Comput. & Ops Res., Vol. 4, pp. 1–12. Pergamon Press, 1977. Printed in Great Britain

A METHOD FOR DETERMINING THE OPTIMAL DISTRICTING IN URBAN EMERGENCY SERVICES*

PAOLA BERTOLAZZI†
Istituto di Automatica, Università di Roma, Via Eudossiana, 18-00184 Roma, Italy

LUCIO BIANCO‡
Centro di Studio dei Sistemi di Controllo e Calcolo Automatici del C.N.R., Via Eudossiana, 18-00184 Roma, Italy

and

SALVATORE RICCIARDELLI§
Centro di Studio dei Sistemi di Controllo e Calcolo Automatici del C.N.R., Via Eudossiana, 18-00184 Roma, Italy

Scope and purpose—This study is part of a research on the system theory and operations research application to the public services. A research program on urban emergency services, especially referring to the fire department service of Rome, was begun in 1975.

In this paper we are dealing with the problem of determining the primary response areas of the individual stations or units in a certain region in order to improve the service performance.

After a brief review of the most relevant studies that have been reported within the past years [1, 2, 4, 5] we propose a formulation of the problem that minimizes the overall travel time within the region under consideration, taking into account the spatial and temporal distribution of the calls.

The mathematical model has been applied to the city of Rome. The results show that the distribution of the responsibility for first intervention among the various districts, obtained with the mathematical model, could permit a reduction in the overall travel time of about 26%.

The essential aim of this work is to suggest an approach to the districting problem in terms of optimization of the entire system, which may help the people in charge of emergency services.

As a matter of fact, the proposed method could permit an improvement in emergency services, without changing the available resources.

Abstract—In this paper a method is suggested for dealing with one of the main allocation problems often appearing in urban emergency services, that of districting.

A mathematical model is developed in terms of constrained optimization with binary variables.

An algorithm, derived from Balas' filter-method, which seems to be quite efficient for this class of problems, has been utilized.

The objective function, considered here, is the overall travel time in the region under consideration.

The optimal travel time, in the case of the Rome fire service, turns out to be meaningfully lower than that derived from the actual situation. At the end is indicated a way of applying the method to the assignment of response areas to the different stations as a function of the current state of the system.

1. INTRODUCTION

The districting problem is one of the most important and recurring allocation problems in urban emergency services. It can be formulated as follows: "Given a region with a known spatial distribution of demands for service and given N response units, whose location is also known; how should the region be partitioned into areas of primary responsibility (districts), so that the quality service be the best possible?"

The approach generally followed is that of determining a response area for every unit. That is, each unit responds to all the alarms coming from its own area, unless it is already busy.

*This work was partially supported by the Consiglio Nazionale delle Ricerche (C.N.R.).

†Paola Bertolazzi holds a degree in electronic engineering from the University of Rome. She has obtained a scholarship of the Italian Department of Education (Ministero della Pubblica Istruzione) at the Istituto di Automatica of the School of Engineering, University of Rome, where she carries on her activity in the operations research group, working on territorial systems.

‡Dr. Bianco heads the Operations Research group of The Centro di Studio dei Sistemi di Controllo e Calcolo Automatici. He is also Associate Professor of mechanics at the Department of Engineering of Naples and holds the same rank in operations research at the University of Rome. He is a member and officer of a number of Italian National Science and Engineering Boards and committees, and has published extensively on Control Theory and Operations Research. He holds degrees in both electronics and engineering and aerospace engineering from the University of Rome.

§Dr. Salvatore Ricciardelli is associate professor of operations research at the University of Rome. He holds degrees in both electronic engineering and in aerospace engineering from the University of Rome. He has worked extensively in control theory and its applications to communications satellites and has presented several papers on this subject, together with Dr. Bianco. These include papers at the 12th and 13th European Space Symposea and the 14th International Astronautical Congress. He is now working on urban problems.

If it is, it is up to the other available units to respond on the basis of a predetermined dispatching policy.

Traditionally, emergency services planning regarding the districting problem has been made by partitioning the region into regular areas (square or circular), according to the minimum distance criterion. This meets the requirements of a dispatching policy assigning to each call the closest available unit[1]. Such a districting policy does not reflect the true situation because:

(a) it does not take into account the actual changes in travelling time along the different routes of a large city;

(b) it does not take into account natural barriers (parks, rivers, railways etc.) and the actual street network that makes the route distribution discrete instead of continuous.

For point (b), mathematical models, considering the actual situation have been developed[2].

As regards point (a), it is necessary to replace the "distance" concept with the "travel time" concept. This means that all points on the dividing line between two districts are at the same travel time from the respective units.

Even interpreting "closest" in the sense of shortest travel time, the above mentioned criterion is not, all the same, fully satisfactory. Although the average travel time of each unit is made minimum, that does not necessarily imply that the average travel time of the whole system (that is the average response time) will be the minimum. As a matter of fact, operating in this way, due consideration is not given to the call rate, which is a basic parameter for minimizing the overall average travel time and balancing the workload among the units. Besides, if the mathematical model includes the call rate as well, the conclusion can be reached that dispatching the closest available units does not necessarily minimize the system average travel time[3]. Carter, Chaiken and Ignall[4] analysed the case of two fixed-position response units and rigorously derived the optimal districts, considering the stochastic behaviour of the system and assuming a simplified form of interdistrict cooperation. Furthermore, they considered two performance parameters: average travel time and workload balance between the units.

