Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Technical Note—A Branch-and-Bound Algorithm for Seeking the P-Median

Pertti Järvinen, Jaakko Rajala, Heikki Sinervo,

12. E. L. KELLER, *Quadratic Optimization and Linear Complementarity*, Doctoral Dissertation, University of Michigan, Ann Arbor, 1969.
13. C. E. LEMKE, "Bimatrix Equilibrium Points and Mathematical Programming," *Management Sci.* **11,** 681–689 (1965).
14. ———, "On Complementary Pivot Theory," in (G. B. DANTZIG AND A. F. VEINOTT JR., eds.) *Mathematics of Decision Sciences*, Part I, pp. 95–114, American Mathematical Society, Providence, R. I., 1968.
15. O. L. MANGASARIAN, "Pseudo-Convex Functions," *SIAM J. of Control* **3,** 281–290 (1965).
16. ———, *Nonlinear Programming*, McGraw-Hill, New York, N. Y., 1969.
17. B. MARTOS, "Subdefinite Matrices and Quadratic Forms," *SIAM J. Appl. Math.* **17,** 1215–1223 (1969).
18. ———, "Quadratic Programming with a Quasiconvex Objective Function," *Opns. Res.*, **19,** 87–97 (1971).
19. W. C. MYLANDER, "Nonconvex Quadratic Programming by a Modification of Lemke's Method," Technical Paper RAC-TP-414, Research Analysis Corp., McLean, Virginia, 1971.
20. K. RITTER, "A Method for Solving Maximum Problems with a Nonconcave Quadratic Function," *Z. Wahrscheimlich-Keithstheorie verw. Geb.* **4,** 340–351 (1966).
21. C. VAN DE PANNE AND A. WHINSTON, "A Parametric Simplicial Formulation of Houthakker's Capacity Method," *Econometrica* **34,** 354–380 (1966).
22. P. WOLFE, "The Simplex Method for Quadratic Programming," *Econometrica* **27,** 382–398 (1959).

A Branch-and-Bound Algorithm for Seeking the *p*-Median

Pertti Järvinen, Jaakko Rajala, and Heikki Sinervo

University of Tampere, Finland

(Received March 6, 1970)

We construct a branch-and-bound algorithm for seeking the *p*-median, and make comparisons with the vertex-substitution method of TEITZ AND BART [*Opns. Res.* 16, 955-961 (1968)]. We show how the vertex-substitution method can lead to a local optimum, and give a heuristic method for finding a good initial solution for this method.

THE CONCEPT of the p-median of a graph has been defined by HAKIMI,[3] who generalized the concept of the median of a graph.[4] REVELLE, MARKS, AND LIEBMAN[5] showed how the p-median is related to other location models. If we consider more than just these 'private- and public-sector location models' we see that finding the p-median solves a certain clustering task, because the p-median has the following property: There is a set of p points consisting entirely of models of the graph, that minimizes the sum of weighted distances to the closest of any p points on the graph. (However, another set of p points, not all nodes, could possibly provide the same minimum.) The nodes correspond to the units that must be clustered.

PROPERTIES OF THE p-MEDIAN

LET V^p be a set of p points $v_{i_1}, v_{i_2}, \cdots, v_{i_p}$ on a weighted graph $G=(V, E)$ and let

$$d(V^p, v_j)=\min\{d(v_{i_1}, v_j),\quad d(v_{i_2}, v_j), \cdots, d(v_{i_p}, v_j)\},$$

where $d(v_{i_k}, v_j)$ is the length of the shortest path from vertex v_{i_k} to vertex v_j and $v_{i_k}\epsilon V^p \subset V$, $k=1, 2, \cdots, p$; $v_j \epsilon V$, $j=1, 2, \cdots, n$.

