# A tree search algorithm for the $p$-median problem

N. CHRISTOFIDES and J.E. BEASLEY
*Department of Management Science, Imperial College, London SW7 2BX, United Kingdom*

Received February 1981
Revised November 1981

In this paper we present two lower bounds for the $p$-median problem, the problem of locating $p$ facilities (medians) on a network. These bounds are based on two separate lagrangean relaxations of a zero-one formulation of the problem with subgradient optimisation being used to maximise these bounds. Penalty tests based on these lower bounds and a heuristically determined upper bound to the problem are developed and shown to result in a large reduction in problem size. The incorporation of the lower bounds and the penalty tests into a tree search procedure is described and computational results are given for problems with an arbitrary number of medians and having up to 200 vertices. A comparison is also made between these algorithms and the dual-based algorithm of Erlenkotter.

## 1. Introduction

The $p$-median problem is the problem of locating $p$ facilities (medians) on a network so as to minimise the sum of all the distances from each vertex to its nearest facility. This problem occurs quite frequently in practical situations, e.g. the location of supply depots on a road network where the vertices of the network represent the customers that are to be supplied from the depots.

A number of algorithms have been developed for this problem:

(i) Tree search procedures such as Efroymson and Ray [2], and Khumawala [7];

(ii) Linear programming based approaches such as Spielberg [15], Garfinkel et al. [5], Rosing and Revelle [13], and Schrage [14];

(iii) Heuristic methods such as Maranzana [8], and Teitz and Bart [16].

It is well known that this problem is NP-hard [17], hence the use of tree search procedures for its solution. Lagrangean relaxation in conjunction with subgradient optimisation can be used to compute lower bounds for use in tree search procedures, and has been applied successfully by other authors to location problems (e.g. Neebe and Rao [11], Cornuejols et al. [1], Narula et al. [9], Neebe [12] and Nauss [10]). Here we extend this work by comparing two different lagrangean relaxations of the problem, by developing "penalty tests" for reducing the problem size and by incorporating the resulting lower bounds into a tree search procedure where branching is controlled by the structure of the solution to the lagrangean problem. Computational results are given for large networks (up to 200 vertices).

## 2. Problem formulation

The $p$-median problem can be formulated as a zero-one program as follows:

Let $d_{ij}$ ($\geq 0$) represent the cost of allocating vertex $j$ to vertex $i$, with $V$ the entire vertex set. Let

$$a_{ij} = \begin{cases} 0 & \text{if vertex } j \text{ cannot be allocated to vertex } i, \\ 1 & \text{otherwise.} \end{cases}$$

(Note that $a_{ij} = 0$ is equivalent to setting $d_{ij} = \infty$.)
Let

$$x_{ij} = \begin{cases} 1 & \text{if vertex } j \text{ is allocated to vertex } i \\ & \text{(which implies that } i \text{ is a median vertex),} \\ 0 & \text{otherwise} \end{cases}$$

The problem then is

$$\min \quad z = \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}, \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \tag{2}$$

$$\sum_{i \in V} x_{ii} = p, \tag{3}$$

North-Holland Publishing Company
European Journal of Operational Research 10 (1982) 196–204

$$\sum_{\substack{j\in V\\ j\neq i}} a_{ij}x_{ij} \leq n_i x_{ii} \quad \forall i \in V, \tag{4}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in V, j \in V. \tag{5}$$

Equation (2) ensures that each vertex is allocated to a median and (3) that there are $p$ median vertices. In (4) we define

$$n_i = \sum_{\substack{j\in V\\ j\neq i}} a_{ij} \quad \forall i \in V. \tag{6}$$

The usual formulation of this problem is $a_{ij} = 1$ $\forall i \in V$, $j \in V$ and $n_i = |V| - 1$ $\forall i \in V$. Equation (4) ensures that nothing can be allocated to a vertex unless that vertex is a median.

The two relaxations of this problem that we investigate are

1. LR1 – relaxation of equation (2);
2. LR2 – relaxation of equation (4).

## 3. The lagrangean dual programs

In formulating the lagrangean dual programs we introduce $K_1$ as the set of vertices that have been positively identified as medians and $K_0$ as the set of vertices that have been positively identified as non-medians to facilitate incorporating the bounds into a tree search procedure.

