TIMS Studies in the Management Sciences 22 (1986) 121–153
© Elsevier Science Publishers B.V.

EXTENDING AND APPLYING THE HYPERCUBE QUEUEING MODEL TO DEPLOY AMBULANCES IN BOSTON

Margaret L. BRANDEAU
Stanford University

and

Richard C. LARSON
Massachusetts Institute of Technology

The Hypercube queueing model is a planning tool which can be used to aid in long-term deployment of emergency service vehicles. By computing selected performance characteristics of an urban emergency service system, the Hypercube model can assist the decision maker in determining the appropriate number of vehicles to meet a city's particular service standards and in determining appropriate districts and home locations for those units.

This paper presents an overview of the Hypercube model and describes the incorporation of three technical improvements – a varying service times option, a mean service time calibration feature, and an improved travel time estimation algorithm – into the model. The paper then details an application of the improved model by the City of Boston for ambulance deployment. The model has now been successfully used for more than 2 years by Boston planners to deploy ambulances and has resulted in an estimated savings of $150000 per year. The paper concludes by summarizing the lessons learned from such an application and then describes possible further extensions to the Hypercube model.

1. Introduction

The location of emergency service units such as police or fire vehicles, ambulances, or emergency repair vehicles is an important resource allocation problem which has been subject to serious analytical attention only during the last two decades. The interrelated problems of determining appropriate locations for units as well as appropriate response areas can be quite complex, for several reasons. First, emergency service systems themselves are complex systems with many service vehicles, requests for service occurring both temporally and spatially, cooperating servers who often respond to calls in a certain dispatch preference order, multiple dispatches of servers to calls, varying travel times in different parts of the service region, and possible queue delays, so that a realistic analytic model of such a system can be quite difficult to devise. Second, even for a moderately sized city with 10 ambulances (or police cars, or fire vehicles) and 100 points of demand, the number of possible vehicle location/service area configurations is staggeringly large. Third, emergency

service planners typically wish to satisfy diverse – and often conflicting – system performance objectives such as minimized response time, equal unit workloads, and uniform levels of service to all districts, so that no single mathematical technique is likely to take into account all of these factors.

Researchers have approached the problem of locating urban emergency units in a variety of ways [5]. Some researchers [12] have concerned themselves only with locations of units within specified response areas, while others [17] have considered the problem of designing districts, given unit locations. A variety of approaches have been suggested for simultaneously determining appropriate unit locations and districts for emergency service vehicles. Toregas et al. [23] have modeled the problem as a set covering problem in which the objective is to locate the minimum number of units such that each demand point is within a specified travel time or distance from the nearest server. Daskin and Stern [7] have also used a set covering approach, but their objective takes into account not only the number of units required to cover the region, but also the extent of multiple coverage.

Realizing that service delay, especially during busy periods, may be an important part of emergency service systems, some researchers have developed queuing models to describe them. Chaiken [4] used a simple infinite-server queuing model to determine the probability that a specified number of fire vehicles in New York City would be simultaneously busy. Fitzsimmons [9] developed a more complex queuing model for ambulances which takes into account such factors as travel time from each server to each call; within the model, simulation is used to estimate the conditional response time to certain calls, given that the primary ambulance for the district is already busy. The output of the model is the expected citywide distribution of response times. Fitzsimmons applied the model to the Los Angeles ambulance system with the objective of reducing the average response time to calls; he noted, however, that because the output of the model is a distribution of response times (rather than just the mean response time), the model could also be used to minimize a fractile-type criterion (for example, the probability of a response time greater than six minutes).

Another approach to the problem has been through pure simulation models. Savas [21] used a simulation model of the New York City ambulance system to show that the average response time could be reduced by redeploying ambulances, while Carter and Ignall [2] used simulation to study fire department allocation policies. Swoveland et al. [22] developed a simulation model to determine the average citywide response time for ambulances in Vancouver, British Columbia and then used a branch and bound ("probabilistic enumeration") approach to determine new locations and new district configurations which would reduce the average response time.

One limitation of some of the above approaches is that they do not take into account the probabilistic nature of an emergency service system; that is, they

do not take into account the fact that the closest server to a call will not always be available. Another limitation is that most of these approaches consider only one (or at most two) performance objectives, typically average response time. The Hypercube Queueing Model [16] is a descriptive, analytical model which calculates a variety of performance measures, given a particular emergency service system configuration. It does not select an "optimal" configuration, but only assists the decision maker by estimating the operational performance of any particular set of vehicle locations and districts. In this way the model helps the emergency service planner determine the appropriate number of service vehicles to meet a city's particular service standards and appropriate home locations and primary response areas for those units. Developed by Larson in 1973, the Hypercube model has been used for police and ambulance deployment in a number of cities in the US and abroad [1,3,6,9,14].

This paper describes three recent improvements – a varying service times option, a mean service time calibration feature, and an improved travel time estimation algorithm – which were incorporated to make the model more realistic and hence more useful to the emergency service planner. We then describe how the improved model was used by the City of Boston for strategic ambulance deployment. Now in use for more than two years by Boston ambulance planners, the model has resulted in an estimated annual savings of $150 000 per year. We conclude the paper by discussing the lessons learned from such an application.

Before describing the improvements to the Hypercube model and its application in Boston, however, we first review briefly the use of the model and the mathematical structure behind it.

2. Hypercube model overview

2.1. System description and model use

In an emergency ambulance system, emergency calls arrive at different times from different parts of a region at a central dispatch number. If available, a vehicle is dispatched immediately to the incident; otherwise, a vehicle is dispatched as soon as one becomes available. After arriving at the incident, the vehicle typically spends some time at the scene, transports the patient to the hospital if necessary, and then returns to its home location (if different from the hospital). This sequence of events is depicted in fig. 1.

The Hypercube model is used to describe such a system, as follows: The user partitions the city into a number of statistical reporting areas or "geographical atoms." The user collects data describing current system operation such as the number of service requests from each atom, the average service time of each vehicle or *response unit*, and the average travel time between every

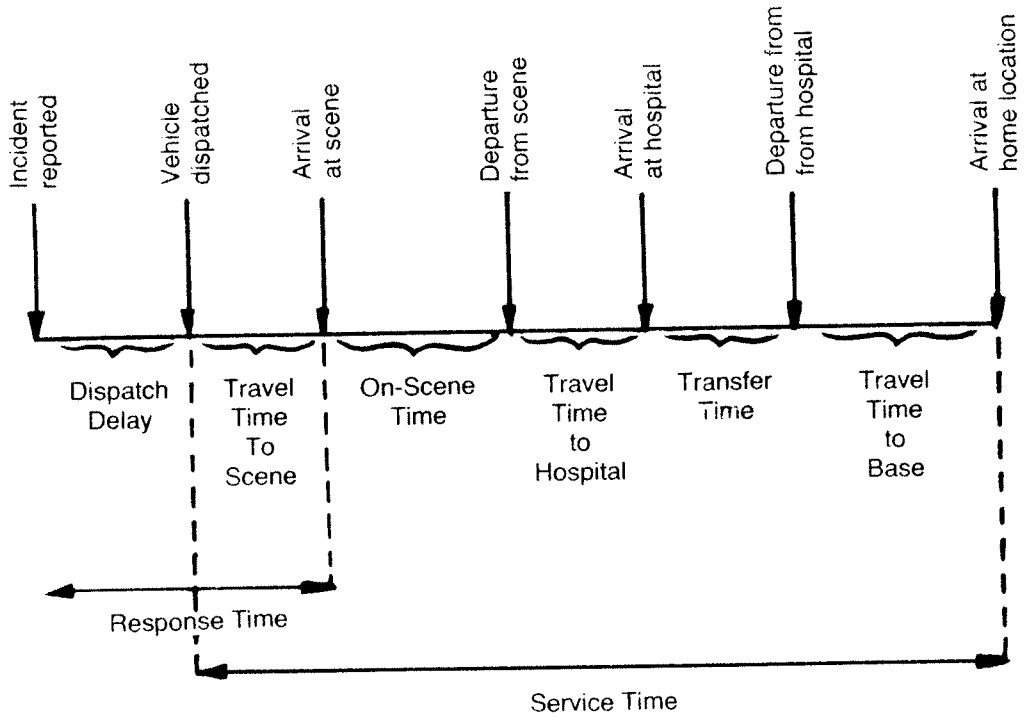

Fig. 1. Sequence of events in an emergency medical vehicle response.

pair of atoms. The model is calibrated using these data. Once calibrated, the model can be used to compute selected performance measures relating to alternative configurations of the system. For each configuration considered, the model computes performance characteristics at four different levels of aggregation, as shown in table 1. Then, based on his assessment of the output, the user may decide to alter the system configuration and rerun the model. In particular, he may change the idle location of one or more response units in the system and/or the total number of response units in the system. This process continues until the user has created a system configuration which is considered satisfactory.

It is important to note that the Hypercube model does not indicate to the user which system configuration is "optimal." Rather, based on his or her own particular service requirements, the user decides which system configuration represents the most appropriate trade-off among the model's numerous performance measures. Most likely, the trade-off embodies a balance among the following objectives: minimizing the average system-wide response time (subject to the total number of response units available), equalizing response unit workloads, minimizing the expected maximum response time to any particular area, minimizing the average fraction of dispatches which are inter-district, and minimizing the fraction of calls handled by backup units [15].

2.2. Mathematics of the hypercube model

2.2.1. Exact model

As mentioned above, the Hypercube model assumes that the area being modelled can be broken down into K geographical atoms. Each geographical atom k generates some fraction f_k of the total system-wide number of calls for service ($\sum_{k=1}^{K} f_k = 1$). The mean travel time from atom i to atom j is denoted by τ_{ij}.

There are N response units or servers. The conditional probability that server i is located in atom j while available is l_{ij} ($\sum_{j=1}^{K} l_{ij} = 1$). Response units may have either fixed or mobile locations while idle. (Typically police vehicles patrol a district when idle, while emergency medical vehicles usually remain at a home base.)

Calls for service are assumed to occur throughout the region being modelled in a Poisson manner at a mean rate λ per hour, with each atom k acting as an independent Poisson generator with mean rate $\lambda f_k = \lambda_k$. If one is not concerned with the identity of busy servers, the queuing system is simply an $M/M/N$ system, with either zero-line capacity ($M/M/N/0$) or infinite-line capacity ($M/M/N/\infty$). The $M/M/N$ model implies the following assumptions:

- Exactly one response unit is assigned to every call that is serviced;
- The service time of any response unit for any call for service has a negative exponential distribution with mean $1/\mu$;
- The service time is independent of the identity of the server, the location of the customer, and the history of the system;
- For the zero-line capacity case, any call for service that arrives while all N response units are busy is either lost or (more likely in practice) serviced from outside the region or by special reserve units from within the region;
- For the infinite-line capacity case, any call for service that arrives while all N response units are busy is entered at the end of a queue of calls that is depleted in a *first-come-first-served* (FCFS) manner.