The set of points $V_0{}^p$ is a p-median of G if, for every $V_k{}^p$ on G,

$$\sum_{j=1}^{j=n} h_j d(V_0{}^p, v_j) \leqq \sum_{j=1}^{j=n} h_j d(V_k{}^p, v_j).$$

We use the notation $D_{ij}=h_j d(v_i, v_j)$. Then for p-median $V_0{}^p$, we have

$$\sum_{j=1}^{j=n} h_j d(V_0{}^p, v_j)=\sum_{j=1}^{j=n} (\min_{v_i \epsilon V_0{}^p} D_{ij})=S.$$

If we consider the $n\times n$ matrix D_{ij}, then the sum S for the p-median $V_0{}^p$ has the following properties:

Property 1. In the sum S there exists one and only one element from every column of the D-matrix.

Property 2. All terms in S are located in only p rows of the D-matrix.

Property 3. The sum S consists of at least p zero terms, which correspond to elements D_{ii} in the main diagonal and $v_i \epsilon V_0{}^p$.

THE BRANCH-AND-BOUND ALGORITHM

THIS ALGORITHM is based mainly on property 2. We try to choose the $n-p$ vertices that do not belong to $V_0{}^p$. Our partition rule (*see* BALAS[2]) is the following: We first take one vertex away from the p-median, then two, three, $\cdots$, and finally $n-p$ vertices.

For the computation of a lower bound of the sum S, let us assume that r vertices $v_{k_1}, v_{k_2}, \cdots, v_{k_r}$ have been taken away $(1\leqq r\leqq n-p)$. Then we have $n-r$ vertices from which p vertices must be chosen. Thus, we have two sets of vertices

$$V^r=\{v_{k_1}, v_{k_2}, \cdots, v_{k_r}\} \quad \text{and} \quad V^{n-r}=\{v_{j_1}, v_{j_2}, \cdots, v_{j_{n-r}}\}=V-V^r$$

and corresponding sets of indices

$$K=\{k_1, k_2, \cdots, k_r\}, \qquad J=\{j_1, j_2, \cdots, j_{n-r}\}.$$

For the columns of the D-matrix (property 1), that corresponds to the set K, we compute for column $k\epsilon K$

$$s_k=\min_{i\epsilon J} D_{ik}, \tag{1}$$

and, for another column $j\epsilon J$,

$$s_j=\min_{i\epsilon J, i\neq j} D_{ij}. \tag{2}$$

The lower bound of the sum S is $LB(K)=S_K+S_J$, where $S_K=\sum_{k\epsilon K} s_k$ and $S_J=$ the sum of the $n-r-p$ smallest of $s_j(j\epsilon J)$, because of property 3.

The algorithm has the following steps:

Step **1.** $r=1$, $S=\infty$; generate index sets $K=\{1\}, \{2\}, \cdots, \{n\}$; go to step 3.

Step 2. Let $K'=\{k_1, k_2, \cdots, k_r\}$. Set $r=r+1$, passivate the lower bound $LB(K')$, and generate new index sets K, where every index set consists of indices of K' and one new index.

Step 3. Compute $LB(K)$ for every new index set.

Step 4. If $r<n-p$, then choose the smallest $LB(K)$ from the newest index sets and denote this set by K' and go to step 2.

Step 5. If $r=n-p$, we have a feasible solution of our problem (when we first arrive there, let us call this solution branch-and-bound without backtracking). Let $LB(K')$ be the smallest $LB(K)$ from the newest index sets. If $LB(K')<S$, then set $S=LB(K')$ and put the corresponding p-median in memory and passivate the lower bounds whose $LB(K)\geqq S$. If after passivation no active ones exist, then the algorithm terminates and the minimum sum is in S. Otherwise, choose the smallest $LB(K)$ from the active lower bounds and denote this set by K' and go to step 2.

It is possible that one could already have found a feasible p-median in step 4 by a closer study of which rows in (1) and (2) produce these minima. If the number of these rows $i\epsilon J$ is at most p, we then have a feasible p-median. Perhaps a closer study of this kind will increase computing time.

EXAMPLE OF A LOCAL MINIMUM IN THE SUBSTITUTION METHOD

REVELLE ET AL. say that "although TEITZ AND BART[6] offer no assurance of optimality at termination, it can be argued that the procedure should terminate optimally. This is because successive node substitution is completely analogous to the vector substitution which takes place in the simplex procedure of linear programming."[5] We show that this is not true as follows.

