Socio-Econ. Plan Sci., Vol. 13, pp 321–326

0038-0121/79/1201-0321/$02 00/0

TECHNIQUES FOR DEFINING GEOGRAPHIC BOUNDARIES FOR HEALTH REGIONS†

J. WILLIAM THOMAS

Department of Medical Care Organization, School of Public Health, University of Michigan, 109 Observatory Street, Ann Arbor, MI 48109, U.S.A.

(*Received* 5 *January* 1979)

Abstract—Many federal and state programs require the geographic partitioning of states into regions for health services planning, monitoring, and/or administration. A common consideration for such programs is that region boundaries should be drawn so as to maximize the proportion of the state's population that receives health care services in its region of residence. Defining region boundaries thus may be viewed as a problem of partitioning a set of N small areal units (e.g. counties) into M subsets (regions) so as to minimize interactions (patient flow) among subsets. This paper describes three algorithms for region design and compares them in terms of computer-processing efficiency and solution value based on results from a number of test cases. Application of two of the algorithms, one based on the greedy heuristic and the other incorporating a max-flow/min-cut procedure, to a problem of dividing a metropolitan region into separate service areas for clusters of hospitals is also described.

BACKGROUND

The concept of organizing health services on a regional, or multi-community basis in the United States was first proposed in 1932[1]. However, regionalization was not promoted as official national policy until the late 1960s when areawide comprehensive health planning agencies were mandated under Section 314(b) of the Public Health Service Act. Since then, regionalization has become a cornerstone of Federal health policy and is reflected in legislation for, among other things, utilization and quality review (P.L. 92-603, 1972), health manpower (P.L. 92-585, 1972), emergency medical services systems (P.L. 93-641, 1974). By coordinating resources on a regional basis, it is expected that the availability, quality, and efficiency of health services will be enhanced[1].

Each of the programs cited above, as well as others recently initiated by Federal and state governments, requires that states be partitioned geographically into regions for health services planning, monitoring, and/or administration.‡ Partitioning typically is performed along county lines, so that each region is composed of several counties. While some region design criteria such as limits on individual region size may vary from program to program, a common consideration for health services regions is that boundaries should reflect existing patient flow patterns; i.e. boundaries should be drawn so as to maximize the proportion of the state's population that receives health services in its region of residence.

Several measures of patient flow have been suggested. However, because of data availability, the only commonly used measure is transit patterns of hospital inpatients. By sampling patient records at each hospital and noting county of residency for patients in the sample, a matrix can be developed showing the number of patients residing in county i who obtain medical care in county j. In a number of states, hospitals routinely report to the state health department detailed information on all inpatient admissions, and the health department is able to publish revised patient flow statistics annually.

While Federal regional health programs normally require that an entire state be divided into multi-county regions, it is sometimes necessary to partition a city, a county, or a multi-county region into smaller geographic subunits. For example, large cities often are divided into health districts for decentralized administration of municipal health department programs. Also, individual counties or regions containing several counties may be divided into discrete service areas for individual hospitals or clusters of hospitals, with area boundaries drawn so as to reflect observed hospital use patterns of the population[2]. Where districts are formed by grouping zip codes, census tracts, minor civil divisions or other small areal units, and district boundaries are selected to minimize out-of-district patient flow, district design is equivalent to the health region design problem at the state level.

TECHNIQUES FOR DEFINING HEALTH REGION BOUNDARIES

Defining boundaries for health regions may be viewed as a problem of partitioning a set of N areal units (e.g. counties) into M subsets (regions) so as to minimize interactions (patient flow) among subsets. As noted above, additional criteria, such as location of at least one major hospital in each region or a minimum population level for each region, may also be applied. The problem is expressable as a 0/1 integer program, but solutions even for small states with few counties and regions involve a prohibitively large number of constraints and integer variables[3]. Emphasis thus has been on heuristic techniques for locating approximately optimal solutions.