Consequently we can define

$$\beta_j = \min_{i\in K_1} d_{ij} \quad \forall j \in V - K_1 \tag{7}$$

as the maximum allocation cost of each vertex

### 3.1. Relaxation LR1

For the relaxation LR1 we introduce lagrange multipliers $\lambda_j \geq 0$ ($j \in V$) for (2) in the zero-one formulation of the problem to obtain the lagrangean dual program

$$\min \quad z_D = \sum_{i\in V-K_0} \sum_{\substack{j\in V-K_1\\ d_{ij}\leq\beta_j}} (d_{ij} - \lambda_j)\, x_{ij} + \sum_{j\in K_1} (d_{jj} - \lambda_j) + \sum_{j\in V} \lambda_j, \tag{8}$$

$$\text{s.t.} \quad \sum_{i\in V-K_0-K_1} x_{ii} = p - |K_1|, \tag{9}$$

$$\sum_{\substack{j\in V-K_1\\ d_{ij}\leq\beta_j\\ j\neq i}} a_{ij}x_{ij} \leq n_i x_{ii} \quad \forall i \in V \tag{10}$$

$$x_{ii} = 0 \quad \forall i \in K_0, \tag{11}$$

$$x_{ii} = 1 \quad \forall i \in K_1, \tag{12}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in V, j \in V, \tag{13}$$

This program can be easily solved. Suppose a set of values for the $\lambda_j$ are given and consider the effect on the lagrangean dual of specifying that $k$ is a median vertex. The best set of allocations of vertices to $k$ that can be achieved is

$$x_{kk} = 1,$$

$$x_{kj} = \begin{cases} 1 & \text{if } (d_{kj} - \lambda_j) \leq 0 \text{ and } j \neq k \\ & \text{and } d_{kj} \leq \beta_j \text{ and } j \in V - K_1, \\ 0 & \text{otherwise } (j \neq k). \end{cases} \tag{14}$$

The contribution to the dual objective function $\alpha_k$ is therefore given by

$$\alpha_k = (d_{kk} - \lambda_k) + \sum_{\substack{j\in V-K_1\\ d_{kj}\leq\beta_j\\ j\neq k}} \min(0, d_{kj} - \lambda_j)$$

$$\forall k \in V - K_0. \tag{15}$$

The lagrangean dual program (equations (8) to (13)) then reduces to

$$\min \quad z_D = \sum_{i\in V-K_0} \alpha_i x_{ii} + \sum_{j\in V} \lambda_j, \tag{16}$$

$$\text{s.t.} \quad \sum_{i\in V-K_0-K_1} x_{ii} = p - |K_1|, \tag{17}$$

$$x_{ii} = 1 \quad \forall i \in K_1, \tag{18}$$

$$x_{ii} \in \{0,1\} \quad \forall i \in V - K_0. \tag{19}$$

The optimal solution to this program is found by setting $x_{ii}$ to one for $i \in K_1$ and the $p - |K_1|$ $x_{ii}$ with the smallest $\alpha_i$ ($i \in V - K_0 - K_1$) to one with all other $x_{ii}$ set to zero. Let $(x^*_{ii})$ represent the values given to the $(x_{ii})$ in the optimal solution of the dual program defined by (16) to (19), then the values of the allocation variables $(x^*_{ij})$ are given by

$$x^*_{ij} = \begin{cases} 1 & \text{if } x^*_{ii} = 1 \text{ and } i \neq j \text{ and } (d_{ij} - \lambda_j) \leq 0 \\ & \text{and } d_{ij} \leq \beta_j \text{ and } j \in V - K_1, \\ 0 & \text{otherwise } (i \neq j). \end{cases} \tag{20}$$

The dual solution value $z_D^*$ is given by

$$z_D^* = \sum_{i \in V} \alpha_i x_{ii}^* + \sum_{j \in V} \lambda_j. \tag{21}$$

### 3.2. Relaxation LR2

For the relaxation LR2 we introduce lagrange multipliers $\lambda_i \geq 0$ $(i \in V)$ for (4) in the zero-one formulation of the problem to obtain the lagrangean dual program

$$\min \quad z_D = \sum_{i \in V-K_0} \sum_{\substack{j \in V-K_1 \\ d_{ij} \leq \beta_i \\ j \neq i}} (d_{ij} + a_{ij}\lambda_i)\, x_{ij} + \sum_{i \in V-K_0} (d_{ii} - n_i\lambda_i)\, x_{ii}, \tag{22}$$