Given the geographical atom of the call, the dispatcher's selection policy is assumed to be one of *fixed preference*. For such a policy one specifies that some unit i, if available, would be the first preference to dispatch to atom k, unit j would be the second preference, unit l the third preference, etc. The dispatcher always selects the most preferred *available* unit.

Given the above assumptions, one can characterize the system as a continuous-time Markov process with 2^N states, corresponding to all combinations of busy and idle servers. (In addition, if the system has infinite-line capacity, the state space includes an "infinite tail.") The discussion here focuses on the $M/M/N/0$ system, in which calls arriving when all servers are busy are assumed to be handled by backup units outside of the system; however, similar results are contained in the literature for the $M/M/N/\infty$ system [16].

Table 1
Performance measures calculated by the hypercube model

Region-wide

1. Mean travel time to calls
2. Mean response time to calls
3. Mean workload [a]
4. Maximum workload imbalance
5. Fraction of dispatches which remove units from their primary response areas
6. Fraction of calls answered by backup units

Response unit-specific

1. Average workload
2. Fraction of dispatches out of the unit's primary response area
3. Average travel time to calls
4. Average response time to calls

District-specific [b]

1. Calls per hour
2. Average workload
3. Fraction of calls answered by non-primary units
4. Average travel time to calls
5. Averager response time to calls

Atom-specific

1. Calls per hour
2. Average travel time to calls
3. Average response time to calls
4. Fraction of dispatches handled by each unit

[a] Defined to be the average fraction of time a vehicle is busy.
[b] A unit's district in the case of EMS vehicles represents the set of atoms for which that unit would be assigned to service requests, if the unit is available, regardless of the status of all other units.

Representing an idle state by a "0" and a busy state by a "1", the system states can be represented as the set of all possible binary vectors of length N. These correspond to the set of vertices of an N-dimensional unit hypercube in the positive orthant. Because of the $M/M/N$ assumptions, and because only one unit is assigned to any call, no two servers can change status simultaneously. This means that any system state transitions must occur along the edges of the N-dimensional hypercube. With this assumption, and using the dispatch preference orderings for each atom, the steady-state probability of each state can be computed. Then the 2^N system state probabilities are used to

compute the various system performance measures which are shown in Fig. 2. For example, workload for each server k is calculated as the sum of the probabilities of all states in which server k is busy.

2.2.2. Approximate hypercube model

The exact Hypercube model described above requires the solution of 2^N simultaneous linear equations, a formidable task when the number of servers is larger than, say, 8 or 10. Because of this, Larson has developed an approximation procedure for the Hypercube model which requires the solution of only N simultaneous nonlinear equations [18]. In this case, the 2^N system states are not calculated, but the response unit workloads are estimated directly.

The idea behind the approximate model is that it is sometimes too difficult to calculate all of the 2^N system states, and that to calculate the system performance measures we only need to know the probability that the jth preferred server is dispatched to a call from any atom. To calculate these quantities, it is assumed that the probability of dispatching the jth preferred unit to a particular atom can be approximated to be proportional to the product of the workloads of the first $(j-1)$ preferred servers and the availability factor of the jth preferred unit. The proportionality constant depends on j and is determined by considering the simple $M/M/N$ queuing model, assuming a situation in which j servers are selected randomly without replacement from the $M/M/N$ system. Using these assumptions, N simultaneous equations can be generated which relate the N unknown workloads to the dispatch policy and the call rates from the various geographical atoms. These equations can be solved iteratively, thereby yielding estimates of the workloads of the units. The workloads can be used to estimate the fraction of dispatches that send server i to atom j, for all i and j, which can then be used to obtain estimates of the values of the desired performance measures.

The performance measures computed by the approximate version of the model are typically within 1 or 2% of those calculated by the exact version of the model [16].

3. Hypercube model improvements

This section describes the three improvements – a varying service times option, a mean service time calibration option, and an improved travel time estimation algorithm – which were incorporated into the approximate Hypercube model prior to implementation in Boston. The Boston application focuses on ambulances; however, the concepts described below are also applicable to other emergency service applications such as police or emergency repair services.

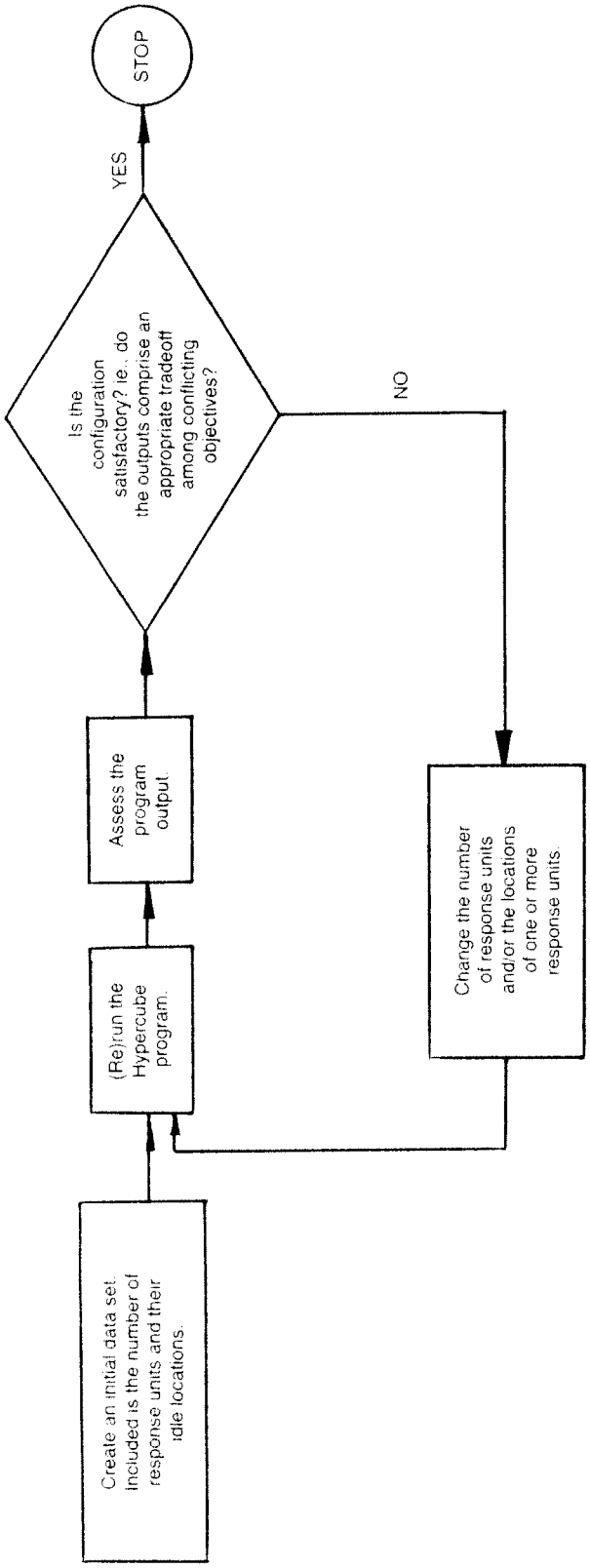

Fig. 2. Using the hypercube model: An iterative process.

3.1. Varying service times option

The exact version of the Hypercube model allows for non-equal response unit mean service times, but the approximate version – which is computationally more tractable – had the limitation of assuming that all emergency response units have the same average total service time. Service time is made up of travel time and non-travel time components, as can be seen from fig. 1. In cities with large or varied area, service time differences may be especially significant, since downtown units may have relatively small travel times while units farther away from the center of the city may have quite large travel times.

The varying service times option incorporated into the approximate version of the Hypercube model is based on an algorithm devised by Jarvis [13] which allows for non-equal mean service times as a part of the approximate version of the Hypercube model. This option can only be used in the $M/M/N/0$ system.

Briefly, the approximation algorithm uses the same kind of reasoning as the equal service time model described earlier. As before, the workload of each server is thought of in terms of the joint probability that more preferred servers for any particular atom are busy and unit i is available. Then, much the same as with equal service times, the probability that server i answers a call from atom j can be replaced by the product of the independent probabilities that server i is available and the more preferred servers are all busy, modified by the appropriate correction factor. While the correction factors from the homogeneous service time model are not strictly correct when used with varying service times, Jarvis had difficulty estimating the factors for this generalized system, and found that the correction factors for the equal mean service time system yielded reasonably accurate results in the generalized system.

As in the equal service time case, the workloads are solved iteratively. For a starting solution, the initial average service time (assumed in the first iteration to be equal for all servers), unit workloads, and average utilization are given by

$$\frac{1}{\mu} = \sum_{j=1}^{K} \frac{\lambda_j}{\lambda} \frac{1}{\mu_{DP(1,j)}}, \tag{1}$$

$$\rho_i = \sum_{j:\, DP(1,j)} \lambda_j \frac{1}{\mu_i} \quad i = 1, 2, \ldots, N, \tag{2}$$

$$\rho = \lambda/\mu N. \tag{3}$$

This varying service times algorithm was implemented as an optional feature in the approximate Hypercube model. While optional, however, varying service times can increase the accuracy of the model results in most actual

applications. The impact of varying service times on an actual application of the Hypercube model will be discussed below in section 4 which describes use of the model in Boston.

3.2. Mean service time calibration

Mean service time calibration (MSTC) arises because the response unit service times depend on the spatial locations of servers and service requests. Because of this, the output performance measures computed by the Hypercube model have direct implications regarding the values of the input data; in particular, the model-computed travel times for each of the servers, based in part on the *user-input service times*, imply values for *model-computed service times*. Proposed by Larson [19], Jarvis [13], and Fitzsimmons [9], MSTC is an iterative procedure by which the Hypercube model cycles through its computation process, replacing the input service times with the model-computed service times until the output service times are equal to the input service times. The performance measures output by the model, then, are based on the final, calibrated service times.

In order to understand MSTC, let us examine the workings of the Hypercube model more closely. As described in section 2.2., the approximate Hypercube model estimates unit workloads based on (i) the total number of response units, (ii) the call rate from each geographical atom, (iii) the service time(s) of the response units, and (iv) the dispatch preference orderings for each atom. The estimated unit workloads are then used to compute the probability that a call from atom j is serviced by server i. These dispatch probabilities, in turn, are used to compute the various output performance measures and, in particular, to compute the average travel time to calls for each unit.