This mathematical model is not utilizable when the units are more than two, but it provides useful insight into certain aspects of the districting problem for emergency services with fixed unit location.

More recently Larson worked out a sophisticated model (valid both for fixed unit and mobile unit systems) based on the queuing theory, including probabilistic phenomena and interdistrict interactions as involved in an actual system[5]. This model investigates the system from various points of view. In fact, it defines and determines several system performance measures.

However, the Larson approach is not an optimization model, because it is based on pre-established dispatching policies, which do not include an objective function optimization such as, for instance, the average response time. Furthermore computational difficulties arise when the whole number of units is more than $10 \div 12$.

In our paper we suggest coping with the districting problem in terms of constrained optimization. The objective function, considered here, is the system average travel time; the unit workload balance is expressed in the constrained equations.

2. MATHEMATICAL MODEL

In the districting problem the emergency units can be considered the system resources, while the various region points are the potential users because each of them can cause an emergency requiring the intervention of one or more units. The geographical description of the region under study and the resources location representation, valid for both fixed and mobile locations, constitute the main problem. In practice, the whole region can be subdivided into geographical atoms. The atoms (the system users) can be as small as necessary to avoid unreasonable quantization error, and they can assume any geometrical shape. Associated with each atom is the fraction of alarm-calls generated in the region under consideration. The geographical distribution of the units, on the other hand, can be described by specifying a location matrix $L = (l_{ij})$, where l_{ij} is the probability a generic unit i is located in an atom j, when available. The L matrix is required to be a stochastic one, so that $\sum_j l_{ij} = 1$ for all the indices i.

In the case of ambulance or fire service, the units are concentrated in a number of fixed stations, thus $l_{ij} = 1$ for an atom j and $l_{ik} = 0$ for every $k \neq j$.

In the following, for the sake of convenience, we shall keep on using the word "station" for mobile unit systems too, referring in such a case to the single unit.

Therefore the problem is the assignment of the calls to the stations, taking care to give primary intervention responsibility for a certain atom to only one station (that implies that if the generic term of the L matrix $l_{ik} \neq 0$, $l_{jk} = 0$ for every $j \neq i$) and sharing the workload, on an average, equally among the units.

For this purpose we define the following variables:

t_{ij} = average travel time from the station i to the atom j;

x_{ij} = binary variable that represents the decision of assigning ($x_{ij} = 1$) or not ($x_{ij} = 0$) the atom j to the station i;

λ = average rate of calls for service (calls per hour) generated from within the entire region of interest;

$1/\mu$ = average service time per call;

n_i = number of the response unit belonging to the station i ($n_i = 1$ in the case of mobile unit system)

f_j = fraction of calls generated from atom $j(\sum_{j\in J} f_j = 1)$;

τ = average percentage of time the unit is busy a day.

So the problem can be formalized as an integer linear programming problem, as follows:

Define the two sets $I = \{1, 2, \ldots, N\}$ and $J = \{1, 2, \ldots, M\}$ respectively associated with the stations located in the region considered and with the atoms into which it is partitioned.

Find the vector

$$\mathbf{x} = \{x_{11}, \ldots, x_{N1}, x_{12}, \ldots, x_{N2}, \ldots, x_{1M}, \ldots, x_{NM}\}$$

minimizing

$$z = \sum_{i\in I} \sum_{j\in J} t_{ij} x_{ij} \tag{1}$$

subject to the constraints:

$$\sum_{j\in J} f_j x_{ij} \leq b_i = \tau \frac{\mu}{\lambda} n_i, \quad \forall\, i \in I \tag{2}$$

$$\sum_{i\in I} x_{ij} = 1, \qquad \forall\, j \in J \tag{3}$$

$$x_{ij} = 0, 1, \quad \forall\, i \in I,\ \forall\, j \in J. \tag{4}$$

In equation (2), without loss of generality, we may assume $f_1 \geq f_2 \geq \cdots f_M$.

It is immediately clear that, with this formulation, travel time statistics play an important role.

All mean travel times are computed from a travel time matrix, whose generic element is τ_{kj}, that is, the mean travel time from atom k to atom j (in general $\tau_{kj} \neq \tau_{jk}$).

The knowledge of (τ_{kj}) makes it possible to express the objective function z coefficients in the following way:

$$t_{ij} = \sum_{k=1}^{M} l_{ik} \tau_{kj} \quad \forall\, i \in I,\ \forall\, j \in J. \tag{5}$$

A vector $\mathbf{x}$ satisfying the constraints (3) and (4) is called "solution", a vector $\mathbf{x}$, satisfying the constraints (2), (3) and (4) is called "feasible solution" and eventually a feasible solution that minimizes (1) is called "optimal solution".