$$\text{s.t.} \quad \sum_{i \in V-K_0-K_1} x_{ij} = 1 \quad \forall j \in V-K_1, \tag{23}$$

$$\sum_{i \in V-K_0-K_1} x_{ii} = p - |K_1|, \tag{24}$$

$$\lambda_i = 0 \qquad \forall i \in K_1, \tag{25}$$

$$x_{ii} = 0 \qquad \forall i \in K_0, \tag{26}$$

$$x_{ii} = 1 \qquad \forall i \in K_1, \tag{27}$$

$$x_{ij} \in \{0,1\} \quad \forall i \in V, j \in V. \tag{28}$$

This program can be easily solved – define $\alpha_j$ as the minimum cost of allocation of vertex $j$ other than to itself, i.e.

$$\alpha_j = \min_{\substack{i \in V-K_0 \\ i \neq j \\ d_{ij} \leq \beta_i}} (d_{ij} + a_{ij}\lambda_i) \quad \forall j \in V-K_1. \tag{29}$$

The lagrangean duel program (equations (22) to (28)) then reduces to

$$\min \quad z_D = \sum_{j \in V-K_1} \alpha_j + \sum_{j \in K_1} d_{jj} + \sum_{j \in V-K_0-K_1} (d_{jj} - n_j\lambda_j - \alpha_j)\, x_{jj}, \tag{30}$$

$$\text{s.t.} \quad \sum_{i \in V-K_0-K_1} x_{ii} = p - |K_1|, \tag{31}$$

$$x_{ii} = 1 \qquad \forall i \in K_1, \tag{32}$$

$$x_{ii} \in \{0,1\} \quad \forall i \in V-K_0. \tag{33}$$

The optimal solution to this program is found by setting $x_{ii}$ to one for $i \in K_1$ and the $p - |K_1|$ $x_{ii}$ with the smallest $\alpha_i$ $(i \in V-K_0-K_1)$ to one with all other $x_{ii}$ set to zero. Let $(x_{ii}^*)$ represent the values given to the $(x_{ii})$ in the optimal solution of the dual program defined by (30) to (33), then the values of the allocation variables $(x_{ij}^*)$ are given by

$$x_{ij}^* = \begin{cases} 1 & \text{if } x_{jj}^* = 0 \text{ and } i \text{ corresponds to} \\ & \text{the minimum in (29) for } \alpha_j \ (i \neq j), \\ 0 & \text{otherwise } (i \neq j) \end{cases} \tag{34}$$

The dual solution value $z_D^*$ is given by

$$z_D^* = \sum_{j \in V-K_1} \alpha_j + \sum_{j \in K_1} d_{jj} + \sum_{j \in V-K_0-K_1} (d_{jj} - n_j\lambda_j - \alpha_j)\, x_{jj}^*. \tag{35}$$

This relaxation can be considerably strengthened by judicious choice of the $a_{ij}$ because $n_i$ $(= \Sigma_{j \in V, j \neq i} a_{ij})$ appears in the dual objective function equation (22) with a negative sign and so decreasing $n_i$ will (in general) increase the bound. We adopted the following approach to solving the lagrangean dual:

(1) Let $a_{ij} = 1$ $\forall i \in V, j \in V$ and determine $\alpha_j$ as given in (29);

(2) Reset all $a_{ij}$ by

$$a_{ij} = \begin{cases} \max\left(0, \dfrac{\alpha_j - d_{ij}}{\lambda_i}\right), & \lambda_i \neq 0, \forall i \in V-K_0, \\ & \forall j \in V-K_1, d_{ij} \leq \beta_j, \\ 0, & \text{otherwise} \end{cases} \tag{36}$$

(i.e. decrease all $a_{ij}$ whilst leaving $\alpha_j$ unchanged) and solve the lagrangean dual using these new $a_{ij}$ values (and new $n_i$ values).