Before implementation of MSTC, the Hypercube model carried out the computation process described above and then stopped once the values for the output performance measures were calculated. However, since the response unit travel times output by the model imply new (model-computed) service times, the program output measures may not be consistent with the input data. If $1/\mu_i \equiv$ total average service time for unit i, $\bar{\tau}_i \equiv$ total average travel time for unit i, and $C_i \equiv$ total average non-travel service time for unit i, we have

$$1/\mu_i = \bar{\tau}_i + C_i \quad i = 1, 2, \ldots, N. \tag{4}$$

If $\tau_i \equiv$ average travel time to calls for unit i, and c is a constant representing round trip travel time [1], then in the steady state, total travel time can be

[1] In the Boston ambulance application, this constant was found from historical ambulance data to be equal to 2.1; that is, the total ambulance travel time on the average was found to be approximately 2.1 times the average travel time to the scene.

expressed as

$$\bar{\tau}_i = c \cdot \tau_i \quad i = 1, 2, \ldots, N. \tag{5}$$

Substituting this into eq. (4) gives a relationship between the model-computed travel times to calls and the model-implied service times:

$$\frac{1}{\mu_i}(\text{model-implied}) = (c \cdot \tau_i) + C_i \quad i = 1, 2, \ldots, N. \tag{6}$$

By sequentially replacing the input service times with the model-implied service times until equilibrium is reached, MSTC ensures that the model input is consistent with model output.

MSTC assists the Hypercube model user in two ways: first, by calculating the average system-wide service time, and second, by calculating the service time of each response unit based on response unit travel patterns. In a typical application, the emergency service planner will have imperfect input information regarding response unit service times; MSTC can calibrate the service times used by the model so that they are consistent with the current deployment pattern. The significance of MSTC in an actual application of the Hypercube model will be described in section 4.

3.3. Barriers algorithm

A third improvement to the Hypercube program was the introduction of a new travel time estimation procedure known as the "Barriers algorithm". One of the key data inputs to the Hypercube program is the travel time between every pair of geographical atoms. These travel times are used directly in determining the dispatch preference orderings for each geographical atom and directly or indirectly in the calculations of the majority of the performance measures output by the model.

As a rule, however, the Hypercube model user will not know the travel time between every pair of geographical atoms in his city, as this requires a sizeable amount of data. In an application with 50 geographical atoms, for example, the user would need to have 2500 empirically obtained travel times. Since most users of the Hypercube model would not have such finely detailed data, the model has an estimation option which can be used to *approximate* the travel time between every pair of geographical atoms. The travel time estimation algorithm computes travel times based on a Manhattan, or right-angle, distance metric using the relative (x, y) positions of the geographical atoms and the average vehicle response speed.

The improved travel time estimation algorithm, developed by Larson and Li [20], also uses a right-angle travel metric, but allows for the possibility of

barriers to travel. Given a set of origin–destination points in the plane and a set of polygonal barriers to travel, the Larson/Li algorithm finds minimal distance feasible paths between the points, assuming that all travel occurs according to the Manhattan distance metric.

In a typical city, there will be a number of areas in which a vehicle must travel a distance longer than a right-angle distance to go from one point to another. Generally this increased travel distance is due to barriers such as bodies of water, parks, and railroad tracks – all of which can be crossed only at certain points. By allowing for barriers to travel in the computation of inter-atom travel times, the Barriers algorithm enhances the realism of the Hypercube model input (and hence the realism of the model output).

4. Implementing the hypercube model in Boston

4.1. Background

The city of Boston, shown in fig. 3, is a diverse urban area of 52 square miles which encompasses both downtown and suburban sections. The resident population of 650 000 increases to approximately 1 200 000 during the work day. Each year approximately 85 000 calls for emergency medical services are generated via the "911" central access number, of which 55 000 require emergency response. The city's pre-hospital emergency medical system includes 111 Emergency Medical Technicians, 20 EMT-Paramedics, and 17 Boston area teaching hospitals which are coordinated through a radio communications network.

In 1978 the Boston Department of Health and Hospitals initiated a three-pronged expansion plan for the Emergency Ambulance Service [12] which included upgrading the training of personnel, and improving the deployment of ambulances. At the time the city had 7 ambulances on duty at any one time. As part of the expansion plan, ambulance system managers wanted to increase the number of ambulances at least to 10 during the next year. System planners wanted to determine where the 10 ambulances should be located, what primary response areas for those units should be, and how many more (if any) ambulances would need to be deployed to meet the city's service standards.

Objectives for the ambulance service differed by interest group [12]. Department fiscal managers wanted to achieve an acceptable level of service at least cost. Political interest groups demanded equitable service to each of the city's neighborhoods. Members of the medical community wanted minimum response time as well as maintenance of existing patient referral patterns. Accordingly, ambulance system managers defined the following primary performance objectives [12]:

(i) to reduce citywide average response times to emergency calls;
(ii) to reduce citywide inequities in ambulance availability.

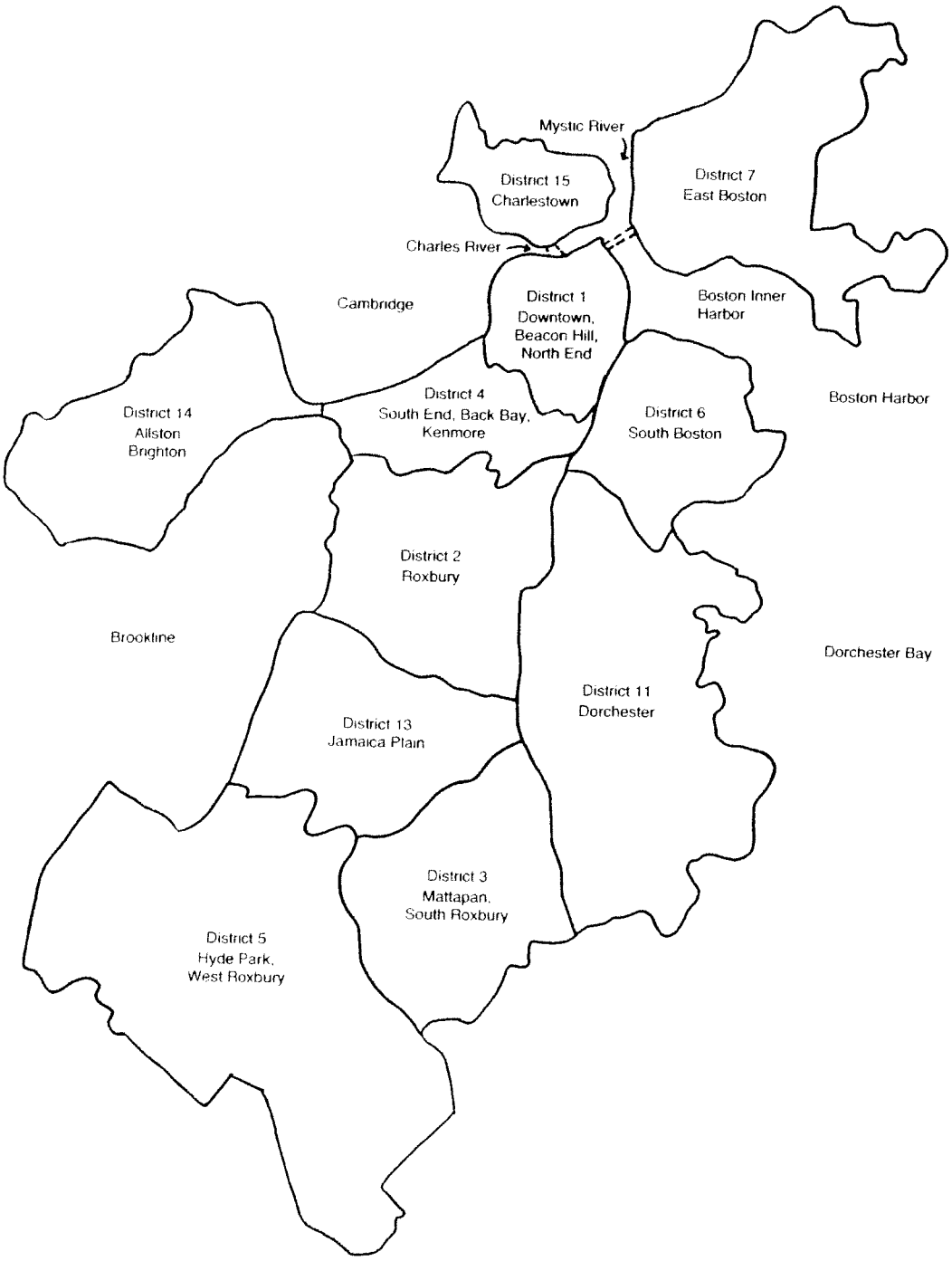

Fig. 3. City of Boston by police district.

Secondary performance goals deemed important were [1]:

(iii) to minimize ambulance workload imbalance;

(iv) to minimize the fraction of cells handled by backup units;

(v) to minimize the fraction of dispatches which are inter-district.

