travel, "neighborhoods," main streets, etc., but the model provides him with detailed information about performance measures. This information permits the planner to identify the failings of any proposed beat design, and leads him to construct a sequence of improvements, ultimately resulting in an acceptable design. When calculating the cfs workload of units, a model takes out-of-beat dispatching into account, and thereby gives much better estimates than can be obtained by counting the number of calls for service in each beat.

The model "understands" that the burden of out-of-beat dispatches falls more heavily on some units than on others, depending on their locations. For example, it is apparent that a patrol unit whose beat is in the center of a command will be the dispatcher's second choice (after the local beat unit) for more locations than will a unit whose beat is on the boundary. If the beat in the center and the beat on the boundary both have the same call rate, the workload of the centrally located unit will nonetheless be higher, because it will have more out-of-beat dispatches.

### Structure and Output of the Models

Ordinarily, one precinct is studied at a time. The precinct must be divided into small "reporting areas," which are approximately the size of a few city blocks. Beats may be designed in any way desired as collections of (usually adjacent) reporting areas, and they may overlap partially or fully. The models solve equations that determine the steady-state probability that any particular collection of patrol cars is unavailable to be dispatched, while the remaining ones are available. Therefore, they are analytic models, not simulations.

A key difference between the models is the way in which they perform these calculations. Larson's model actually includes two different procedures, and the user selects whichever one he wants to use. One is called the *exact* hypercube model [121]; it assumes that all unavailabilities arise from calls for service, which are assumed to have an exponentially distributed service time. It then iteratively solves a complicated system of queuing equations to find the required probabilities. The second is called the *approximate* hypercube model [123]; it solves a less complicated system of equations to give results that are very close to those of the exact model. Bammi's model assumes that cars are unavailable for non-cfs work as well as call-for-service work, and that the unavailability of each car is independent of the unavailabilities of other cars. Neither Larson's assumptions nor Bammi's are exactly correct; they both yield approximations to the real performance of patrol cars, but different approximations.

Once the required probabilities have been determined, the workload of each patrol car can be determined and provided as output. Average travel times to incidents can also be calculated by knowing how long it takes to travel from one reporting area to another, along with the previously determined probabilities that the beat car is available, or the first alternate car is available (but not the beat car), or the second alternate is available (but not the first two), and so on. To calculate the travel times between reporting areas, the hypercube model permits two choices. The user can specify the travel speed of the cars and the coordinates of the center of each reporting area, in which case the program estimates travel times, or the user can calculate some or all of the travel times by some other means and provide them as input. Bammi's program permits the user to specify travel distances and travel speeds between *adjacent* reporting areas only, and then the program calculates all the travel times, a convenience to the user.

Bammi's program includes a heuristic optimization procedure for finding a beat design that minimizes the average travel time in the precinct. (*Heuristic* means that the design is not guaranteed to yield the true minimum.) Larson's program has no such feature, but later versions of the program, now partially documented [37], generate beat designs that minimize workload *imbalance* among patrol cars or minimize travel-time *imbalance* among beats. Here we see a difference of opinion among model builders as to what constitutes a "good" beat design.

### Data Base Required

The essential data for each model consists of information needed to describe the location of each reporting area (or its distance from other reporting areas), the reporting areas that constitute each car's beat, the number of calls for service in each reporting area, and service times. Optional data include the relative amount of time each car spends patrolling each reporting area, and the order in which the dispatcher favors each car to respond into a given area. (If these data are not provided, the program makes certain default assumptions, which vary between the models and according to other input.) Bammi's program permits calls to be divided into two priority levels.

### Cost and Computer Requirements

Larson's Hypercube Model is written in PL/I and is generally available as a batch program. An interactive monitor has been designed for the model [187], and it has been successfully operated on the IBM system at the Massachusetts Institute of Technology and on several other systems. The amount of core storage space and execution time varies with the number of reporting areas and the number of cars. Substantially more space is required for the exact hypercube model than for the approximate model, and the exact model cannot be operated with more than 15 cars. At this limit, the exact model can cost as much as $100 for one run. However, typical runs of the approximate model are claimed [126] to cost under $5 and require less than 300K bytes of core. Most departments would use the exact model only once or twice to check the accuracy of the approximate model, and then use only the less expensive model.

Bammi's model is written in FORTRAN H and has been operated on both an IBM 370/155 computer and a Univac 1108. Bammi [8] describes an example that required 250K bytes of core storage and required 1384 seconds of computer time. The cost is not specified but typically might be around $200 for a job this size. This cannot



be directly compared with Larson's figures, since each optimization run of Bammi's program is equivalent to several iterations of Larson's. However, the computer costs of Bammi's program appear large enough to be considered as a possible factor in choosing between the models.

### Validation and Verification

Both models have been carefully verified. Bammi's has been validated in a limited way by comparing the output with the results of a simulation; an attempt to validate using real data from Aurora, Illinois, encountered problems due to a change in the data collection system between 1972 and 1973. Larson's model has been validated against real data in New Haven, Connecticut [38]. The two models have never been compared with the same input to determine the extent of differences between them.

### Implementation and Impact

Larson's hypercube model was implemented in Boston [122], Quincy [154], and Arlington, Massachusetts [99], and Bammi's beat design model in Aurora, Illinois [5], with redesigned patrol beats being accepted in each instance. Both models are currently being implemented in additional cities.

### Limitations

One car is assumed to respond to each call for service. The calculations are steady-state and therefore cannot reflect changes in call rates or other characteristics over time. The number of patrol cars must be fixed in each run of the program. The capabilities to handle priorities are severely limited.

### Transferability

Larson's model has been shown to be easily transferred to a variety of computer systems. Bammi's does not appear to present any problems in this regard, and the designer is willing to provide assistance to interested users.

### Documentation

Larson's model is extremely well documented in two conceptual papers [121, 123], an executive summary [30], a user's manual [126], and a program listing [127]. Bammi's model is well documented conceptually and has a program listing [5,6,7,8, 170,171], but there is no user's manual.

## DYNAMIC QUEUING MODEL FOR PATROL SYSTEMS

### History

This model was developed in 1973 by researchers at The New York City-Rand Institute under a contract with the Police Department of New York City. The original model was purely descriptive.

In 1974, the model was modified to run in a prescriptive mode. This option provides for the generation of the minimum hourly car requirements that provide a specified service level.

**Policy Issues Addressed by the Model**

The model calculates the dynamic characteristics over time of a patrol system that has a time-varying but cyclic rate of calls for service and changing numbers of patrol cars on duty at different times.

In its descriptive mode, for a given call-rate pattern, the model shows the user the dynamic effects of:

- Changes in tour starting times
- Changes in the number of cars assigned to each tour
- Changes in the pattern of scheduling patrol car meal hours.

The descriptive model can be used to evaluate any proposed schedule of patrol system operation. The key issue this model addresses that other models do not is the time-varying nature of the patrol system.

It can answer such questions as:

- Are there periods of the day for which the service level, say the probability that a call for service must wait for car assignment, is unusually high?
- Can patrol effort be allocated in a way that matches more closely the pattern of demand for service?

In its prescriptive mode, the model tells the user the minimum number of cars required hour by hour to insure that the probability that a call for service must wait for a car assignment never exceeds a specified level.

**Structure of the Model**

The model is a set of equations that are solved numerically to provide, for each point in time, the probability distribution of the number of calls for service that are in the system, from which all queuing statistics can be generated.

The basic assumptions of both the prescriptive and descriptive models are:

- Each call for service is assigned exactly one car.
- Calls that are not serviced immediately are placed in a queue and are then assigned to cars in the order of their arrival.
- There is no job priority structure.
- The time between arrivals of calls for service is exponentially distributed with a time-varying mean.
- Service times are exponential with a constant mean.

The prescriptive model assumes that:

- There is a desired service level that should be met at all times of the day. Service level is defined to be the probability that a call for service must wait for car assignment.
- The number of cars can be changed at the start of each hour.



- There is no limit on the number of cars available for assignment to the patrol system.

**Data Base Requirement**

The only data required are the mean service time (a constant), the arrival rate of calls for service during each hour of the period to be examined, and the number of cars to be assigned to the patrol system in each hour. The last data are not needed for the prescriptive mode, which requires only a desired service level.

**Output of the Model**

The model output is a time trace of the following performance measures that, if desired, can be plotted on a graph:

- The expected queue size
- The expected waiting time
- The probability that all cars are busy
- The probability distribution of the number of jobs in the system.

In addition to these time-dependent values, statistics are calculated summarizing the entire period of the run. They are:

- The probability that a random call must wait for car assignment
- The probability that all cars are busy at a random time
- The expected total patrol time
- The expected number of cars on patrol at a random time
- The expected number of cars on patrol at the time of a random call.

**Cost and Computer Requirements**

The program is written in the Continuous System Modeling Program Language, CSMP, an IBM program product designed especially for the numerical integration of differential equations. A run imitating 48 hours of activity required 102K bytes of storage and 15.8 CPU seconds on an IBM 360/85.

**Validation and Verification**

The program has internal checks for convergence of solutions.

Validation of this model has not been attempted. To do so would require that one observe a real system for a period of time sufficient to accumulate enough data to form estimates of the true probability distributions of the number of calls for service in the system at *each point in time.* It would be impossible to find a real system that had a constant demand pattern for a long enough period.

**Implementation and Impact**

The descriptive model was used in 1973 to demonstrate the benefits of a reallocation of patrol effort and the addition of a fourth tour for one precinct in Brooklyn, New York [113]. In another report [112], the model is discussed in conjunction with the linear programming model described in the next section.

In 1974, the prescriptive mode was used to generate minimal hourly car requirements for 24 high-demand precincts for the New York Police Department.

In an unpublished study, the descriptive model was used to evaluate schedules for ambulance drivers in Washington, D.C.

Impact on the New York Police Department is not yet known. They do not have easy access to a CSMP computer compiler program, and all analysis has been performed at The New York City-Rand Institute. The planning and analysis staff of the department are interested in using the model, as evidenced by their requests for the prescriptive model runs mentioned above.

### Limitations of the Model

Some of the limitations are:

- Only one car is assigned to each job.
- No geographical or car-by-car workload information is provided. In particular, travel times and preventive patrol frequencies, which are calculated by recent patrol car allocation models, are not provided by this model.
- Mean service time is constant over time.
- In the mathematical structure of the model it is necessary to assume that when a reduction in the number of cars is made, work on jobs that were being serviced by the removed cars is discontinued. They remain in the system and will be reassigned at a later time. This is not a major limitation, for this situation is seldom encountered.

### Transferability of the Model

The model is easy to transfer. Data requirements are minimal. The user must have access to a CSMP compiler and should have sufficient knowledge of systems analysis to be able to interpret and use the model.

### Documentation

The descriptive model has been documented along with a program listing [113]. Documentation of the prescriptive model is not available at this time.

## A LINEAR PROGRAMMING MODEL FOR SCHEDULING PATROL CARS

### History

In 1974, The New York City-Rand Institute developed this model under a contract with the New York City Police Department.

### Policy Issues Addressed by the Model

The major policy issues this model addresses are:



- What are the best times of day to schedule tours? Should there be overlapping tours?
- What is the effect on total number of cars required to achieve a given performance level, as the possible meal break hours are changed?
- How does the number of required car-tours vary with different service levels?

### Structure of the Model

Basic assumptions of the model are:

- Each tour lasts eight hours.
- Each car can take a one-hour meal break, starting at the beginning of any hour, during its tour.
- Tours begin (and end) on the hour.

The user must specify a set of hourly car requirements, i.e., the smallest number of cars that will be permitted to be on duty during each hour of the planning period.

The user must also specify the set of possible tour starting times, i.e., hours at which tours are permitted to begin, and a set of constraints on the hours of the tour in which meal breaks may be taken. The latter are specified in the form of intervals, such as the scheduling of meals at any hour from the third through the fifth hour of the tour.

The model is a linear programming model with integer constraints on the number of cars scheduled for duty during each tour and the number of meal breaks taken during any hour. The program finds the number of cars to be assigned to duty during each tour and their meal break assignments so that the hourly car requirements are met using the minimum number of car-tour assignments (the sum of cars over all tours).

The program can be used to produce schedules with up to 168 hours, the number of hours in a week.

### Data Base Required

The user inputs are specified above. The only question remaining is: How does one determine the hourly car requirements? One way that insures a relatively constant service level is to use steady-state results to calculate the smallest number of cars for each hour that meets specified performance standards. This can be accomplished by operating a patrol car allocation model, treating every hour as if it were a tour. Another method is to run the Dynamic Queuing Model (described above) in its prescriptive mode. This will generate hourly car requirements for any service level, but only queuing standards will be met.

### Output

The output of the program is:

- The number of cars assigned to each tour
- The number of cars from each tour that take a meal break at each hour of the tour
- The total number of car-tours required.

The output of the model does not show the actual dynamic characteristics of the schedule it generates, and thus the user does not know how closely actual service levels will match the desired service levels used to determine the hourly car requirements. This can be checked using the Dynamic Queuing Model described above. If the actual service level is not satisfactory, changes in the hourly car requirements can be made and the scheduling model can be run again. Repeating this process will lead to a minimal car schedule with desired service level characteristics.

In example schedules produced for New York City precincts, it was found that generated schedules had service characteristics very close to those desired.

### Cost and Computer Requirements

The program is divided into two routines. The first is a FORTRAN program that takes as input the tour starting times and possible meal breaks. It generates the matrix of coefficients required for the linear program. The second routine solves the problem. It uses the IBM program product, Mathematical Programming System-Extended (MPSX), with the mixed integer programming feature (MIP).

A program to find a schedule for a 24-hour period that allowed tours to start at any of the 24 hours and meal breaks to be taken during any hour of a tour required 200K bytes storage and 10.4 seconds CPU time on an IBM 370 model 85.

### Validation and Verification

The program has been completely debugged. Validation is not an issue with this type of prescriptive model, since it does not generate any performance measures that can be compared with the real world.

### Implementation and Impact

The model has been used to study the effects of adding new tours and changing the meal break hours in New York City precincts. The Police Department has not yet adopted the additional tours and more flexible meal hours indicated by the results, primarily because such changes in New York must be negotiated as part of the labor contract with the Patrolmen's Benevolent Association.

### Limitations of the Model

One limitation is that the model reflects the service level of the patrol system only through the hourly car requirements. However, use of a patrol car allocation model or the prescriptive mode of the Dynamic Queuing Model partially overcomes this limitation.

Another limitation is that the model is not constrained by available resource levels. The number of cars it recommends for tours may exceed the number of cars the precinct actually has available. In this case the service level must be reduced and the model run again to obtain schedules that fall within car availability. The model cannot be used, except in a heuristic way, to determine the best allocation of a given number of cars over precincts or over tours.



### Transferability

Successful transfer of this model requires access to suitable compilers and appropriately trained personnel who can understand, explain, and run the program, measure system workloads, and determine the hourly car requirements.

### Documentation

A program listing is available from The Rand Corporation. A report describing the interactive use of this model and the Dynamic Queuing Model has been written [112], and another report describes the results of an extensive investigation of tour start times and meal break hours [111].

## MANPOWER SCHEDULING MODELS

### Background

The two models discussed here are designed to determine working schedules for police personnel. The first is restricted to determining schedules for patrol personnel while the second considers the problem in a more general framework. It can be used to schedule any group of personnel, from small units to entire forces.

The first model is a result of a 1973 study undertaken for the Illinois Law Enforcement Commission by the Illinois Institute of Technology using Aurora, Illinois, as a prototype city. It is part of a set of models referred to as Superbeat [170].

The second model was developed as a Ph.D. dissertation by Nelson Heller, who later developed a set of computer programs and used them to develop schedules for personnel of the St. Louis Police Department [83,84].

### Policy Issues

The major issues these models address are:

- The allocation of available manpower in the manner that best matches workload requirements over different tours and days of the week
- The working schedules of police personnel.

The second model in particular focuses on the determination of schedules that provide good patterns of recreation days, which are the nonworking days falling between working days. In particular, it may be possible to obtain more desirable recreation patterns without changing tour coverage or the total number of recreation days.

### The Superbeat Model

**Structure of the Model.** Superbeat is an optimization model. It derives a schedule for patrol forces that minimizes average response time over all tours of a week, given a fixed number of available personnel and a specified set of work and recreation patterns. Three tours of 8 hours each are used.

The structure of the model fits the work and recreation patterns of the Aurora, Illinois, Police Department and is adaptable to the existing patterns in many other departments. Patrol personnel work on a recreation pattern that is the same each week. For example, each officer could work five days a week with two consecutive vacation days—Sunday-Monday, Monday-Tuesday, etc. Some officers rotate shifts, i.e., they work a weekly pattern for one month on one tour and then shift to another tour. The condition that every tour change takes place monthly means that the number of *rotating* officers assigned to each tour must be the same for each tour. Other officers work fixed tours.

The variables of the program are the number of officers assigned to work fixed and rotating tours and their specific recreation day patterns.

The objective function in the program is the average call-for-service response time (queue wait plus travel time) over all tours of a week. For a given assignment of men to work schedules, the program estimates these average response times using a combination of queuing theory and travel-time approximations. Response time to high-priority calls can be weighted more heavily than response to low-priority calls in the objective function, or the user can substitute some other objective function that can be calculated for each tour and then averaged.

**Data Base Required.** The following information is required to run the model:

- Allowable patterns of work days and vacation days
- The total number of officers available for fixed and rotating schedules
- The dimensions of the city in miles
- The arrival rates and average service times of high- and low-priority calls (citywide) for each tour
- Unavailability parameters permitting an estimate of the amount of non-cfs work (called downtime) in each tour
- Response speed of patrol cars for each tour.

**Output of the Model.** The output is the number of officers to be assigned to fixed and rotating tours for each of the recreation patterns.

**Cost and Computer Requirements.** The program has five routines. Four are written in FORTRAN V. The fifth, the optimization model, is in UMPIRE, an optimization package available through the Computer Science Corporation. No cost or computer requirements are specified in the available documentation.

**Validation and Verification.** No comparison of the average response time for current schedules to the response time predicted by the model has been made. The programs have been debugged.

**Implementation and Impact.** The model was used to develop patrol force schedules for Aurora, Illinois. These schedules were very close to the schedules then in operation; slight schedule changes were made starting on May 1, 1973.

**Limitations of the Model.** The particular recreation and rotation patterns in this model, while matching the *existing* patterns in many departments, do not permit the user to consider any novel schedules that might be substantially better. (See the following model.)

Another aspect of the model that may be a limitation is that the calculation of response time requires so many assumptions and simplifications that it may be inaccurate. This can only be determined by a careful validation.

An obvious limitation is that the model only schedules patrol forces. In many departments, work schedules of patrol forces cannot be separated from schedules of other personnel.



**Transferability of the Model.** The model could be transferred easily. The adopting department would need a systems analyst to supervise the running of the programs and access to a FORTRAN V computer facility and the UMPIRE optimization model. Other optimization programs could be used with program modifications.

**Documentation.** The computer programs are documented in Ref. 171.

### The St. Louis Scheduling Model

**Structure of the Model.** This is a general scheduling model with three basic parts. The first part takes a set of workload measures for each hour of the week and uses a quadratic programming algorithm to determine the best matching of available eight-hour tour assignments to the workload pattern. Workload values need not be absolute measures of actual work. They may be simple relative measures of the work in different hours of the week. In performing the allocation, the program accounts for user-supplied constraints on specific tour manning levels.

The second part of the program takes the assignment of men to tours from the first part and for each tour develops a set of alternative work schedules that all meet the tour allocation requirements. Each of the alternative work schedules provides the same manpower coverage; the only differences among them are the arrangements of recreation days.

The third part of the program merges the tour-by-tour schedules of the second part into a set of full schedules. The schedules are called proportional rotating schedules—*proportional* because the number of men working on a given tour is proportional to that tour's workload requirement, and *rotating* because at the end of each week, each rotating man moves to the next week of the schedule (not necessarily changing tours). Fixed tours are also allowed. This model differs from the Superbeat model described above in that it *generates* schedules (rather than simply allocating officers to schedules supplied as input) and considers a much wider range of possibilities for designing schedules.

The number of weeks in the full schedule cycle is equal to the number of men (or squads if men are grouped into equal-sized units) that are scheduled.

**Data Base Requirements.** The following is a list of the inputs to the program:

- The number of tours and their starting times. At most, five tours are allowed in each day.
- The total number of men to be scheduled.
- The number of recreation days per man per year. An average of two recreation days per week are built into the program, but additional paid holidays can be specified.
- A measure of the workload for each hour of the week (168 hours).
- Upper and lower bounds on the manning levels desired for each tour of the week.
- Minimum and maximum number of consecutive days for recreation periods.
- Minimum and maximum number of consecutive days for work periods.
- Minimum and maximum number of recreation days at the beginning and end of each tour.
- Maximum number of consecutive working weekends to be allowed.

- • The length of a special minivacation period.
- Number of desired schedules.

**Output of the Program.** The program provides the following output:

- The optimal allocation of men to tours.
- A percentage comparison of workload requirements and tour manning levels.
- A summary of the number of men on duty and off duty for each tour.
- A set of schedules, ranked according to the number of recreation weekends, maximum number of consecutive working weekends, length of longest work period, number of maximum length work periods, and other schedule attributes.

**Cost and Computer Requirements.** The computer program is written in FORTRAN IV and is available for a modest copying and postage charge from the Law Enforcement Assistance Administration.

In one application, the use of the second part of the program to prepare five sets of tour schedules required 102K bytes core storage and about 9 CPU seconds on an IBM 360 Model 65. Typical complete schedule designs are claimed [85] to require a total of 6 to 12 minutes of computer time.

**Validation and Verification.** The program has been partially debugged by running both real and test problems. Since the program is still being used by the designer, updates are made to correct any errors brought to his attention.

**Implementation and Impact.** This model has been used to develop schedules for four units of the St. Louis Police Department, the Evidence Technician Unit and three Traffic Safety Units. The schedules developed were more acceptable to personnel than previous schedules [83]. The model was also used to schedule about 45 officers in one precinct of the St. Louis County Police Department.

The program was also used by The New York City-Rand Institute to design work schedules for units of the New York City Police Department. These include:

- An eight-man computer supervisory unit
- Some four- and five-man transport driver crews
- The entire field services force.

At the time of the study, the field services force of the New York Department had over 18,000 men. Acceptance of a new work schedule for these men is a question that is subject to labor negotiations between the city and the patrolman's union, the PBA. If the issues of the number of days off per year and pay levels can be resolved, there is a possibility that one of the computer-generated schedules will be adopted.

**Limitations of the Model.** The computer program limits the user in the following ways:

- No more than six different tour changes may take place in one complete cycle of the schedule. This may be a problem if (as in New York) the officers are required to change tours every week.
- The second part of the program can only consider five tours at one time. If six are desired, the program must be run twice and the outputs merged for input to the third part of the program.



- If the user is not careful, the input he prepares may lead to excessive running times. This occurs especially with tours that have a large number of weeks, say over 10.
- Changes from one tour to another are allowed only after a recreation day. (This is a desirable feature for a new schedule, but it may not match current departmental practices.)
- Since the model is in three parts, the user must insure that output from one part is prepared correctly for input to the next part.

**Documentation.** The program is documented by an executive summary [85], a conceptual description [83], and a complete program user's manual [84].

# Chapter 5

# COURT MODELS

## INTRODUCTION

Most court models have been designed to aid administrators in managing the system. They permit examination of policy options such as moving personnel (e.g., judges), from one function to another or reorganizing the sequence of steps taken by a typical defendant in passing through the courts. In general, the objectives of such changes are to increase the number of defendants processed per unit of resources, or to decrease the delay times experienced by defendants.