## 4. Problem reduction

From the lagrangean dual program we can estimate the increase in the lower bound that would result from forcing vertices to be median/non-medians and from forcing an arc (i.e. an allocation) to be in the solution. If the lower bound that results from imposing some condition in the lagrangean dual program is above some heuristically determined upper bound to the $p$-

Table 1
Penalties for LR1

| | $j \in V - M^*$ | | $j \in M^* - K_1$ | |
|---|---|---|---|---|
| $i \in V - M^* - K_0$ | $\alpha_i - \max_{k \in M^* - K_1} \alpha_k$ | $i = j$ | $\alpha_i - \alpha_j + \max(0, d_{ij} - \lambda_j)$ | |
| | $\alpha_i - \max_{k \in M^* - K_1} \alpha_k + \max(0, d_{ij} - \lambda_j)$ | $i \neq j$ | | |
| $i \in M^*$ | $\max(0, d_{ij} - \lambda_j)$ | | 0 | $i = j$ |
| | | | $\min_{k \in V - M^* - K_0} \alpha_k - \alpha_j + \max(0, d_{ij} - \lambda_j)$ | $i \neq j$ |

median solution, then that condition cannot be satisfied in the optimal $p$-median solution.

Let $M^*$ be the set of lagrangean median vertices i.e.

$$M^* = \{i \mid x_{ii}^* = 1\}. \tag{37}$$

We can then develop penalties for setting $x_{ij}$ to one in the lagrangean dual solution (i.e. forcing the allocation of $j$ to $i$). These are given in Table 1 for relaxation LR1 and Table 2 for relaxation LR2, e.g. for Table 1 with $i \in V - M^* - K_0$ and $j \in M^* - K_1$ where $i$ is a non-median and $j$ is a median, the penalty is composed from the cost of making $j$ a non-median ($-\alpha_j$), making $i$ a median ($+\alpha_i$) and allocating $j$ to $i$ if it is not already allocated ($\max(0, d_{ij} - \lambda_j)$).

In a similar manner penalties can be developed for $x_{ij} = 0$ (given that $x_{ij}^* = 1$). One further set of penalties that are worthy of mention in connection with LR2 relate to ($\beta_j$) the set of maximum allocation costs. For relaxation LR2 it is possible to show that

$$\beta_j \leq z_{UB} - z_D^* + \alpha_j, \quad j \notin M^*, \tag{38}$$

$$\beta_j \leq z_{UB} - z_D^* + (d_{jj} - n_j \lambda_j) + \min_{k \in V - M^* - K_0} (d_{kk} - n_k \lambda_k - \alpha_k), \quad j \in M^* \tag{39}$$

where $z_{UB}$ is any upper bound on the problem.

Obviously the penalties given above can be strengthened, albeit at an increase in computational effort but we have found the simple penalties outlined above to be quite effective in reducing the size of the problem by fixing variables.

## 5. The subgradient procedure

Subgradient optimisation is a technique for progressively updating lagrange multipliers in a systematic way in an attempt to maximise the value

Table 2
Penalties for LR2

| | $j \in V - M^*$ | | $j \in M^* - K_1$ | |
|---|---|---|---|---|
| $i \in V - M^* - K_0$ | $(d_{ii} - n_i \lambda_i - \alpha_i) - \max_{k \in M^* - K_1} (d_{kk} - n_k \lambda_k - \alpha_k)$ | $i = j$ | $(d_{ii} - n_i \lambda_i - \alpha_i) - (d_{jj} - n_j \lambda_j) + (d_{ij} + a_{ij} \lambda_i)$ | |
| | $(d_{ii} - n_i \lambda_i - \alpha_i) + (d_{ij} + a_{ij} \lambda_i) - \max_{k \in M^* - K_1} (d_{kk} - n_k \lambda_k - \alpha_k) - \alpha_j$ | $i \neq j$ | | |
| $i \in M^*$ | $-\alpha_j + (d_{ij} + a_{ij} \lambda_i)$ | | 0 | $i = j$ |
| | | | $-(d_{jj} - n_j \lambda_j) + (d_{ij} + a_{ij} \lambda_i) + \min_{k \in V - M^* - K_0} (d_{kk} - n_k \lambda_k - \alpha_k)$ | $i \neq j$ |

of the lagrangean dual program. The subgradient procedure, incorporating the penalty tests developed above, is as follows:

(1) Determine an initial value for $z_{UB}$ – the upper bound on the problem. This can be done using any heuristic for the $p$-median problem (e.g. Teitz and Bart [16]). Let $z_L^* = -\infty$ where $z_L^*$ is the maximum lower bound found during the subgradient procedure.

(2) Choose an initial value for the $(\lambda_j)$ – we used

$$\lambda_j = \min_{\substack{i \in V \\ i \neq j}} d_{ij} \quad \forall j \in V \quad \text{LR1}, \tag{40}$$

$$\lambda_i = 0 \quad \forall i \in V \quad \text{LR2}. \tag{41}$$

(3) For the current set of lagrange multipliers $(\lambda_j)$ determine the optimal lagrangean dual solution $z_D^*$ with $M^*$ the associated median set and $(x_{ij}^*)$ the associated variable values.