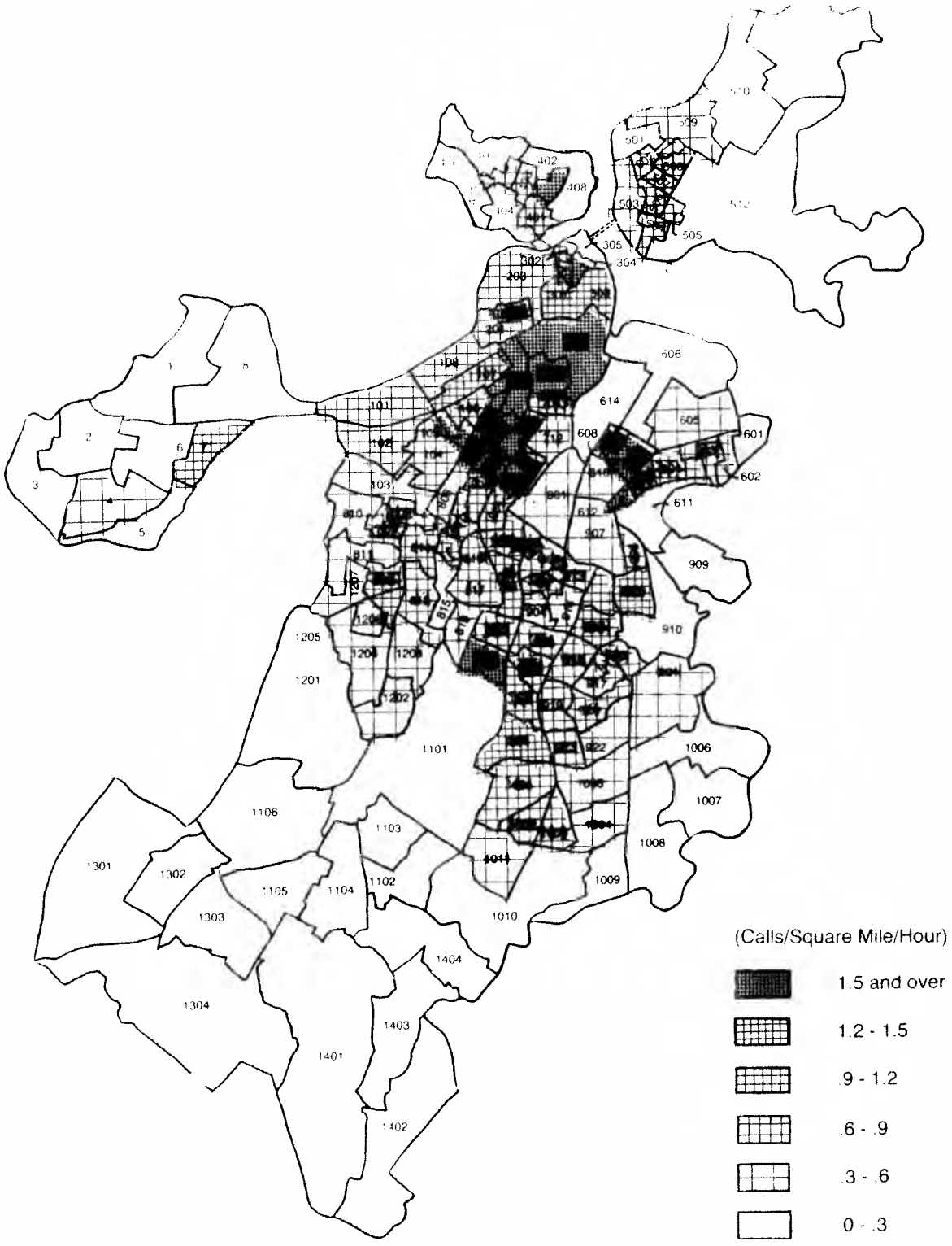

Fig. 4. Boston ambulance demand density by census tract.

4.2. Data collection and validation

Six basic data items are required as a minimum to run the Hypercube model:

(1) The city must be broken down into a number of reporting areas or *geographical atoms*.
(2) The relative *workload* of each of these atoms must be known.
(3) The user must know either
   (a) the average *travel time* between every pair of geographical atoms, or
   (b) the *(X, Y) coordinates* of each atom and the mean vehicle travel *speed*.
(4) The mean *total service time* and the mean *non travel service time* for the response units must be known.
(5) The idle locations or *satellite locations* of the response units must be known.
(6) The hourly system-wide *call rate* must be known.

4.2.1. Geographical atoms

In a typical application of the Hypercube model, the geographical atoms might correspond to police sectors, census tracts, or small collections of city blocks. For the Boston application, the city's 147 census tracts were used. Although current ambulance data were available only by police sector, census tracts were chosen as the geographical atoms so that model runs would be compatible with data from the city's new computer-aided dispatch system, which in the future would collect data by census tract.

4.2.2. Atom workloads

The relative workload of each of the geographical atoms shown in fig. 4 was obtained from one year of historical data (52000 emergency responses). To convert the data from police sector to census tract, conversion factors were obtained by placing a celluloid tracing of the Boston police sectors over a map of the census tracts and estimating the relative fraction of each police sector in each census tract. As can be seen from fig. 4, the highest density of emergency ambulance calls is generated in the relatively small downtown area, while the larger residential areas have a much lower call density.

4.2.3. Inter-atom travel times

In the original version of the Hypercube model the travel distance between every pair of geographical atoms was estimated as the right-angle distance between the center of the atoms, under the assumption that urban response units travel along mutually parallel and perpendicular paths (e.g., north–south and east–west streets). Such an assumption was clearly inadequate in a city like Boston, where irregular growth patterns (e.g., residential areas with large parks and cemeteries), natural obstacles (e.g., Boston Harbor and the Charles

River), and highway access (e.g., the Boston Expressway) presented significant barriers to travel, as shown in fig. 5. Barriers 1, 2, 3, 7, and 8 correspond to parks. Barriers 4, 5, and 6 correspond to large cemeteries. Barriers 9, 10, 11, and 12 are bodies of water. Barrier 13 represents the city of Cambridge (which is separated from Boston by the Charles River) and Barrier 14 represents an area of the city where there are no streets.

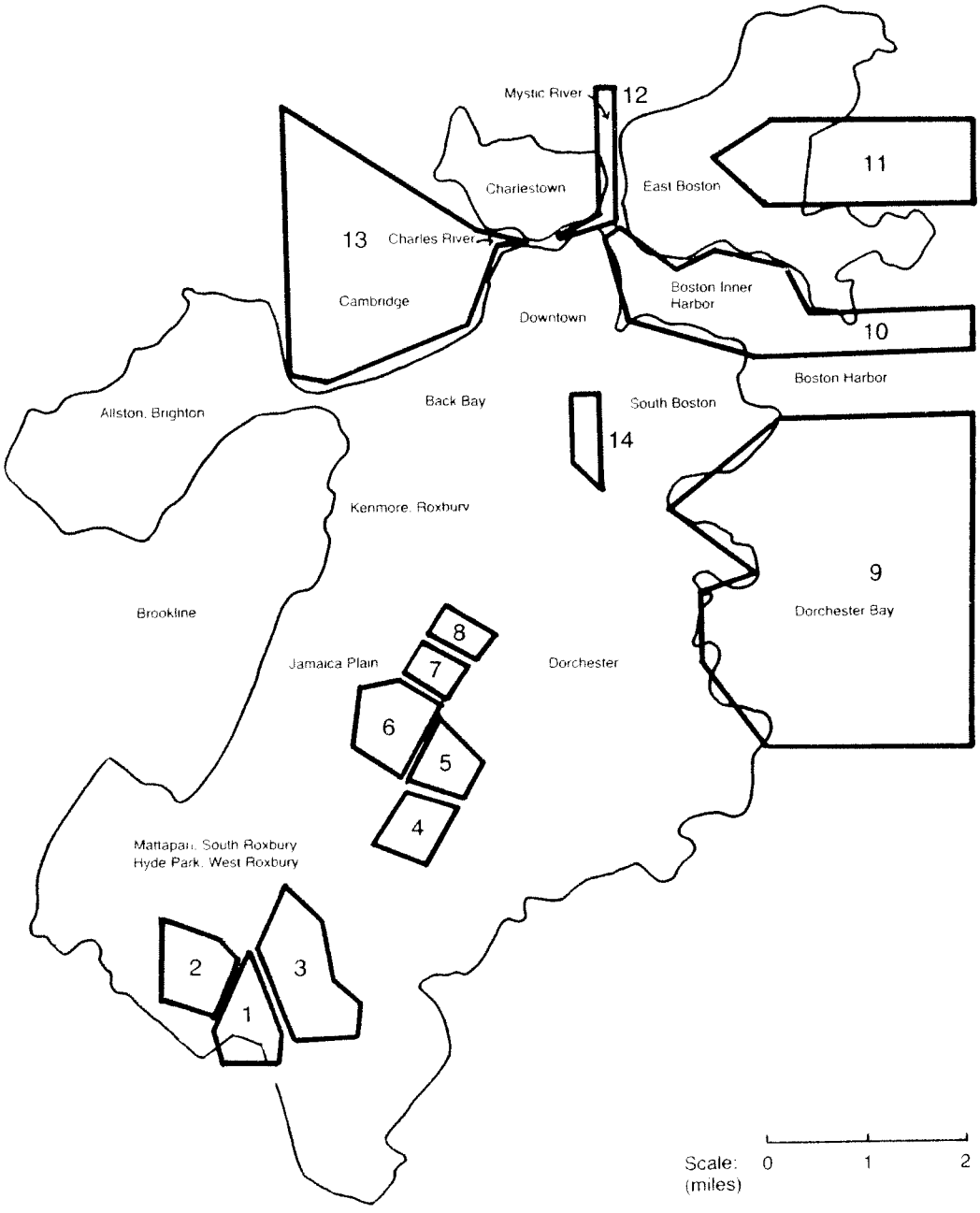

Fig. 5. The city of Boston showing barriers used in the barriers algorithm.

The inter-atom travel distances were estimated using the POLYPATH algorithm of Larson and Li [20] which calculates the distance between any pair of atoms as the right-angle distance plus any extra distance incurred by the presence of barriers. The travel distance totally within any geographical atom, rather than being set to zero as a simple right-angle calculation would assume, was estimated by the POLYPATH algorithm as

$\frac{1}{2}$(area of the census tract)$^{1/2}$.

Finally, the travel distances were transformed to travel times by dividing by a constant average travel speed of 17.3 miles per hour (obtained from a specially conducted study of 45 Boston ambulance trips).

4.2.4. Service time

Since the Mean Service Time Calibration feature of the model automatically computes the individual response times for each vehicle, it was only necessary to obtain an input value for the overall average service time and for the non-travel-related component of service time (see fig. 1). The total average service time, obtained from Boston computer data on 19000 emergency ambulance responses, was found to be 27.89 min. From the same sample, the total non-travel-related component of service time (17.00 min) was calculated as the sum of the average on-scene time (6.55 min) plus the average hospital time (10.45 min).

4.2.5. Satellite locations of ambulances

When idle, city ambulances in Boston remain at fixed home or "satellite" locations. These locations are often fire houses, hospitals, police stations, or civic centers. Idle locations for the ambulances were chosen separately for individual runs of the model, and will be shown below. It was assumed that all of the 147 geographical atoms were candidate locations for ambulance satellites since for all census tracts there was some type of center which could potentially house an ambulance.

4.2.6. Ambulance call rate

An average hourly call rate of 5.87 calls per hour was obtained by dividing the total number of emergency ambulance responses in 1977 (51418) by the total number of hours in the year (8760). However, because the level of emergency ambulance calls varies significantly by time of day (with evening hours the busiest and early morning hours the least busy), workload was calculated for each of the three daily ambulance shifts and was found to be:

Night Shift	(12 p.m.–8 p.m.) – 3.91 calls/h
Day Shift	(8 a.m.–4 p.m.) – 5.87 calls/h
Evening Shift	(4 p.m.–12 p.m.) – 7.83 calls/h.