The problem of data collection, present for all types of models, is of particular interest for court models because of the primitive nature of most court information systems. It is entirely possible that if an administrator were made aware of the processing delays for various types of defendants or in various sections of his court system, he would know what remedies to take without the need for a model. To illuminate this issue, we discuss in this chapter not only two models, but also an information system.

## THE CANCOURT MODEL

### Historical Background

The planning for the study leading to the development of the CANCOURT model began in 1968. At that time "The application of systems analysis research to what is compendiously described as the criminal justice system was almost a totally neglected subject" [72]. Early debate in the Centre of Criminology concentrated upon the choice between analyzing the system in its entirety and restricting the first study to a particular segment of the total operation. The final decision favored the more intensive examination of a single, cohesive, administrative system—the criminal courts. The Canada Council was persuaded to provide financial support for a feasibility study that would test a systems analysis approach to the operations of criminal courts. Joint financial backing was made possible by The Ford Foundation, a consistent supporter of the Centre's efforts to apply multidisciplinary research to issues in criminology.

The project was identified as the "Court Section" of the "Economics of Crime" project. A first purpose was to develop a theoretical framework within which one could start to develop operational definitions of the goals of the criminal justice system in order to apply the techniques of planning/programming/budgeting systems (PPBS). A second and more important objective was to identify factors that significantly affect defense and prosecution activities and behavior in the court subsystem. This pointed the project in the direction of a model rather than a PPBS. The research places a high weight on an operational approach, emphasizing problems facing people involved in the system either as defendants, defense lawyers,





court administrators, Crown attorneys (prosecutors), or judges. The analysis is presented in a form intended to lead eventually to operationally feasible solutions to those problems.

The project team, under the leadership of Robert G. Hann, was evidently aware of the need for a systems approach rather than "isolated in-depth analyses of particular problems." The team was also aware of past systems work, especially work that grew out of the U.S. President's Commission on Law Enforcement and the Administration of Justice—specifically by Navarro and Taylor [179] and Blumstein and Larson [19]. This work "formally set the stage for the application of a set of research tools designed for a type of analysis that was the antithesis of piecemeal research and instead was geared to looking at the . . . criminal justice system as . . . a system." Previous attempts at a systems perspective were still faced with the problem of "not having available models powerful enough to assimilate and analyze the data collected within an integrated systems context."

### Policy Issues Addressed

This model permits the examination of many policy issues related to court structures and procedures as they affect the flow of offenders and the resources required. Some key issues: Where should a given type of case be tried? When should there be preliminary inquiries? Are more judges really needed? How many appearances should be allowed? What scheduling policies should be implemented? What changes should be made in the Criminal Procedure Law (for example, in the areas of bail and narcotics)?

### Structure of Model

The structure of the Canadian system of criminal courts is represented by Fig. 4. Each box refers to a major activity that could be carried out in dealing with a case, starting at the top of the diagram and moving down by one of the possible routes. Six major types of activities are represented: apprehension, sorting, preliminary inquiries, grand jury hearings, trials, and appeals. "Sorting" refers to those activities that are carried out in the courtroom prior to an accused giving his "intent to plea" or actual plea. A case may experience from two (apprehension and sorting) to all six of these activities.

CANCOURT is a case-by-case simulation. Such a model could have been designed to allocate separate sections of computer memory for each of the activities in Fig. 4 and for each of the courts. It was deemed more efficient, however, to develop a model that took advantage of the similarities of activities in the various courts. Therefore it was decided to build a model with different main modules for each of the main categories of activities carried out within the system. The basic modular construction of CANCOURT is shown in Fig. 5. This is a simplification for two reasons: each module may represent a large number of separate activities in various courts and many of the possible routes between modules have been omitted for simplicity.

The modules are described next in the order in which they might be used in the model. Module names are in capitals.

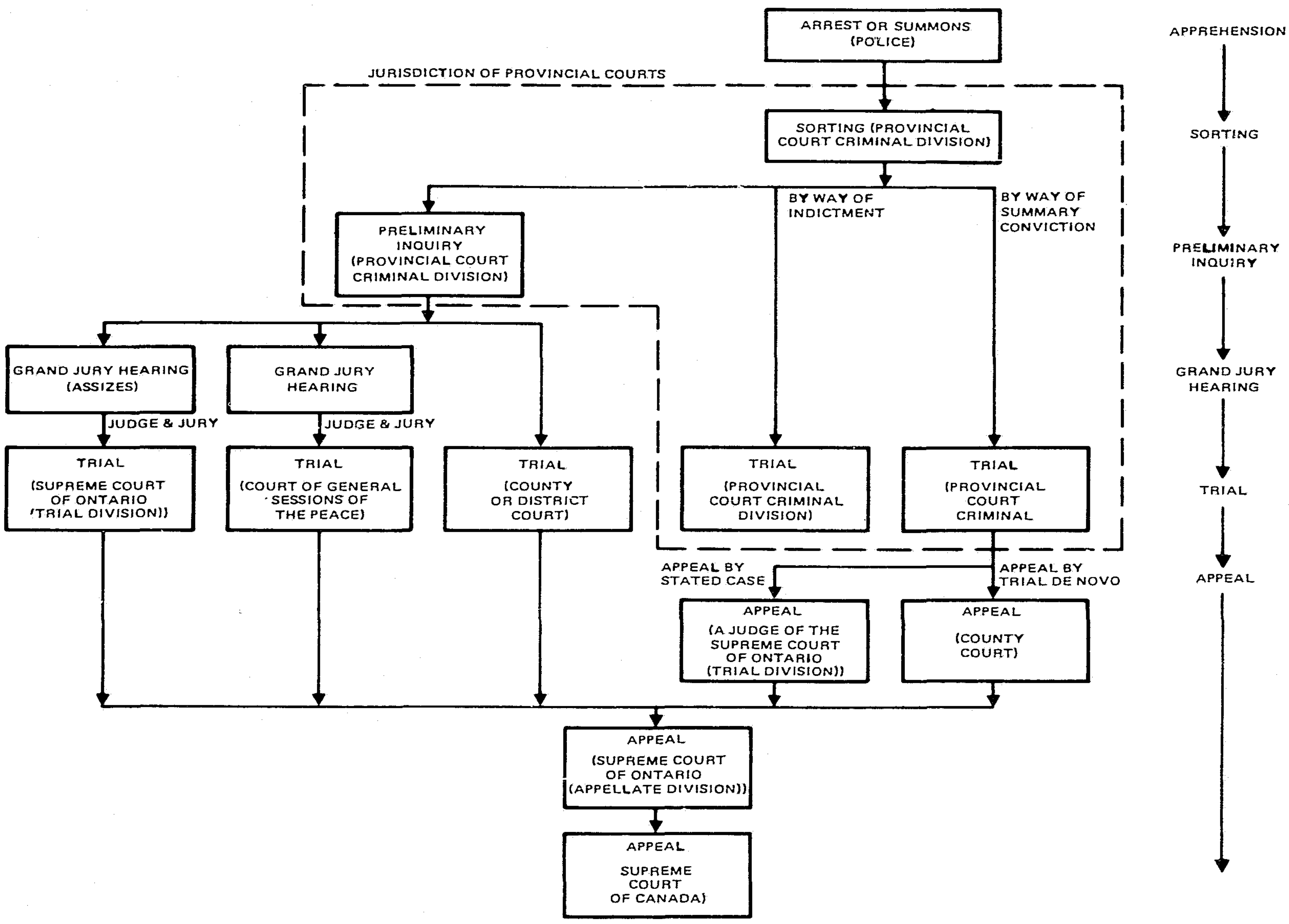


Fig. 4—Flow diagram of the Canadian system of criminal courts



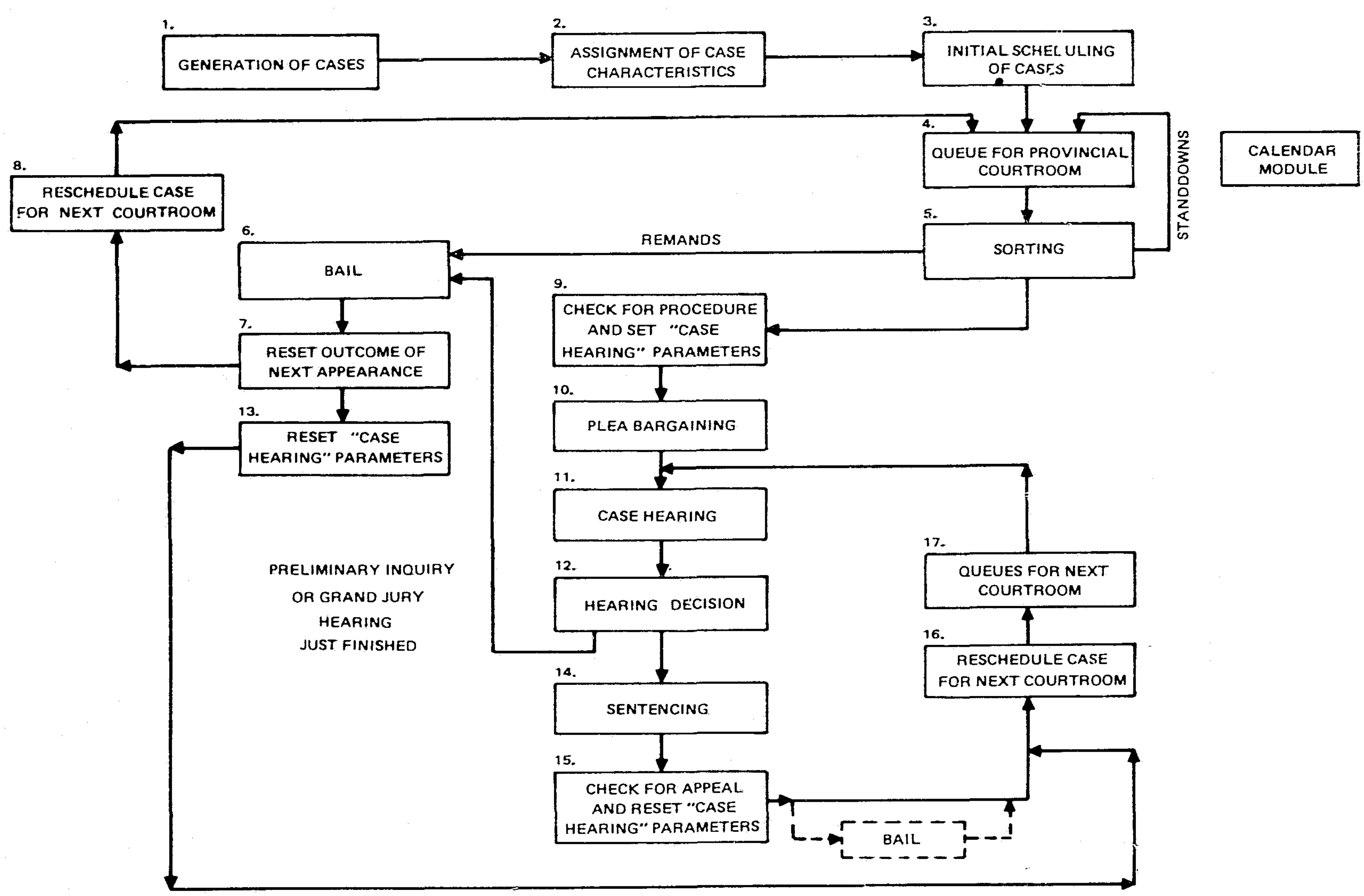


Fig. 5—Summary internal modular configuration of the CANCOURT model

**GENERATION OF CASES Module**

This module creates and simulates the entry of new cases into the model. By varying the rate of case creation, one could use the model to estimate the effects of different crime rates, reporting and charging practices, and police efficiency in solving crimes. At present the module generates cases separately for each of eight case offense types: narcotics, liquor, municipal by-laws, traffic, criminal code cases that will be tried by summary conviction, criminal code cases that will be tried by indictment, and criminal code traffic offenses.

**ASSIGNMENT OF CASE CHARACTERISTICS Module**

After a case has been generated and assigned a case offense type, this module then assigns, on the basis of probabilities calculated from data collected and analyzed previously, the following characteristics to each case: the number of accused; the number of counts; whether the accused are arrested or summonsed and in custody or not at first appearance; the sex of the key accused in the case.

This module also assigns to each case, again based on historical data, parameters determining how the case will be processed at later stages, including: the actual procedure that the case will follow if it goes on to trial; the maximum number of appearances; the outcome of the first appearance in Provincial Court; the court decision that will be handed down at the first case hearing; and whether the decision will be appealed or not. The passage of time for defense lawyers and Crown attorneys to prepare for the first appearance is also simulated.

**INITIAL SCHEDULING OF CASES Module**

This module assigns cases to one of the available Provincial courtrooms for their first appearance in the court system. Cases are assigned to one of 17 courtrooms in the proportions observed in a sample of cases on the basis of sex of the accused and the type of crime.

**QUEUES FOR PROVINCIAL COURTROOMS Module**

After a case is assigned to a courtroom of the Provincial Courts, it enters a separate queue for that courtroom and then waits until the courtroom is empty. Cases are queued within priority classes. Priorities are assigned according to whether the case is a new case, a case previously remanded, a case that has been "stood down" for processing later on in the same court day (or first thing the next day), or a case transferred from another court for processing the same day. Stood-down cases get the highest priority, new cases get next priority, and remanded cases get the lowest priority.

**SORTING Module**

When all cases before it in the queue have been processed, the case is simulated as "seizing" a Provincial courtroom in the SORTING module. SORTING activities include: standing down a case for later appearance the same day; transferring a case to another court for same-day appearance; remanding a case if the accused does not show up; issuing a bench warrant or reissuing a summons; remands for various other purposes.



**BAIL Module**

If a case has been simulated as remanded, it will be sent to the BAIL module, which simulates activities such as considering bail and granting and setting or denying bail for the accused. The probabilities of bail being considered, granted, or denied are based on crime type and whether the accused is in the sorting or trial stages of his court career, as well as whether the accused is in custody at that appearance. After the bail decisions have been simulated, the custody status parameters of the case are adjusted accordingly. Later decisions in the model, such as the decision of guilt or innocence, can be made dependent on whether the accused has been in custody.

**Reset OUTCOME of Next Appearance**

After the case has been through the BAIL module it is transferred to this module, which determines, on the basis of past statistics, what is likely to happen to the case at the next appearance simulated. The possible outcomes are those described under SORTING.

**RESCHEDULE Case for Next Provincial Courtroom Module**

After the outcome of the next appearance for the case has been determined, the case is rescheduled to a Provincial courtroom. If a case has been remanded for trial in Provincial Court, the module is more likely to assign a courtroom that specializes in trials rather than sorting appearances. The next court assigned is dependent on the type of appearance to be simulated next and the last court of appearance.

After the case has been assigned a new courtroom or administrative office for the next appearance, it is put in the queue of cases waiting for that facility, as described under the QUEUES module.

**Check for PROCEDURE and Set CASE HEARING Parameters**

When all cases before this one have had their next appearance simulated, the model will simulate the next court appearance in the SORTING module. If the case is simulated as being remanded again, the case will repeat the sequence just described until the model simulates the case as being remanded for trial or a preliminary inquiry. The case then enters this PROCEDURE module which checks the offense type and the procedural parameters (as given in ASSIGNMENT OF CASE CHARACTERISTICS) to see whether the next step will be a trial or a preliminary inquiry. If the case is to go to trial, the PROCEDURE module also determines whether any Crown or accused elections are to be simulated and whether the case is to be tried by way of indictment or summarily.

**PLEA BARGAINING Module**

After the CASE HANDLING parameters are set to appropriate values, the case proceeds to a section simulating its "deterioration" because of some or all of the charges either being withdrawn or dropped. The model user must specify how long he wants the model to run. He controls this length of run by specifying a maximum value for one of five variables:

- Number of cases generated, by offense type
- Number of cases disposed of
- Number of offenses disposed of
- Number of accused disposed of
- Number of calendar days simulated.

It is necessary, therefore, to increase each of these variables accordingly when a case is disposed of in the model. One place this is done is in the PLEA BARGAINING module which simulates, on the basis of data estimated for past cases, the substitution, dropping, or withdrawal of some or all of the charges in a case as a result of Crown and/or defense activity.

**CASE HEARING Module**

After processing through the PLEA BARGAINING module, the case is ready to be sent to the CASE HEARING module, which simulates the activities taking place during either:

- A preliminary inquiry in Provincial Court for indictable offense cases to be tried in certain other courts
- A grand jury hearing for indictable offense cases to be tried in certain other courts
- A trial for cases tried by way of indictment in certain courts, or tried by way of summary conviction in Provincial Court
- Appeals in certain courts for various reasons.

Activities included in this module include (depending on type and results of the hearing):

- Remanding to locate the accused
- Remanding for other reasons during the hearing
- Reading the charge
- Swearing in a jury
- Presentation of the case for the Crown
- Motion for dismissal by the defense
- Crown's and defense's summing up
- Remanding for hearing a verdict.

**HEARING DECISION Module**

After CASE HEARING, the case is processed through this module, which simulates the passing of:

- A "sufficient or insufficient evidence to proceed" decision if a preliminary inquiry has just been simulated
- A "true bill or no bill" decision if a grand jury hearing



- A "guilty," "not guilty," or "dismissed" verdict for a trial
- An "allowed or disallowed" verdict for an appeal.

Each hearing decision is dependent on the type of crime and the type of hearing.

**RESET CASE HEARING Parameters**

If a preliminary inquiry or grand jury hearing has just been simulated, the case is moved through the BAIL module and the passage of time for the last hearing is simulated. If the case is to proceed further, it is then sent through a RESET module which resets the CASE HEARING parameters so that either a grand jury hearing or a trial—whichever is appropriate—will be simulated on the next pass through the CASE HEARING module. The case is then sent to a second type of RESCHEDULING module described below.

**SENTENCING Module**

If a trial or appeal hearing has just been simulated, the case is moved next through the SENTENCING module. Allowance is made for activities such as remanding for sentencing or for pre-sentence reports but, once the case is ready to receive a sentence, it would be assigned one on the basis of past sentencing behavior observed in the courts. In the case of an appeal this might refer to a modification of the previous sentence.

**CHECK for Appeal and Reset CASE HEARING Parameters Module**

After sentencing, this module checks the type of decision passed and whether it is appealed. If the case is not to have further proceedings, this module records the statistics desired by the user relating to the processing of the case, then ceases to deal with the case. If an appeal is lodged, the module resets the CASE HEARING parameters to values that will ensure that the appropriate appeal hearing will be simulated next time through the CASE HEARING module.

**RESCHEDULE Case for Next Courtroom Module**

If the case has been appealed from trial or has just had a preliminary inquiry or grand jury hearing, it is assigned to the queue for a specific courtroom hearing.

**QUEUE for Next Courtroom Module**

The case is then placed in a queue to simulate the necessary waiting before the next court appearance. If a certain amount of time is required, such as to obtain a report or transcript, the case will not leave the queue until this amount of time has passed.

**Data Base Required**

The data requirements for this type of detailed, case-by-case simulation are considerable.

The main purpose in data collection for CANCOURT was to provide accurate estimates of the parameters needed to make the CANCOURT model a true representation of the actual court system. This meant that at every significant decision point or activity in the model, it was necessary to obtain data to estimate two types of model parameters, resource and behavioral. The main variable used to simulate the use of resources in performing activities carried out in the courtroom is the amount of courtroom time required for the activity. It would be possible to estimate the cost of a unit of courtroom time by adding up salaries and depreciation of all resources that must be used to operate the courts per unit of time.

The behavioral parameters can be divided into two subsets. The first of these would include the data needed to estimate parameters controlling whether or not this particular decision is to be made at the particular court appearance being simulated. For example, it was necessary to collect data to estimate the appearance number at which one could expect the accused's election of venue to take place. This timing of activities is likely to vary with such factors as type of offense, how the accused pleaded, whether the accused had legal representation, and whether or not he was in custody. The data collected, therefore, had to include information not only on when the election took place, but under what circumstances it took place.

The second subset of behavioral parameters estimated are those relating to the possible results of each activity—specifically, the probabilities of each of the allowable results occurring, given that the activity took place. Again, data must be collected to estimate not only what the relevant average probabilities are, but also how these probabilities vary with the circumstances surrounding the decision.

An iterative procedure was adopted in order to specify the needed decision points. After a study of existing court procedures, a preliminary version of CANCOURT was programmed. This required a close examination of data requirements. The next step was to sample existing court records or make direct observations to determine the feasibility of obtaining the needed data. When certain data were found to be unobtainable, it was necessary to reprogram the model and begin the cycle again. Fortunately, this cycle had to be repeated only once. This was, however, mainly due to the relatively large portion of resources devoted to data collection. (Over $50,000 out of the total budget of approximately $100,000 was spent collecting, coding, and cleaning the data.)

Empirical data for the CANCOURT project were compiled using four main methods:

1. Direct observation in Provincial Court
2. Sampling of the existing information files for cases disposed of in Provincial Court
3. Informal discussions with court personnel
4. In addition, for specific information, reliance was also placed on existing published documents.

Data on court operations were collected by four members of the project, each assigned randomly to one of the 17 Provincial courtrooms each day the courts were open for a period of five weeks. The observers were present in the courtroom from the time it was open to the public until it closed for the day. This gave usable data for a total of 104 court days—approximately 25 percent of all the court days during that period. The observers used a detailed coding system to record what happened to each of 7,529 "count appearances" that were listed on the dockets of the courts observed. Count appearances were computed from the number of charges times the number of defendants times the number of appearances. For example, if two defendants appeared in court charged with three offenses each and both defendants had their cases heard three times, then the number of count appearances would be $2 \times 3 \times 3 = 18$. A "case" then becomes a set of count appearances that were dealt with as a unit in the courtroom. The list of variables observed included the main resource variable—the time in hours, minutes, and seconds that each case began and ended—and included also the variables describing whether or not a particular activity occurred and, if it occurred, the result of the activity. This included variables describing charges being read or withdrawn, Crown and defense elections, remands or standdowns, bail being granted, and handing down a verdict and/or passing sentence. At the end of the court observation period, additional information was obtained from the dockets prepared by the court officers and from a summary of the Criminal Code and other statutes. This additional information includes the section of the statute allegedly contravened and the age, sex, and arrest date of each accused.



All of these data were then transferred from the court observation schedules to specially designed optical scanning machines. These were scanned and converted to magnetic tape and then transferred to a disk pack for cleaning and storage. Many of the cleaning or error-checking procedures were carried out by computer programs which checked both whether the codes for each variable were within allowable ranges and whether the different codes were internally consistent within a case or within a count appearance. The data were then stored in a manner making it possible to extract the value any particular variable took for a particular appearance of a particular defendant on a particular charge.

It was also necessary, in order for the model to accurately simulate a court career, to obtain additional information regarding the timing of each of the decisions during that court career. Information was also needed to determine whether the timing and results of such decisions were dependent upon activities performed in previous appearances or to be expected in future appearances. Data were also needed on conditions prevailing generally in the court system before, during, and after the sampling period, since general conditions might also have affected the timing and the results of the particular activities observed. Additional information on court careers was obtained from the information files in the Provincial Court Clerk's office. Altogether, data were collected on some 130 variables for each of the 1,708 case careers in the intensive sample. These cases represent some 8,766 different count appearances.

### Cost/Computer Requirements

It was estimated that a computer run (IBM 370) covering the 104 court-days for which data were gathered would cost approximately $30.