3. THE ALGORITHM

For solving integer linear programming problems, several algorithms are available. Some of them are based on traditional methods (for instance, Gomory's algorithm and enumerative methods), others are based on a combinatory approach, to which, Balas especially has made a great contribution [6]. From a comparison of these methods, it seemed preferable to use an algorithm developed by De Maio and Roveda [7], some years ago, to solve a specific class of industrial problems.

This algorithm, in fact, follows in some aspects the filter method of Balas, but the second stage, which involves lengthy computations, is avoided by making use of equations (3) as a filter.

This constraint causes some subvectors to be identified in the vector $\mathbf{x}$; for each subvector only one component may assume the value of 1, while the others are constrained to assume the value of 0.

Hence the solution number is not $2^{N\cdot M}$, but N^M which may be a much smaller number. The essential idea of the algorithm is that of generating a sequence of solutions $\{x^k\}$ such as to satisfy these conditions:

(a) the sequence of the values z_k associated with x^k through the equation (1) is not decreasing;

(b) when a solution x^s corresponding to z_s is reached, all the solutions x^t such as $z_t < z_s$ have been implicitly or explicitly considered.

So it is evident that the first feasible solution obtained is also an optimal one. The procedure for generating solutions starts from the initial solution x^0, associated with a minimum absolute cost z_0.

This solution is determined in the following way: for every $j \in J$ determine the index $i_j \in I$ such that $t_{i_j,j} = \min_{(I)}\{t_{ij}\}$ and when there is more than one index i_j, satisfying the previous relation, choose $\bar{i}_j = \min\{i_j\}$. Then set $x_{i_j,j} = 1$ and $x_{ij} = 0$ for every $i \in I$ $(i \neq i_j)$.

With the sequence of the subsequent solutions it is possible to construct a tree T, made up of nodes, representing the solutions x^k associated with their cost z_k generated step by step, and of branches connecting solutions generated in two subsequent iterations. The initial node is obviously x^0. At a generic step s the partial tree T^s consists of all the solutions generated up to the relative iteration.

We may call active nodes, all the T^s terminal nodes that have not been eliminated by feasibility test No. 2 (see below). At the generic step s two sets are defined:

$$X^s = \{x^k | x^k \text{ is an active node of } T^s\}$$
$$Z^s = \{Z_K | x^k \text{ is an active node of } T^s\}.$$

The algorithm is applied to these two sets. The basic structure of the algorithm is the following:

Starting phase. Compute, as before mentioned, the minimal cost solution x^0 and form the sets

$$X^0 = \{x^0\}$$
$$Z^0 = \{z_0\}.$$

Iterative phase. The iterative phase is developed in the following steps:

Choice step. At the $s+1$ iteration, if $Z^s \neq \phi$ choose:

$$z_k = \min_{z_i \in Z^s} \{z_i\}$$

(if some tie exists, choose one of the tied z's, in an arbitrary way, for instance the one having the smallest index). Determine the corresponding solution x^k and cancel z_k from Z^s and x^k from X^s. If $Z^s = \phi$, no feasible solution exists and the algorithm terminates.

Feasibility step. Test No. 1. Compute

$$\hat{b}_i = -b_i + \sum_{j \in J} f_j x_{ij}^k, \quad \forall\, i \in I.$$

If $\hat{b}_i \leq 0$ $\forall$ $i \in I$, the test is passed, x^k is optimal and the algorithm ends.

If $\hat{b}_i > 0$ for some $i \in I$, form the sets

$$M^k = \{i | \hat{b}_i > 0\}$$
$$N^i = \{j | x_{ij}^k = 1\}, \quad \forall\, i \in M^k$$

and apply the next test.

Test No. 2. For every $i \in M^k$ and every $j \in N^i$, check if there is an index $l \neq i$ such that $t_{lj} \geq t_{ij}$ and form the sets: $J^i = \{j | j \in N^i,\ i \in M^k\} \exists\, l \neq i$ such that $t_{lj} > t_{ij}$ or $\exists\, l > i$ such that $t_{lj} = t_{ij}$.

Then compute

$$\Delta_i = \sum_{j \in J^i} f_j \quad \forall\, i \in M^k.$$

If $\Delta_i < \hat{b}_i$ even for only one $i \in M^k$, the test is not passed: cancel the node k and go back to the choice step.

If $\Delta_i \geq \hat{b}_i\ \forall\ i \in M^k$, compute

$$\delta^l_{ij} = t_{lj} - t_{ij} \quad \forall\, i \in M^k, j \in J^i, l \in I$$

and form the sets

$$L_{ij} = \{l | \delta^l_{ij} > 0 \text{ and } l \neq i,\ \delta^l_{ij} = 0 \text{ and } l > i\}$$

$$I^i = \{t | \delta^t_{ij} = \min_{l \in L_{ij}} \delta^l_{ij}\}.$$

Then go to the next step.