(The fact that the number of calls per hour in the day shift was the same as the average daily number of calls per hour was quite by chance.)

4.2.7. Data validation

In order to validate the data, an initial 9-car run was made which was designed to reflect as closely as possible the actual ambulance operating conditions in Boston. (When the study was initiated, the city had 7 ambulances on duty, but over the next few months added two more vehicles, so that at the time the first Hypercube run was made, there were 9 available ambulances.) In an ideal situation the results of such a run could be used to closely calibrate and validate the model; that is, if the performance measures calculated by the model do not correspond to the various (known) performance measures with reasonable accuracy, the model data and/or calculations could be adjusted to make the model output consistent with actual data. Strict mathematical validation of this kind could not be carried out in Boston, however, since different numbers of ambulances were run at different times of day so that the dispatch preference list used by city dispatchers did not always correspond to that assumed by the model.

Despite the fact that strict mathematical validation could not be carried out, other more general means were used to examine the validity of the model output. First, the primary response area (or "district") for each ambulance as created by the model (based on closest distance) was almost the same as the actual district in all cases except one. Second, as expected, units in downtown areas were found by the model to have significantly higher workloads than units in residential areas. Third, the model-computed service time was found to be 29.30 min, which was only 5% higher than the input (actual average) service time of 27.89 min. Together, these three facts served to lend strong credibility to the model output (both in the views of the analysts and, more importantly, in the view of the Boston ambulance planners).

4.3. Model runs

During the course of the initial ambulance study, Hypercube model runs were made with 9, 10, 11, 12, 13, and 15-car configurations. The purpose of these runs was twofold: first, to assist Boston ambulance planners in determining appropriate locations and districts for the current 9 ambulances as well as for planned future numbers of ambulances, and; second, to educate the Boston planners in the use of the Hypercube model to enable them to use the model for future deployment planning. In this section we will describe the series of 9-car runs – the first of which corresponded to the city's actual deployment pattern at the time – and we will show how the model provided insights into possible operational changes which would improve overall system performance without adding more vehicles.

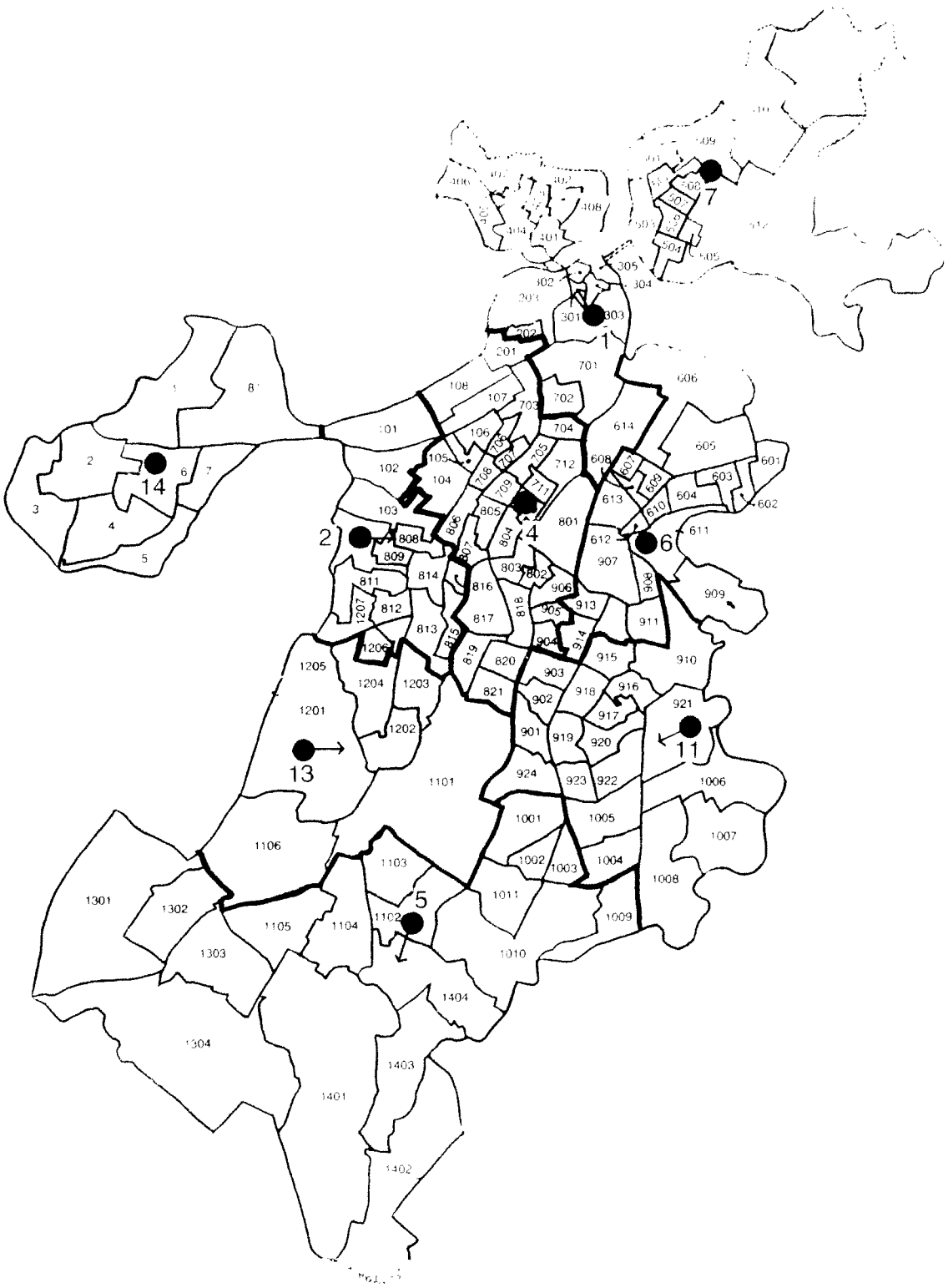

Fig. 6. First 9-car run.

The first 9-car run, shown in fig. 6, was made with ambulances located at the existing satellites. (We note that the ambulance's numbers in fig. 5 are not sequentially ordered between 1 and 9, but rather range between 1 and 14; this is because at the time of the study, Boston ambulances were labeled by the central "911" dispatcher according to the number of the police district they were in.) Looking at the selected performance measures in table 2, we can see that in this configuration District 5 (the residential areas of Hyde Park and West Roxbury) has a very high average response time. This suggested that ambulance 5 should be moved further south. However, while such a move would decrease the average system-wide response time and would help to equalize service accessibility to all neighborhoods, it would also increase the ambulance workload imbalance since the ambulance call density is lower than average in the southern part of the city. Thus a tradeoff must be made between the conflicting objectives. In this case, the first two objectives were deemed more important than the latter; ambulance planners felt that an average estimated response time of 10.1 min to any district was too high. They also believed that there was a certain amount of *latent demand* in that district – that due to the current high response times of city ambulances, many people living in that area relied on private ambulances, but that if response time were reduced, demand for city ambulances in that district would increase.

It also appeared that ambulances 2, 13, and 11 were all too close to the outer edges of the city and that it would make more sense to place them in more central areas. From the map one can see that there is a significant gap in ambulance coverage in the central area between ambulances 2, 4, 6, 11, 5, and 13. Moving ambulances 2 and 13 east and ambulance 11 west would help reduce the average ambulance response time in two ways: each of these three districts would have lower response times since the ambulances would be located more centrally within them, and all of the districts (especially 2, 4, 6, 11, 5, and 13) would have lower response times because the distance traveled in interdistrict dispatches would be lower.

The response time in parts of Charlestown (the northern part of District 1) was also somewhat high, so it was decided to move ambulance 1 slightly farther north; likewise ambulance 7 could be moved northeast to decrease the very high response time in census tract 511. District 7 is an example of an area of the city in which ambulance planners must make a tradeoff between reducing the average district-wide response time and equalizing service accessibility to the different geographical atoms. The average response time in District 7 would be minimized if the ambulance 7 satellite were located in census tract 508. However, then the average response time to census tract 511 would be 13.6 min. If the ambulance satellite were moved northeast, the response times to the different census tracts in the district would be more equal but the average district-wide response time would be increased. In this case it was decided to move ambulance 7 slightly northeast.

Table 2
Selected performance measures from three nine-car runs

District	First 9-car run			Second 9-car run			Third 9-car run		
	Average response time to district	Unit work-load	Maximum response time to district	Average response time to district	Unit work load	Maximum response time to district	Average response time to district	Unit work-load	Maximum response time to district
1	7.045	0.367	12.34	6.897	0.332	11.20	6.418	0.318	10.78
2	7.113	0.339	8.69	6.696	0.380	8.62	7.327	0.382	8.24
4	7.298	0.479	9.57	7.049	0.455	8.99	6.417	0.437	9.02
5	10.054	0.309	15.74	8.728	0.236	14.62	8.886	0.241	14.44
6	7.246	0.380	9.07	7.322	0.365	11.60	6.631	0.354	10.36
7	7.193	0.186	13.59	7.487	0.177	12.96	7.202	0.162	12.55
11	8.339	0.354	10.45	7.997	0.359	10.98	8.433	0.374	11.16
13	8.232	0.271	8.96	7.863	0.313	8.45	8.117	0.336	9.13
14	6.834	0.318	9.13	7.139	0.173	9.43	7.158	0.174	9.50
Average or maximum	7.816	0.318	15.74	7.479	0.310	14.62	7.408	0.309	14.44