Techniques have been reported for such related applications as school districts[4–9], election districts[10–12], police patrol sectors[13], and fire inspection districts[14]. However, since these problems do not involve criteria for interaction among geographic subunits, the proposed methods are not generally applicable for selecting health region boundaries. Three algorithms which can accommodate interaction criteria and are

†Portions of the research reported in this paper were included in the author's doctoral dissertation for the Department of Decision Sciences, The Wharton School, University of Pennsylvania.

‡Small or sparsely populated states sometimes may be treated as a single region.

therefore relevant for health region were investigated in the study reported here. These include an implicit enumeration algorithm, an algorithm based on the greedy heuristic, and one based on Ford and Fulkerson's max-flow/min-cut theorem.

Implicit enumeration algorithm. Implicit enumeration is a combinatorial technique wherein every possible assignment (in this case, of counties to regions) is explicitly or implicitly evaluated. If it can be determined, for certain partially completed assignments, that no pattern of assigning the remaining unassigned counties will yield a completed solution that is feasible in terms of included constraints or one that represents an improvement over previously identified solutions, then further enumeration along that branch of solutions may be abandoned.

Based on an approach first proposed by Graves and Winston[15], the mean and standard deviation of solution values associated with all completions of each partial assignment are employed to determine, within a specified probability range, whether the partial assignment will lead to any completed solution that is better than the current best solution. This optimality test is applied each time one of the N counties is considered for assignment to one of the M regions. Each of these potential assignments is also evaluated for feasibility with respect to (a) contiguity of counties assigned to a region and (b) a requirement for at least one hospital in each region.

Because of the optimality test, each newly completed solution will be superior, in terms of the proportion of patients receiving health care services in their regions of residence, than previously identified solutions. However, the procedure cannot assure that a globally optimal solution will be located, since the branch on which that solution lies may be prematurely terminated with a small but finite probability through application of the optimality test.

Similar implicit enumeration approaches have been used by Graves and Winston for quadratic assignment problems[16], Duncan and Scott for a virtual storage computer paging problem[17], and Liggett for a school districting problem[9].

Greedy algorithm. Assume that an initial assignment of one county to each region has been specified. The greedy heuristic then may be used to assign each of the $(N\text{-}M)$ remaining counties. At each step of the algorithm, the unassigned county and the developing region that share the *greatest* amount of patient flow are identified and the county is assigned to that region. Thus regions "grow" as counties are assigned one by one. The procedure terminates when all counties have been assigned.

Variants of this heuristic were employed by Taliaferro and Remmers[18] and Transaction Systems, Inc.[19] for selecting health region boundaries, and it is the basis for a manual approach to defining health region boundaries described by Ciocco and Altman in 1954[20]. The greedy heuristic also has been employed in algorithms for the uncapacitated plant location problem[21], the capacitated plant location problem[22, 23], the knapsack problem[24], and the p-median problem[25].

Max-flow/min-cut. Instead of building up regions through sequential assignments of counties (as with the greedy heuristic) or by using a combinatorial approach (as in implicit enumeration), an alternative method is to define regions by "cutting" the state into ever-smaller pieces. As a first step, the state is divided into two regions. Another partition is then made to yield three regions. The next partition yields four regions; etc.

Let each county of the state to be partitioned represent one node of a network. The capacity of the arc connecting counties (nodes) i and j is defined to be the sum of the patient flows from i to j and from j to i. Ford and Fulkerson's max-flow/min-cut theorem, which states that the maximum possible flow from the source node to the sink node of any network equals the minimal cut capacity of all arc cuts separating the source and sink, then provides a basis for defining the partitions[26].

Assuming that N counties are to be partitioned into M regions, M of the counties are selected to serve as source and sink nodes. The first of these counties is designated as the source node and the second as the sink node for the first cut. The minimum cut partition is then determined using the max-flow/min-cut procedure. Next, the third county on the list is selected as the sink for the second cut. The source node for the second cut is the previous end node, either the source or sink of the first cut, that is located in the same partition as the new sink. The max-flow/min-cut procedure is used to divide this partition into two parts, yielding a total of three partitions.