### Implementation and Impact

When the project reports were published in 1973, they became controversial in the legal community because of publicity claiming that they asserted that "lawyers are useless." (The reports do assert that additional judges are not needed.) This controversy resulted in a "hands-off" attitude by the judges and by the Attorney-General's office, factors which prevented the complete implementation of CANCOURT. The model builders now realize that there should have been continuous coordination with the agencies involved throughout the development of CANCOURT, even though it was developed under university and foundation funds. The model has had no impact except on the knowledge of those who built the models or read the reports.

### Limitations of the Model

The model builders now feel that CANCOURT is too complex. If they had an opportunity to revise it, they would make it considerably simpler, probably designing it to deal with the lower courts only and to focus on the scheduling problem, which is a high-priority problem at present.

### Transferability

CANCOURT is transferable, with effort. Given its modular construction, it is designed for structural change and for the introduction of new data.

## A JUROR MANAGEMENT MODEL

A simulation model of a typical four-judge district court utilizing a jury pool was constructed at the University of South Florida by Michael J. White. This is an interesting example of a fairly detailed model (perhaps not detailed enough) of a small component of the court system, namely the jury pool and the process of selecting jurors. Interestingly, this component is not included in broader models such as CANCOURT, which apparently assume an inexhaustible supply of jurors and are not concerned with the resources required to maintain that supply.

One of the problems in juror management is to maintain a sufficiently large pool of competent and qualified jurors to service the caseload going through the courts. Before he is accepted as qualified for any given case, a juror must undergo "voir dire" examination in the courtroom to determine his competency for that particular case.

### Structure of the Model

There are four courtrooms in the simulation, and these courtrooms must be tied up by the voir dire hearings before the cases can be tried. Each case requires 14 qualified jurors—12 to form the jury and 2 as alternates. Under the present system, the voir dire for each case is handled separately as the preliminary part of the case proceedings. As soon as a courtroom is available, 30 jurors leave the jury pool for voir dire, after which 16 "strikes" return to the jury pool and the remaining 14 jurors stay for the trial. The first version of the model was based on this procedure; then a "multiple voir dire" approach was tried and the model was reprogrammed accordingly. Under this second concept, no cases are tried until all cases on the docket have had juries chosen.



### Output

Comparative runs were made with each of these two versions of the juror simulation model. In both cases there were four courtrooms, a backlog of 250 cases, and a pool of 120 jurors. On successive runs of each simulation, the juror pools were reduced in decrements of 10. Two main measures of performance were monitored: the average utilization of jurors and the total duration of the 250 cases. It was found that the multiple voir dire approach permitted a greater reduction in the size of the jury pool than the conventional approach before there was a significant increase in the total duration of the cases. In addition, jury utilization was substantially higher, and there was a better utilization of court facilities with the multiple voir dire approach.

### Limitations of the Model

The author claims to have demonstrated the superiority of the multiple voir dire approach; however, he mentions several limitations. No restriction was placed on the number of times a juror could be used for either a voir dire or a trial. In the case of a court using the multiple voir dire approach, the possibility exists that the juror could be needed to participate in more than one case at the same time. This would require careful scheduling of cases. This problem could be handled alternatively by restricting a juror's term of service to one case per trial docket (there were five dockets of 50 cases each); however, the model was not rerun with that restriction, which might have affected the end results. Another real-life restriction involves the vulnerability of a case to jury tampering if the jurors selected are identified prior to the actual trial. This might require a conventional selection of juries for certain cases.

The above qualifications were made by the author. One also wonders why several other factors were not taken into account by these particular models. For one thing, if a juror is to be used on more than one case, then it would seem that this type of model should be linked to a court scheduling model to be sure that there is a feasible schedule for any given set of juror assignments. The maximum number of jurors needed at any one time is 56 (14 for each of the four courtrooms). However, one wonders if even a pool of 120 jurors would always be sufficient, given the fact that for some cases it might be difficult to qualify 14 jurors. Some features could have been incorporated into the models to take into account, by case type, the problem of qualifying jurors. This would have added considerable realism, and it might well have changed the outcome of the comparison of the two approaches.

### Language and Documentation

These models were written in the GPSS/360 language. The complete programs are given in the basic reference [188].

## INFORMATION SYSTEMS VERSUS MODELS

To clarify the difference between computerized information systems and computer models, we shall describe one of the former, the Criminal Court Status Information System, and then compare it with criminal court system models.

**IBM's Basic Court System**

Utilizing federal grants, the supreme bench of the city of Baltimore adopted IBM's Basic Court System program and installed the computerized system for the recording of criminal cases and their scheduling and disposition. This is an on-line system supplemented with batch reports for managerial purposes [14]. The system has been developed and implemented and is already providing faster and more accurate information for case calendaring. The scheduling of cases is still done manually, however. The system was renamed the "Criminal Court Status Information System" and encompasses the city jail, the police department, the supreme bench, the state's attorney's office, and the interface between the district court, the public defender's office, and the supreme bench. Most of the effort involved took place in the Criminal Assignment Office under the direction of A. LaMar Benson, Commissioner.

The system includes three basic computerized files within its memory bank, as follows:

1. *Case History File.* This is a master file within the system. Basic information pertaining to the case can be retrieved by entering the case number on the terminal. This file includes the court division ("1" for Supreme Bench Criminal Court), case number, entitlement (case name), status, filing date, case type (charge), bail or warrant information, defendant's race, sex, date of birth, docket event, and date of motions and other non-calendar activity, name of person connected with the case (i.e., title, connection with defendant, witness, police officer), identification number, arrest register number, location (jail, bail, etc.), location number and commitment date, calendar/document date showing part, room, reason, parties, description, estimated time, actual time, and disposition.

2. *Name File.* This file contains information about defendants, lawyers, prosecutors, witnesses, police officers, bail bondsmen, and other individuals connected with the case. The name file information includes: name, title, connection code (defendant, witness, etc.), court division, case number, entitlement (case name), filing date, address, identification.

3. *Court Calendar File.* The court calendar file permits the court to keep a centralized, up-to-the-minute record of its calendar. As dispositions, additions, changes and deletions are received from the courtroom's clerks and other sources, the clerk's office terminal operators can immediately update the affected records. The calendar information contains the following: calendar date, part, room, court division, case number, reason (arraignment, court trial, jury, etc.), parties, estimated time, actual time, and disposition.

As one can see from the listing in Table 2, this is an information system and not a model. It is an active rather than a passive information system; in other words, certain notices and reports are issued automatically. As an example of this, there is a first notice of trial date sent to the defense and state's attorney's offices and then, 28 days before trial, there is a second notice, and 8 days before trial, a third notice—these last two notices with additional addressees. Some special numbering and coding systems developed for this information system would probably facilitate the monitoring of the system to gather data for use with court models. For example, a system of numbering charging documents was developed with 8 digits: the first identifies the nature of the charging document, second and third the calendar year, and the remaining 5 digits identify a sequential number of charging documents



Table 2

AUTOMATED CRIMINAL ASSIGNMENT INFORMATION SYSTEM

| | |
|---|---|
| **A. Batch Reports — Managerial** | |
| 1. New case listing | Daily |
| 2. Audit trail | Daily |
| 3. Daily docket activity report | Daily |
| 4. Appearance filed | Daily |
| 5. Attorney conflict list | Weekly |
| 6. Courtroom docket | Daily |
| 7. Future courtroom docket | Weekly |
| 8. Case no action report | Weekly |
| 9. Active case listing report | Monthly |
| 10. Closed case listing report | Monthly |
| 11. Sub curia listing | Monthly |
| 12. Inactivation listing | Monthly |
| 13. Appeals listing | Monthly |
| 14. Prisoner control listing | Daily |
| 15. Case summary and case history | Daily |
| 16. Statistical report | — |
| **B. Docket and Notices Sent** | |
| 1. Courtroom docket | Daily |
| 2. Defense counsel and State's Attorney's Office notice (later in duplicate) | Daily |
| 3. Witness notices | Daily |
| 4. Defendant and bailbondsman notice | Daily |
| 5. Detained defendant notice | Daily |

according to category (indictments, appeals, warrants, etc.). The overall coding system includes the following:

1. Case status codes—6 categories
2. Calendar part codes—14 categories
3. Case charge codes—93 categories
4. Room number codes—14 categories
5. Docket reason codes—35 categories (Examples: abated by death, appeal filed, bail hearing, pretrial conference)
6. Calendar disposition codes—67 categories (Examples: appeal withdrawn, denied, satisfied bail forfeiture, mistrial, not guilty)
7. Defendant location codes—11 categories (Examples: bail, hospital, city jail, own recognition)
8. Police address codes—34 categories

It was claimed that installation of this computerized information system resulted in substantial improvements. For example, when the project started in July 1973 there were 6,337 open documents and 3,413 defendants. By June 1974, there were 3,092 documents and 1,720 defendants. However, other things were happening during this time which tend to obscure the exact effects of the information system. In September 1973, three nonjury courts were established because of the case backlog. A new arraignment policy went into effect in October, eliminating arraignments from two parts of the court (those parts identified as high-impact courts). A new liaison system was established between the supreme bench and the police department to assure the presence of police witnesses. During a four-month period (summer 1973) a special night task force was set up, operating five computer consoles to enter all open cases since January 1972. Also, the Criminal Assignment Office had four temporary employees during this same four-month period to expedite the scheduling for all criminal courts of the supreme bench. These policy changes and special manpower allocations must have had some impact on court backlogs; therefore the improvements cannot be wholly attributed to the improved information system.

### Relationship to Court Models

Let us turn now to the comparison of an information system, such as this one, to a true court model, such as CANCOURT. The information system provides a wealth of data on each case as it enters and passes through the court system, and generates managerial information on a daily, weekly, and monthly basis on each of these cases. Statistical reports on court performance, however, were prepared manually, no doubt using data from the information system, but there is no evidence in the reference that any of these statistical reports were generated by the computer, using special programs. There was some discussion of a critical path program, but insofar as can be determined, this was never developed.

What the information system provides is basically a snapshot of case status at various points in time. It does not collect statistical data on the variations in flow between types of cases. Given the proper support, however, it would not be an unreasonable task to "instrument" the court information system at designated decision points and keep records of the flow by type of case. This computerized bookkeeping could automatically be translated into flow probabilities or branching ratios for use in models such as CANCOURT and JUSSIM. The developers of CANCOURT went to great pains to estimate the probabilities of a case disposition as a function of its previous history. Yet this specialized data collection effort extended only over a relatively short period of time and therefore could not account for seasonal fluctuations or for the effect of various court loads on the probabilities computed. Programming an information system to generate these probabilities on a periodic basis over a long period of time might be an extremely effective and relatively inexpensive way of gathering data for input to court models. On the other hand, without such special instrumentation, information systems such as Basic Court are almost useless as sources of data for models, and one is forced to go back to the basic case folders in order to trace the history of individual cases and develop the needed probabilities.

In the future, more care should be given to the design of criminal justice information systems so that they include the necessary program elements to develop model input data.

## LEADICS

### Historical Background

The LEADICS computer model is a small part of the overall LEADICS project undertaken by Notre Dame University Law School and College of Engineering [178].



The purpose of the model itself was to develop a means of calculating time delays for the overall processing of cases and offenders through the court system. (The name is an acronym for Law-Engineering Analysis of Delay in Court Systems.)

The authors of the report indicated several options they felt they had in undertaking the computer model. There are several commercially available simulation languages, such as GPSS, GASP, or GERTS, any one of which could have been used to develop estimates of the transit time through the court system. Each of these commercially available computer program products had certain deficiencies. Additionally, the designers wanted to operate their model on a small computer in an interactive mode, which would allow for more immediate feedback to the user. None of the commercial simulation languages had this capability.

### Structure of the Model

LEADICS is a numerically based model that calculates the characteristics of the total transiting time more accurately and quickly than could any simulation. The characteristic of any of the various transiting or passage times is best described by its probability density function. With this density function one could calculate any of the various moments (expected outcomes, variances, etc.) or median of the distribution or obtain good estimates for any confidence limits or error bound one might want for the transiting time. Not only is the characteristic of the total time available but also the characteristics of any of the intermediate delay segments could be found as measured by a density function using the model.

The actual technique used is a numerical procedure that was developed by electrical engineers and systems engineers, called "fast Fourier transforms." The sum of the various time delays throughout the court system is a sum of individual random variables, generally speaking, each of them independent and non-identically distributed, and the total is then this sum over all these components in the system. While it is easy to find the expected value of the sum of random variables, and in many cases of the variance of the sum of random variables, the problem is made much more complex when there are various probabilistic branching and feedback loops in a system such as the court system. For these reasons the Fourier transform method is a fairly efficient numerical approach to finding the distribution function for the total time. The procedure that was used first estimated the first ten moments of each of the component random variables and the branching probabilities from the different processing stages in the court network.

Some severe data problems were experienced and did delay the implementation and the trial runs of the model. The designers found that data forms were error-ridden, and often contained illegible handwriting, a common problem for criminal justice system data files. The model was programmed in FORTRAN IV in an interactive version and installed on one of the smaller IBM machines, an 1130. The user sits at a terminal and can propose changes in the court system. These changes, such as branching ratio parameter changes, can be tested. Branching ratios indicate the fraction of cases leaving a processing stage for each of the other stages. Time delays can be changed to estimate the effect on the total transit time through the system.

### Output

The model acts as a numerical calculating program and produces statistical characteristics of the time delays.

### Limitations of the Model

One of the several criticisms that can be made of this model is that queuing delays are not modeled directly. It is not possible to test for the addition of more judges, more prosecutors, more defense attorneys, or more courtrooms, because the model itself only takes as input the total delay time distribution and the flow system graph and branching parameters. One would have to use a side calculation or analysis external to the LEADICS computer model to determine how the addition of resources would affect delay time in any processing stage.

A second criticism is that independence is assumed among the random variables. A simple example demonstrates why this is not always correct. A long case in one court is likely to become a slow and tedious case in a subsequent appeals court. In a fast case, with the case being solidly against the defendant, the defendant may pass quite quickly through all subsequent court delay stages. These are complex technical problems, and could be overcome only with some difficulty.

# Chapter 6
# CORRECTIONS MODELS

Gass [65] recently reported finding "a paucity of models and research into decisionmaking as applied to correctional problems," a judgment with which we concur. In fact, we did not locate a single model in the field of corrections that was as fully developed and tested as the others described in this report. In order to give some notion of the types of issues that could be addressed using a suitable model, we shall describe one exploratory effort that appears to show some promise.

## FCSM (FEDERAL CORRECTION SIMULATION MODEL)

A simulation model and several simpler techniques for estimating expected population in the Canadian correctional system were developed in the first stage of a project called the Offender Prediction Study. The work was performed by Systems Dimensions, Ltd., for the Department of the Solicitor General of Canada [71,73].

### Historical Background

In recent years Canada has experienced a large increase in penitentiary inmate population. This has greatly enlarged resource requirements, not only for the institutions themselves, but also for other parts of the correctional system such as the parole supervision process. These increased demands came at a time when greater demands were occurring in other parts of the public sector in Canada, and when there has been considerable pressure for a revamping of the whole correctional process.

It was recognized that, to achieve more efficient allocation of limited resources, planners would need more effective tools to use in policy formulation. For this reason, the Solicitor General contracted with Systems Dimensions to work with Ministry personnel to develop new planning tools. The specific objective of the study was to develop a capability for forecasting two of the main determinants of demands for resources in the Ministry: (1) populations of offenders in the system, and (2) flows within and between the components of the criminal justice system responsible to the Solicitor General.

It was recognized at the outset that there was a requirement for the development of tools that could not only forecast populations and flows under existing programs and policies, but also could forecast effects of proposed policy changes. It was clear that a rather sophisticated model was necessary for the latter purpose. However, a need was felt to develop simpler techniques for planning tools during the period when the more sophisticated model was being developed. The study effort, therefore, examined and developed both types of techniques.

The study was seen as a two-stage effort, with the first stage being devoted to a general investigation of other attempts to use prediction models and evaluation of the feasibility of developing such models for use by the Ministry. The results from this stage would recommend work to be done in the second stage.

## Policy Issues Addressed

The key issue is, given fluctuations in parole rates and transit times (times spent in penitentiary), how does the Solicitor General plan future penitentiary requirements? Before the start of the study, a simple forecasting model was used for this purpose; it simply extrapolated past trends. Even though accurate, it could not be used to study policy in areas such as parole or sentencing, because its predictions were independent of policy. Two examples of present policy are: (1) an inmate is eligible for parole after serving one-third of his sentence; (2) assuming good behavior, an inmate generally has to serve only about two-thirds of his sentence. While it might appear easy to forecast penitentiary populations as a function of parole granting rate (the percent of those eligible who are granted parole), in fact some parolees violate their parole and return to correction facilities. The number who do so might depend on the parole granting rate, and certainly depends on how rigidly parole violations are handled. More complicated interactions might also occur. For example, judges might change their sentencing practices in the light of new parole policies.

## Structure of Model

This section will discuss the preliminary work done on the more sophisticated model and mention briefly some of the less-complicated interim planning tools that were developed.

The study report gives a particularly lucid description of the potential role of models, and the comparative advantages of simple conceptual models as opposed to complex quantitative models. It outlines a planning process model highlighting three levels of planning: (1) policy planning (comprised of normative policy planning, program policy planning, and administrative policy planning); (2) program planning and budgeting; and (3) operation management. The authors attempt to show that this planning process in these three planning levels is built on a hierarchy of objectives; objectives or ends at one level being the means for accomplishing ends at a higher level. They placed considerable stress on the role of evaluation at each of the three levels in the planning process. Information as to the probable effects of different policies or programs is necessary in order to choose among alternative policy strategies.

They see two kinds of evaluations: retrospective and prospective. Retrospective evaluation attempts to estimate the impact of present or former programs or policies on the process of goal attainment in the past and present. For example, a model could be used to determine whether changes in parole granting policies in the past had caused any part of the current increase in penitentiary populations. Prospective evaluation attempts to forecast the impact of current or proposed programs or policies on future events.

Using this definition of evaluation, most of the prediction models developed in the course of the study are, in fact, evaluation models. The model builders attempted to design the models in such a way as to be able to perform both retrospective and prospective evaluation.



As indicated above, the policy planning level can be divided into three functions: normative policy, program policy, and administrative policy. As an example of *normative policy*, decisionmakers may be faced with a change in government policy that places more emphasis on rehabilitation of criminals to become productive members of the community, rather than prevention and deterrence through custodial activities. Before such normative policy changes can be made, the decisionmakers must have information regarding the resource requirements implicit in various levels of program objectives to see which are likely to be feasible. Thus, predictions are necessary even at this highest level of planning effort.

In *planning program policy*, the analyst must have similar prediction information of both offender flows and populations and resource requirements. If, for example, predictions indicate that existing resources for rehabilitation would not be sufficient, the program policymaker might be required to consider new programs and restructuring of activities in order to accomplish the desired normative policy change. *Administrative policy changes* might be required in order to achieve different program objectives. To continue the rehabilitation example, if increased emphasis on rehabilitation is desired, and if that requires more contact with the community to which the offender will return, then changes in administrative policy that emphasize such things as regionalization or decentralization of decisionmaking down to the community level might be required.

The study concluded that although powerful prediction tools are necessary at the policy planning level, the most immediate short-term benefits from the use of such models would be at the program planning and budgeting level.

Program planning and budgeting personnel are required to take programs flowing from the higher level planning process and translate them into estimates of resource needs and to prepare budgets for programs which flow from these policies.

At the lowest level of planning—that of *planning management of operations*—particular resources must be secured and policies and plans finally translated into action. Planning at this level must, of course, be far more specific than at the higher levels. As a result, there is a demand for considerably more disaggregated prediction information. Annual estimates of flows of populations or actual population levels would be of only very general use to planners at this level where the need is to schedule and ensure the availability of resources for particular geographic locations.

At this level, it became apparent to the people doing the study that there is a fundamental requirement for any prediction model to be able to predict the dynamics of change through time as well as the ultimate end result of any change in policy. While it may be enough for program planning and budgeting purposes to know how much of the total ultimate effect of a policy change will be felt year by year during the course of a program, for operations management purposes it is necessary to have similar information for periods of much shorter duration—perhaps even for periods of less than a month.

The major model developed during the course of the study to provide the sort of prediction information necessary to support decisionmaking within the Federal Correction System was a computer model which attempted to simulate populations and flows of offenders within the Federal Correction System. It simulates the dynamic flow of offenders into, within, and from four main components of the Federal Correction System. The Ministry of the Solicitor General is responsible for three of these—Federal Penitentiary, the National Parole Program, and the Mandatory

Supervision Program. The model also simulates offenders staying within the community, not under Federal Correction System supervision, along with their possible return to a penitentiary. Thus, it has a fourth component which roughly corresponds to that part of the criminal justice system not under the direct authority of the Ministry. An important design characteristic was that the simulation was not intended to estimate steady-state results. Rather, it begins with population characteristics at a specified point in time (e.g., the present or, for retrospective evaluation, some year in the past) and simulates forward in time.

Figure 6 is a flow diagram of the Federal Correction System Model (FCSM). The model is programmed in GPSS/360X to simulate the flow of offenders through the Federal Correction System and includes various feedback loops which offenders may follow if they are returned to the penitentiary after release. Inputs to the system from outside are determined exogenously.

Within the model, decisions as to whether or not an offender is to receive parole are determined as random events, either based on observed data (that is, empirical probabilities) or user-specified probabilities for each offender passing through the model. Time delays for each offender at the various stages in the flow process are also randomly determined, based either on observed data or on user-specified time distribution.

As the figure shows, the exogenous model offender input is the number of new entries (that is, persons with no previous penitentiary histories) received by federal penitentiaries. Other input to penitentiaries (recidivism) is internally simulated within the model).

Before being placed in a penitentiary, an offender is assigned a sentence which is drawn from the distribution of actual sentence lengths determined empirically. Next, a decision that determines whether or not an offender will be paroled is simulated. Since the parole grant rate is not constant from year to year, an algorithm is used to determine a probability of parole as a function of sentence length and input data giving each year's parole grant rates. If the offender is paroled, he serves a certain amount of his nominal sentence length before being released. This determination is made by drawing from a distribution of time served before being released to National Parole. If an offender is not given parole, he serves a given fraction of his sentence length based on the assumption that he does not lose any of his statutory remission time and earns all other remission time.

After serving time in a penitentiary, the offenders who receive parole leave the penitentiary and are placed on the National Parole program. A decision determining whether or not the offender will be successful on parole is then simulated, using user-input probabilities. If the result is failure, then the model returns the offender to the penitentiary with a new sentence length determined by an existing regulation regarding revocation of parole.

If the offender is successful on parole, he serves his entire parole period before being removed from the National Parole Program.

After an offender has successfully completed National Parole, the model makes a decision to determine whether or not the offender will be returned to a penitentiary. This probability is again user-input. Should the model determine that the offender will recidivate, he spends a fraction of his time in the community free from Federal Correction System supervision before being returned to the sentencing section at the beginning of the model. If the offender does not recidivate, he is removed from the model.