Branching step. For every $\hat{i} \in M^k$, generate all the solutions $x^{\hat{i}_t}$ that satisfy the following conditions:

(a) $x^{\hat{i}_t}_{ij} = x^k_{ij}, \quad \forall\, j \in (J - J^{\hat{i}})$ and $\forall\, i \in I$

(b) $x^{\hat{i}_t}_{lj} = 1, \quad \forall\, j \in J^{\hat{i}}$

either for $l = \hat{i}$ (no shifting of an 1 happens in the relative column) or for l equal to the element of $I^{\hat{i}}$ corresponding to the element j here considered.

(c) $\sum_{j \in J} f_j x^{\hat{i}_t}_{ij} \leq b_{\hat{i}}$

(d) Do not consider redundant solutions, that is, given two solutions $x^{\hat{i}_t}$ and $x^{\hat{i}_v}$ and defined the two sets

$$E^{\hat{i}}_t = \{i | j \in J^{\hat{i}},\ x^{\hat{i}_t}_{ij} = 0\}$$
$$E^{\hat{i}}_v = \{i | j \in J^{\hat{i}},\ x^{\hat{i}_v}_{ij} = 0\}$$

both $E^{\hat{i}}_t \not\subset E^{\hat{i}}_v$ and $E^{\hat{i}}_v \not\subset E^{\hat{i}}_t$ must hold.

The algorithm for generating nonredundant solutions is given in the Appendix.

By making use of the solutions $x^{\hat{i}_t}$, previously generated, form all the solutions x^{k_a} such that:

$$x^{k_a}_{ij} = \begin{cases} x^{h_t}_{ij},\ \forall\, i \in I,\ \forall\, j \in J^h,\ \forall\, h \in M^k \\ x^k_{ij},\ \forall\, i \in I,\ \forall\, j \in (J - \bigcup_{h \in M^k} J^h). \end{cases}$$

Then update the sets Z^s and X^s with the solutions obtained and go back to the choice step.

Of course, all the solutions generated in this way, that are identical with an existing one, are deleted. The flow chart of the algorithm is shown in Fig. 1.

4. APPLICATION TO THE ROME FIRE DEPARTMENT

A validity proof of this kind of districting problem formulation has been made by applying the algorithm to the Rome department.

The application required four subsequent steps:

(1) choice of the region, data collection and analysis;
(2) data processing in a form suitable to the particular algorithm;
(3) computational program;
(4) analysis of the results.

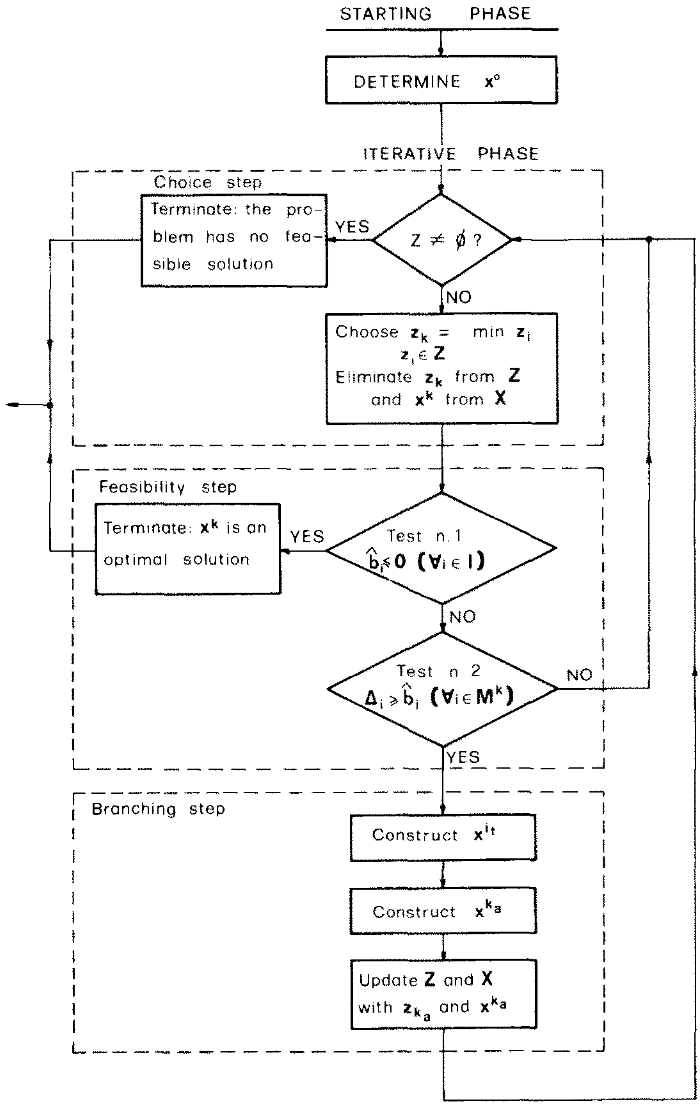

Fig. 1. Flow chart of the algorithm.

(1) *Choice of the region, data collection and analysis*

We have chosen the urban area of Rome, included inside the "Raccordo Anulare" (Ring Road) and those suburbs outside the "Raccordo Anulare", whose service requirements weigh on the city fire stations.

This region is not homogeneous as it includes strongly urbanized areas with high population density, industrialized and agricultural and woody areas.