Suppose that in a graph like the one in Fig. 1 we try to seek 2-medians of the graph. We assume that all weights are equal to one. Our D-matrix is then

	1	2	3	4	5	6
1	0	2	4	2	3	2
2	2	0	2	4	2	3
3	4	2	0	2	4	5
4	2	4	2	0	5	4
5	3	2	4	5	0	1
6	2	3	5	4	1	0

If the initial solution is $\{v_1, v_2\}$, then $S=8$. When we now try to substitute some other vertex for either vertex v_1 or vertex v_2, each substitution will produce $S=9$. Therefore, the set $\{v_1, v_2\}$ would seem to be 2-medians of the graph in Fig. 1. We can, however, find two better solutions, namely the sets $\{v_3, v_6\}$ and $\{v_4, v_5\}$, whose S-values are equal to 7.

HEURISTIC METHOD FOR SEEKING THE p-MEDIAN

We seek p rows of the D-matrix (property 2) in such a way that as many second smallest s-values (formula 2) as possible are included. We construct the algorithm as follows.

Step 1. Set $q=1$, $I=\{1, 2, \cdots, n\}$, $J=\{1, 2, \cdots, n\}$, and $V^p=\phi$.

Step 2. Find, for every $j \epsilon J$, $s_{i'j}=\min_{i \epsilon I, i \neq j} D_{ij}$, where i' is the row in which the second smallest D_{ij} exists. If there exists more than one row in which D_{ij} is equal to the second smallest, then we take all these $s_{i'j}$.

Step 3. Find the row i_0 that has the largest number of terms $s_{i'j}$. If there are many rows with an equal number of s-terms, then choose as row i_0 the one whose sum of s-terms is smallest. Let the corresponding column numbers be $j_1, j_2, \cdots, j_k$. Join the vertex v_{i_0} to set V^p. If $q=p$, then terminate, and V^p will form the p-median. Otherwise $q=q+1$ and go to step 4.

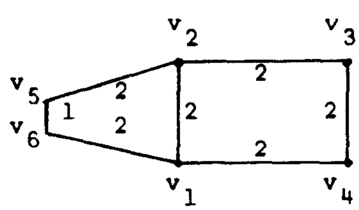

Figure 1

Step 4. Exclude i_0 (see step 3) from the set I and $j_1, j_2, \cdots, j_k$ from the set J, and the terms $s_{i'j_h}$ $(h=1, 2, \cdots, k)$ from the set of $s_{i'j}$. Go to step 3.

COMPUTATIONAL RESULTS

This section is devoted to the computational experiments conducted and their findings. We performed experiments with a 20×20 matrix and tried to find 5-medians, 10-medians, and 15-medians. The entries of symmetric distance matrices were generated at random from a uniform distribution between 1 and 1024, inclusively. The elements in the main diagonal were equal to zero. The weights h_i $(i=1, 2, \cdots, 20)$ were also uniformly distributed between 1 and 1024. The number of D-matrices was 50.

We studied the following five methods: branch-and-bound, branch-and-bound without backtracking, substitution (where the initial solution was formed by the first p nodes), heuristic, and substitution with a heuristic initial solution. The efficiencies of the last four methods are shown in Table I.

The Algorithms were programmed by Autocode for the Elliott 503. The running times are shown in Table II.

TABLE I
EFFICIENCIES AND CORRESPONDING NUMBERS OF PROBLEMS SOLVED BY SUBOPTIMAL-PRODUCING TECHNIQUES

Efficiency	Branch-and-bound without backtracking			Substitution			Heuristic			Substitution with heuristic		
median	5	10	15	5	10	15	5	10	15	5	10	15
1.00	9	25	37	29	31	48				33	44	47
0.95	9	16	5	5	2	0	2			5	4	3
0.90	9	2	6	3	6	2				7	1	
0.85	9	6	2	7	6					0	1	
0.80	3	1		2	4		2			2		
0.75	5			2	0		3	2		2		
0.70	2			2	1		9	2		1		
0.65 and lower	4						34	46	50			

Note. After Ashour,[1] "the quality of a solution obtained by one of the suboptimal-producing methods, referred to as efficiency, is defined as the quotient of the optimal schedule time and that solution."