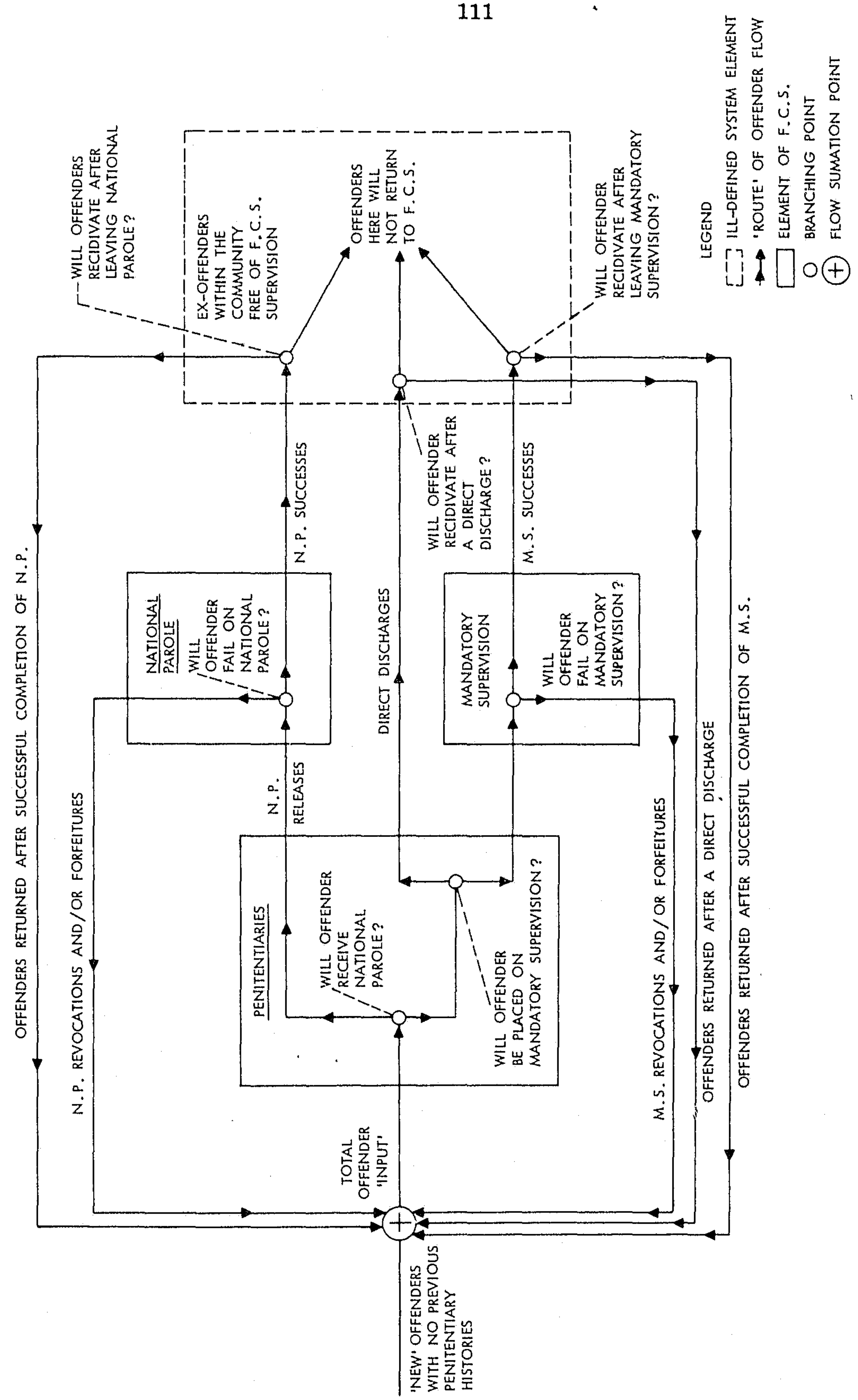


Fig. 6—Flow diagram of the Federal Correction Simulation model (Canada), redrawn from Ref. 71, p. v-8

### Data Base Required

As can be seen in Fig. 6, data are required for each decision point in order to decide which direction a case will take at that point. Severe data problems were encountered prior to the implementation of a computerized information system in 1971. The model now uses routine data generated by the information system. It was difficult to validate the first phase because even such basic data as penitentiary populations were either missing or grossly inconsistent. As of February 1975, a Delphi approach was being planned and investigated as a way of getting estimates of future parole rates and transit times at each federal penitentiary. More than twenty key planning officials were to be involved in this study.

### Output

The February 1975 version would aggregate outputs at the national level, with regional outputs to come later. The basic outputs would be the total populations of federal penitentiaries (those with inmates serving sentences of two years or more) and the flows between the components shown in Fig. 6.

### Cost and Computer Requirements

It was estimated that to run a 10 percent sample for a twenty-year period, say 1964-84, would cost around $40 using the GPSS language 360X, version 5, which operates on an IBM System 360 or 370 computer.

### Validation and Verification

Verification and sensitivity testing were conducted carefully and were still under way in mid-1975. A major effort was made to validate the model, to the credit of the designers. Validation was conducted by starting the model with initial conditions representing no offenders in the population in 1942. The model was then permitted to "warm up" by simulating a twenty-year period. This process generated the presumed characteristics of the penitentiary and parole population and released offenders in the community for the year 1962, which then served as initial conditions for a validation test. (It would be impossible to obtain data describing the relevant characteristics for the year 1962.) The model was then operated to simulate actual policy for the years beginning with 1963, and the output was compared with actual data. The results matched within 2 percent and were particularly accurate near the critical turning points in penitentiary population. (While the predictions obtained from the simple model that extrapolated trends appear visually to be almost as accurate, they erred by an average of 4 percent, so the difference between the two models amounts to approximately one full correction facility.)

### Implementation and Impact

The first priority of implementation will be to ensure that the model actually drives programming and forecasting in the Office of the Commissioner of Penitentiaries. A second priority will be to look at the implications for the Secretariat (which includes policy planning, research, statistics, and management information systems) and for the steering committee (SPAC) made up of the heads of organizations under the Solicitor General (the Secretariat, the Commissioner of the Royal Canadian Mounted Police, the Commissioner of Penitentiaries, the head of the National Parole Service, and the Chairman of the National Parole Board). Complete documentation and demonstration projects were to be ready by August 1975. Judging by the interested participation of representatives from these various offices under the Solicitor General, it is anticipated that this model will actually be used as planned and extended for regional and other types of forecasts and planning.



Proposed interactive versions were resisted by SDL on the grounds that the user should take more time to think about results. However, a compromise was achieved by permitting user input to be entered interactively, with the output received later from a batch printer.

Other possible impacts: education of the client, reduction of paperwork, the ability to make better budget arguments and to avoid short-term fluctuations in budgeting. Long-range forecasting should be of much higher quality and those involved should develop a better understanding of the effects of policy changes such as parole rate.

### Limitations of the Model

No limitations are foreseen if the model is used as designed, in highly aggregated form. The model makes no attempt to quantify such relationships as the effect of parole rate on violation rate.

### Transferability

This model is probably applicable to U.S. federal penitentiary systems with minor changes and new input data. Careful attention should be paid to tailoring the model to the U.S. systems and transforming it into their model.

### Documentation

The final documentation was not available in time to be reviewed by the authors of this report but consists of a summary [74], a user's manual [75], and a program description [76].

# Chapter 7

# IMPLEMENTATION OF MODELS

## INTRODUCTION

This chapter discusses the process by which organizations acquire computer models and put them to use. It also addresses elements of the complementary issue of diffusion—the transfer of knowledge and models among agencies. Based on a series of interviews with builders and users of criminal justice models, it draws from empirical data some generalized lessons concerning these issues.

We attempted to contact users and builders for each of the models found to be of interest for this study, as described in Chapter 1. (These models are listed in the appendix.) While these may not be representative of all models for which an implementation was attempted, the total sample of 39 cases includes contacts with all parts of the criminal justice system, including the police, corrections departments, court administrations, state planning agencies, and researchers. Tables 3 and 4 describe the data base.

As we have mentioned in Chapter 2, any discussion of implementation must distinguish between models intended for *one-shot* policy decisions and those intended for *recurring* decisions. The LEADICS model, described in Chapter 5, is an example of a one-shot model, since it was intended to develop one set of recommendations to decrease delays in the courts of two Indiana counties. Patrol allocation models, described in Chapter 4, are designed to support recurring decisions, although a one-shot implementation is also possible.

One way to get full value from the investment in a computer model is to reuse it, to amortize its cost over many decisions. In addition, a way to measure a model's success is to look at the number of decisions in which it has been used. For models that support recurring decisions, each use within one agency spreads the cost out. In the sense that it is another period of time added to the model's use, it is also an indication of successful implementation. For one-shot models, on the other hand, diffusion to others is required for repeated applications. To evaluate model use in light of these differences, we must recognize that different emphases in examining diffusion and continued use should be placed on each type of model.

In this chapter we discuss how models come to be introduced to user agencies and either used or not used. Thus we are primarily interested in models that already exist and have been verified, or in models that are developed at the specific request of an identifiable user agency. Prior to the activities discussed in this chapter, there may be an effort by the model builder to find a real-life trial for his invention. This start-up initiative might be an extension of the scientific enterprise of model development, rather than an intentional diffusion exercise per se. If the host agency for such an exercise views itself as cooperating with the model builder rather than as considering the possibility of using the model, we do not include this process as part of the implementation sequence.





Table 3

DESCRIPTION OF DATA BASE OF MODEL APPLICATIONS

| Type of User Agency | (a) Number of Models Examined | (b) Number of Models With Interviews | (c) Percent of Models With Interviews 100 × (b)/(a) | (d) Number of Model Users Known | (e) Number of Model Users Interviewed | (f) Percent of Model Users Interviewed 100 × (e)/(d) |
|---|---|---|---|---|---|---|
| Police | 15 | 6 | 40 | 25 | 10 | 40 |
| Courts and corrections | 20 | 5 | 25 | 8 | 6 | 75 |
| Overall | 10 | 6 | 60 | 37 | 23 | 62 |
| Total | 45 | 17 | 38 | 70 | 39 | 55 |

Table 4

DESCRIPTION OF MODEL BUILDER DATA BASE

| Model Type | Number of Models Examined | Number of Model Builders Contacted | Percent of Builders Contacted |
|---|---|---|---|
| Overall | 10 | 6 | 60 |
| Police | 15 | 5 | 33 |
| Courts and corrections | 20 | 5 | 25 |
| Total | 45 | 16 | 36 |

## THE IMPLEMENTATION SEQUENCE

Based on both our observations and the literature of organizational innovation (see, for example, Refs. 69 and 195), the process of implementing a model can be viewed as occurring in four stages that correspond roughly to chronological order:

*Stage I. Preliminaries*
- Presence of enabling conditions
- Assertion of need

*Stage II. Structuring*
- Remedy specification
- Search

*Stage III. Choice*
- Selection
- Trial introduction

*Stage IV. Implementation*
- Sustained use of the model

Within each stage, occurrence of the indicated steps can overlap and follow varying orders.

In Stage I, a set of conditions must be present within the organization to make it receptive toward innovation. A favorable environment for such new ideas can come from two states of the organization in terms of its performance levels. If its operations are regarded as unsatisfactory, there is what Downs [58] has called a *performance gap* that will cause a search for improvements. Such a reaction is called *crisis innovation,* although many criminal justice agencies use the term *crisis* to refer to a problem that must be resolved in a few hours or days. If, instead, the agency's goals are being attained or surpassed, then Cyert and March [51] suggest there is a tendency to seek higher levels of performance by exploring new ideas and methods. Such an initiative comes from spare resources, and is commonly called *slack innovation.*

A hypothetical court administrator's office will illustrate both cases. Using speed of processing and volume of cases handled as indicators, a performance gap would be perceived if long queues and excessive delays in adjudication were common. This condition might move the administrator to innovate in the hope of bringing about some relief. This is crisis innovation. On the other hand, if queues and delays were within acceptable limits, the administrator might be receptive to innovation as a means of doing even better and of employing spare resources. This is slack innovation.

The latter has the feature that it is not undertaken in response to some pressing need; instead it might replace or improve a process that is generally considered adequate. There is a potential problem here, as we shall see, with obtaining interest and support among decisionmakers later on. In addition, slack innovations normally arise as opportunities that are not planned or provided for. As a result, resources required for their support may be in marginal and uncertain supply. As they relate to these two characteristics, many of the cases of attempted model implementation that we found can be characterized as slack innovation.

Within Stage I, the term *enabling conditions* refers to providing a foundation of necessary elements permitting the innovation to proceed. This can mean a wide variety of requirements, including some slack in personnel time and other resources allowing an initial search for a model; availability of information sources on models; receptiveness, or neutrality, toward the idea of using a model on the part of the agency's management; and the presence of resources to support whatever innovation might arise from using the model (commonly, money which is provided by some outside funding agency like LEAA). *Assertion of need* refers to the process by which someone in the agency develops a constituency for exploring the opportunities for change.

Occurring in varying order in Stage II are *problem formulation*—a more precise definition of what policy issue is to be addressed—and some specification of what *remedies* might be suitable as further information is obtained. There is also a *search* for an appropriate model that includes considering alternative responses as study continues.

The search itself consists of reviews of the literature, visits to conferences and seminars, and investigations of word-of-mouth leads. It has been suggested by Knight [106] that search in a slack situation will be wide, looking at many alternatives before one is selected. Our interviews, however, indicate that the idea of "satisficing" search [168] is more appropriate for criminal justice models. This means that few alternative models will be considered and that the search will end with the first acceptable choice, rather than looking further to obtain an even better one. For reasons of scarce resources and poor information availability, searches were generally short, and seldom involved more than two alternative models.



In Stage III, the search ends with the choice of some model, and its introduction on an experimental basis. This involves information system set-up, data collection, trial runs, and analysis. The initial use of the model is almost always considered a trial, whether formally called such or not.

Sustained use is the final stage, achieved only where the model is successfully integrated into the organization.

## FACTORS IN THE IMPLEMENTATION OUTCOME

Our general observation in this study was that most implementation efforts tend to stall somewhere in the sequence just described, thereby never achieving the full use intended by the model builder. This situation is summarized in Table 5. Of the 39 contacts in the data base, 7 had models that were working (in the case of models intended for recurring decisions) or had been used to produce one-time recommendations that were at least partially acted on; 7 were in some stage of installation with future use anticipated. The remaining 25 contacts were cases of nonuse.

To explore the reasons for nonuse, we relied primarily on the statements of model builders and users during interviews. Where opinions differed, we attempted to sort them out by additional interviews and reinterviews. In some cases, several factors appeared to have contributed to nonuse, but for the purposes of displaying the data in summary form, we made a judgment as to which factor was primary. The results of this categorization are shown in Table 6. In the discussion that follows, instances of overlap between items in the table will be indicated.

Table 5

MODEL APPLICATIONS BY STATUS OF MODEL USE

| | Type of Model | | | |
|---|---|---|---|---|
| Status of Model | Police | Courts/ Corrections | Overall | Total |
| In use/used in analysis | 2 | 3 | 2 | 7 |
| In progress toward installation | 3 | 1 | 3 | 7 |
| Not adopted | 5 | 2 | 18 | 25 |
| Total | 10 | 6 | 23 | 39 |

Table 6

APPLICATIONS WHERE MODELS WERE NOT USED BY REASON FOR NONADOPTION

| | Type of Model | | | |
|---|---|---|---|---|
| Reason for Nonadoption | Overall | Police | Courts/ Corrections | Total |
| Model attributes: | | | | |
| Computer language | 1 | 1 | 0 | 2 |
| Data requirements | 4 | 0 | 0 | 4 |
| Conceptual complexity | 1 | 0 | 0 | 1 |
| Interpersonal problems: | | | | |
| Staff change in user agency from supportive to non-supportive personnel | 3 | 2 | 1 | 6 |
| User agency decisionmakers never had an interest in the model | 0 | 2 | 0 | 2 |
| Contract disagreement, unrelated to the model | 1 | 0 | 0 | 1 |
| Model acquired for potential use that did not arise | 5 | 0 | 1 | 6 |
| Unknown | 3 | 0 | 0 | 3 |
| Total | 18 | 5 | 2 | 25 |

### Obstacles to Implementation

**Model Attributes.** As we have noted in Chapter 2, models may differ greatly in their logical structure, in whether or not they have been verified and validated, in data requirements, cost, and mode of operation (i.e., interactive or batch). These attributes of models appeared to be important impediments to implementation in somewhat under one-third of the examples where models failed to be used. In two cases, the language in which the computer program was written constituted the primary impediment. In one of these, the programming language could not be compiled on any computer system available to the agency; in the other, the agency had no personnel who had been trained to use the language.

The most important model attribute that served as an impediment to implementation was a requirement for large quantities of currently unavailable data. This occurred in four instances, all of them overall models of the criminal justice system. (Data requirements were also a secondary factor in some nonuses listed elsewhere in Table 6.) The same models have, however, been successfully implemented in other jurisdictions, indicating that an agency's reaction to the need for a major data collection effort will be influenced by the perceived importance of the policy issue to be addressed and potential value of the data for other purposes. These models show promise for increased use in the future, as information systems become available to provide at least a major part of the required data.

In only one instance did we find implementation thwarted by conceptual characteristics of the model. This was a case where the agency personnel never succeeded in mastering the terminology and ideas underlying the model. It was not a case of the agency personnel judging that the model was conceptually unsound.



In other words, the kind of quality characteristics that a model builder has uppermost in his mind during the design phase were not found to play a role in nonuse for any of the models we reviewed, except in the negative sense that if the model builder is too "clever" he may confuse potential users. It should be noted that this example was a model that was successfully implemented elsewhere, and therefore the judgment of excessive complexity is a relative one.

We did not find any examples where nonuse could be attributed primarily to the mode of program operation (batch or interactive). However, this is probably explained by the fact that a model would be rejected on these grounds at such an early stage of the implementation sequence that the potential user would not have come to our attention in connection with this study.

**Interpersonal Elements.** At the transfer point between the organization and the model supplier, there is a fragile personal interaction. The future of a model often depends on certain key actors, with relatively small emphasis on the characteristics of the model or the processes which it is to assist. The failure of implementation could be attributed primarily to this process in over one-third of the cases, indicated as the second item in Table 6.

One important actor is the person in the potential user organization who assumes the "advocate's" role. It was often the case in our data that a single person would see the need for a model, conduct a search, and sponsor his choice before the user agency's decisionmakers. Then he would pursue the implementation with little support from the rest of the organization.

In Table 7, the distribution of contacts by presence of an advocate is shown. Unfortunately, information on the acquisition process does not usually become part of the permanent record of user agencies and can be obtained only if people who were directly involved or otherwise closely associated with the project are still available for interviews. This problem prevented us from determining whether there had been an advocate in 42 percent of contacts. In four other cases the notion of an advocate does not apply because the models were obtained for reasons other than use in agency decisions. Of the 19 cases remaining, the advocate is clearly the dominant pattern, applying in 14 instances.

Unfortunately for implementation prospects, much depends on the judgment and actions of the advocate. Because model support is focused on him, his continued attention and political skills are often crucial. The case of PHILJIM's adoption by the Department of Corrections in Alaska will illustrate these points.

An analyst with the Department of Corrections had some exposure to simulation through a previous job. He saw a potential for its use in the Alaska criminal justice system and made inquiry with the State Planning Agency (SPA). At the time, the staff of the SPA saw themselves as grant brokers rather than planners, and suggested that Corrections acquire and run whatever model they found appropriate. A search was performed which, through some 1968 LEAA reports on COURTSIM, led to the discovery of the existence of the JUSSIM model.

The analyst reviewed the JUSSIM model, but was not convinced it would work in a real-world political environment (essentially because he felt it was unproven in such a situation). Additionally, the JUSSIM people impressed him as interested in research and not much in another application of an established model. They referred him to Government Studies and Systems Corporation—a Mathematica subsidiary—in Philadelphia. The PHILJIM model appeared to have proved itself there, so the

Table 7

RELEVANCE OF MODEL ADVOCATE IN CASES EXAMINED

| Type of Model | Total Number of Contacts | Applications With Advocate | Unknown if Advocate Applies | Applications Without Advocate | Not Applicable[a] |
|---|---|---|---|---|---|
| Police | 10 | 4 | 3 | 3[b] | — |
| Courts and corrections | 6 | 1 | 4 | 1 | — |
| Overall | 23 | 9 | 9 | 1 | 4 |
| Total | 39 | 14 | 16 | 5 | 4 |

[a]Many nonline agencies acquired overall models, so the advocate as described here does not apply; management consulting firms and university research centers or clearinghouses are examples.

[b]These three applications were all with the New York City Police Department through The New York City-Rand Institute. All other applications were with separate agencies; therefore, of five *different* police departments where information was available, the advocate model obtained in four.

Corrections Department let a contract for its installation in Alaska. The model was brought on line and run with a set of prototype data. However, it never found continued use. Clearly, there would be small chance that any model would have reached an operational state without this analyst's initiative in perceiving the need for it and in acquiring it despite the lack of assistance from the planners. In a similar way, this dependence led to the model's present dormant state, as these subsequent events indicate:

1. The analyst's attention turned toward other interests. He thought the SPA would continue PHILJIM after the demonstration period, but they still did not see their role as planners at that point and did not assume responsibility.
2. His grant did not have provisions for running ongoing analyses, and he ran out of money.
3. There were data-base problems. A management information system was in development simultaneously with his acquisition of PHILJIM. It was originally perceived as a statewide, interagency system, but the consultants who were retained to set it up employed a subsystem approach instead. As a result, much PHILJIM data collection remained manual and difficult.

The pattern was much the same in many of the other implementation cases we surveyed. It may be that certain parts of the normal organizational resistance to change can be overcome by an advocate. First, he is a focal point at which all the details of a new project come together; his familiarity with the model is a resource to the rest of the organization; he has chosen to make the investment in energy to overcome the doubts and fears of those in power in the organization. An analyst who helped introduce the JUSSIM system in the Maryland SPA emphasized the advocate's role with his view that there is a need for someone, preferably with experience, to introduce the technology and educate users to assure continued use.



In addition to the person we have identified as the advocate, managers in the sponsoring and client agencies can influence the implementation outcome. Points for potential disagreement among managers include:

- The felt need, or perception, that there is a use for the model
- The view that the model can fill the requirements of its intended use
- Its anticipated impacts in terms of organizational changes
- Familiarity and understandability of the technology
- Comparative judgments about costs, benefits, and consequences of implementation.

A project's continuation, then, may be susceptible to changes in management. The Boston Police Department, for example, contracted for acquisition of an early simulation model for patrol allocation. During the course of work to modify the computer program for the department, a new commissioner was installed. Apparently because the simulation was associated in a political sense with the previous commissioner, work was stopped and the model remains dormant.

A corollary to these points is that vulnerability to changes in personnel increases as time elapsed on a project increases. Unless or until a constituency develops for a model, the transfer of key persons can doom implementation. This circumstance, which was observed in six instances, is clearly unrelated to the inherent attributes of the model.

## Agency Characteristics and World Views

In the early stages of the implementation sequence, a combination of two sets of organizational characteristics plays an important role in the outcomes. These are (1) the presence of technical capability to support the model, and (2) the presence of personnel with a receptive posture toward modeling.

On the first point, model support consists mainly of persons with the ability to adapt the model or the inputs if necessary, and to interpret the output (programmers and analysts). Since we are discussing what Knight [106] has called nonprogrammed innovations—ones that are not planned or provided for by the agency—the availability of these resources is often marginal and far from assured.

On the second point, we have mentioned the role of the advocate in bringing the model to the organization, and of a number of others in its continuation. The world view, or the set of attitudes of all of these persons regarding innovation, is an important determinant in the model's future.

The people in our survey cases who had a part in conceiving or assisting model use were, as a rule, those with training or experience in sophisticated management techniques. Some extraorganizational relationships were also characteristic. Membership in professional organizations, attendance at conferences and conventions, or simply exposure to the professional literature are examples.