The stations in charge of the entire area are normally seven (eight, considering the training school located at "Capannelle") approximately placed on concentric circles with the main station in the middle (see Fig. 2).

The denomination of the stations and the unit distribution among the stations are shown in Table 1. It is easy to note that the main station, located in the town center, has a number of units clearly exceeding that of all other stations. Such a situation certainly affects the optimal districting. The necessary data for the algorithm implementation have been taken from filing cards filled in for every accident. There are recorded the emergency location, the type of unit employed and the time spent between the call and the units' return home after performing their duty.

The analysis has been confined to a peak period of time (August 1974) and some system parameters such as average unit workload, their service mean rate, alarms mean rate (see Table 2) and spatial distribution of demands for service, have been computed through appropriate statistics.

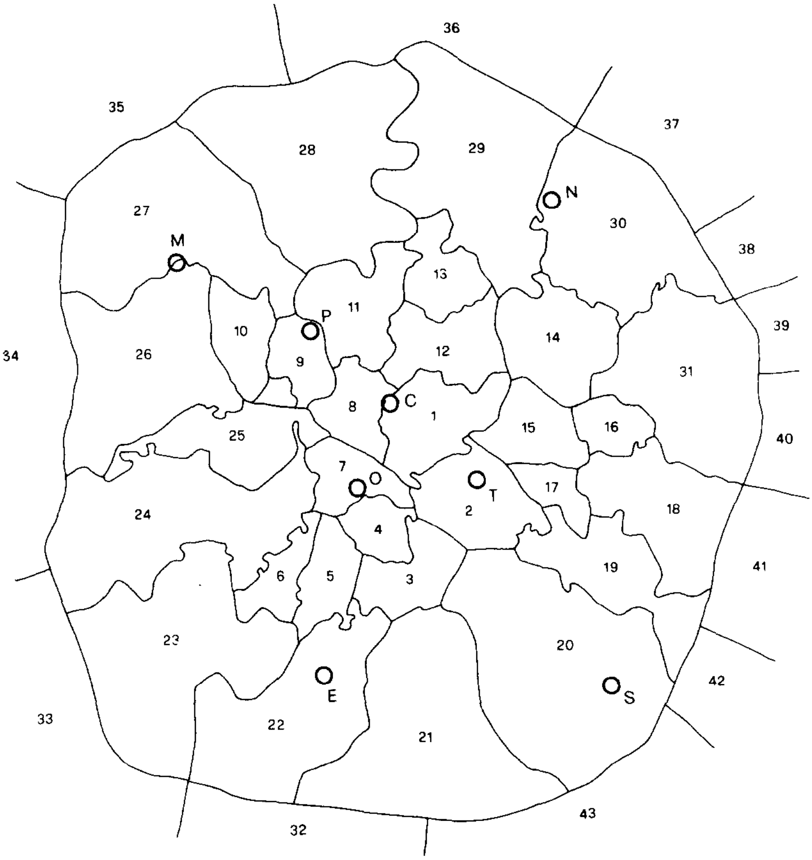

Fig. 2. Atoms subdivision and station distribution.

(2) *Data processing*

Primarily, the region under study has been partitioned into atoms (each individualized by a known number of calls), with the aim of fitting the actual town POSTAL CODE (ZIP CODE in U.S.A.) subdivisions to the proposed method requirements. This partitioning and the number of alarm calls generated from within each atom, in the period considered, are shown in Fig. 2 and Table 2.

Thus, the f_i coefficients in equation (2) are straightforwardly obtained.

The travel time matrix t_{ij} (see Table 3) has been found, by making use of a work performed by the Rome municipal department of urban transportation.

In this work an extended succession of nodes and preferential routes connecting them in the town area, is illustrated. The routes average speeds are also provided.

(3) *Computer program*

The algorithm has been implemented on an UNIVAC 1110 in FORTRAN V.

The computation time does not cause a problem, because the algorithm performs the optimal solution after only 56 iterations and it takes 26" of CPU to complete them.

Table 1. Unit distribution

Stations	Code	Number of units	Typical parameters
Centrale	C	6	
Ostiense	O	2	
Eur	E	1	$\tau = 0.6$
Prati	P	2	$\mu = 0.5$ services/h
Monte Mario	M	1	$\lambda = 4.1$ calls/h
Nomentano	N	2	
Tuscolano	T	1	
School Capannelle	S	2	

Table 2. Number of alarm calls (1–20 August 1974)

Atoms	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
Number of calls	17	8	11	5	8	9	10	10	8	16	11	10	20	14	11	4	5	8	14	32	12	18	14	17	18	46	28	23	20	11	11	6	4	8	22	22	3	13	7	7	15	9	8

Table 3. Travel times (minutes)