This process is repeated until M partitions have been defined. At each step, the next county on the source/sink list is selected to be the new sink. The corresponding source node is specified (an end node of a prior cut, located in the same partition as the new sink), and the max-flow/min-cut procedure is applied. When no counties are left on the source/sink list, the procedure terminates, and the cumulative flow across all cuts is the inter-region flow of the newly defined system of regions.

Let x be the source node and y be the sink. The procedure for locating each minimum-flow cut proceeds iteratively as follows:

(1) Initialize all arc flows g_{ij} for arcs (i, j) to 0.

(2) Beginning at the source x, put a plus (+) sign by each arc (x, j) for which the current arc flow g_{xj} is less than the arc capacity f_{xj}, and label node j with a check mark ($\surd$). Node x is now *scanned*, so mark it with a ($\not\surd$).

(3) Select any node j that is labelled with a ($\surd$). For every arc (j, k) for which (1) g_{jk} is less than f_{jk}, and (2) node k is unlabelled, put a (+) mark on arc (j, k) and label node k with a ($\surd$). After all arcs from j have been checked, mark node j as scanned ($\not\surd$).

(4) Continue the operation of Step 3 until either (Case (a)) sink node y is labelled, or (Case (b)) all labelled nodes have been scanned. *Case (a)*: Breakthrough has occurred, since a flow-augmenting path from x to y has been discovered. This path is identified by tracing back from y to x over those arcs marked with a (+). Calculate the flow augmentation, which is the minimum value of $(f_{ij} - g_{ij})$ over all arcs (i, j) on the flow-augmenting path. Erase all labels ($\surd$, $\not\surd$) and (+) signs, and go back to Step 2. *Case (b)*: If node y is still unlabelled at the completion of Step 3, then the current solution represents the maximal flow through the network. The minimum cut separates all unlabelled nodes, including y, from all nodes labelled ($\not\surd$), including x.

SELECTION OF CORE COUNTIES

The greedy algorithm described above starts with an initial assignment of one county to each of the M regions. Similarly, the max-flow/min-cut algorithm requires an initial specification of M counties to serve as source/sink nodes. Conceptually, the selected counties in each case serve as region "cores" or centers.

If patient flow data are based on transit patterns of hospital inpatients, only those counties which contain

one or more hospitals need be considered for inclusion in $V=\{v^1, v^2, \ldots, v^{m-1}, v^m\}$, the set of region centers. Because region boundaries are defined to reflect existing patient flow patterns, the elements of V should be the M most active counties (containing hospitals), where activity refers to the amount of patient flow that one county shares with all other counties. Letting f_{ik} represent the sum of patient flows from county i to county k and from county k to county i, and H the set of all counties which contain hospitals, and selection rule is:

$$V^1 = \left\{ i \left| \max_{i \in H} \sum_{k \in H} f_{ik} \right. \right\}$$

$$V^2 = \left\{ i \left| \max_{i \in \{H-\{v^1\}\}} \sum_{\substack{k \in H \\ k=v^1}} f_{ik} \right. \right\}$$

$$V^l = \left\{ i \left| \max_{i \in \{H-\{v^1, v^2, \ldots, v^{l-1}\}\}} \sum_{k \in H \atop k \neq V^1, V^2, \ldots, V^{l-1}} f_{ik} \right. \right\}$$

that is, the lth element of V is that county which contains at least one hospital and which has the greatest amount of patient flow interaction with counties not already assigned to V.

TESTS AND APPLICATIONS

Each of the three algorithms were programmed in PL/1, and twelve test problems were run on an IBM 370/168 computer. Results are shown in Table 1. Cases 3 through 10 were based on actual patient flow data compiled in 1974 by the North Dakota State Health Department (case 3 considered only the 23 eastern-most counties), while other cases employed hypothetical data. In the cases considered, computer time requirements for both the greedy and max-flow/min-cut algorithms were observed to increase approximately linearly with problem size. However, the increase in computer time for the implicit enumeration algorithm was approximately of order (N^3), and cost considerations prohibited using the algorithm on the larger test problems. All three algorithms were able to define region boundaries yielding high solution values. However, the max-flow/min-cut solution was in every case equal or marginally superior to solutions from the other algorithms.