Advanced training, use of sophisticated techniques, and a view that extends beyond the immediate organization are all elements in the professionalization of personnel, as described by Zaltman, Duncan, and Holbek [195], among others. A lack of professionalization in this technical sense generally indicates an unreceptive agency. The following comparison of two Massachusetts police departments will illustrate the importance of support capacity and managerial attitudes.

Both of these departments were exposed to Larson's Hypercube Model. One has integrated it in a relatively complete way into its allocation planning—including provision for its use in annual reviews of procedures. The second has tentatively adopted a model-assisted design of patrol beats, but only under external political pressure, and future use is virtually out of the question.

The openness of the first department to new methods predated this experience; it had used management consultants years earlier to look at the possibilities of modeling. The police managers had demonstrated their innovativeness. One concrete result of their approach was a preexisting automated management information system that could provide much of the data needed by the model.

On the other hand, modeling might have gone unnoticed in the second department, except for outside influence. The advocate here was a young assistant city manager who was on a temporary assignment in the police department. His observation that crime incidence and patrol were geographically unbalanced—and his search for a remedy—caused the model's introduction. But the department's limited perspective was evident elsewhere: its information system was manual, with data stored exclusively on index cards, and the like. Processing consisted solely of aggregating incident counts for reporting to the FBI. In short, there was no foundation on which to operate a computer model. The assistant city manager must bear some responsibility for the model's termination, for in essence he was piling high technology onto a relatively primitive management system. The immediate cause of disuse was the lack of computer support. MIT's facility was used during the initial analysis, but this could not be continued. The town's only computer was controlled by the Board of Education, which didn't want to share it.

As we mentioned, the class of potential innovators consists of persons with some technical knowledge and contacts outside the agency with other managers, researchers, etc. The presence of professionalized persons such as these occurs through several vehicles. In planning agencies, such as Regional Planning Commissions, the tasks virtually demand an analytically oriented staff. In line criminal justice agencies like the police, corrections departments, and the courts, this is not as likely. There are, however, two alternatives:

- The agency may have a planning department, i.e., institutional provisions may have been made to draw on the new management technologies. The Office of Programs and Policies of the NYCPD and the Advanced Planning Division of the LAPD are examples.
- Outside researchers or managers may be present. Most commonly the department has agreed to be a host agency for some research and model development. Cases include the St. Louis and New York police departments, where development of patrol allocation models was done in the sixties.

In few of these situations is the potential innovator likely to have decision power over implementation. Thus, a critical transition from innovator to decisionmakers is required in the later stages of the sequence: choice and implementation. The success of the transfer, an exercise in constituency building, depends on several factors:



- The managers' perceptions of the innovating unit, which can range from disdain to frequent consultation.
- The importance of the issues addressed by the model. The Massachusetts police department—and, it's safe to say, most others—do not regard patrol sector redesign as a critical issue. There is a variety of more pressing matters for police administrators to consider, patrol allocation probably seeming among the least important.

  Importance can vary, however, and the application of allocation modeling by the NYPD shows how. It has been useful to them in three specific ways:

  1. For review of staffing levels across precincts citywide. This provides a rational means to allocate the graduating class of the Police Academy, and to decide where attrition and transfer will be permitted to decrease staff.
  2. In negotiations over hours, staffing, and other issues with the patrolmen's union.
  3. In administrative emergencies like cutbacks due to budgetary problems.
- The impact of changes indicated by the model. Substantial savings in resources or other visible results might make model use more attractive, but they may threaten familiar methods, which leads to resistance.
- The appropriateness of the model as seen by the administrators, that is, whether they think it will help with the tasks it addresses. Included are criteria like the usefulness of the information it provides and its logical fit to the real system. These are largely seat-of-the-pants measurements, but ones that test the subjective validity of the model and in large part decide its acceptance.

### Indicators of Successful Implementation

In many ways the examples we found of successful implementation can be characterized by the *absence* of obstacles to implementation already described: the administrator was *not* replaced in the middle of the project by someone who lacked interest in the model, the model's advocate *did* have the political skills needed to build a constituency and was *not* transferred to another job, an appropriate computer system *was* available, and so forth. In addition to these factors, however, several other common characteristics of successful implementations were found. Again, there are no hard-and-fast rules; all the favorable indicators we are about to describe could be present, and yet the implementation effort failed when one of the previously mentioned obstacles arose.

First, in every case of successful implementation in our data base, *the policy issue to be addressed was clearly understood at the start, was considered of high priority by at least some agency administrators, and was well-matched to the capabilities of the model.* This match between the model and the policy issue was achieved either by building the model specifically for the purpose at hand, or by having a good understanding of the characteristics of an existing model.

Second, *the time required to build (or install) the model and collect data for it was known in advance, and the decision to proceed with the model was made in light* *of the anticipated time delay.* This occurred either because the model had been implemented elsewhere previously, so that reasonable estimates could be made, or because the model builder had realistic expectations of the time delays and conveyed them forcefully to the potential user agency. By contrast, models that were not ready for use by deadlines (whether explicit or implicit) were unlikely to be used.

Third, in every case of successful implementation we found, either the *model builder himself or one of his associates (usually either his student or a colleague in the same organization) had a direct personal interaction with the user agency.* In other words, the agency had at least occasional access to someone who had a detailed understanding of the inner workings of the model, not just the capability to understand documentation. While this finding might lead to pessimism about the possibility for widespread implementation of models in the future, we believe it should be interpreted in light of the timing and design of this study. Many of the models described in this report are still too "young" to have had an opportunity to be disseminated very far, and our approach for locating users (namely via written materials and contacts with model builders) was unlikely to turn up examples of implementation unknown to the model builder himself. In any event, the finding raises some questions about the role of the model builder, and these will be explored in the next section.

The final common characteristic of successful implementations was *the availability of a user's manual for the model,* at least in draft form. On the one hand, the production of a user's manual is indicative that the model builder has a generally favorable attitude toward implementation, and therefore reflects a host of other characteristics that we may not have noted explicitly. On the other hand, it is apparent that even the simplest computer model cannot be operated without some sort of instructions on how to do so.

## THE ROLE OF THE MODEL BUILDER

At the present time there is essentially no one to take models before the user public and see to their diffusion, so the task has fallen to the researchers who develop models. There is, however, no assurance that the model builders will always be willing. Indeed, it may be reasonably argued that their involvement should end when the model has had a trial application, since their particular talents lie in the realm of model design. The researchers, then, have been left with a task in which they may have no interest and currently have no identifiable responsibility.

The case of USC's SIMBAD [134], a repeating model for probation decisions, will serve as an example. SIMBAD was developed as a follow-on to research on juvenile justice with no particular agency as a client. When it was completed, some publicizing activity went on in the form of articles and presentation of papers at conferences, and there was one instance of a researcher making site visits to an agency that was particularly interested. On the whole, however, the two-year term of the research itself left the researchers inclined to pursue other projects rather than promote SIMBAD's implementation. With no other proponents, the model has fallen into disuse.

More direct efforts are shown by researchers who contacted potential users. The group at Notre Dame who developed the LEADICS court simulation, for example,



tried to interest courts in neighboring states in their model when work in two Indiana county systems was completed.

An example of active implementation efforts is provided by The New York City-Rand Institute, an organization that assisted the New York City government on a wide range of urban problems and also tested its models in other cities. The Institute supplied continuing model-building skills, technical support, and a genuine commitment to promoting the techniques among line agencies. When combined with favorable conditions among users, the Institute did effect many improvements in a varied mix of city agencies, but it was closed as a consequence of New York City's financial problems.

Other examples are provided by two "centers" for criminal justice modeling: the School of Urban and Public Affairs at Carnegie-Mellon University with JUSSIM, and the Innovative Resource Planning Project (Innovative Resource Planning in Urban Public Safety Systems) at MIT with various police patrol models. Through efforts that are traceable in part to the involvement of researchers, the models developed at these universities have the largest number of applications of all those studied. They are centers in the sense that they distribute information and copies of their models and documentation to users who inquire. In addition, both have periodically conducted their own seminars with representatives of user agencies. To generate interest and familiarize users with the models, these involve lectures on theory and application, comparisons of new and old methods, and the opportunity to operate the programs at the computer terminal. Both centers have provided technical assistance in the past.

JUSSIM diffusion has been an interesting combination of media. Many organizations learned of the model through the literature, conferences, by word of mouth. There were also the seminars at Carnegie-Mellon which, if they did not train modelers, sent more aware and interested analysts back to their home agencies. Finally, graduate students from the University moved into user organizations, carrying the message with them. In the latter two instances, generating the interest in using models was coupled with the important capabilities to operate them and interpret their output.

Looking to the future, it is impossible for the model builders to play as important a role in implementation as they have in the past. Even if they were all active in criminal justice modeling and had continuing funding for implementation purposes, which is not at all the case, widespread implementation cannot be directly and personally assisted by a small number of individuals. The missing element has been an agent to look after model application. By analogy to the physical sciences, there are few *engineers* in the field of criminal justice models.

A possible alternative is for computer software firms and private consultants to fill this gap. The record here has few success stories, however. The cause lies in the propensity to try modeling initially. Technical assistance can help a project along, even save one that might otherwise fail, but it is not the demand-creating force that will interest agency managers in modeling. To date, this demand has not been sufficient to keep such firms viable.

## CONCLUSIONS

Several characteristics of the implementation process have been described as key to the final result of an agency's experience with modeling. First, the introduction of virtually all the models in our sample was a form of *slack innovation*—that which comes from spare personnel time, funds, or other resources, and which is directed at improving operations that are generally considered adequate. Since this kind of change is not planned for, there is frequently the problem that support facilities are in marginal and uncertain supply. And, because the change is to improve on acceptable performance levels, there are strong pressures to resist it and maintain the status quo.

Second, the implementation process is essentially an interpersonal transaction. There is an identifiable advocate in most situations who singlehandedly sees a use for a model, investigates the set of possibilities, and brings one into the organization. The dependence of implementation on his political judgment and energies makes for a precarious situation. The experiences, personalities, and views of managers and researchers also affect the outcome.

Third, the willingness to accept new models is predicated on previous experience or training in the use of sophisticated management techniques. Such preparation appears not only in the decision to try out the model, but in laying the groundwork by previously introducing support facilities like information systems. (Support conditions may be an obstacle even where there is some planning unit willing to give the model a tryout.)

Fourth, an analogy was made, in reference to diffusion, to the physical sciences, suggesting that there are no *engineers* in criminal justice modeling. This means there is no one to take the technology developed by researchers and fit it to particular applications. So far, model builders have filled this gap—when it has been filled—but with some problems. In addition, the search for an appropriate model by user agencies is likely to be brief and rather narrow. In short, diffusion is an uncertain and currently inefficient process.

The thrust of our results with regard to the impact of modeling is that its contribution is still a potential one. The set of agencies that have tried it is a very small part of the total, and the proportion that continued to use it past some initial trial period is minute.

If there is one perspective that policymakers should keep while considering the future of criminal justice modeling, it is that this is one part of a large effort in the improvement of all planning. Models represent a powerful but advanced technology that must be considered in context with complementary techniques and the state of analysis in the line agencies. As part of a planning process, it is clear that such sophistication demands a groundwork of experience and education that, as a rule, is not yet present.

For policymakers who perceive the research into modeling as an investment toward improving the operation of criminal justice agencies, and who are convinced of its potential, there are several courses available. One clear, if complex, policy priority is to promote acceptance of new management techniques by the administrators of line agencies. The suggestion goes beyond modeling per se, touching on the entire area of planning and analysis techniques. This is because effective model use requires some preliminary conditions such that an agency that does not seek innovation can effectively avoid most contact with it.

The problem is a far-reaching one, calling for a change in firmly set management approaches, but at least two responses seem useful. They are (1) to provide training



by exposing upper agency managers (police chiefs, court administrators) to the analytic and planning methods available; and (2) to give them experience with the techniques by promoting researcher-host relationships. The idea is to put more research projects into line agencies where the work itself and the interaction of managers and analysts might reduce some resistance.

The second policy area is the diffusion of models. There is a need for an entity to go between builders and users, providing assistance to agencies seeking out modeling and information on the area for those who are unaware. This will be discussed further in the next chapter.

# Chapter 8

# POLICY IMPLICATIONS

As with most modeling for public policy applications, criminal justice modeling is a "young" field. Its beginnings coincided with the work of the President's Crime Commission in the mid-1960's. For this reason, it is too early to have adequate textbooks on the subject or a large enough group of applications to discern general conclusions about the types of criminal justice models that will prove useful. Optimism about the future course of the field is based on the successes that have been achieved by models in such other applications as business planning, architectural design, and military studies. But whether these examples are actually comparable to the criminal justice system is not really known at this time.

Our review of criminal justice models forces us to the conclusion that very few of them have been used for making policy decisions, and even where they have been so used, there is some question whether the decisions were important enough or different enough from whatever would have been done without a model to justify the cost and effort involved. There are many more examples of abortive applications and optimistic forecasts of future applications that never occurred than there are of successful applications.

We have identified a number of factors that appear to have contributed to this situation. The question is whether the obstacles can be overcome in the future or whether they are inherent characteristics of the criminal justice system that will continue to defeat modeling efforts. In this chapter we review our observations, bringing to bear also the findings of other researchers, in an effort to illuminate the kinds of changes that will have to take place if criminal justice modeling is to become more fruitful. (See especially Ref. 64.)

## THE GENESIS OF MODELS

A great deal of model building is done for its own sake, i.e., it is *self-motivated.* To an analyst, model building is fun and educational, and, if he is a student, it permits him to demonstrate mastery of a technique (such as integer programming) or a computer language (such as SIMSCRIPT or GPSS).

This situation presents both an opportunity for the progress of criminal justice modeling and a danger. The *opportunity* arises from the fact that a model initiated without a policymaker specifying in advance the problem to be solved can be a very good one, or at least a strong conceptual foundation for future models. Many of the criminal justice models reviewed in this report are based on the work of students who were preparing Ph.D. dissertations. These include Heller's police manpower scheduling model, patrol allocation models based on Larson's work, Bammi's beat design model, and many parts of the JUSSIM models.

The *danger* lies in the fact that it is much more interesting for an analyst to design a model from scratch than to use somebody else's. Thus it is possible to have endless cycles of reinventing the wheel, without any advances being made in either





the conceptual underpinnings of the model or its ease of use. In addition, the self-motivated model builder may be entirely satisfied once the model is programmed and verified; he may have no interest whatever in whether it is ever implemented.

At the opposite extreme from self-motivated models are those that are designed specifically for the use of a particular criminal justice agency. While it stands to reason, and has been shown by other research [64], that such models are more likely to be implemented than self-motivated models, the likelihood of implementation does not necessarily extend to agencies other than the one for which the model is designed. Indeed, the computer program itself may incorporate certain unique features of the intended user agency, or the documentation of the program may be in a format understandable only to that agency. Even such simple characteristics as the number and names of the precincts in a police department that contracts for a model can be built into the computer program in such a way that modification for another department may be very difficult. Such a design is easier for the model builder than a more flexible one, and he is likely to pursue it unless he has some motivation to consider the generalizability of the model from the start.

Perhaps the best case is a model that is designed with generality in mind but in conjunction with some "host" agency. The actual problems and data availability in the host agency then serve as a guide to the features that should be included in or excluded from the model, and the possibility of validation exists by virtue of the "host" relationship. But characteristics unique to the host would not necessarily be modeled, or might be included in the computer program as an option.

From the point of view of federal or state criminal justice funding agencies, procedures are needed to encourage the following outcomes at the genesis stage for models:

1. Innovative and potentially useful proposed models are identified and funded.
2. The genesis of the model is explicitly recognized as the first step in a lengthy development process, whether the model builder himself is interested in development or not.
3. Proposed models that duplicate the capabilities of existing models are not funded, or are funded only if they will be modifications of an existing model.
4. The likelihood of eventual implementation is as great as possible.

In general, the Law Enforcement Assistance Administration (LEAA) has not in the past actively supported modeling efforts by Ph.D. students, although some work of this type has been funded as part of a grant to the thesis advisor for a specified modeling project. Traditionally, fellowship grants have been more properly the role of the National Science Foundation (NSF), but in the case of criminal justice models NSF may rightly conclude that the proposal falls in LEAA's domain. Thus, the proposal can "fall through the cracks." (LEAA has a fellowship program, but most fellowship funds have been for individuals who plan to pursue careers within the criminal justice system.) Thus, if this type of work is to be encouraged, LEAA's research arm, the National Institute of Law Enforcement and Criminal Justice, must develop an enhanced capability for reviewing and funding such proposals.

To meet the objectives specified above, proposals for designing new models, whatever their source, should be required to meet the following conditions:

1. Demonstrate a familiarity with existing models generally related to the problem area being addressed.
2. Either demonstrate clearly that no existing model has been designed for the purposes intended for the new model or give a detailed critique of existing models intended for the same purpose, showing why they are inadequate and why they cannot be modified.
3. Identify a suitable host agency[1] and propose funding to support the necessary activities of the host (e.g., meetings with the model builders, computer runs, special data collection efforts, etc.). It is not reasonable to expect the selected host agency to bear the costs of cooperating with the model builder.
4. Indicate how the development stage following design of the model can be carried out. This may be either a commitment by the user to continue work on the model past the design stage, if funded at a later date for this purpose, or a concrete proposal to produce user's manuals and program listings that would permit others to perform this work.
5. Specify the computer programming language to be used for the model. (It appears that most instances in which a model was written in a language unavailable to potential users occurred because no one gave this matter any prior consideration.)

If at all possible, proposals to design new models should be subjected to peer review. This method for evaluating proposals, which is common practice in some federal agencies but not LEAA, involves gathering together a panel of experts to rank proposals. Although we recognize that adoption of such procedures would result in treating proposals related to models differently from others and might delay the granting process, it is important to note that no one reviewer would necessarily know enough details about possible alternatives to a proposed model for him to make a valid determination of whether it should be funded. But several reviewers, each familiar with a number of existing models, could come to such a determination through discussion. In addition, the reviewers may have some familiarity with the proposed host agency.

## DEVELOPMENT AND TESTING

Once a model has been designed and tested in one agency, it is by no means necessarily ready for use by other agencies. First, even if an effort has been made to build a model of general applicability, it may unknowingly have features that are unique to the host agency. Therefore, applications in other jurisdictions are required before the model can be convincingly shown to be generally useful. Second, even if validation of the model has been conducted once, this is not an adequate test. The second phase of model development provides an opportunity for genuine validation (or first validation, if this step was omitted in the design phase).

Third, it is rare for a model builder to have a clear understanding of the steps needed to collect data for his model after testing it in only one agency. The development phase permits these procedures to be discovered and codified for others.

Funding agencies should be prepared from the start to undertake testing and

[1] This may be automatic if the proposal is made by an operating agency.



development of any models whose design they support (contingent on the model turning out to be a good one). As we have mentioned, the model designer may not be interested in performing this task, or he may not be particularly well-suited for it. If so, this should be understood and taken into account when design of the model is first funded. If grants are made available for developing and testing existing models, it seems reasonable to believe that competent organizations and individuals will be found to play this role.

The products of the development and testing phase should be complete user's manuals and program documentation, together with case studies telling what was done in each agency. Such case studies permit readers to make a judgment of whether the impact of the model justifies its cost, instill some confidence that the model actually works and can be understood by agency personnel, and provide guidance for estimating the time delay and cost involved in installing the model. (Obtaining such estimates from the first test application is practically impossible.)

These case studies should be required whether the implementation effort is a success or a failure. While documenting a failure is painful for the grantee (and perhaps also for the funding agency), we will continue to have great difficulty understanding the obstacles to model implementation and how they can be overcome in the future unless such case studies are demanded. One of the major disappointments in preparing the overview of models presented in this report was the difficulty of getting a clear picture of what happened to promising models that were never completed or implemented. Opinions differed among those involved and seemed to be self-serving. Perhaps a contemporaneous account of events as they occurred would have been more understandable and less subject to the distortions of time. Funding agencies do not advance the state of model building by sweeping the failures under the rug.

One reasonable possibility is to have an independent evaluator assigned the sole responsibility for preparing case studies of selected implementation efforts. Without any direct interest in whether the effort succeeds or fails, the evaluator can be expected to complete his task in either case.

## DISSEMINATION

We have already pointed out in Chapter 7 that traditional attitudes in criminal justice agencies often preclude even an attempt to locate a model to assist in a policy decision, and that when a search is made it tends to be "satisficing," stopping when the first reasonable possibility is found. This points to an urgent need for an entity to go between model builders and potential users, to serve as the engineers and salesmen of the model trade. Unfortunately, of all the methods that have been tried to enhance dissemination of models, none has yet shown any notable success. We are therefore left with a clearly identified need without a satisfactory solution.

One problem is that whoever plays the role of the engineer will necessarily have a less adequate understanding of the model than its designer and, to stay in business, he or his firm may be forced to behave in ways that are contrary to the interests of the user agency. Brewer [22], in a more general context, has discussed these individuals, whom he calls entrepreneurs:

Entrepreneurs serve as an important catalyst between policy-makers and those who possess highly specialized talents and skills. To the extent that brokerage dominates a problem-solving activity, confusion and distortion between the research and policymaking processes may be sufficient to defeat the effective prosecution of both.

Contracts are let to solve policy problems within a period of time for an agreed-upon amount. When time and money are used up, a report containing the "answer" to the problem must be produced. Unfortunately, to secure the contract an entrepreneur's vague and overoptimistic self-assessments frequently confuse what the problem is and distort expectations about what an answer might be.

Entrepreneurs are not necessarily evil or pernicious. Moreover, such undesirable types of behavior are rather easily detected and dealt with: the real problem is subtler and consequently more difficult to understand. Salesmen of problem solutions are seldom versed in the technical intricacies of the products they sell. It is probably unreasonable to expect a simulation salesman to be an expert computer technician. Likewise, one does not expect an automobile salesman to have an engineering degree. There are critical differences, however. When one buys an automobile, questions about the technical proficiency of the designers and the excellence of manufacture are more or less resolved because of external professional standards that guide the former and quality-control procedures that assure the latter. Such is not the case with "simulators" and simulations. Not only are there few discernible scientific standards available to aid in one's evaluation of a computer simulation, there is little agreement among professionals in the trade as to what standards are pertinent or ought to be developed. Certain low levels of misfeasance and abuse are tolerable for a profession; however, as the stakes increase, such laxity may become too costly by any measure. For example, building a large-scale simulation and then reselling all or part of it to other users and clients is rational entrepreneurial behavior whose more general consequences may be intolerable. Without adequate standards and procedures for quality control, efforts to maintain proprietary control over a computer simulation may only mask and perpetuate an ill-conceived and poorly executed product. A rational entrepreneurial desire to build a general-purpose urban computer simulation is no guarantee that one can or should be built.

Another result of the fragmented problem-solving process involving users, builders, and entrepreneurs may be a "can do for fee" syndrome. The user buys a complex model or simulation, but he or the person who inherits it does not necessarily understand the whole model, if he ever did. At some point the user wants the model to do something he thought it could or should be able to do, but, for a variety of reasons, he finds out that it cannot. Then he must go back to the salesman and his model-building team, who respond by writing a modification, or an extra subroutine, or entirely new models... for a fee. The procedure has an open-endedness that assures the salesman continuing business; however, it appears that it is the user who is getting "the business."