Atoms / Stations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
1	1	4	7	5	6	7	3	1	4	6	4	2	5	1	4	6	5	7	7	12	12	9	10	7	5	10	8	7	7	8	9	22	29	25	27	20	25	25	17	17	23	23	22
2	1	6	4	1	5	5	1	3	5	7	7	6	8	9	6	8	7	9	10	13	9	5	8	5	6	10	9	10	11	12	11	20	27	26	30	23	29	27	20	20	30	26	22
3	9	13	3	5	3	6	6	8	14	13	13	11	14	13	10	12	10	13	16	9	2	1	9	7	5	14	14	16	17	18	15	13	23	31	43	29	34	31	23	31	34	35	22
4	5	7	9	7	8	8	5	3	2	4	2	5	7	9	8	10	8	11	10	14	14	13	11	6	4	7	5	4	9	11	13	25	30	29	24	17	28	29	21	20	27	26	23
5	11	13	15	12	13	11	10	8	6	3	7	11	9	15	16	17	14	20	16	19	18	16	12	9	7	2	2	5	12	15	21	38	35	28	20	18	35	26	29	35	45	32	35
6	10	10	14	13	15	15	11	9	14	15	10	6	5	8	10	10	10	13	14	19	18	18	18	21	15	17	12	9	3	6	11	30	46	39	29	15	19	21	20	19	30	31	30
7	2	1	5	4	9	9	5	4	7	9	7	5	8	7	3	5	3	5	4	4	9	8	12	9	8	12	12	10	9	11	8	21	29	27	31	23	27	25	16	35	23	19	16
8	8	6	11	13	15	15	15	11	13	15	13	13	16	11	10	9	7	6	4	5	9	14	18	17	18	20	18	15	17	17	12	23	33	40	50	36	23	29	20	19	22	18	10

We met more difficulties, on the other hand, dealing with the storage problem, because the algorithm solution is based on branch and bound techniques.

So that it generates a number of solutions growing rapidly with the dimensions of the problem (that is, M and N). Although the algorithm quickly converges and the number of the solutions generated before the optimal is quite small, we had to turn each vector of 43 components into a vector of 4 components through a suitable conversion.

In this way the memory area dimensions to be reserved, have been drastically cut down (see Table 4).

Table 4. Computational results

N	M	S	S.A.	tc	ME	MEO
8	43	5785	56	26″	344,000	32,000

N: number of the matrix rows; M: number of the matrix columns; S: number of the generated solutions; SA: number of analyzed solutions before finding the optimal; tc: time of CPU; ME: memory area for storing the solutions without the conversion; MEO: memory area reserved for storing the solutions with the conversion.

(4) *Analysis of the results*

The data, performed in the above mentioned way, were introduced as inputs in the computer program. The results of the computer run showed quite interesting results, as one can see by simply making a comparison between Fig. 3 and Fig. 4. They respectively show the current primary response area subdivision, empirically obtained, and the one derived from the proposed optimization method.

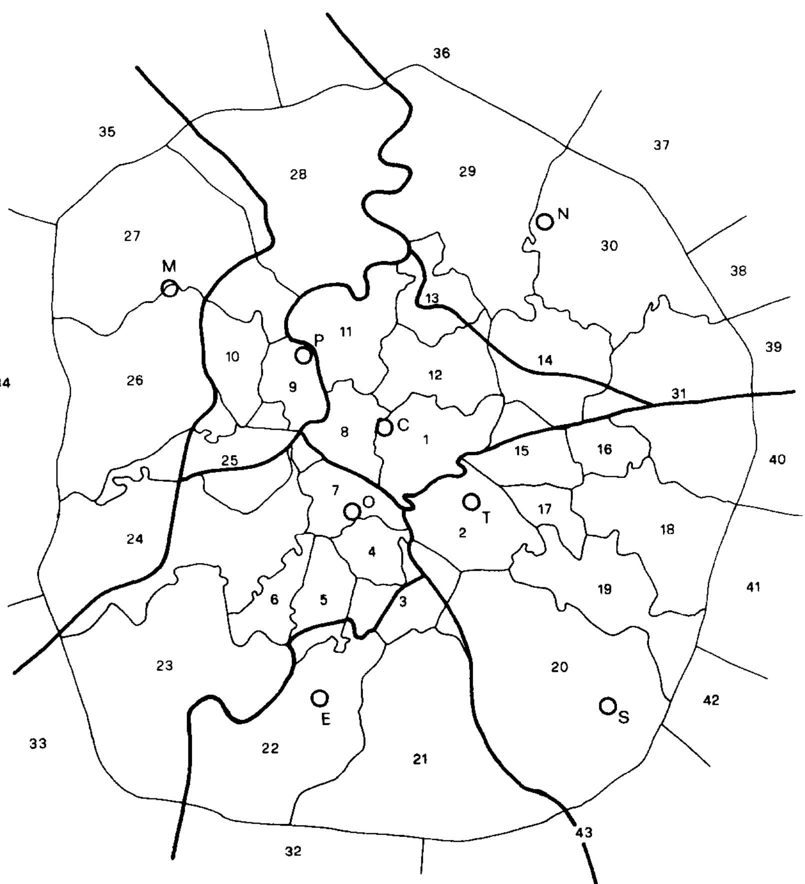

Fig. 3. Current response areas.