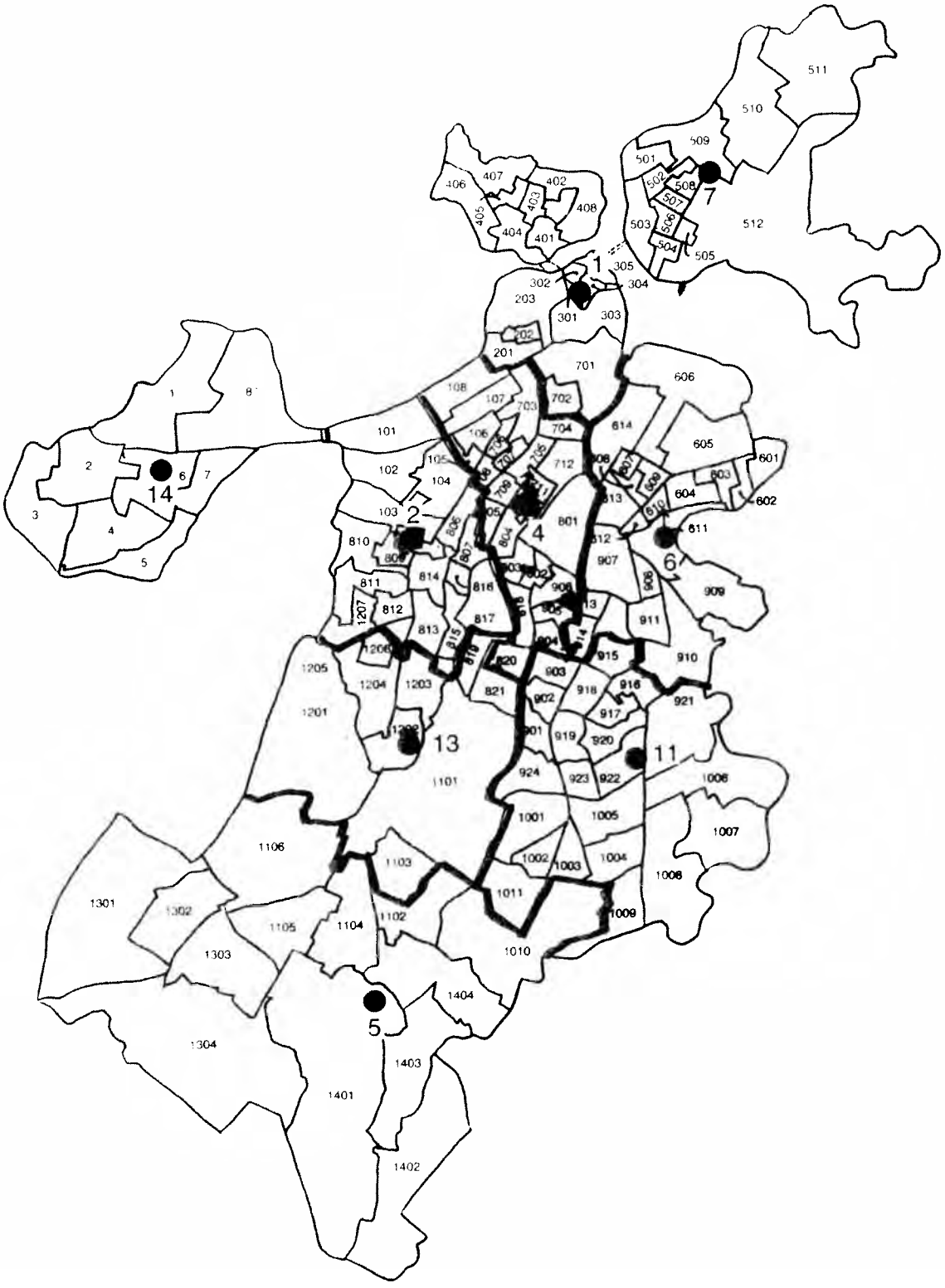

Fig. 7. Second 9-car run.

Because there was no geographical atom located slightly northeast of census tract 508, a *dummy atom* was added to the model. This atom is located at the intersection of census tracts 508, 509, and 512. In the same way, a dummy atom was added to District 5; it is located in the northeastern corner of census tract 1401, just south of the boundary between census tracts 1104 and 1404. These dummy atoms have zero ambulance demand; they are only included to be used as possible locations for an ambulance satellite.

The five changes discussed here, indicated by arrows in fig. 6, were made to the input data and the model was rerun. Fig. 7 shows the configuration for the second nine-car run. One can see from table 2 that the changes in the satellite locations had the effect of reducing the average system-wide response time from 7.8 min to 7.5 min. Furthermore, the average ambulance workload was decreased from 0.318 to 0.310 and the maximum average response time to any one census tract decreased from 15.74 min to 14.62 min. Although the workload of ambulance 5 decreased, the standard deviation of the ambulance workloads (which is one measure of the workload variability) did not increase. Thus, the changes made from the first nine-car run improved *all* of the key system performance characteristics.

Can the second nine-car configuration now be improved upon? From table 2 one can see that ambulance 4 is by far the busiest ambulance in this configuration. If ambulance 4 were less busy, other ambulances would not have to answer calls in District 4 as often, and this would probably help to decrease the average response time. It was therefore decided to move ambulance 4 north to census tract 705 to give it a smaller primary response area. It was also decided to move ambulance 6 north one census tract where it would be more centrally located within its district and could more easily serve as a backup for calls from District 4. These two changes were made to the model and the model was rerun.

The changes between the second and third runs were not as dramatic as those seen between the first and second runs, but one can see from table 2 that the two changes did improve the overall system performance characteristics. The average ambulance response time decreased from 7.5 min to 7.4 min and the average ambulance workload decreased from 0.310 to 0.309, while the standard deviation of the workload did not increase. At this point it was deemed that the third nine-car run was as good a configuration as could be expected, given nine ambulances.

4.4. Results

As a result of analyzing the Boston ambulance system with the above 9-car runs, several existing ambulances were moved to the new locations suggested in the second and third model runs; in particular, ambulance satellites near the edges of the city were moved inward. A second operational change came about

somewhat indirectly from the model results. One input to the Hypercube model is a dispatch preference list which is calculated automatically using the vehicle home locations and the travel time matrix, and which is used to calculate many of the system performance measures. For each geographical atom, the model generates a list of ambulances in preferred order of response, where the most preferred unit is the one whose home base is closest to the atom, the second preferred unit is second closest, etc. In examining the model-created dispatch preference list, it was discovered that Boston dispatchers did not always dispatch the closest available ambulance since the dispatch system they were using had been designed for the police department. This system, generated as a part of the city's computer aided dispatch system, was designed around police sectors and police districts and gave only a rather crude representation of ambulance districts and ambulance interdistrict response patterns. To rectify this situation, the dispatch preferences for interdistrict dispatches generated by the model were printed out in a way that could be transferred to the dispatcher's station, and the dispatcher was directed to utilize this information in making interdistrict dispatch decisions.

To analyze the effects of these two changes, response time statistics for the city's emergency ambulance service were analyzed for two successive one-week periods in 1979 [12]. In the first week no changes were made, while in the second week both changes were implemented. As shown in fig. 8, average citywide response time decreased from 7.54 to 7.09 min. Furthermore, the fraction of responses greater than 15 min substantially decreased, so that more uniform service was provided. These improvements in the ambulance system performance were achieved without adding any new vehicles, but merely by redeploying the existing 9 vehicles. Hill et al. [12] have estimated that in order to achieve the same service improvements without changing the original 9-car configuration, at least one ambulance would have had to have been added to the available fleet, so that by acting on the insights gained through the use of the model, the city was able to save $150 000 per year. The model continued to be used by Boston ambulance planners after the initial study, both as a tool for planning ambulance deployment and as a source of data for ongoing efficiency studies.

4.5. Effect of model improvements

We conclude our discussion of the Boston ambulance study by describing the effect of the three Hypercube model improvements on the realism of the model results.

4.5.1. Varying service times and MSTC

Table 3 shows the results of the first 9-car run, with and without the varying service times and MSTC features. For the run with MSTC (Run 1) and the

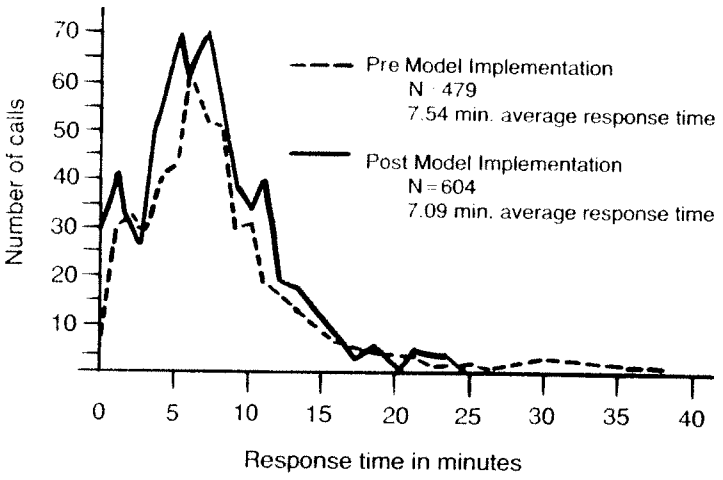

Fig. 8. Response time frequency distribution.

first non-calibrated run (Run 2), mean service time input to the model was 27.89 min. As can be seen from table 3, the service time values and related performance measures from the non-calibrated run differed significantly from the model-calibrated values obtained using MSTC. Perhaps the most important difference is that the overall averages of the performance measures are roughly 5% lower in the non-calibrated version.

Now, suppose that the average service time is manually calculated from the output of Run 2 by the ambulance planner to be 29.30 min (thus providing a better guess of mean service time), and the model is rerun with equal service times for all servers of 29.30 min (but no MSTC). In this case, even though the *average* service time value is equal to that from the MSTC run, the model-computed performance measures still vary significantly from the MSTC output measures because in the non-calibrated run, all service times are assumed to be equal. For example, as shown in table 3, the workload of ambulance 1 is 7% lower in the calibrated run than in the non-calibrated run, reflecting the fact that ambulance 1 has a lower than average mean travel time. Likewise, ambulance 8 has a workload 11% higher in the MSTC run, in accordance with its higher than average travel time. Thus, while the average service time is the same in both runs, the ambulance-specific and district-specific performance measures are not the same; even when the overall average service times are equal, the MSTC results are more accurate.

MSTC was also used in 10, 11, 12, and 15-car runs. In these runs, it was known that the actual service time would be less than 27.89 min, but it was not known by how much, since actual data were available only for the 9-ambulance system; thus, input service times for all servers of 27.89 min were used. Again, MSTC proved useful in calibrating the overall average service time and in calculating a mean service time for each server.