The essential point is this: Any entrepreneur worth his salt will behave in these ways if he expects to stay in business. Salesmanship may well be the undoing of what promises to be a highly useful problem-solving technique.[2]

These observations, together with a history of infrequent and unsuccessful ventures by commercial firms into the business of selling criminal justice models, suggests that other means must be found for disseminating models. Dissemination consists of several distinguishable activities, including (1) making potential users aware of the existence and utility of models, (2) arranging for the computer programs and documentation to be available when an agency wants them, and (3) providing assistance in implementing them.

[2] From *Politicians, Bureaucrats, and the Consultant: A Critique of Urban Problem Solving*, by Garry D. Brewer, © 1973 by Basic Books, Inc., Publishers, New York.



Even modest efforts toward increasing awareness of models might be fruitful. We have already noted that model builders tend to announce their products in media that are rarely accessed by criminal justice agency personnel, namely technical journals, research conferences, and reports having limited circulation. Funding agencies can guarantee that a larger audience of potential users is reached by negotiating with each grantee a jointly agreeable list of publications and conferences that are of interest to potential users and in which the grantee will be required to disseminate his findings.

To make computer programs and documentation readily available, a single federal center should be established specifically for this purpose. Such a center could be part of a federal agency, similar to the National Criminal Justice Reference Service, or it could be an independent organization such as a university. The personnel of this center should have the capability to identify which models (if any) meet a requester's needs and to provide copies of programs in a form that can be read by the requester's computer system. However, they need not necessarily be able to assist users in installing the model and collecting data. Instead, a list of individuals and organizations that perform such a function could be maintained by the center, together with a list of criminal justice agencies that have already used each model.

Such a center would greatly diminish the phenomenon of a "satisficing" search by providing information about all reasonable alternatives to meet the requesters's needs. In addition, the mere existence of the center might encourage agencies to consider the possibility that a model might be useful to them.

One dissemination device that we believe has already demonstrated great value is the training course in which students have an opportunity for "hands on" use of models. In many instances the documentation of a model may appear quite forbidding to anyone without a technical background, and yet only a few minutes at a computer terminal is required to master the operation of the model. The student can therefore achieve complete confidence that he will be able to operate the model once it is installed and can understand exactly what the model will or will not do. In addition, he has made personal contact with the instructor, who he knows will be able to help him if he runs into any difficulties. After returning to his home agency, the student is very likely to become the "advocate" we described earlier and to see the model through to implementation.

Criminal justice funding agencies can do much to encourage an increase in the number of such training courses. First, they could include a course on police models, or court models, or whatever, in their existing training programs. Second, they could welcome proposals from universities, research firms, and practitioners' organizations to conduct such courses. And finally, they could make funds available to criminal justice agency personnel to cover their travel and incidental expenses for attending such courses. Possibly the federal center that would make computer programs and documentation available could also serve as a training center, inviting appropriate experts to serve as faculty from time to time. Courses held at the center would not only benefit the attendees but also keep the center's staff abreast of the latest developments.

Well-trained users are likely to avoid misapplications of models, such as underlie some examples of nonimplementation, and to have realistic expectations about the time delays that will be incurred in implementing a model.

## RELEVANT CHARACTERISTICS OF USER AGENCY PERSONNEL

We have described how personnel turnover and lack of sustained interest in innovation in criminal justice agencies have thwarted many efforts to implement criminal justice models. No easy or rapid solutions to these problems can be expected. However, the training courses that we have just mentioned as a fruitful dissemination device could also help to establish a better climate for implementation of models. First, they would cause a number of criminal justice agency personnel to view model-related activities as important and worthy of a substantial personal commitment. At the present time, the staff of many planning agencies consists mostly or entirely of individuals who perceive their current assignment as a brief stop on the way to some ultimate career goal. If asked, they will state that in a year or so they expect to have moved to a new position or agency. No wonder, then, that they have little interest in a model that will not be fully implemented for three years!

However, by gaining some expertise with models and meeting similarly placed individuals in other agencies who are engaged in the same type of activities, some of the planners may come to view the successful implementation of a model as worthy of a sustained career commitment. Such a development might do much to reduce the phenomenon of the "vanishing advocate."

Second, it should be anticipated that some of the students at training courses will be agency administrators rather than planners. If they are convinced of the utility of a model, they can do much to instill a sense of purpose in the planners and to assure that the model will be integrated into the policy planning process.

Finally, there are subtle but vital communication gaps between model builders and agency personnel. They speak different languages, have different life styles and modes of dress, and may not share common values and ideals. The opportunity for both groups to meet socially, as occurs in a several-day training course, can help to minimize these gaps, although they will never be totally eliminated.

Even when students of a training course fail to implement the models introduced to them, they may have benefited in other ways. We have already pointed out that one of the major impacts of criminal justice models to date has been to enhance the users' understanding of the operations and data needs of their own agencies. These benefits can accrue even if the model is operated with a data base from some agency unfamiliar to the student, as would ordinarily be the case in a training course. General principles, such as that a certain kind of change *never* leads to improved values of performance measures, can be forcefully impressed on the student by letting him try several examples on an interactive model. Even without having modeled his own agency, he will have gained insight into the policy variables with the greatest leverage for improving performance and the management information he really needs for his work.



## DATA RESOURCES

Until very recently, the field of criminal justice has not had a tradition of producing statistical abstracts or central data repositories that could permit researchers to try out new models under a variety of circumstances. By virtue of a purposeful attack by LEAA, this problem area appears to be gradually diminishing, although much remains to be done. New data sources should serve as a spur to improved model design, as well as serving the other purposes for which they are intended.

There still remains the problem that information systems, even recently developed ones, can be incompatible with the data requirements of criminal justice models. No reasonably imaginable mechanism can assure that information systems now under development will support models yet to be designed, but there is a clear need for the data requirements of existing models to be formulated and described in such a way that they are accessible to the designers of information systems. We do not know any way that this can be accomplished other than to fund a project specifically for the purpose of addressing this problem and developing suitable manuals and other publications.

## CONCLUSION

Criminal justice modeling is a tool that has not been widely used but appears to have considerable potential. We have adopted an optimistic attitude in this chapter, suggesting ways to increase the opportunities for successful use of models in the future. The availability of this report is itself intended to help further this goal. If a reasonably sustained effort is made to foster the dissemination of the best models that already exist, we believe that only a few years will be required to determine whether the current obstacles can be overcome or whether the truth of the matter is that models do not have a cost/effective role to play in the criminal justice system.

# Appendix

# BRIEF DESCRIPTIONS OF MODELS

The models described here in alphabetical order under the headings are those initially selected in 1974 as meeting the criteria listed in Chapter 1. The information included is a short model name, organization building the model, principal persons involved, description, and references. In some cases, the descriptions refer to a family of models designed by a single person or organization. The components are discussed as separate models, where appropriate, in the text. Note that references for models not reviewed in the text have not been updated since 1974. See bibliography for references.

## OVERALL MODELS OF THE CRIMINAL JUSTICE SYSTEM

1. CALIFORNIA CRIMINAL JUSTICE SYSTEM
   Space-General Corp.
   John Kuhn

A systems analysis and cost/effectiveness study of the California system of criminal justice which attempts to apply the techniques of systems engineering to the problems of crime and delinquency. Computer simulation was used to calculate the cost/effectiveness of system policies or of other operating conditions.
*Ref.:* *172.*

2. CANJUS
   Ministry of the Solicitor General and Secretariat
   of Treasury Board
   R. George Hopkinson

This is a Canadian application of JUSSIM (see Model 7).

3. CRIMINAL JUSTICE SYSTEM TRAINING MODEL
   Georgia Institute of Technology
   School of Industrial and Systems Engineering
   Willard R. Fey, Harrison M. Wadsworth, Donovan B. Young

An interactive computer model of the CJ system suitable for use in training of CJ planners. Based on Atlanta, Part I crimes. Consists of a group of equations representing the forces and influences between conditions and decisions in a CJ system. Outputs crime rate, number of police officers, corrections budget, etc., starting at a specified year. For training—not to be used for policy decisions.
*Refs.:* *61,62.*

4. CRIMINAL JUSTICE SYSTEMS MODEL
   Denver Regional Council of Governments
   Lloyd J. Alvarado

This is an operations research model for "full-scale criminal justice planning." Using a steady-state, open Markovian type formulation, the model positions the important decision variables and their associated constraints, for example, limitations in resources, according to functional stages considered sequentially. The only information about previous stages relevant to selecting policy values for the current decision variables is summarized by an indicator which may be n-dimensional. The efficacy of a policy change is judged on its impact in the present stage and in all subsequent stages.

This model is being integrated into the planning process. The Colorado State Standards and Goals Task Force on Systems will serve as advisory board charged with recommending further utilization.

*Ref.: 1.*

5. CRIMINAL JUSTICE SYSTEM SIMULATION FOR LOS ANGELES COUNTY
   University of Southern California
   Alexander McEachern

Objective was to develop a prototypical planning model on the basis of information available describing the operation of the CJ system in Los Angeles County. Would be used for developing realistic projections of activity levels at different points in the CJ system and for developing and evaluating objectives and actions to achieve them.

*Ref.: 133.*

6. DOTSIM (VENTURA COUNTY)
   Public Safety Systems, Inc.
   Robert W. Poole

A CJ resource allocation model simulating the movement of offenders through the system (DOTSIM is an acronym for Dynamic Offender Tracking Simulation). The user can evaluate operation and interactions of the total CJ system or any portion, including feedback of repeat offenders; queuing effect; offender processing delays; resources and costs; performance measures of agencies, programs, policies; random behavior of crime occurrence, processing time, etc.; processing strategies which differentiate between recidivists and first offenders. DOTSIM differs from JUSSIM in that JUSSIM is a linear model using aggregate flows which divide according to branching ratios, while DOTSIM simulates the movement of individual offenders, allowing calculation of delays, queuing, etc.

*Refs.: 94,96.*

7. JUSSIM
   Carnegie-Mellon University
   Alfred Blumstein

An interactive computer program enabling CJ system designers to assess the consequences of possible changes. The model focuses on the flow of offenders through CJS processing stages. Offenders are aggregated into groups according to crime type or other characteristics relevant to how they will be processed. Resources are applied at each stage or flow path; these represent costs or manpower needed. The



planner provides a base case file describing the current system and then generates and evaluates test cases interactively. Requires data on branching ratios at each point in the CJ system. JUSSIM II is an extension of the original model which adds the feedback effects of recidivism. Model is widely implemented, sometimes under different names.

*Refs.: 10,11,12,43,63,173.*

*Related Refs.: 18,19,21,28,29,158.*

8. JUVENILE JUSTICE SYSTEM
   Ohio State University, College of Engineering
   Ned B. Wilson

A flow model describes the movement of juveniles through the system, using a Markovian representation. There are also police and juvenile court submodels. This model was used in 1971 by the Ohio Youth Commission. A new director of OYC was appointed soon after, and he did not continue its use. The model has been dormant for about three years.

*Ref.: 191.*

9. PHILJIM
   Government Studies & Systems (Mathematica)
   Charles I. Goldman, Benjamin H. Renshaw

A model designed for overall CJ system simulation, developed after acquisition of an early version of JUSSIM. Improvements were made in the representational features of JUSSIM, in ease of use and understandability by users, and as required for planning rather than instructional purposes. PHILJIM is also being used in Alaska, for the Department of Corrections, in developing five adult and five juvenile regional models. The Denver Council of Governments has acquired PHILJIM and used it. Also in various stages of use in Sacramento, California, and Austin, Texas. PHILJIM was chosen to be included in the PROMIS system package.

*Refs.: 21,43,158.*

10. PRISON/PAROLE SYSTEM SIMULATION
    Naval Postgraduate School, Monterey, California
    Davis W. Anderson, Edward A. Brill

A simulation model, viewing a prison/parole system as a feedback process for criminal offenders. Transitions among the states in which an offender might be located are assumed to be in accordance with a discrete time semi-Markov process. Projected prison and parole populations for sample data and applications of the model are discussed. It is possible to estimate future prison/parole population as a function of first offenders per year. By varying parameters (discharge rate and time from prison) one can assess changes in prison/parole/ex-convict populations or one can estimate average cost per year of different systems. This model has been used only with arbitrary parameters and has had no actual application.

*Ref.: 3.*

## POLICE MODELS

11. DETECTIVE ALLOCATION MODEL
University of Illinois
Deepak Bammi

This model is used to maximize the expected number of cases solved in any division of the Detective Bureau by the optimal allocation of detectives to cases. The allocation is based on the historical probability of solving a case as a function of the type of case and the number of days spent on it to date. The probabilities are obtained from past experience. If a case is delayed either before or after investigation has begun, the historical probabilities are multiplied by an exponential decay factor. Detectives who solve cases during a day are assigned to new cases. A computer program had been written but no documentation was available as of December 1974.

12. HYPERCUBE QUEUING MODEL
The New York City-Rand Institute and
Massachusetts Institute of Technology
Richard Larson

This is a spatially distributed queuing model that can be useful in aiding police planners to locate their patrol units and to design response districts for their city. The exact hypercube model is a computationally efficient algorithm that evaluates numerically the performance of systems having up to 15 patrol units. The approximate hypercube model allows almost any number of cars to be handled. The measures of performance computed include: regionwide mean travel time; workload imbalance; fractions of dispatches that are interdistrict; workloads of each patrol unit; mean travel time to each small geographical reporting area and to each district; mean travel time of each patrol unit; fraction of responses that are interdistrict. These vary according to the user's specification of the reporting areas belonging to the patrol beat of each unit (beats may overlap) and the relative amount of patrol time spent in each area. Available as a batch program or with an interactive monitor.

Larson also designed a patrol allocation model that served as a basis for Model 16, and a simulation model (Model 20).
*Refs.: 24,30,37,99,121,123,126,127,154,187.*

13. LEMRAS
IBM (withdrawn)

The Law Enforcement Manpower Resource Allocation System (LEMRAS) analyzes information relating to called-for-service activity over user-defined geographic areas and time periods. The forecasted activity may be analyzed to derive information on the number of patrol units required to answer the calls. LEMRAS was withdrawn at the end of 1974. According to IBM it was a useful but not widely used program. It does not run on current IBM computer systems and, as a batch program, was not considered suitable for current requirements of law enforcement agencies.
*Ref.: 93.*



14. LINEAR PROGRAMMING APPROACH TO POLICE SCHEDULING
The New York City-Rand Institute
Peter Kolesar, Kenneth Rider, Thomas Crabill, Warren Walker

This approach to the problem of matching the number of patrol cars on duty to the demand for service focuses on varying two sets of decision variables—the number of cars assigned to work in a specific tour and their mealtime assignments. First, queuing or allocation models are used to estimate the number of cars required on duty during each hour of the day so that a specified standard of service is maintained during that hour. Second, this model solves an integer linear programming problem to obtain the tour assignments and mealtimes which will use the fewest cars to meet these requirements. The linear program also assures that the schedules satisfy other constraints specified by the Police Department. This procedure can be repeated until feasible and desirable schedules are produced.
*Refs.: 111,112.*

15. A PARAMETRIC MODEL FOR RADIO CAR ALLOCATION
The New York City-Rand Institute
Kenneth Rider

This model was intended to circumvent some of the difficulties encountered with standard allocation models, enabling the user to incorporate a trade-off between "efficient" and "equitable" service policies into the allocation procedure. It was never fully developed.
*Ref.: 162.*

16. PATROL CAR ALLOCATION MODEL
The New York City-Rand Institute
Jan Chaiken, Peter Dormont

Allocates patrol units to precincts based on measures of performance related to the objectives of patrol operations, including: the amount of time units spend on preventive patrol; average response time to calls for service; average time a call is delayed before a unit is dispatched. PCAM is an analytic model that can be used in either the interactive or the batch mode. It is assumed that units engage in three types of activity: answering calls (or coming across incidents); preventive patrol; administrative activities that place the unit out of service (meals, arrest processing, etc.). A descriptive mode calculates performance for the given assignment of units. A prescriptive mode calculates the number of units needed in each precinct (a) to satisfy specified minimum performance requirements, or (b) to minimize a specified performance measure, subject to available resources.
*Refs.: 32,33,34.*

The model is an outgrowth of work by others reported in Refs. 2,50,93,118,119,136,-137,144,145,167,182.

17. PATROL CAR SIMULATION
The New York City-Rand Institute
Peter Kolesar, Warren Walker

This simulation, written in SIMSCRIPT II.5, is an imitation through time of the events which occur during police patrol operations. The program maintains a map

of the region being simulated, monitoring the location of incidents and patrol units simulated. A simulation run covers a fixed time period such as a series of duty tours over a series of days. The user specifies conditions at the start of the time period, including the number and location of units and the operating and dispatching rules being followed. Then, using an internal timing mechanism, the program carries out the assignment of units to jobs, their travel times to the jobs, the queuing of calls, if any, etc. While carrying out this sequence of events, the program records and stores statistical summaries on response times, patrol availability, workloads, etc. Job streams can be developed separately from actual job histories, projections of future call patterns, or results of probabilistic models of call generation. Scheduled unavailabilities, such as meals, can also be simulated.
*Refs.: 47,48,49,80,114,115.*

18. PATROL RESPONSE SIMULATION
California State University, Fullerton, Department of
Quantitative Methods, School of Business Administration
and Economics
William Heitzman

A simulation model was written in FORTRAN IV to simulate patrol operations in the Van Nuys Division of the Los Angeles Police Department. The model utilizes a patrol reserve for high-priority calls. (Calls can belong to one of four priority levels.) The effect of the size of the reserve number for the highest priority call with various total patrol allocations is investigated. Given the response-time criteria for all four priorities, an optimal policy can be selected that minimizes total force size. The arrival rate of calls, response speeds, and service rates can be varied in the model.

19. PATROL SCHEDULING AND ALLOCATION MODELS
St. Louis Police Department
Nelson Heller, Thomas McEwen

This project resulted in the development of LEMRAS by IBM. The St. Louis work on police models is probably the longest effort of its type, dating back to 1964 when the department acquired a computer to improve its information system. A Resource Allocation Project was established in July 1966, stemming from Crowther's 1964 study of how to use the computer for police manpower allocation.

The allocation programs have two components, prediction and queuing. For prediction the city was divided into 490 small areas about the size of several blocks. Dispatchers' records were coded according to the area of occurrence and the program counted how many incidents in each of eight crime categories occurred in each area. These counts were projected into the future by means of exponential smoothing in order to predict the call rate in each area for each hour of the week. Service times were similarly predicted. A precinct or other geographical area can be constructed from the small areas. The queuing model calculates from the predicted workload the percentage of callers who will experience a delay, given how many cars are in the field. By consulting computer-generated tables, one can estimate how many cars should be fielded in order to hold the percent of calls delayed to any standard selected.

The St. Louis analysts also developed computerized techniques for constructing



work schedules for police officers. These techniques allocated manpower by watch and day of week in proportion to demand for service, assigned on-duty and off-duty days, and provided ways to schedule compensatory days off for overtime worked and to minimize the schedule's sensitivity to absences. The scheduling techniques were first applied to the Evidence Technician Unit.
*Refs.: 25,50,81,82,83,84,85,136,137,167.*

20. PATROL SIMULATION
Massachusetts Institute of Technology
Richard C. Larson

This simulation model was writtin in the MAD language, which is no longer supported on any computer system. It was reprogrammed in PL/I by Urban Sciences, Inc., and applied to the Boston Police Department. Many of its design characteristics were to have been incorporated in the Mathematica simulation (Model 21).

The city (or portion of the city) is divided into small geographical areas called *atoms*. Patrol sectors are defined as collections of atoms and may overlap. Jobs are generated internally according to average arrival statistics provided by the user. Based on information about dispatching policies, for which a variety of options are permitted, the simulation tracks the response of patrol cars, queuing of calls, etc., and calculates aggregate statistics on workloads, travel times, queuing delays, inter-district dispatches, and other performance measures. Service on a low-priority call can be preempted by a higher-priority call. The user can specify the relative amount of preventive patrol time spent by each car in each atom.

Larson also designed a patrol allocation model that served as a basis for Model 16, and the Hypercube Queuing Model (Model 12).
*Ref.: 119,181.*

21. PATROL SIMULATION
Mathematica, Inc.
Saul Gass

In order to research the patrol and dispatch functions of urban police departments, a general-purpose computer simulation was to be developed to provide a means for testing and evaluating proposed alternative policies. Application was planned with the Washington, D.C., Police Department. After a one-year effort concluded early in 1973, the six-month follow-on to apply the simulation was delayed until mid-1974. The model was never completed.
*Ref.: 107.*

22. POLICE EMERGENCY RESPONSE SIMULATION
Polytechnic Institute of Brooklyn
Norbert Hauser, Gilbert Gordon, Julius Surkis

Several computer simulation models were developed for the New York Police Department under LEAA grant. Two separate models were developed: (1) the turret board model representing the operation of the input processing sector of the Communications Center; (2) the dispatching-field resource model representing the selec-

tion, assignment, and dispatching of field resources, the field response after dispatching, and the final disposition of the call.
*Ref.:* 77.

23. POLICE PATROL FORCE DISTRIBUTION
Miami University, Department of Economics, Oxford, Ohio
J. Fred Giertz

The purpose is to use techniques of economic analysis to investigate the distribution of police among the various districts of a city. The model investigates police distribution versus index crimes and demographic/economic characteristics in Chicago. The reference also discusses alternate patrol distribution plans in Los Angeles. It establishes a decision model utilizing the hypothesized preventive effect on crime of police patrol to analyze the various plans, and proposes a more precise measure of the benefits of police protection involving the reduction in the cost of index crimes to society.
*Ref.:* 67.

24. SUPERBEAT
Illinois Institute of Technology
Spencer Smith, Deepak Bammi

Superbeat is a collection of several models that will be described individually.

*Beat design:* This is an analytic model that calculates performance measures of a system of patrol cars, taking into account the geographical distribution of the cars, and recommends a beat design that minimizes response time for the study region. The region is divided into reporting areas, and the user specifies the travel time between adjacent areas. Patrol beats are defined as collections of reporting areas and may overlap. Cars may be unavailable for calls for service or for other unavailablities. The model assumes that the unavailabilities of different cars are independent.

A heuristic iterative procedure is used to identify a beat design that minimizes a measure of average response time. This can be an average for all calls or a weighted average of the response time for priority and nonpriority calls. Optimization is subject to constraints on the maximum travel time within beats.
*Refs.:* 5,6,7,8,170,171.

*Simulation:* Written in FORTRAN, this model has a geographical structure similar to the beat design model. The city (or portion of the city) is divided into reporting areas, and the user specifies as input the travel distances between areas. Travel speeds may vary by time of day and type of call. The job stream, representing calls for service and other unavailabilities, is ordinarily generated externally. The model tracks the response of cars to jobs, queuing delays, etc., and calculates summary statistics on workload, travel times, and waiting times. Service on low-priority calls can be preempted to respond to higher priority calls.
*Refs.:* 170,171.

*Manpower scheduling:* This is an optimization model for obtaining a schedule for patrol forces that minimizes average response time over all tours of the week, given a fixed pattern of available personnel and a specified set of work and recreation patterns. Some of the men work fixed tours and others work rotating tours. The program specifies how many officers should be assigned to each of the work-recreation patterns input by the user.
*Refs.:* 170,171.



25. TIME-DEPENDENT (DYNAMIC) QUEUING MODEL FOR RADIO CAR ASSIGNMENT
The New York City-Rand Institute
Peter Kolesar, Kenneth Rider, Warren Walker

Calculates how queuing delays vary over time when the number of patrol cars on duty and the call rate vary by hour. Used to evaluate schedules for patrol cars.
*Refs.:* 112,113.