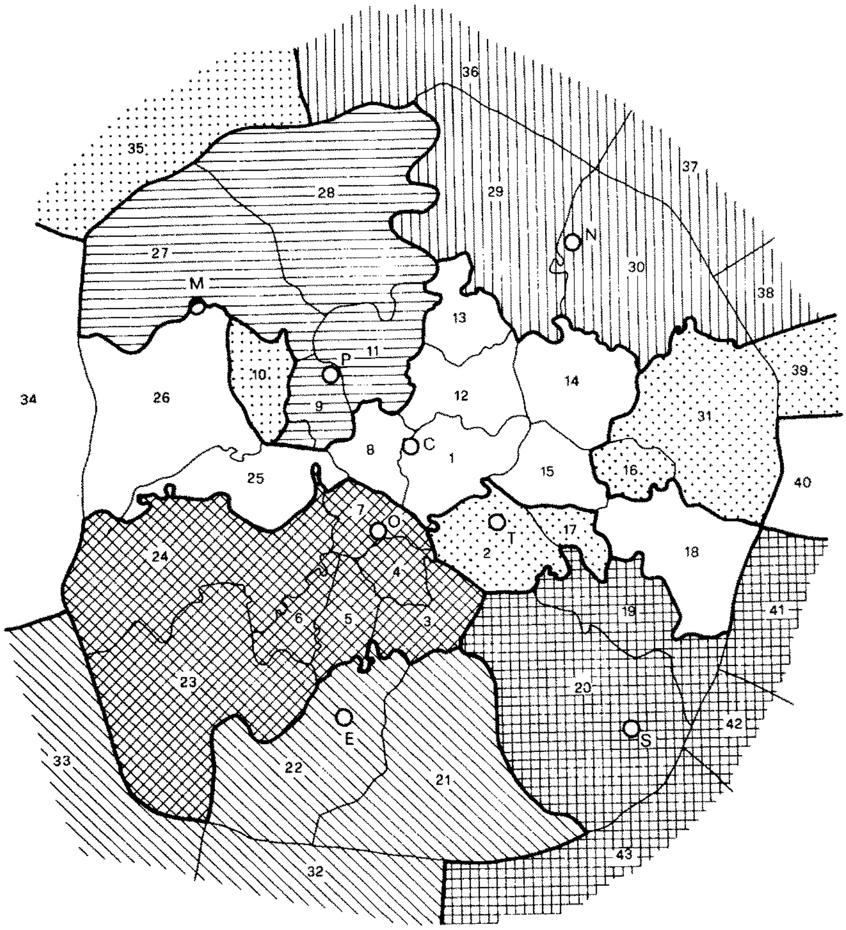

Fig. 4. Optimized response areas.

Table 5. Optimal assignments

Stations	Code	Atoms of the relevant response areas
Centrale	C	1, 8, 12, 13, 14, 15, 18, 25, 26, 40
Ostiense	O	3, 4, 5, 6, 7, 23, 24
Eur	E	21, 22, 32, 33
Prati	P	9, 11, 27, 28
Monte Mario	M	10, 35
Nomentano	N	29, 30, 36, 37, 38
Tuscolano	T	2, 16, 17, 31, 39
School Capannelle	S	19, 20, 41, 42, 43

The associated values of z are 465 min in the former case and 343 min in the latter case. Therefore the system travel time, referring to the actual situation, is reduced by 26%, that is, by about 2 h.

As was foreseeable the most significant changes concern above all the main station, but Monte Mario, Nomentano and Tuscolano stations also present considerable variations.

It is obvious that if it were actually necessary to redistrict the primary response areas, phases (1) and (2) should be refined, by collecting a more extensive data series and keeping in mind the cooperation and coordination requirements with the provincial stations.

However, in our opinion, the results of this work can also provide useful insight for a more elaborate approach directed towards obtaining an optimal districting for the Rome fire service.

Besides, they give a good test of the efficiency of the proposed method.

5. CONCLUSION

This approach to the districting problem makes it possible to find the optimal solution under the hypothesis that all units are available. In this way we solve a static problem. Nevertheless, in theory, the proposed method can also be utilized to make a dynamic assignment of the response areas, by considering the current state of the system.

To do that, we need to apply the algorithm to all the possible states of the system, and to determine the optimal response areas for each state.

The solutions obtained could be stored in a computer and then dealt with in such a way that, given a call from a certain atom and given the system state at that moment, it would be possible to determine at once the optimal station responsible for the atom in question.

REFERENCES

1. R. L. Keeney, A method for districting among facilities, *Ops Res.* **20**, 613 (1972).
2. R. Larson, *Response of Emergency Units: The Effects of Barriers, Discrete Streets, and Oneway Streets.* The New-York city Rand Institute, April (1971).
3. J. M. Chaiken and R. Larson, *Methods for Allocating Urban Emergency Units.* The New-York city Rand Institute, May (1971).
4. G. M. Carter, J. M. Chaiken and E. Ignall, *Response Areas for Two Emergency Units.* The New-York city Rand Institute, March (1971).
5. R. Larson, A hypercube queuing model for facility location and redistricting in urban emergency services. *Comput. Ops Res.* **1**, 67 (1974).
6. E. Balas, An additive algorithm for solving linear programs with zero-one variables. *Ops Res.* **13**, 517 (1965).
7. A. de Maio and C. Roveda, An all zero-one algorithm for a certain class of transportation problems. *Ops Res.* **19**, 1406 (1971).