We can conclude that the efficiencies of branch-and-bound with backtracking, substitution, and substitution with a heuristic initial solution increases when $p \to n$, but the heuristic method itself behaves in the opposite way. When we take the initial solution in the substitution technique with the heuristic method, we get, on the average better results in faster time than by the original substitution method.

TABLE II
THE RUNNING TIMES OF VARIOUS TECHNIQUES
(The times are in seconds.)

	Branch-and-bound				Substitution			
	Mean	St. dev.	Min	Max	Mean	St. dev.	Min	Max
5-median	109.26	59.88	18.0	275.0	2.09	0.65	1.1	4.3
10-median	46.26	41.65	5.8	226.8	2.27	0.66	0.9	3.8
15-median	2.519	0.46	1.7	3.8	1.799	0.26	1.4	2.7

	Substitution with heuristic			
	Mean	St.dev.	Min	Max
5-median	1.48	0.48	0.7	2.7
10-median	1.84	0.22	1.5	2.2
15-median	1.86	0.25	1.5	2.6

Note. The branch-and-bound method without backtracking took 1.3–2.8 sec and the heuristic method 0.3–0.4 sec.

REFERENCES

1. SAID ASHOUR, "An Experimental Investigation and Comparative Evaluation of Flow-Shop Scheduling Techniques," *Opns. Res.* **18,** 541–549 (1970).
2. E. BALAS, "A Note on the Branch-and-Bound Principle," *Opns. Res.* **16,** 442–444 (1968).
3. S. L. HAKIMI, "Optimum Location of Switching Centers and Absolute Centers and Medians of a Graph." *Opns. Res.* **12,** 450–459 (1964).
4. ———, "Optimum Distribution of Switching Centers in a Communications Network and Some Related Problems," *Opns. Res.* **13,** 462–475 (1965).
5. C. REVELLE, D. MARKS AND J. C. LIEBMAN, "An Analysis of Private and Public Sector Location Models," *Management Sci.* **16,** 692–707 (1970).
6. M. B. TEITZ AND P. BART, "Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph," *Opns. Res.* **16,** 955–961 (1968).

Strengthened Dantzig Cuts for Integer Programming

David S. Rubin

University of North Carolina, Chapel Hill, North Carolina

and

Robert L. Graves

University of Chicago, Chicago, Illinois

(Received December 30, 1970)

In 1959, Dantzig proposed a particularly simple cut for integer programming. However, in 1963, Gomory and Hoffman showed that, in general, this cut does not provide a finite algorithm. In 1968, Bowman and Nemhauser showed that a slightly modified version of the Dantzig cut does provide a finite procedure. We show how this latter cut can be strengthened through the use of group-theoretic techniques.

CONSIDER the integer program IP: max $z=c'x$, subject to $Ax=b$, $x\geqq 0$ and integer. We assume that all the components of A, b, and c are themselves integers. Dropping the integrality requirements on x leaves the associated linear program LP. Let B be the LP optimal basis. Partition A as (B, N), and c and x accordingly. Then Tucker's optimal tableau is

$$\begin{array}{r|l} & -x'_N \\ \hline z=c_BB^{-1}b & c_B'B^{-1}N-c_N', \\ x_B=B^{-1}b & B^{-1}N, \\ x_N=0 & -I. \end{array}$$

Denote the typical element of the tableau by y_{ij}, $i=0, 1, \cdots, m+n; j=0, 1, \cdots, n$. Let f_{ij} $(=y_{ij}-[y_{ij}])$ be its fractional part. Let y_j be the jth column of the tableau, and f_j its fractional part. The corresponding quantities in the new tableau resulting from a single pivot are denoted y'_{ij}, f'_{ij}, y_j', and f_j'.