Table 3
Comparison of MSTC and non-MSTC results

Unit	Service time					Unit workload					District-specific travel time				
	Run 1 [a]	Run 2 [b]	Per cent difference	Run 3 [c]	Per cent difference	Run 1	Run 2	Per cent difference	Run 3	Per cent difference	Run 1	Run 2	Per cent difference	Run 3	Per cent difference
1	26.56	27.89	+5.0%	29.30	+10.3%	0.367	0.377	+2.7%	0.393	+7.1%	5.05	5.20	+3.0%	5.34	+5.7%
2	29.56	27.89	−5.6%	29.30	−0.9%	0.339	0.324	−4.4%	0.342	+0.9%	5.11	4.81	−5.9%	4.92	−3.7%
3	27.22	27.89	+2.4%	29.30	+7.6%	0.479	0.483	+0.8%	0.500	+4.4%	5.30	5.43	−2.5%	5.54	+4.5%
4	31.89	27.89	−12.5%	29.30	−8.1%	0.309	0.278	−1.0%	0.290	−6.1%	8.05	8.37	+4.0%	8.49	+5.5%
5	29.19	27.89	−4.5%	29.30	+0.4%	0.380	0.367	−3.4%	0.386	+1.6%	5.25	5.16	−1.7%	5.29	+0.8%
6	30.55	27.89	−8.7%	29.30	−4.1%	0.186	0.175	−5.9%	0.188	+1.1%	5.19	5.60	+7.9%	5.76	+11.0%
7	29.69	27.89	−6.1%	29.30	−1.3%	0.354	0.336	−5.1%	0.351	−0.8%	6.34	6.41	+1.1%	6.53	+3.0%
8	34.05	27.89	−18.1%	29.30	−14.0%	0.271	0.226	−16.6%	0.242	−10.7%	6.23	5.32	−14.6%	5.41	−13.2%
9	28.53	27.89	−2.2%	29.30	+2.7%	0.180	0.170	−5.6%	0.181	+0.6%	4.83	4.75	−1.7%	4.88	+1.0%
Average	29.30	27.89	−4.8%	29.30	+0.0%	0.318	0.304	−4.4%	0.319	+0.3%	–	–	–	–	–

[a] Run with initial service time = 27.89, MSTC.
[b] Run with service times = 27.89, no MSTC.
[c] Run with service times = 29.30, no MSTC.

The introduction of mean service time calibration raises a number of important questions regarding the robustness and accuracy of the MSTC procedure. Will the mean service time calibration procedure always converge? If not, under what conditions does it not converge? How do the input service time values affect the final calibrated service time values (and thus the related performance measures) which are output by the model?

In order to investigate these issues, six runs of the model were made using actual Boston ambulance data. The runs were identical except for the initial input service times. The first five runs used uniform input service times. Taking 27.89 min as a "best guess" of the average service time, four additional runs were made with uniform input service times that were 10% high (30.83 min), 5% low (26.50 min), 10% low (25.10 min), and 25% low (20.85 min). For all runs the response unit service times for each server converged to the *same value* (within a reasonable roundoff error) in every case – no matter what the value of the input service time. Likewise, the output performance measures converged to the same values each time (this is less surprising, however, since the only differences in the input to the various runs were the service times).

Then, a sixth run of the model was made with non-uniform input service times. Three servers (selected at random) were assigned input service times equal to the constant non-travel time component of the service time (17.00 min) and the remaining six servers were assigned very large service times (36.00 min). The service times again converged to the same values computed in the first five test runs.

In all of the above runs, the constant portion of the service time (C_i) was approximately 60% of the total service time. In order to further test the convergence properties of MSTC, the model was run with the non-travel component of the service time set equal to zero – that is, with

$$1/\mu_i = \bar{\tau}_i = 2.1(\tau_i). \tag{7}$$

An initial value for the response unit service times was obtained by estimating the travel time as the sum of intra-district and inter-district travel, as described by Larson [18]. Assuming that response unit districts are roughly rectangular and that each vehicle, while idle, is located at the center of its district, yields the following approximation:

$$\hat{\tau}_i = \left[\tfrac{1}{2}\cdot V^{-1}(A/N)^{1/2}\right] + \left[\tfrac{1}{2}\cdot V^{-1}\cdot(A/N)^{1/2}\tfrac{3}{4}\rho\right], \tag{8}$$

where V is the average vehicle response speed, and A is the total area of the city, and ρ is the average utilization rate. The estimated average service time then becomes

$$1/\hat{\mu}_i = 1/\hat{\mu} = (2.1)(\tfrac{1}{2})\cdot V^{-1}\cdot(A/N)^{1/2}(1+\tfrac{3}{8}\rho). \tag{9}$$

Using this estimate for the initial average service time of each server, the Hypercube model was run under a range of utilization rates (from roughly 0 to 60%). In every case, the MSTC procedure converged within 6 or fewer iterations. Furthermore, the above service time approximation was shown to be

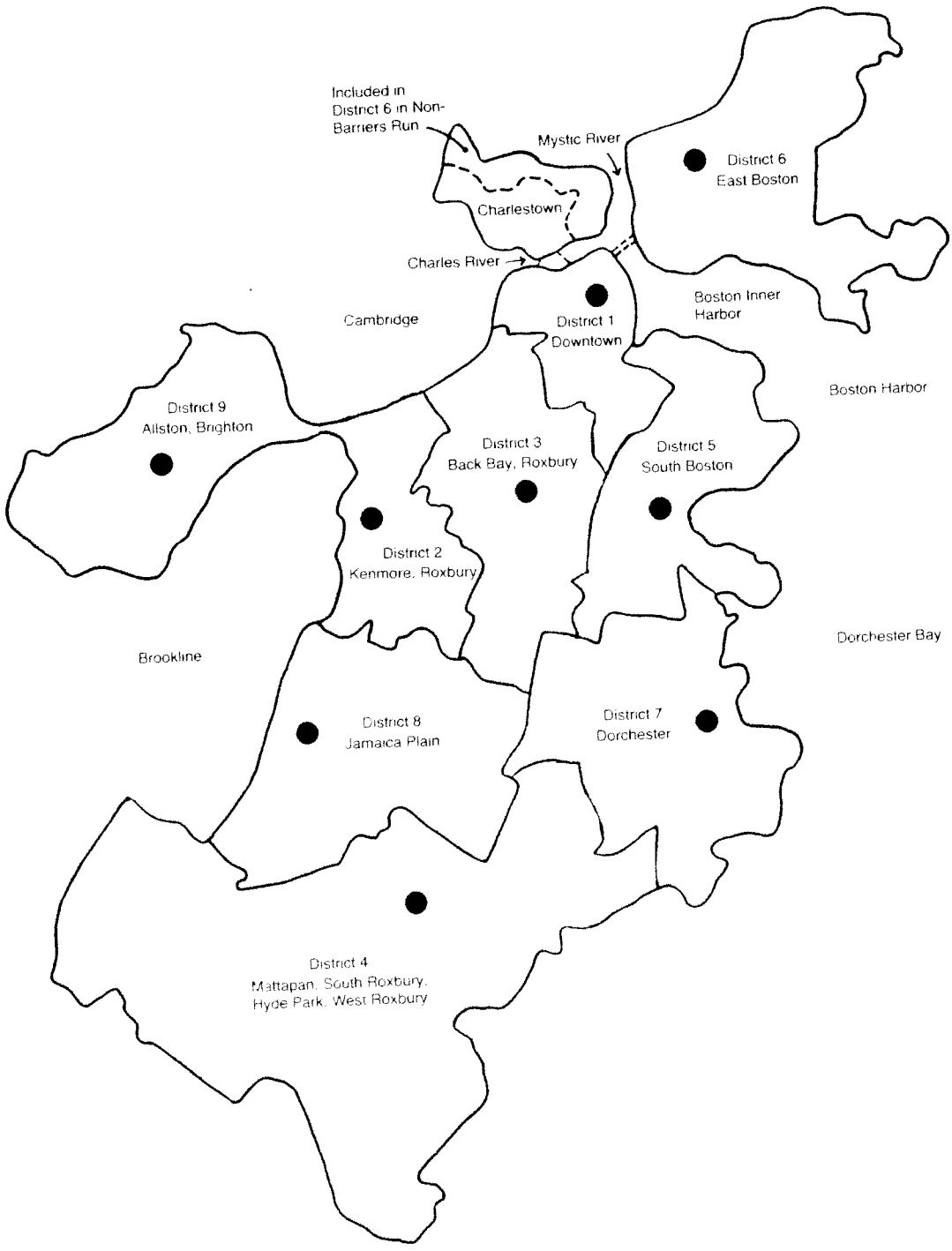

Fig. 9. District configurations for two 9-car runs.

quite close to the final calibrated average in cases with utilization less than 40%.

MSTC has been shown to converge under a wide range of conditions and substantially improves the accuracy of the Hypercube computations. In cases where the average service time of the response vehicles is not known, a good initial estimate of the service time is given by

$$1/\hat{\mu}_i = 1/\hat{\mu} = C_i + c\hat{\tau}_i, \tag{10}$$

where $\hat{\tau}_i$ is given by eq. (8) [2].

4.5.2. Barriers algorithm

To determine the effect of barriers on model-computed travel times, two runs of the Hypercube model with measured Boston ambulance data were made, with one using the Barriers algorithm and one ignoring the barriers using the previous (right-angle) travel time estimation algorithm. For each run identical satellite locations for 9 ambulances were input. The Hypercube then created primary response areas or districts for each unit so that each census tract is in the district of the ambulance estimated to be closest.

As can be seen in fig. 9, the two district configurations are identical except that in the non-barriers run, an area of Charlestown (corresponding to two geographical atoms) is included in District 6 when it should actually be included in District 1 (since travel time across the Mystic River/Boston Harbor is actually much larger than that estimated by the simple right-angle algorithm). There are also differences between preference orderings for non-primary units caused by inclusion of the barriers.

The overall effect of the Barriers algorithm was to increase the model-computed travel times [3] (as they should be) but more importantly, to create more accurate dispatch preference orderings for each area of the city. The Hypercube model results using the Barriers algorithm for the input travel times are more realistic – and thus more useful to the emergency service planner – than those obtained using the simple right-angle algorithm.

5. Summary and conclusions

The Hypercube model was successfully used to improve ambulance deployment in Boston. The model satisfied the Boston planners' original criterion of

[2] A similar travel time estimate for applications with non-stationary or patrolling vehicles is contained in [18].

[3] The two runs here represent light workload conditions; under a heavier system workload, however, the travel time differences would be much more significant because non-primary units would be dispatched more often.

finding ambulance satellite locations and primary response areas which would meet their service objectives better than the current deployment plan. The original Boston ambulance districts had been designed around police districts and did not always reflect ambulance demand patterns. By considering ambulance demand separately, as well as potential barriers to travel, the Hypercube model helped Boston planners determine new ambulance districts which paralleled, rather than crossed, major barriers to travel and which were specifically designed around ambulance demand. Use of the model also had the secondary (unforeseen) result of pinpointing suboptimal dispatch decisions, so that a new dispatch preference ordering for non-primary units was developed.