(*Received* 10 *October* 1976)

APPENDIX

The problem considered here is to determine all the binary nonredundant solutions of inequalities

$$\text{(a)}\quad \sum_{j=1}^{n} d_j y_j \geq D$$

$$\text{(b)}\quad \sum_{j=1}^{n} d_j y_j \leq D$$

where $D > 0$; $d_1 \geq d_2 \geq \cdots \geq d_n$; $y_j = 0, 1$.

By nonredundant solutions, we mean those solutions $y = (y_1, y_2, \ldots, y_n)$ for which, having defined a set of indices $E = \{j | y_j = 1\}$, given two solutions y^h and y^k, the following conditions are satisfied

$$E^h \not\subset E^k \quad \text{and} \quad E^k \not\subset E^h.$$

To find the solutions we have to check if some $d_h \geq D$ exists; for such a d_h we find at once that the solution is defined by $E^h = \{h\}$ in case (a), while in case (b) the associated variable must be at all times zero.

Therefore, it is possible to cross out the variable y_h in the inequalities and solve them with the remaining variables. Therefore we suppose $d_1 < D$. Besides we set

$$\sum_{j=1}^{n} d_j > D.$$

If it is not so, we will have:

$$(1)\quad \sum_{j=1}^{n} d_j = D,$$

only one solution with $y_j = 1$ for $j = 1, 2, \ldots, n$;

$$(2)\quad \sum_{j=1}^{n} d_j < D,$$

no solution in case (a) and only one solution in case (b) with $y_j = 1$ for $j = 1, 2, \ldots, n$.

The algorithm, here proposed for the solution is a search method based essentially on the concept of "partial solution" S^h. We mean, by a partial solution, a binary assignment of the first h variables (with $h < n$). By a "solution" on the other hand, we mean a binary assignment of all variables y_j.

The method consists of generating a feasible solution of the problem, and then deriving some partial solutions; these are successively tested by a feasibility test to verify if a complete solution exists obtainable from them, which is feasible.

The partial solutions that pass the test are called active, those that do not pass the test are eliminated together with all the other solutions derivable from them. Now let us show the algorithm structure in the two cases (a) and (b).

Initialization. The algorithm starts on the active partial solutions S^1, set $T^1 = \{S^1 | S^1 \text{ is an active partial solution}\}$.

Iterative phase. Each iteration i includes the following steps:

(1) Form the set

$$T^i = \{S^h | S^h \text{ are active partial solutions}\}$$

when $T^i = \{\phi\}$, the process ends.

(2) Choose at random any one partial solution in T^i and let it be S^k, eliminate it from T^i and then form a complete solution in the following way:

case (a)

$$y^i = \begin{cases} y_j^i \text{ as in } S^k & \text{for } j = 1, 2, \ldots, k \\ y_j^i = 1 & \text{for } j = k+1, \ldots, h \\ y_j^i = 0 & \text{for } j = h+1, \ldots, n \end{cases}$$

where

$$h = \min\left\{t \Big| \sum_{j=1}^{k} d_j y_j^k + \sum_{j=k+1}^{t} d_j \geq D\right\}, \quad k+1 \leq t \leq n;$$

case (b)

$$y^i = \begin{cases} y_j^i \text{ as in } S^k & \text{for } j = 1, 2, \ldots, k \\ y_j^i = 0 & \text{for } j = k+1, \ldots, h \\ y_j^i = 1 & \text{for } j = h+1, \ldots, n \end{cases}$$

where

$$h = \min\left\{t \Big| \sum_{j=1}^{k} d_j y_j^k + \sum_{j=t+1}^{n} d_j \leq D\right\}, \quad k+1 \leq t \leq n.$$

(3) If $h = n$, no partial solution is generated from y^i and we pass to step (1) beginning again a new iteration starting on the residual T^i.

(4) *If* $h < n$ the following partial solutions are made up:

case (a)

$$S_i^h = \begin{cases} y_j^h = y_j^i & 1 \leq j \leq h, j \neq r \\ y_j^h = 0 & j = r \end{cases}$$

where $r = k+1, \ldots, h$;

case (b)

$$S_i^h = \begin{cases} y_j^h = y_j^i & 1 \leq j \leq h, j \neq r \\ y_j^h = 1 & j = r \end{cases}$$

where $r = k+1, \ldots, h$.

(5) Subject every S_i^h, constructed in this way, to the feasibility test:

case (a)

$$\sum_{j=1}^{h} d_j y_j^h + \sum_{j=h+1}^{n} d_j \geq D$$

case (b)

$$\sum_{j=1}^{h} d_j y_j^h \leq D.$$

If the test is checked, S_i^h is an active partial solution. If the test is not checked, S_i^h is eliminated and with it also the derivable solutions.

(6) Go back to step (1) updating set T^i with the active partial solutions determined at step (5).