Figure 1 illustrates the max-flow/min-cut and greedy algorithm solutions for partitioning North Dakota into eight regions. The two algorithms yielded the same four-region solution for North Dakota, and this was identical to the one defined earlier by State officials who utilized the relatively time-consuming manual procedure of Ciocco and Altman[19].

As a part of a project concerned with demonstrating new methods of monitoring hospital performance, the greedy and max-flow/min-cut algorithms were employed to identify natural groupings or clusters of hospitals and the geographic service area associated with each cluster.† The initial application involved partitioning the seven-county metropolitan Detroit region, which includes 82 hospitals and is composed of 174 zip code areas, 84 zips within Detroit and its immediate suburbs and 90 in the outlying counties. Data from 1975 giving the number of patients from each zip code area that utilized each hospital were obtained by summarizing patient discharge information routinely reported by all of the hospitals.

Attempts to partition the entire 174 zip code region into cluster service areas were not successful. After five minutes of cpu time on the University of Michigan's Amdahl 470 V/6 computer, the max-flow/min-cut algorithm had managed to complete only three cuts.

Problem size for the max-flow/min-cut algorithm is determined not only by the number of areal units (zip codes) and the number of cuts to be made, but also by the number of node-to-node (in this case, residence zip code to hospital zip code) connections defined by the patient flow data. In densely populated cities like Detroit, geographic zip code areas are small, and patients often bypass nearby community hospitals, travelling on the freeway system across many zip code areas in order to utilize one of the major medical institutions located downtown. (Several large hospitals in downtown Detroit draw patients from all over the city and suburbs.) Con-

†For a discussion of the concepts underlying measurement of hospital performance, see Griffith[2]. The Hospital Performance Measures Project is financed by the W. K. Kellogg Foundation and sponsored by the Michigan Health Data Corporation in collaboration with the Bureau of Hospital Administration, School of Public Health, University of Michigan.

Table 1. Computation time and solution value comparisons based on twelve test problems

Case	No. of Counties	No. of Regions	Greedy CPU Sec.	Greedy Solution Value(%)†	Max-Flow/Min-Cut CPU Sec.	Max-Flow/Min-Cut Solution Value(%)†	Implicit Enumeration CPU Sec.	Implicit Enumeration Solution Value(%)†
1	10	2	1.4	97.7	1.3	97.7	3.2	97.7
2	10	4	1.4	94.2	1.3	94.2	15.4	94.2
3	23	2	1.6	92.5	1.4	96.9	48.1	96.3
4	53	2	3.1	97.1	2.9	98.1	*	
5	53	3	3.2	96.4	3.0	96.7	*	
6	53	4	3.4	95.4	3.0	95.4	*	
7	53	5	3.6	93.5	3.1	93.8	*	
8	53	6	3.8	92.6	3.1	92.8	*	
9	53	7	3.9	91.6	3.2	92.1	*	
10	53	8	4.0	91.1	3.2	91.4	*	
11	90	4	6.7	96.8	5.6	97.4	*	
12	90	8	8.2	95.0	6.0	95.7	*	

†Proportion of patients residing in the region in which they receive medical care

*Not run because of excessive computer time requirements

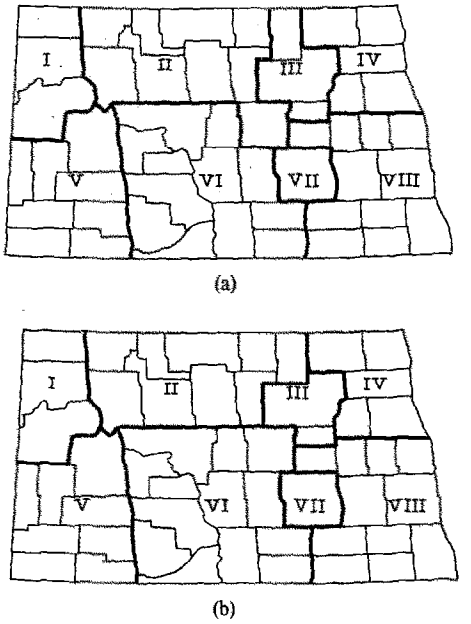

Fig. 1. Hypothetical eight-region partitions of North Dakota. (a) Max-flow/min-cut algorithm solution, (b) Greedy algorithm solution.

sequently, the number of patient flow connections among the city zip codes is quite large.