## COURT MODELS

26. ANALYTIC MODELS OF CRIMINAL COURT OPERATIONS
Cornell University School of Business Administration
Norman Lyons

In his Ph.D. thesis, Lyons developed a series of analytic models applicable to the work planning problems of a large criminal court system and tested these models on a simulated version of an actual criminal court system in order to gain insight into court system problems. Two simulation problems were developed—one for long-run and one for short-run planning. These were used to test the projections of the analytic models, then, together with the analytic models, to examine the impact of pure scheduling policies and of system policy changes. A number of system policy recommendations are made. The analytic models include: queuing theory models for long-run planning; linear programming models for long-run planning of judge requirements; chance-constrained programming models for long-run planning; a short-run scheduling model. The Allegheny County (Pennsylvania) court system is used as an example.
*Ref.:* 132.

27. BASIC COURT SYSTEM
IBM (withdrawn)

The Basic Court System is superseded by System/370 Justice (see Model 37) and is no longer available. See Model 32 for a successful application in Baltimore.

28. CANCOURT I
University of Toronto, Centre of Criminology
Robert G. Hann, Lorne P. Salzman

A computerized Monte Carlo court system simulation model capable of simulating the simultaneous processing of a large number of cases through a court system consisting of a number of groups of courts or quasi-courts, each group having different or overlapping jurisdictions for handling pretrial, preliminary inquiry, grand jury, trial and/or appeal functions. The model simulates the resources used and the

backlogs and dispositions experienced by cases at major points in the system.
*Refs.: 70,71,165.*

29. COURTEX
Funded by California Council on Criminal Justice
Peter Haynes

This is a noncomputerized gaming simulation consisting of three exercises: (1) a court policies exercise which addresses decisionmaking in a trial court situation; (2) a felony case processing exercise dealing with problems in criminal case calendar management; (3) a civil case processing exercise dealing with civil case calendar management. These are training exercises to illustrate real-world problems in court administration. The feasibility of adequately simulating the court system (as in LEADICS and JUSSIM) is questioned because of lack of understanding of aspects that are important but lie beyond the formal court process (e.g., operations of attorneys). Present simulations are described as "simplifications which have some utility but which require extensive development to accurately reflect case processing operations of a court." It is pointed out that such simulations are also expensive. The approach here is a gaming simulation utilizing a basic model of operations which is correct but not excessively detailed, together with extensive freedom for role actors to make decisions about the operations of the organization. Their decisions have a quantitative impact upon the court system which "has to be lived with."

According to the reference, COURTEX has proven useful for teaching about court administration in the contexts of both a university program and a training program.
*Refs.: 78,79.*

30. COURT FLOW MODELS
The New York City-Rand Institute
John Jennings

These models include analytic and simulation models of case flow, scheduling, and courtroom activities, which, however, were not documented as computer programs accessible to others.
*Refs.: 100,101,102,103.*

31. COURTSIM
Institute for Defense Analyses
Joseph A. Navarro, Jean G. Taylor, Robert H. Cohen

COURTSIM simulates the processing of adult felony cases in the District of Columbia. The GPSS programming language and 1965 data were used. Runs were made to test effects of changes such as increasing grand jury resources and varying procedural rules. The outputs consisted mainly of time delay reports.
*Ref.: 179.*

32. CRIMINAL COURT STATUS INFORMATION SYSTEM
Supreme Bench of Baltimore City
A. LaMar Benson

Under several LEAA grants, Chief Judge Dulany Foster obtained IBM's Basic Court System and adapted it to a Criminal Court Status Information System-Case Scheduling Model to assist the orderly and expeditious flow of criminal cases through Baltimore courts. Fully operational, it may be expanded to include critical path calculations. Court backlog was substantially reduced in one year; however, this was due to better information flow and not to the use of a model. The scheduling was still done manually.
*Ref.: 14.*



33. COURT ADMINISTRATION
The Institute of Judicial Administration, Inc.
Paul Nejelski

The references allude to several court models but no details were obtained.
*Refs.: 140,146,174.*

34. JUDICIAL SYSTEM SIMULATION
Arizona State University, Center of Criminal Justice
J. Kent Butler, G. H. Bruns

A current study to develop an applied computer simulation model of the flow of civil cases through a specific judicial system, utilizing variations of the GERTS language. Data from the Superior Courts of Maricopa County will be collected to support the model. Solutions to congestion and delay will include more strict enforcement of certificates of readiness; master calendaring; the preparation of impact statements for proposed legislation. It is planned to develop as a management aid for planners and administrators.
*Ref.: 160.*

35. JUROR MANAGEMENT
Case Western University and
Court Management Project, Cleveland Bar Association
Leon S. Lasdon

After extensive study of juror utilization in the Cuyahoga County Court of Common Pleas, Cleveland, Ohio, juror use and statistical programs were developed and used to simulate jurors in use during a day and to show possible juror-day savings. Some of the author's recommendations were accepted, and significant improvements resulted ($89,000 annually, or about 28 percent of total juror cost in 1972).
*Refs.: 105,128.*

36. JUROR MANAGEMENT SIMULATION
University of South Florida, College of Business Administration
Michael White

A simulation model of a typical four-judge district court utilizing a jury pool was constructed. The model was modified to test the effects of a change to a "multiple voir dire" approach to juror management. Programmed in GPSS. Durations were simulated as uniform random variables.
*Ref.: 188.*

37. JUSTICE/370
IBM

This is basically a management information system providing an automated means for maintaining and reporting the status of each person and case involved in the justice process. It can provide a data base for, and can serve as the first step toward, a total justice information system encompassing the courts, law enforcement agencies, and correctional institutions. The system can also serve as a base for the user who plans to add on-line programs to run in a terminal-oriented environment. Reasons for case backlogs and delays may be analyzed by user-written programs. Available in batch mode as a replacement for the IBM Basic Court System.
*Ref.: 176.*

38. LEADICS
University of Notre Dame
Leslie Foschio, James Daschbach

A systems analysis directed at identifying the causes and possible cures of unnecessary delay in the processing of criminal cases in state courts resulted in a simulation model to facilitate evaluation of the effects of proposed changes in the system. Both legal and engineering skills were used. (The name stands for Law-Engineering Analysis of Delay in Court Systems.)

The Superior Court of New Jersey, Hudson County, had a minicomputer-based information system and made a study of tailoring LEADICS for use with this system (under LEAA grant 72-DF-02-0022). The LEADICS simulation is not perceived as a decision tool for continuous use, but as a part of ad hoc court studies for two Indiana counties.
*Refs.: 141,178.*

39. OKLAHOMA CITY COURT SIMULATION
University of Texas at Dallas
Raymond P. Lutz, Jerry G. Metcalf

A GERT network was developed for the Municipal Court of Record of Oklahoma City. It analyzed the judicial time required to try the alcohol-related cases involving motor vehicles. Objectives were to define the judicial process, determine actual and potential procedural bottlenecks, analyze and forecast judicial hours required under existing and proposed procedures. The results were used to change court operations to significantly increase the number of individuals tried per month and to stabilize the court backlog.
*Ref.: 131.*

40 PAROLE DECISION-MAKING
National Council on Crime and Delinquency
Leslie T. Wilkins

Done in collaboration with the U.S. Board of Parole, this project aimed to develop, test, and demonstrate programs of improved information for decisionmaking by providing objective, relevant information for individual case decisions, and by summarizing experience with parole as an aid to improved policy decisions. The use of an on-line system was explored. Further aims included the definition of



paroling objectives, the description of paroling decisions, the testing of relations between information available for decisions and the decision outcomes, the evaluation of new procedures, and the dissemination of results to parole systems in the United States.

"Simulation" in this case refers to a one-time social experiment (something like a management game) in which parole-involved administrators made decisions on a set of fictionalized parole cases. This interesting and high-quality work has not been discussed in detail in the text because the use of computers was incidental to the modeling effort, and the computer programs themselves are not models.
*Refs.: 68,87,88.*

41. PROMIS
Peat, Marwick, Mitchell & Co.

A Prosecutor's Management Information System (PROMIS) was developed for the U.S. Attorney's Office, District of Columbia. The only "modeling" component of PROMIS is a method for judging the importance of cases so that the more important cases will not inadvertently fail to be tried. Four criteria were established: (1) the seriousness of the crime using a version of the Sellin-Wolfgang scale which primarily measures the amount of personal injury and property loss; (2) the seriousness of the defendant's criminal record based on the Gottfredson scale which measures primarily the number and density of prior arrests, indications of a history of drug abuse, and the offense committed; (3) the age of the case; (4) the probability of winning the case, assessed subjectively by the prosecuting attorney who originally screens the case and files the charges. Fourteen days before each trial date the computer provides a ranking and summary of each case on the calendar. This is updated one day before trial.
*Ref.: 177.*

42. SCHEDULING OF NEW YORK CRIMINAL CASES
Cornell University
Steven Patent

The purpose of this research for Patent's M.S. thesis was to investigate whether improved performance relative to due-date can be obtained with fixed resources for a court system handling criminal cases. The data and structure modeled by computer simulation are based on the courts that handle felony cases in Manhattan. The main concern is the time taken in processing a defendant's case.
*Ref.: 149.*

43. SIMBAD
University of Southern California
Alexander McEachern

The basic objective was to introduce new knowledge and new technology into the practice of probation. Participating departments were to have remote, real-time access to a computer which would provide estimates of success for disposition and treatment decisions at any point in the probation process. SIMBAD is an acronym for "Simulation as a Basis of Social Agents' Decisions."

At one point, SIMBAD was operational on USC's 1130 computer, on-line. SIM-

BAD had moved through development to readiness for field testing in 1968; however, no follow-on funding was secured from LEAA. The program was mounted in Oregon and some data collection started, but the program was not implemented.
*Refs.: 133,134.*

## CORRECTIONS MODELS

44. EVALUATION OF STATE CORRECTIONAL PROGRAMS
Georgia Institute of Technology, School of
Industrial and Systems Engineering
James T. Pittman

Applied quantitative techniques to evaluation of correctional programs in Georgia, using the Markov assumption. Outputs are for four classes of felonies (assult, burglary, larceny, and robbery) and include: expected proportion of current convicted population that will be in prison at any future date; expected number of subsequent crimes for which the average member of the current convicted population will be incarcerated; the equivalent annual cost to society, per criminal career. Estimates of first-offender input are used with the model to predict system population for a ten-year period.
*Ref.: 153.*

45. FEDERAL CORRECTION SIMULATION MODEL
Solicitor General of Canada
Robert Hann (Systems Dimensions, Ltd.)

FCSM is a dynamic stochastic model. Although the model pays particular emphasis to the flows of offenders through the various elements of a correctional system (institutions, parole, etc.), it also contains sections dealing with recidivism and sentencing. The model will be used for two purposes—short- and long-range policy planning, and predicting (for budgeting purposes) the populations and flows of offenders within and between the various elements of the Canadian Correctional System.
*Refs.: 71,73,74,75,76.*

46. JAIL POPULATION MANAGEMENT
American Justice Institute
Santa Clara Criminal Justice Pilot Program
Emmett J. Burke, Robert C. Cushman,
William A. McConnell, Robert V. Ragsac

The use of Markov chain analysis permitted the construction of two general types of predictive jail models which can take into account the various types of random input (bookings) and output (releases). The manner in which the jail population progresses from one level to another and the predictability of these transitions is described. Numerical data illustrate the use of the model as a diagnostic and predictive tool, and the basic requirements for a formalized control and decision model are presented.
*Ref.: 23.*

# BIBLIOGRAPHY


1. Alvarado, Lloyd J., "On the Utility of a Certain Type of Operations Research Model for Full-Scale Criminal Justice Planning," presented at the 44th national meeting of Operations Research Society of America (ORSA), November, 1973.
2. *An Analysis of the Los Angeles Police Department's Patrol Car Deployment Methods*, Engineering School report, University of California at Los Angeles, June 1974.
3. Anderson, D. W., and E. A. Brill, *A Prison/Parole System Simulation Model*, Naval Postgraduate School, NTIS[1] AD-764 544, March 1973.
4. Ansoff, H. Igor, and Robert L. Hayes, "Role of Models in Corporate Decision-Making," *OR '72*, North-Holland Publishing Co., 1973, pp. 131-162.
5. Bammi, D., "Design of Police Patrol Beats To Minimize Response Time to Calls for Service," Ph.D. thesis, Illinois Institute of Technology, December 1972.
6. Bammi, D., "Allocation of Police Beats to Patrol Units To Minimize Response Time to Calls for Service," *International Journal of Computers and Operations Research*, Vol. 2, 1975, pp. 1-12.
7. Bammi, D., and N Thomopoulos, "A Predictive Model for the Police Response Function," University of Illinois, submitted to *Operational Research Quarterly*.
8. Bammi, D., and Spencer B. Smith, "Design of Police Patrol Beats to Minimize Response Time to Calls for Service," submitted to *Operations Research*.
9. Belkin, J., R. Brauner, and R. G. Cassidy, "Programming Approach to the Criminal Justice System," *Infor*, Vol. 13, No. 3, October 1975, pp. 227-247.
10. Belkin, J., and A. Blumstein, *Methodology for the Analysis of Total Criminal Justice Systems*, Urban Systems Institute, Carnegie-Mellon University, November 1970.
11. Belkin, J., A. Blumstein, and W. Glass, *JUSSIM, An Interactive Computer Program for Analysis of Criminal Justice Systems*, Urban Systems Institute, Carnegie-Mellon University, rev. July 1974
12. Belkin, J., A. Blumstein, and W. Glass, *JUSSIM II, An Interactive Feedback Model for Criminal Justice Planning*, Urban Systems Institute, Carnegie-Mellon University, June 1973.
13. Bennett, Wayne, et al., *The Use of Probability Theory in the Assignment of Police Patrol Cars*, Edina Police Department, Edina, Minnesota, July 1970.
14. Benson, A. L., *Manual for Case Scheduling, Criminal Assignment Office*, Supreme Bench, City of Baltimore, January 1974.
15. Blumstein, A., "Management Science to Aid the Manager: An Example from the Criminal Justice System," *Sloan Management Review*, Massachusetts Institute of Technology, Vol. 15, No. 1, Fall 1973, pp. 35-48.
16. Blumstein, Alfred, *Seriousness Weights in an Index of Crime*, Carnegie-Mellon University, June 1974.


[1] NTIS = National Technical Information Service, U.S. Department of Commerce, Springfield, Va.

17. Blumstein, A., R. Christensen, S. Johnson, and R. Larson, "Analysis of Crime and the Overall CJS," Chapter 5 of *Task Force Report: Science and Technology,* President's Commission on Law Enforcement and Administration of Justice, GPO, June 1967.
18. Blumstein, A., and R. Larson, "Models of a Total Criminal Justice System," *Operations Research,* Vol. 17, 1969, pp. 199-232.
19. Blumstein, A., and R. Larson, *Models of a Total Criminal Justice System,* Institute for Defense Analyses, NTIS PB 208 096, November 1970.
20. Bohigian, H. E., *The Foundations and Mathematical Models of Operations Research with Extensions to the Criminal Justice System,* John Jay College of Criminal Justice, Ph.D. dissertation, 1971, The Gazette Press, Inc.
21. Braybrook, W., C. Goldman, and B. Renshaw, "Application of an Indicator Based Planning System and Simulation Modelling to a Criminal Justice Planning Process: A Philadelphia First Try," presented at the 41st National Operations Research Society of America Meeting at New Orleans, April 1972.
22. Brewer, Garry D., *Politicians, Bureaucrats and the Consultant: A Critique of Urban Problem Solving,* Basic Books, New York, 1973.
23. Burke, E. J., et al., *Jail Population Management: Diagnosing and Predicting Changes in Occupancy Levels,* prepared for NILECJ by the American Justice Institute, March 1973.
24. Campbell, Gregory L., *A Spatially Distributed Queuing Model for Police Patrol Sector Design,* Massachusetts Institute of Technology, NTIS PB 226 133, June 1972.
25. Capaul, J., N. Heller, and E. Meisenheimer, "A Stochastic Model for Allocating Police Patrol Units to Districts which Reflects Each District's Rates of Injury, Property Loss, and Fear," St. Louis Metropolitan Police Department, presented to the 38th national ORSA meeting, 1970.
26. Carlin, Jerry L., Colin L. Moodie, "A Comparison of Some Patrol Methods," *Police,* Vol. 16, No. 12, August 1972, pp. 27-31.
27. Casey, Patrick J., *Determining Police Patrol Car Requirements by Computer SIM,* Arizona State University, Department of Industrial Engineering, January 1968.
28. Cassidy, R. G., "A Systems Approach to Planning and Evaluation in Criminal Justice Systems," CANJUS Project Report 18, Ministry of the Solicitor General, Statistics Division, Toronto, October 1974.
29. Cassidy, R. G., and B. W. Johnson, "Use of the CANJUS Model for Planning and Evaluation in the Canadian Criminal Justice System," Report 5/73, Ministry of the Solicitor General, Statistics Division, December 1973.
30. Chaiken, Jan M., *Hypercube Queuing Model: Executive Summary,* The Rand Corporation, R-1688/1-HUD, July 1975.
31. Chaiken, Jan M., *Patrol Allocation Methodology for Police Departments,* The Rand Corporation, R-1852-HUD, August 1975.
32. Chaiken, Jan M., and Peter Dormont, *Patrol Car Allocation Model: Executive Summary,* The Rand Corporation, R-1786/1-HUD, September 1975.
33. Chaiken, Jan M., and Peter Dormont, *Patrol Car Allocation Model: User's Manual,* The Rand Corporation, R-1786/2-HUD, September 1975.
34. Chaiken, Jan M., and Peter Dormont, *Patrol Car Allocation Model: Program Description,* The Rand Corporation, R-1786/3-HUD, September 1975.



35. Chaiken, Jan, and Richard Larson, "Methods for Allocating Urban Emergency Units: A Survey," *Management Science,* Vol. 19, No. 4, December 1972, pp. 110-130.
36. *The Challenge of Crime in a Free Society,* President's Commission on Law Enforcement and Administration of Justice, GPO, 1967.
37. Chelst, Kenneth, *An Interactive Approach to Police Sector Design,* Innovative Resource Planning in Urban Public Safety Systems, Massachusetts Institute of Technology, WP-03-74, March 1974.
38. Chelst, Kenneth, *Implementing the Hypercube Queueing Model in the New Haven Department of Police Services: A Case Study in Technology Transfer,* The Rand Corporation, R-1566/6-HUD, 1975.
39. Cho, Yong Hyo, "A Multiple Regression Model for the Measurement of the Public Policy Impact on Big City Crime," *Policy Sciences,* Vol. 3, No. 4, December 1972, pp. 435-455.
40. Clarke, Stevens H., *Crime and Delinquency in North Carolina,* Institute of Government, University of North Carolina, June 1974.
41. Clawson, Calvin, "A Theoretical Approach to the Allocation of Police Preventive Patrol," *The Police Chief,* July 1973, pp. 53-59.
42. Cohen, J., K. Fields, M. Lettre, R. Stafford, and C. Walker, "Implementation of the JUSSIM Model in a Criminal Justice Planning Agency," *Journal of Research in Crime and Delinquency,* Vol. 10, No. 2, July 1973, pp. 117-131.
43. Cohen, J., M. Lettre, and R. Stafford, *Analysis of the Allegheny County Criminal Justice System: Present Operations and Alternative Programs,* The Urban Systems Institute, Carnegie-Mellon University, February 1972.
44. *Computer Applications and Quantitative Techniques in Law Enforcement and Criminal Justice, Selected References,* The Traffic Institute, Northwestern University, February 1972.
45. Cooper, Gary (ed.), *Proceedings of the International Symposium on Criminal Justice Information and Statistics Systems,* sponsored by Project SEARCH and LEAA, October 1972.
46. Cordrey, J. B., *Utilization of Statistical Techniques in Criminal Decision Making,* for LEAA by Community Research, Inc., Dayton/Montgomery County Pilot Cities Program, January 1973.
47. Crabill, T. B., and J. Hagouel, "Documentation of the New York City Police Patrol Simulation Model," The New York City-Rand Institute, unpublished, 1974.
48. Crabill, T. B., "Dispatch Command Region Design—An Investigation of Combining Two New York City Police Precincts into a Single Dispatch Command Region," The New York City-Rand Institute, unpublished mimeo, February 1975.
49. Crabill, Thomas B., Peter Kolesar, and Warren Walker, *Validation of a Police Patrol Simulation Model,* The Rand Corporation, P-5464, June 1975.
50. Crowther, R. F., *The Use of a Computer System for Police Manpower Allocation in St. Louis, Missouri, Part I, Manpower Requirements for Call Answering Services,* Indiana University, Department of Police Administration, 1964.
51. Cyert, R., and J. March, *A Behavioral Theory of the Firm,* Prentice-Hall, Englewood Cliffs, New Jersey, 1963.
52. Dalkey, N. C., *The Delphi Method: An Experimental Study of Group Opinion,* The Rand Corporation, RM-5888-PR, June 1969.