With an estimated cost-benefit ratio of six to one in the first year [12], and with minimal expense for model upkeep, use of the model proved to be an important factor in increasing the productivity of the Boston ambulance system. Application of the model was successful for several reasons. First, the model provided an objective means of evaluating different deployment plans which could be defended before civic, health, and political interest groups. Second, by not focusing on any single performance measure, the model allowed ambulance planners to explicitly determine appropriate tradeoffs between sometimes conflicting objectives. Third, since Boston ambulance planners were trained during the initial study in the use of the Hypercube model, the model became an "in house" model which continued to be used for deployment planning after the initial study. Finally, Boston ambulance planners were firmly committed to the idea of using an analytic tool to help them make deployment decisions, so that they were willing to defend the use of the model and the insights gained from it before groups such as the Boston City Council.

The three improvements incorporated into the model also helped to enhance its realism, as described earlier. Two possible improvements which could be incorporated into the model for future applications are as follows: First, a "layering" feature could be introduced to distinguish between call and vehicle types. Such a layering feature could be used when there is more than one kind of emergency service vehicle in the system being modelled, and different vehicles can provide different levels of service.

For example, suppose that an ambulance service has both Basic Life Support (BLS) and Advanced Life Support (ALS) units. The ALS units can provide better care to critical incidents and should thus be first-preferred responders for such incidents, while BLS units should be first-preferred for non-critical incidents. By extending the Hypercube model to include different EMS call types (e.g., trauma, cardiac, respiratory, burns, spinal cord injuries, etc.) and different vehicle types (e.g., ALS and BLS), the ambulance planner could use the Hypercube model to create a much more effective deployment plan for the vehicles than he could by considering all calls and servers to be the same or by using the model separately for each response unit type. This would

be especially useful for modelling urban regions where downtown and poorer areas typically have many more critical EMS (or police) calls than outlying or more affluent areas.

If an ambulance planner were to use the Hypercube model with both ALS and BLS units treated equally, the resultant deployment plan could not fully exploit the unique capabilities of the different units. If the planner were to run the Hypercube two separate times, once with the BLS units and non-critical calls, and once with the ALS units and critical calls, he would not be taking into account potential workload sharing between the two vehicle types. By categorizing the call and vehicle types, the Hypercube model can distinguish between vehicle capabilities while at the same time allowing any necessary workload sharing.

A layering extension to the Hypercube model would also be useful for police applications in which a city has both one-officer and two-officer patrol units.

A second possible extension to the Hypercube model would be to introduce "survival functions" for the response time. Currently one measure of system effectiveness computed by the Hypercube is the average vehicle response time to incidents [4]. However, response time is in fact a *surrogate measure*; what the EMS planner is actually interested in is lives saved or hospital days averted. Survival functions transform response time into a more pertinent performance measure and place appropriate weights on different response time values.

Survival functions could most usefully be incorporated into the Hypercube model extended by call-type and vehicle-type, as described above. Suppose that there are two vehicle types i and a number of different incident types j. Then for each call type j handled by unit i there would be a survival function S_{ij} such that

$S_{ij}(t)$ = expected number of lives saved when a vehicle of

type i responds to a call of type j in time t.

Two such functions for a given call type j might look like those shown in fig. 10.

Call-type and vehicle type categorizations coupled with survival functions for response time would enable the emergency service planner to more accurately evaluate the utility of various deployment plans in situations with distinguishable servers. The ambulance planner, for example, could detemine the *actual effectiveness* of different response time levels for critical and non-

[4] Although, as mentioned earlier, the Hypercube model is non-optimizing, so that the user may decide upon a "best" deployment plan based not only on response time but also on unit workload, the fraction of calls handled by backup units, etc.

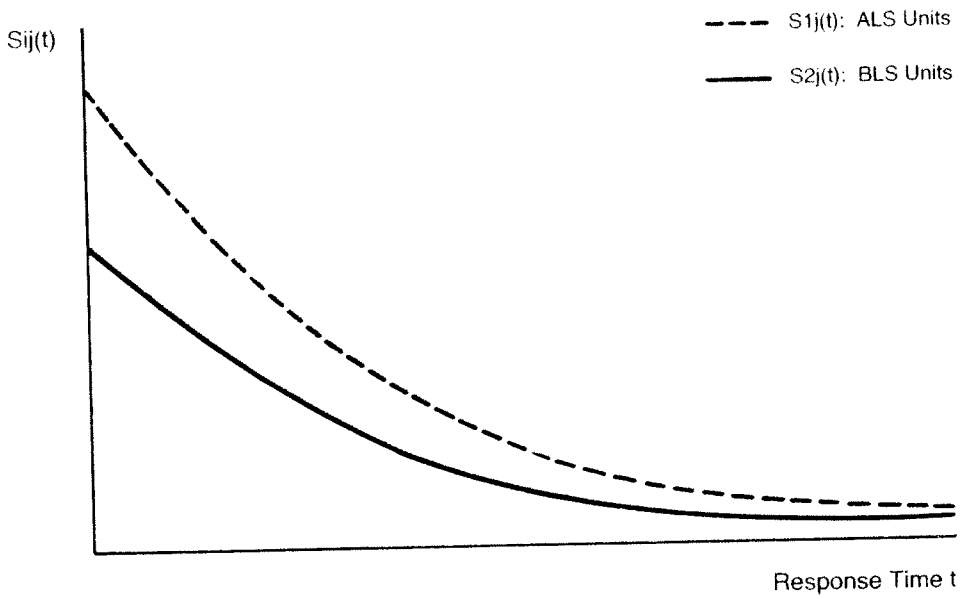

Fig. 10. Possible survival functions for call type j.

critical incidents, and could use the Hypercube model to assist in creating a deployment plan in which critical incidents are responded to more quickly (on the average) than non-critical incidents.

The major problem with the use of survival functions is the lack of adequate data for such functions. However, as a crude first step it should at least be possible to estimate survival functions for critical and non-critical calls.

Acknowledgement

We are indebted to Dr. Lenworth M. Jacobs and to E.D. Hill and J.L. Hill for making possible the Boston implementation of the Hypercube model.

References

[1] M.L. Brandeau and R.C. Larson, "Implementing the Hypercube Queuing Model to Plan Ambulance Districts in Boston", Public Systems Evaluation, Inc., Cambridge, Massachusetts (September 1978).

[2] A. Carter and E. Ignall, "A Simulation Model of Fire Department Operations: Design and Preliminary Results", *IEEE Transactions on Systems Science and Cybernetics* SSC-6 (1970) 282–293.

[3] J.M. Chaiken, "Implementation of Emergency Service Deployment Models in Operating Agencies", Rand Corporation, Santa Monica, California (1979) (Rand Paper Series P-5870).

[4] J.M. Chaiken, *Number of Emergency Units Busy at Alarms Which Require Multiple Servers* (Rand Institute, R-531-NYC/HUD, New York City, New York, 1971).

[5] J.M. Chaiken and R.C. Larson, "Methods for Allocating Urban Emergency Units: A Survey", *Management Science* 19, No. 4, Part II (1972) 110–130.
[6] K.R. Chelst, "Implementing the Hypercube Queuing Model in the New Haven Department of Police Services: A Case Study in Technology Transfer", Rand Corporation, Santa Monica, California (July 1975) (Rand Paper Series R-1566/6-HUD).
[7] M.S. Daskin and E.H. Stern, "A Hierarchical Objective Set Covering Model for Emergency Medical Service Vehicle Deployment", *Transportation Science* 15 (1981) 137–152.
[8] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik* 1 (1959) 269–271.
[9] F.X. Finn, H.W. Findley and J.F. Molloy, "Quincy (Massachusetts) Police Department: Application of the Hypercube Model for Sector Design Analysis", in: Richard C. Larson, ed., *Police Deployment* (Lexington Books, Lexington, Massachusetts, 1978) pp. 139–158.
[10] J.A. Fitzsimmons, "A Methodology for Emergency Ambulance Deployment", *Management Science* 19, No. 6 (1973) 627–636.
[11] J. Halpern, "The Accuracy of Estimates for the Performance Criteria in Certain Emergency Service Queuing Systems", *Transportation Science* 2, No. 3 (August 1977) 223–242.
[12] E.D. Hill, J.L. Hill and L.M. Jacobs, "Planning for Emergency Ambulance Systems", City of Boston, Department of Health and Hospitals, mimeographed (1981).
[13] J.M. Hogg, "The Siting of Fire Stations", *Operational Research Quarterly* 19 (1968) 275–287.
[14] J.P. Jarvis, "Optimization in Stochastic Service Systems with Distinguishable Servers", Innovative Resource Planning in Urban Public Safety Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts (June 1975).
[15] J.P. Jarvis and M.A. McKnew, "Applying the Hypercube in Arlington, Massachusetts", in: Richard C. Larson, ed., *Police Deployment* (Lexington Books, Lexington, Massachusetts, 1978) pp. 123–137.
[16] R.C. Larson, "Illustrative Police Sector Redesign in District 4 in Boston", *Urban Analysis* 2 (1974) 51–91.
[17] R.C. Larson, "A Hypercube Queuing Model for Facility Location and Redistricting in Urban Emergency Services", *Computers and Operations Research* 1, No. 1 (March 1974) 67–95.
[18] R.C. Larson, *Urban Police Patrol Analysis* (MIT Press, Cambridge, Massachusetts, 1972).
[19] R.C. Larson, "Approximating the Performance of Urban Emergency Service Systems", *Operations Research* 23, No. 5 (September–October 1975) 845–868.
[20] R.C. Larson, "Structural System Models for Locational Decisions: An Example Using the Hypercube Queuing Model", in: K.B. Haley, ed., *Operations Research*; 1978 – Proceedings of the 8th IFORS International Conference on Operations Research (North-Holland, Amsterdam, 1979).
[21] R.C. Larson and V.O.K. Li, "Finding Minimum Rectilinear Distance Paths in the Presence of Barriers", *Networks* 11 (1981) 285.
[22] E.S. Savas, "Simulation and Cost-Effectiveness Analysis of New York's Emergency Ambulance Service", *Management Science* 15, No. 12 (1969) B608–B627.
[23] C. Swoveland et al., "Ambulance Location: A Probabilistic Enumeration Approach", *Management Science* 20, No. 4, Part II (1973) 686–698.
[24] C. Toregas et al., "The Location of Emergency Service Facilities", *Operations Research* 19 (1971) 1363–1373.