While the greedy algorithm was able to define 15 service areas in 21.5 cpu sec, the compactness and relative size of these areas were considered unsatisfactory. Again the problem was traceable to characteristics of the patient flow data, specifically the impact of the large downdown hospitals. The algorithm defines service areas that reflect patient flow patterns; and where a few hospitals located in close proximity to each other attract large numbers of patients from diverse parts of the city, some of the "natural" service areas identified by the algorithm turn out to be very large.

To make the analysis more tractable, the algorithms were applied independently to the 84 zip codes comprising Detroit and its close suburbs and to the 90 zip codes making up the outlying region. The max-flow/min-cut algorithm, in partitioning the Detroit zip codes into thirteen cluster areas, consumed 122 cpu sec and located a solution where 48% of patients utilized hospitals in the patients' areas of residence. The greedy algorithm took only 4.7 sec to find a solution with a value of 44%. Neither solution was considered acceptable, however, because each contained a number of "enclaves" (single zip code areas embedded in larger cluster areas) and each included one excessively large area containing approximately one-third of the zip codes and hospitals. Other runs were made, some with certain of the larger hospitals removed from the data. While six clusters on the periphery of the city remained fairly stable during these runs, problems with enclaves and unacceptably large cluster areas in the central city continued to occur.

Both algorithms performed significantly better when applied to the 90 outlying zip code areas. The min-cut algorithm consumed 7.7 cpu sec to define six partitions with an associated solution value of 94%; and the greedy algorithm in 3.7 sec located a slightly different solution that also had a value of 94%. These solutions are shown

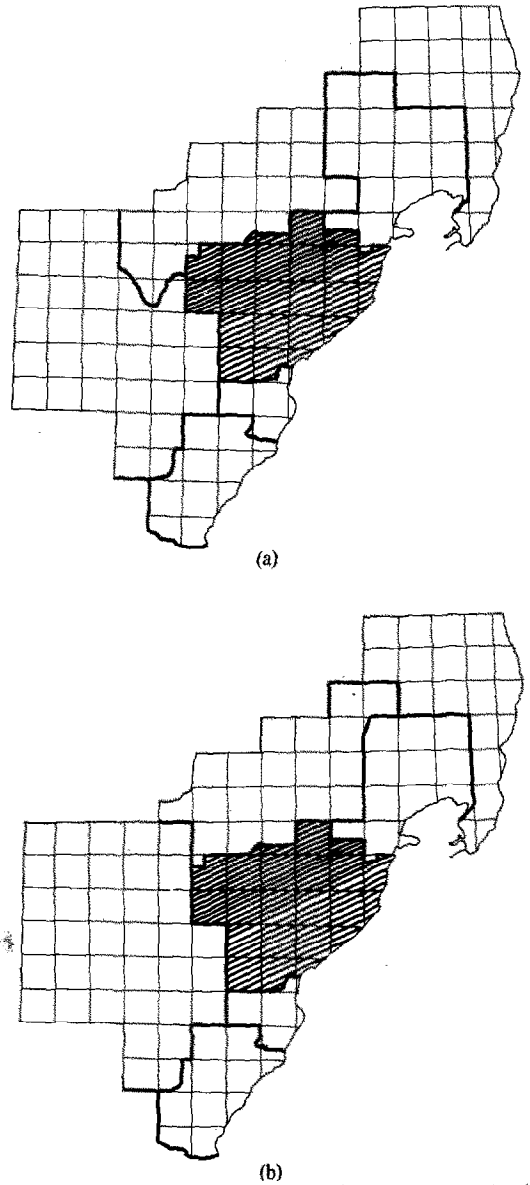

Fig. 2. Alternative six-cluster-area partitions of zip codes in Southeast Michigan (excluding Detroit and its close suburbs). (a) Max-flow/min-cut algorithm solution, (b) Greedy algorithm solution.

in the unshaded areas of Fig. 2. Table 2 lists cpu time requirements and solution values for alternative runs of the greedy and max-flow/min-cut algorithms.