53. Dalkey, N. C., *Studies in the Quality of Life: Delphi and Decision-Making,* Lexington Books (Heath), Lexington, Massachusetts, 1972.
54. *Designing Statewide Criminal Justice Statistics Systems—The Demonstration of a Prototype,* Project SEARCH Statistical Steering Committee, Technical Report No. 3, November 1970.
55. *Designing Statewide Criminal Justice Statistics Systems—An Examination of the Five State Implementation,* prepared for LEAA by the California Crime Technological Research Foundation, submitted by the Project SEARCH Statistical Steering Committee, Technical Report No. 5, December 1972.
56. *Directory of Automated Criminal Justice Information Systems,* Law Enforcement Assistance Administration, U.S. Department of Justice, December 1972.
57. Distler, A. S., *The Potential for Operations Research Techniques in Corrections,* The Mitre Corporation, M74-64, July 1974.
58. Downs, A., *Inside Bureaucracy,* Little, Brown, Boston, 1966.
59. Drabek, Thomas E., *Laboratory Simulation of a Police Communications System under Stress,* The Ohio State University, College of Administrative Science, 1969.
60. Fath, Frederick A., *Scheduling Techniques for Municipal Court Traffic Sessions,* prepared for Tacoma Municipal Court by Boeing Aerospace Co., Seattle, April 1973.
61. Fey, W. R., H. M. Wadsworth, and D. B. Young, *Criminal Justice System Training Model,* final report, Georgia Institute of Technology, School of Industrial and Systems Engineering, prepared for LEAA, May 1974.
62. Fey, W. R., et al., *Criminal Justice System Trainee's Manual,* School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, 1974.
63. Fields, K., M. Lettre, and R. Stafford, *A Description of the Allegheny County Criminal Justice System,* The Urban Systems Institute, Carnegie-Mellon University, February 1972.
64. Fromm, Gary, et al., *Federally Supported Mathematical Models: Survey and Analysis,* prepared for the National Science Foundation by Data Resources, Inc., and Abt Associates, Inc., June 1974.
65. Gass, Saul I., et al., *A Guide to Models in Governmental Planning and Operation,* Chapter 8, "Models in Law Enforcement and Criminal Justice," prepared for the Environmental Protection Agency by Mathematica, Inc., August 1974.
66. Gass, S., and J. Dawson, *An Evaluation of Policy-Related Research: Reviews and Critical Discussions of Policy-Related Research in the Field of Police Protection,* prepared for the National Science Foundation by Mathematica, Inc., October 1974.
67. Giertz, J. Fred, *An Economic Analysis of the Distribution of Police Patrol Forces,* Miami University, NTIS PB 194 985, April 1970.
68. Gottfredson, D., L. Wilkins, P. Hoffman, and S. Singer, *The Utilization of Experience in Parole Decision-Making, A Progress Report,* NCCD Research Center, National Council on Crime and Delinquency, June 1973. There are twelve supplemental reports, including: *Development of a Data Base for Parole Decision-Making; Parole Decision-Making Coding Manual; The Problem of Overlap in Experience Table Construction; Information Selection and Use in Parole Decision-Making; Paroling Policy Guidelines: A Matter of Equity; Parole Selection: A Balance of Two Types of Error; The Reliability of Information in the Parole Decision-Making Study.*



69. Hage, J., and M. Aiken, *Social Change in Complex Organizations,* Random House, New York, 1970.
70. Hann, R. G., *Court System Simulation Models: Promises and Caveats,* University of Toronto, December 1973.
71. Hann, R. G., J. S. Bailey, T. N. Burrell, and A. C. Taylor, *Offender Prediction Study: Stage One,* prepared by Systems Dimensions, Ltd., for the Solicitor General of Canada, May 1974.
72. Hann, R. G., and L. P. Salzman, *CANCOURT—I: A Computerized System Simulation Model to Support Planning in Court Systems,* University of Toronto, Centre of Criminology, May 1974.
73. Hann, Robert G., John S. Bailey, and Jamal Hassan, *Offender Prediction System: Overview,* prepared by Systems Dimensions, Ltd., for the Solicitor General of Canada, Research and Systems Development Division, 340 Laurier Avenue, Ottawa, 1975.
74. Hann, Robert G., John S. Bailey, and Jamal Hassan, *FCSM: Introduction,* prepared by Systems Dimensions, Ltd., for the Solicitor General of Canada, Research and Systems Development Division, 340 Laurier Avenue, Ottawa, 1975.
75. Hann, Robert G., John S. Bailey, and Jamal Hassan, *FCSM: User's Manual,* prepared by Systems Dimensions, Ltd., for the Solicitor General of Canada, Research and Systems Development Division, 340 Laurier Avenue, Ottawa, 1975.
76. Hann, Robert G., John S. Bailey, and Jamal Hassan, *FCSM: System Reference Manual,* prepared by Systems Dimensions, Ltd., for the Solicitor General of Canada, Research and Systems Development Division, 340 Laurier Avenue, Ottawa, 1975.
77. Hauser, N., et al., "Computer Simulation of a Police Emergency Response System," Polytechnic Institute of Brooklyn, Report PB-211-311, National Technical Information Service, U.S. Department of Commerce, Washington, D.C., September 1969.
78. Haynes, Peter, Thomas Cameron, and Sandra Ricci, *Construction and Testing of a Simulation Exercise of the Felony Courts of the Twentieth Judicial Circuit, Florida,* for Criminal Courts Technical Assistance Project, Washington, D.C., April 1974.
79. Haynes, Peter, J. Fazio, A. Forrest, and M. VanderVelde, *COURTEX, A Series of Exercises for Training Court and Other Justice Administrators,* funded by California Council on Criminal Justice, 1974.
80. Held, Joan, and Peter Kolesar, "A Method of Constructing Coordinates for the New York City Police Department Patrol Simulation and for the Generation of Computer Incident Maps," The New York City-Rand Institute, unpublished mimeograph, May 1973.
81. Heller, N., B. Phegley, J. Rother, and K. Schmidt, "Computer Design and Evaluation of Police Patrol Beats," presented at 38th national ORSA meeting, October 1970.
82. Heller, N., R. Markland, and J. Brockelmeyer, "Partitioning of Police Dis-

tricts into Optimal Patrol Beats Using a Political Districting Algorithm: Model Design and Validation," University of Missouri, St. Louis, presented at the 40th national ORSA meeting, October 1971.
83. Heller, N., J. T. McEwen, and W. Stenzel, *Computerized Scheduling of Police Manpower, Volume 1: Methods and Conclusions,* Report PB-232-071, National Technical Information Service, U.S. Department of Commerce, March 1973.
84. Heller, N., J. T. McEwen, and W. Stenzel, *Computerized Scheduling of Police Manpower, Volume 2: Evaluations and Program Users Manual,* National Technical Information Service, U.S. Department of Commerce, Report PB-232-072, March 1973.
85. Heller, Nelson B., "What Law Enforcement Can Gain from Computer Designed Work Schedules," National Institute of Law Enforcement and Criminal Justice, November 1974.
86. Helmer, Olaf, *Social Technology,* Basic Books, Inc., New York, 1966.
87. Hoffman, P., and J. Beck, *Parole Decision-Making: A Salient Factor Score,* U.S. Board of Parole Research Unit, Report Two, April 1974.
88. Hoffman, P. B., and L. K. DeGostin, *Parole Decision-Making: Structuring Discretion,* U.S. Board of Parole Research Unit, Report Five, June 1974.
89. Hoffman, Richard B., *Technology and the Provision of Police Services,* State University of New York at Buffalo, School of Management, October 1969.
90. Holliday, Leo P., "A Time-Sharing Version of the NYCRI Radio Car Simulation Model," The New York City-Rand Institute, unpublished mimeograph, January 1973.
91. Holliday, Leo P., "A Methodology for Radio Car Planning," The New York City-Rand Institute, unpublished mimeograph, January 1974.
92. Hopkinson, R. G., *A Preliminary Description of the Canadian Criminal Justice System,* 2 volumes, Ministry of the Solicitor General, Statistics Division, Report 4/73, December 1973.
93. 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.
94. Jago, W. H., "'DOTSIM' A Dynamic Offender Tracking Simulation," presented at the 44th national Operations Research Society of America Meeting, November 12-14, 1973.
95. Jago, W. H., et al., *Criminal Justice Resource Allocation,* Public Safety Systems, Inc., Report 012-009, July 1973.
96. Jago, William H., and Tim O. Ozenne, *Mathematical Modeling of Criminal Justice Processing,* prepared for Ventura County Model CJ System Project by Public Safety Systems, Inc., Report 012-010, September 1973.
97. Jarvis, James P., *Optimal Dispatch Policies for Urban Server Systems,* Innovative Resource Planning in Urban Public Safety Systems, Massachusetts Institute of Technology, TR-02-73, September 1973.
98. Jarvis, James P., and Richard C. Larson, *Optimal Server Assignment Policies in M/M/N/O Queuing Systems with Distinguishable Servers and Customer Classes,* Innovative Resource Planning in Urban Public Safety Systems, Massachusetts Institute of Technology, WP-06-74, April 1974.
99. Jarvis, James, Mark McKnew, and Larry Deetjen, *Data Collection and Computer Analysis for Police Manpower Allocation in Arlington, Massachusetts,* Innovative Resource Planning in Urban Public Safety Systems, Massachusetts Institute of Technology (to appear).



100. Jennings, John B., *The Flow of Arrested Adult Defendants Through the Manhattan Criminal Court in 1968 and 1969,* The New York City-Rand Institute, R-638-NYC, January 1971.
101. Jennings, John B., *Quantitative Models of Criminal Courts,* The New York City-Rand Institute, P-4641, May 1971.
102. Jennings, John B., *Evaluation of the Manhattan Criminal Court's Master Calendar Project: Phase 1—February 1-June 30, 1971,* The New York City-Rand Institute, R-1013-NYC, January 1972.
103. Jennings, John B., *Analysis of the Night and Weekend Arraignment Parts in the Bronx and Queens Criminal Courts,* The New York City-Rand Institute, R-1236-NYC, April 1973.
104. Jones, R., and K. Joscelyn, *Computerized Allocation of Police Traffic Services, A Demonstration Study,* Indiana University Institute for Research in Public Safety, for the National Highway Traffic Safety Administration, NTIS PB 213 025, March 1972.
105. *Jury Management Program, Systems Description, Vol. II,* Court Management Project, Cleveland Bar Association (no date).
106. Knight, K., "A Descriptive Model of the Intra-Firm Innovation Process," *Journal of Business,* Vol. 40, 1967, pp. 478-496.
107. Knoppers, Bastiaan A., *Simulation of Police Patrol and Dispatch,* prepared for LEAA by Mathematica, Inc. (no date).
108. Kobrin, S., S. Lubeck, E. Hanse, and R. Yeaman, *The Deterrent Effectiveness of Criminal Justice Sanction Strategies,* final report, Public Systems Research Institute, University of Southern California, NTIS PB 214 570, September 1972.
109. Kolesar, P., "A Simulation Analysis of the Effects of a Car Locator System," The New York City-Rand Institute, unpublished mimeograph, 1975.
110. Kolesar, Peter, and Edward Blum, "Square Root Laws for Fire Engine Response Distances," *Management Science,* Vol. 19, 1973, pp. 1368-1378. (Revised and updated version, *Square Root Laws for Fire Company Travel Distances,* The Rand Corporation, R-895-NYC, 1975.)
111. Kolesar, P., K. Rider, and T. Crabill, *Guidelines for Scheduling Police Patrol Cars,* The New York City-Rand Institute, R-1803, forthcoming.
112. Kolesar, P., K. Rider, T. Crabill, and W. Walker, *A Queuing-Linear Programming Approach to Scheduling Police Patrol Cars,* The New York City-Rand Institute, P-5260-1, 1974 (to appear, *Operations Research).*
113. Kolesar, P., K. Rider, and W. E. Walker, "Analysis of Police Patrol Tours by Numerical Solution of a Time Dependent Queue," The New York City-Rand Institute, unpublished mimeograph, June 1973.
114. Kolesar, Peter, and Warren E. Walker, *A Simulation Model of Police Patrol Operations: Executive Summary,* The New York City-Rand Institute, R-1625/1-HUD, March 1975.
115. Kolesar, Peter, and Warren E. Walker, *A Simulation Model of Police Patrol Operations: Program Description,* The New York City-Rand Institute, R-1625/2-HUD/NYC, February 1975.

116. Kolton, Kent W., "The Dedicated Police Computer—Does it Really Make a Difference?," *The Bureaucrat*, Vol. 1, No. 4, Winter 1972, pp. 357-365.
117. Kolton, Kent W., "Use of Computers by Police: Patterns of Success and Failure," *Urban Data Service Report*, Vol. 4, No. 4, April 1973.
118. Larson, Richard C., *Models for the Allocation of Urban Police Patrol Forces*, Massachusetts Institute of Technology, Operations Research Center, Technical Report 44, 1969.
119. Larson, Richard C., *Urban Police Patrol Analysis*, MIT Press, Cambridge, Massachusetts, 1972.
120. Larson, Richard C., *Decision-Aiding Tools in Urban Public Safety Systems*, Innovative Resource Planning in Urban Public Safety Systems, Massachusetts Institute of Technology, JR-01-73, Winter 1972-73.
121. Larson, Richard C., *A Hypercube Queuing Model for Facility Location and Redistricting in Urban Emergency Services*, The New York City-Rand Institute, R-1238-HUD, March 1973. (Revised version in *Computers and Operations Research*, Vol. 1, 1974, pp. 67-95.)
122. Larson, Richard C., "Illustrative Police Sector Redesign in District 4 in Boston," *Urban Analysis*, Vol. 2, 1974, pp. 51-91.
123. Larson, Richard C., *Urban Emergency Service Systems: An Iterative Procedure for Approximating Performance Characteristics*, The New York City-Rand Institute, R-1493-HUD, January 1974. (Revised version in *Operations Research*, Vol. 23, 1975, pp. 845-868.)
124. Larson, Richard C., *Resource Allocation in Public Safety Services*, Innovative Resource Planning in Urban Public Safety Systems, Massachusetts Institute of Technology, JR-05-75, May 1974.
125. Larson, Richard C., "Resource Planning for Urban Public Safety Systems," *Technology Review*, June 1974, Vol. 76, No. 7, pp. 20-29.
126. Larson, Richard C., *Hypercube Queuing Model: User's Manual*, The New York City-Rand Institute, R-1688/2-HUD, July 1975.
127. Larson, Richard C., *Hypercube Queuing Model: Program Description*, The New York City-Rand Institute, R-1688/3-HUD, July 1975.
128. Lasdon, L. S., A. D. Waren, and S. J. Madson, "Juror Management in a Metropolitan Trial Court," *Judicature*, Vol. 57, No. 9, April 1974, pp. 402-409.
129. Lipsett, F. R., and J. G. Arnold, "Computer Simulation of Patrol Operations of a Semi-Rural Police Force," *Journal of Police Science and Administration*, Vol. 2, No. 2, June 1974, pp. 190-207.
130. Locke, J. W., et al., *Compilation and Use of Criminal Court Data in Relation to Pre-Trial Release of Defendants: Pilot Study*, National Bureau of Standards, Technical Note 535, August 1970.
131. Lutz, Raymond P., and Jerry G. Metcalf, "Application of GERT to a Municipal Court System," presented at 44th national meeting of ORSA, November 1973.
132. Lyons, Norman, "Analytic Models of Criminal Court Operations," Ph.D. thesis, Carnegie-Mellon University, Business Administration, 1972.
133. McEachern, A. W., *Criminal Justice Simulation Study, A Preliminary Planning System*, prepared for the Los Angeles Regional Criminal Justice Planning Board, The Public Systems Research Institute, University of Southern California (no date).
134. McEachern, A. W., Edward M. Taylor, J. Robert Newman, and Ann E. Ashford, "The Juvenile Justice System: Simulation for Research and Decision-Making," *American Behavioral Scientist*, January-February 1968, pp. 1-48.
135. McEachern, Alexander W., and J. Robert Newman, "A System for Computer-Aided Probation Decision-Making," *Journal of Research on Crime and Delinquency*, Vol. 6, No. 2, July 1969.
136. McEwen, J. Thomas, Project Director, *Allocation of Patrol Manpower Resources in the Saint Louis Police Department*, prepared for LEAA by Resource Allocation Unit, St. Louis Police Department, two volumes, 1968.
137. McEwen, J. Thomas, *A Mathematical Model for Prediction of Police Patrol Workload*, Resource Allocation Project, St. Louis Police Department, May 1968.
138. McEwen, J. Thomas, and R. C. Larson, "Patrol Planning in the Rotterdam Police Department," *Journal of Criminal Justice*, Vol. 2, 1974, pp. 235-238.
139. McGuire, James T., "Put the Troopers Where the Action Is," *FBI Law Enforcement Bulletin*, Vol. 39, No. 8, August 1970, pp. 2-5, 26.
140. Merrill, F., and L. Schrage, "Efficient Use of Jurors: A Field Study and Simulation Model of a Court System," *Washington University Law Quarterly*, 1969, p. 151.
141. *A Minicomputer-Oriented Court Simulation Study*, prepared for LEAA by Superior Court of New Jersey, Hudson County, 1972.
142. Mitchell, Phillip S., "Optimal Selection of Police Patrol Beats," *The Journal of Criminal Law, Criminology and Police Science*, Vol. 63, No. 4, December 1972, pp. 577-584.
143. *A Model of and Problems Encountered in a Metropolitan Police Department*, Center for Application of Sciences and Technology, Wayne State University, September 1969.
144. Mudge, Richard, "A Description of the New York City Police Department RMP Allocation Model," The New York City-Rand Institute, unpublished, May 1974.
145. Mudge, Richard, and Peter Dormont, "A Description of the New York City Police Department RMP Allocation Model," The New York City-Rand Institute, unpublished, December 1974.
146. Nejelski, Paul, "Computer Simulation: An Aid to Court Study," *Judicature*, Vol. 55, No. 1, June-July 1971, pp. 17-20.
147. *Omnibus Safe Streets and Crime Control Act of 1968*, P.L. 90-351, June 19, 1968.
148. *Opportunities for Improving Productivity in Police Services*, The National Commission on Productivity, 1973.
149. Patent, Steven, *Courtroom Scheduling of Criminal Cases With a Due-Date*, M.S. thesis, Cornell University.
150. Phillips, L., and H. L. Votey, "An Economic Analysis of the Deterrent Effect of Law Enforcement on Criminal Activity," *The Journal of Criminal Law, Criminology, and Police Science*, Vol. 63, No. 3, 1972.
151. Phillips, Llad, "Crime Control: The Case for Deterrence," from *The Economics of Crime and Punishment*, a conference sponsored by American Enterprise Institute for Public Policy Research (no date).
152. Pittman, James T., "An Analytic Process for the Evaluation of State Correctional Programs," M. S. thesis, Georgia Institute of Technology, August 1972.

153. Pittman, James T., and Paul Gray, *Evaluation of Prison Systems*, University of Southern California, Graduate School of Business Administration (no date).
154. Quincy Police Department, *Application of the Hypercube Sector Design Analysis*, Planning and Research Division, Quincy, Massachusetts 02169.
155. Rardin, Ronald L., and Paul Gray, "Analysis of Crime Control Strategies," *Journal of Criminal Justice*, Vol. 1, 1973.
156. Reed, John H., "Operational Research and the Courts," *Judicature*, Vol. 56, No. 2, August/September 1972, pp. 67-71.
157. Renshaw, B. H., and C. I. Goldman, "Application of an Indicator-Based Planning System and Simulation Modeling to a Criminal Justice Planning Process: A Philadelphia First Try," Government Studies & Systems, Inc., presented at 41st national ORSA meeting, April 1972.
158. Renshaw, B., and C. Goldman (Government Studies and Systems, Inc., Philadelphia), W. Braybrook (Philadelphia Regional Planning Council), and E. Mitchell (Division of Corrections, Alaska), "PHILJIM: The Philadelphia Justice Improvement Model, a General Computer Model for Simulation of the Total Criminal Justice System of States and Cities with Planning Applications in the State of Alaska and the City of Philadelphia," paper presented at the International Symposium in Criminal Justice Information and Statistical Systems, New Orleans, October 1972.
159. *Report on the Criminal Justice System*, National Advisory Commission on Criminal Justice Standards and Goals, January 1973.
160. Research Proposal submitted to National Science Foundation by Arizona State University, January 1974.
161. Riccio, Lucius J., *A Model for Court Resource Planning*, Industrial Engineering Department, Lehigh University, February 1973.
162. Rider, Kenneth L., "A Simple Parametric Model for the Allocation of Radio Motorized Patrol Cars," The New York City-Rand Institute, unpublished, 1974.
163. Rosenshine, Matthew, "Contributions to a Theory of Patrol Scheduling," The Pennsylvania State University, *Operations Research Quarterly*, Vol. 21, No. 1, March 1970, pp. 99-106.
164. Roy, R. H., "An Outline for Research in Penology," *Operations Research*, Vol. 12, No. 1, January-February 1964, pp. 1-12.
165. Salzman, L. P., *A Simulation Approach to Caseflow Management in the Provincial Courts*, University of Toronto, May 1974.
166. Schneible, Daniel C., *A Separable Programming Approach to Allocating Manpower for the California Highway Patrol*, Naval Postgraduate School, March 1973.
167. Shumate, R. P., and R. F. Crowther, "Quantitative Methods for Optimizing the Allocation of Police Resources," *Journal of Criminal Law, Criminology, and Police Science*, Vol. 57, 1966, pp. 197-206.
168. Simon, Herbert, *Administrative Behavior*, The Free Press, New York, 1957.
169. Smith, R. D., *Computer Applications in Police Manpower Distribution*, International Association of Chiefs of Police, April 1964.
170. Smith, S. B., et al., *Superbeat: A System for the Effective Distribution of Police Patrol Units*, Illinois Institute of Technology, October 1973.





171. Smith, S. B., et al., *Superbeat Program Manual*, Illinois Institute of Technology, October 1973.
172. Space General Corporation, *Prevention and Control of Crime and Delinquency*, Report PCCD-7, prepared for the Youth and Adult Corrections Agency, State of California, July 1965.
173. Sterrett, Dennis, Allegheny Regional Planning Commission, personal communication with D. Jaquette, September 1974, and with M. Lawless, December 17, 1974.
174. Stoever, William, "The Expendable Resource: Studies to Improve Juror Utilization," Institute of Judicial Administration, *The Justice System Journal* (no date).
175. Stollmack, Stephen, and Carl M. Harris, "Failure-Rate Analysis Applied to Recidivism," *Operations Research*, Vol. 22, No. 6, November-December 1974, pp. 1192-1205.
176. "System/370 Justice System," brochure, IBM Field Developed Program, GB21-1198-0.
177. *System Overview and Report Formats for PROMIS*, prepared by Peat, Marwick, Mitchell & Co., for the District of Columbia Government, Office of Crime Analysis, under LEAA grant, May 1971.
178. *System Study in Court Delay. LEADICS—Law-Engineering Analysis of Delay in Court Systems*, four volumes, Law School and College of Engineering, University of Notre Dame, January 1972.
179. Taylor, J. G., J. A. Navarro, and R. H. Cohen, "Simulation Applied to a Court System," *IEEE Transactions on Systems Science and Cybernetics*, V. SSC-4, No. 4, November 1968, pp. 376-379.
180. Tullier, Pierre, *Operations Research and the Prison/Parole System, A Markov Example*, UNIVAC Federal Systems Division (no date).
181. Urban Sciences, Inc., "Computer Simulation of the Boston Police Department," 177 Worchester Street, Wellesley, Mass. 02181, 1971.
182. Urban Sciences, Inc., "Police Resource Allocation Program (RAP)," 177 Worchester Street, Wellesley, Mass. 02181, September 1972.
183. Van Vuuren, Nancy, et al., *Toward a Safer Community*, Volume 3, Allegheny Regional Planning Council, no date (1974?).
184. Votey, H. L., and Llad Phillips, "Police Effectiveness and the Production Function for Law Enforcement," *The Journal of Legal Studies*, University of Chicago Law School, Vol. I, No. 2, June 1972.
185. Votey, H. L., and Llad Phillips, "Social Goals and Appropriate Policy for Corrections: An Economic Appraisal," *Journal of Criminal Justice*, Vol. 1, 1973.
186. Walker, Warren E., *The Deployment of Emergency Services: A Guide to Selected Methods and Models*, The New York City-Rand Institute, R-1867-HUD, 1975.
187. Weissberg, Richard, *Using the Interactive Hypercube Model*, Innovative Resource Planning Project, Massachusetts Institute of Technology, 1975.
188. White, Michael J., "A Simulation Analysis of Juror Management," presented at the 43rd national ORSA meeting, College of Business Administration, University of So. Florida, May 1973.
189. White, Thomas, "Plausible Quantitative Estimates of Impact in a Criminal

Justice System: Examples Using Crime Clearance-Rates and Length of Prison Term," The Urban Institute, draft.

190. Wickland, Gary A., *A Research Study Using OR in the Court System*, The University of Iowa, College of Business Administration (no date).
191. Wilson, N. B., A. B. Bishop, J. D. Joscelyn, and M. A. Bell, *Final Report: Systems Analysis and Operations Research Support to the Ohio Youth Commission Juvenile Prevention Project*, Systems Research Group, Department of Industrial and Systems Engineering, The Ohio State University, EES 386X, December 1972.
192. Wilson, Thomas A., and Paul Gray, "File Management in Management Information Systems," *Decision Sciences*, Vol. 5, No. 3, July 1974, pp. 374-378.
193. Wolfgang, Marvin E., and Harvey A. Smith, "Mathematical Methods in Criminolcgy," *International Social Science Journal*, Vol. 18, No. 2, 1966, pp. 200-223.
194. Young, Suzanne (comp.), *Performance Measures for the Allocation of Police Patrol Forces: An Annotated Bibliography*, Innovative Resource Planning Project, Massachusetts Institute of Technology, WP-09-74, August 1974.
195. Zaltman, G., R. Duncan, and J. Holbek, *Innovations and Organizations*, Wiley, New York, 1973.

# END