CONCLUSIONS

From the applications discussed above, several conclusions can be drawn concerning the strengths and limitations of the three algorithms considered.

First, it is clear that the implicit enumeration algorithm is appropriate only for problems of relatively small size. Inclusion of constraints on region population size, number of hospitals per region, etc., will limit the number of feasible combinations that must be considered and thereby improve the algorithm's efficiency. But such constraints often are not a part of the problem specification for health regions.

Second, tests indicate that the greedy algorithm is

Table 2. Computation time and solution value comparisons based on six partitionings of Southeast Michigan zip codes

Case	Description	Greedy		Max-Flow/Min-Cut	
		CPU Sec.	Solution Value(%)†	CPU Sec.	Solution Value(%)†
1	SE Mich. Health Service Area, 174 zip codes, divided into 15 cluster areas	21.5	64	*	*
2	Detroit & close suburbs, 84 zip codes, divided into 13 cluster areas	4.7	44	122.0	48
3	Detroit & close suburbs (without 5 central hospitals), 83 zip codes, divided into 11 cluster areas	4.1	56	139.6	56
4	Same as Case 3, except with 10 cluster areas	4.0	59	131.2	59
5	Outlying region around Detroit, 90 zip codes, divided into 6 cluster areas	3.7	94	7.7	94
6	Same as Case 5, except with 7 cluster areas	4.2	92	10.5	92

†Proportion of patients residing in the cluster area in which they receive medical care

*Run terminated after 5 minutes, only 3 cuts completed

capable of efficiently locating solutions to problems of any practical size. The greedy algorithm is conceptually the simplest of the three and has, in various forms, been employed in other efforts to define health region boundaries.

Third, the computational efficiency of the max-flow/min-cut algorithm is influenced not only by the number of counties (zip codes, etc.) and regions to be considered, but by the density of the county-to-county patient flow matrix. For applications involving the partitioning of urban areas into smaller districts, this algorithm is likely to consume excessive amounts of computer time. However, it is very efficient in partitioning states into multi-county regions and in other applications with low density patient flow matrices.

Fourth, solutions located by the max-flow/min-cut algorithm are superior to those of the other algorithms in terms of the proportion of patients utilizing health services in their regions of residence. Differences in solution values may be small, however.

Fifth, size and shape of individual regions defined by the algorithms are determined by documented patient flow patterns. In situations where urban areas are to be divided into smaller districts, the size and compactness of some of the districts defined by the algorithms might be considered unsatisfactory. This likely will occur if area residents frequently travel outside of their local communities when seeking health care. Although constraints on district size and shape are easily incorporated into the implicit enumeration algorithm, the inability of this algorithm to deal with larger problems severely limits its utility in such cases.

A further consideration, one not previously discussed explicitly but relevant when comparing the relative efficiencies of the algorithms, is that decision makers often do not know in advance the number of regions (districts) that should be defined. A decision on this question is usually made after examining several alternative solutions, each containing a different number of regions. The max-flow/min-cut algorithm produces two regions with its first cut; three regions with its next cut, etc., until M regions are defined. Thus solutions with (M-1) regions, (M-2) regions, etc. are provided as a byproduct of the process of defining M regions. In Table 1, for example, all the solutions for different partitionings of North Dakota (two to eight regions) were available from the final eight-region run. (Separate runs were made for each different number of regions merely to provide cpu time estimates for comparison with the greedy algorithm.) The greedy and implicit enumeration algorithms, by contrast, require separate computer runs for each number of regions considered.

In summary, the max-flow/min-cut and greedy algorithms appear to have greater general applicability than the implicit enumeration algorithm. The greedy algorithm utilizes less computer time for areas characterized by relatively dense patient flow matrices. But the max-flow/min-cut algorithm can usually identify a marginally superior solution, and it can provide solutions for varying numbers of regions at no extra cost.

REFERENCES

1. D. A. Pearson, The Concept of Regionalized Personal Health Services in the United States, 1920–55. In *The Regionalization of Personal Health Services* (Edited by Ernest W. Saward), pp. 3–51. PRODIST, New York (1976).
2. J. R. Griffith, *Measuring Hospital Performance.* INQUIRY, Blue Cross Association, Chicago (1978).
3. J. W. Thomas, Defining substate regions for public service programs. Doctoral Dissertation, University of Pennsylvania (1977).
4. P. D. Belford and D. H. Ratliff, A network-flow model for racially balancing schools. *Ops. Res.* **20**, 619–628 (1972).
5. S. H. Clark and J. Surkis, An operations research approach to radial desegregation of school systems. *Socio-Econ. Plan. Sci.* **1**, 259–272 (1968).

6. A. D. Franklin and E. Koenigsberg, Computed school assignments in a large district. *Ops. Res.* **21**, 413–426 (1973).
7. L. B. Heckman and H. M. Taylor, School rezoning to achieve racial balance: a linear programming approach. *Socio-Econ. Plan. Sci.* **2**, 127–133 (1969).
8. C. A. Holloway *et al.*, An interactive procedure for the school boundary problem with declining enrollment. *Ops. Res.* **23**, 191–206 (1975).
9. R. S. Liggett, The application of an implicit enumeration algorithm to the school desegregation problem. *Management Sci.* **20**, 159–168 (1973).
10. R. S. Garfinkel and G. L. Nemhauser, Optimal political districting by implicit enumeration techniques. *Management Sci.* **16**, B-495–B-508 (1970).
11. S. W. Hess *et al.*, Nonpartisan political districting by computer. *Ops. Res.* **13**, 993–1006 (1965).
12. E. S. Savas, A computer-based system for forming efficient election districts. *Ops. Res.* **19**, 135–155 (1971).
13. R. C. Larson, Illustrative police sector redesign in District 4 in Boston. *Urban Analysis* **2**, 51–91 (1974).
14. D. M. Miller and D. E. Fyffe, Allocating building inspection manpower for fire prevention. *Management Sci.* **22**, 1310–1319 (1976).
15. G. W. Graves and A. Winston, A new approach to discrete mathematical programming. *Management Sci.* **15**, 177–190 (1968).
16. G. W. Graves and A. Winston, An algorithm for the quadratic assignment problem. *Management Sci.* **17**, 453–471 (1970).
17. J. Duncan and L. W. Scott, A branch and bound algorithm for pagination. *Ops. Res.* **23**, 240–259 (1975).
18. J. D. Taliaferro and W. W. Remmers, Identifying integrated regions for health care delivery. *Health Services Rep.* **88**, 337–343 (1973).
19. Transaction Systems, Inc., *Evaluation of Alternative Health Area Definition Methods*. Prepared for Bureau of Health Manpower, Nat. Tech. Inform. Service PB-260 672 (1976).
20. A. Ciocco and I. Altman, *Medical Service Areas and Distances Traveled for Physician Care in Western Pennsylvania*. Public Health Service Pub. 248, U.S. Government Printing Office, Washington, D.C. (1954).
21. G. Corneujols, M. L. Fisher and G. L. Nemhauser, An analysis of heuristics and relaxations for the uncapitated location problem. *Working Paper* 76-02-01, Dept. of Decision Sciences, The Wharton School, University of Pennsylvania (1976).
22. G. Sa, Branch and bound and approximate solutions to the capacitated plant location problem. *Ops. Res.* **17**, 1005–1016 (1969).
23. K. Spielberg, Algorithms for the simple plant location problem with some side conditions. *Ops. Res.* **17**, 85–111 (1969).
24. M. J. Magazine *et al.*, When the greedy solution solves a class of knapsack problem. *Ops. Res.* **23**, 207–217 (1975).
25. P. Jarvinen *et al.*, A branch-and-bound algorithm for seeking the p-medium. *Ops. Res.* **20**, 173–178 (1972).
26. L. R. Ford and D. R. Fulkerson, *Flows in Networks*, pp. 3, 17–22. Princeton University Press, Peinceton, New Jersey (1962).