(4) Determine the cost of the feasible solution associated with $M^*$. This is given by

$$\sum_{i \in M^*} d_{ii} + \sum_{j \in V - M^*} \left( \min_{i \in M^*} (d_{ij}) \right). \tag{42}$$

If this is better than $z_{UB}$ then update $z_{UB}$ accordingly. Similarily set

$$z_L^* = \max(z_L^*, z_D^*). \tag{43}$$

(5) Stop if $z_{UB} = z_L^*$, i.e. the highest lower bound found and the upper bound (corresponding to a feasible solution) coincide; else go to (6).

(6) Perform the penalty tests outlined previously.

(7) Define the subgradient vector $S$ by

$$S_j = 1 - \sum_{i \in M^*} x_{ij}^* \quad \forall j \in V \quad \text{LR1}, \tag{44}$$

$$S_i = \sum_{\substack{j \in V \\ j \neq i}} a_{ij} x_{ij}^* - n_i x_{ii}^* \quad \forall i \in V \quad \text{LR2} \tag{45}$$

(8) Stop if $S_j = 0 \; \forall j \in V$ when an optimal solution $(x_{ij}^*)$ will have been found else go to (9).

(9) Calculate a step size $t$ for use in updating the lagrange multipliers by

$$t = \frac{\pi (z_{UB} - z_D^*)}{\|S\|^2} \tag{46}$$

where $\pi$ a constant $(0 < \pi \leq 2)$ and $\|S\|$ any norm of the subgradient vector – we used the Euclidean norm $(\Sigma_{j \in V} S_j^2)^{1/2}$.

(10) Update the lagrange multipliers by

$$\lambda_j = \max(0, \lambda_j + tS_j) \quad \forall j \in V. \tag{47}$$

(11) Go to (3) to resolve the lagrangean dual program with this new set of multipliers unless some termination condition (such as the maximum number of iterations allowed) is satisfied in which case stop.

In choosing a value for $\pi$ (eq. (46)) we followed the approach of Held, Wolfe and Crowder [6] in letting $\pi = 2$ for $2|V|$ iterations and then successively halving both the value of $\pi$ and the number of iterations until the number of iterations reached a threshold value of 5, $\pi$ was then halved every 5 iterations.

## 6. The tree search procedure

In the event that the subgradient procedure does not optimally solve the problem a reduction in the size of the problem will probably have been achieved because the penalty tests will have fixed certain vertices as medians/non-medians etc. Any of the existing tree search algorithms for the $p$-median problem could then be used on the reduced problem to obtain an optimal solution. We used a depth-first tree search procedure based on the lagrangean relaxations of the problem discussed earlier. The branching strategy used was to pick the vertex $j$ corresponding to

$$\alpha_j = \min_{i \in M^* - K_1} \alpha_i \quad \text{LR1}, \tag{48}$$

$$d_{jj} - n_j \lambda_j - \alpha_j = \min_{i \in M^* - K_1} (d_{ii} - n_i \lambda_i - \alpha_i) \quad \text{LR2} \tag{49}$$

at any node of the tree (ties for $j$ arbitrarily broken) and to branch by setting $x_{jj}$ to one (as intuitively $j$ corresponds to the vertex most likely to be a median in the optimal completion of the current node).

Obviously there is a trade-off between continuing to investigate a node of the tree using the subgradient procedure and abandoning it to branch from that node and we indicate below how we balanced the two alternatives in the algorithm used to obtain the computational results reported in the next section.

### 6.1. The initial node

We carried out the subgradient procedure at the initial tree node until one or more of the following conditions were satisfied:

(i) All subgradients were zero (i.e. an optimal solution had been obtained);

(ii) The minimum feasible solution $z_{UB}$ and the maximum lower bound $z_L^*$ coincided (i.e. $z_{UB}$ corresponds to an optimal solution);

(iii) $\pi$ (eq. (46)) dropped below 0.005.

Computational experience indicated that, for both relaxation LR1 and LR2, computing the penalties for all possible allocations at each subgradient iteration had little effect on the convergence of the procedure. Accordingly the penalty tests carried out at each subgradient iteration were:

(1) For LR1 and LR2 the tests to identify vertices as medians/non-medians;

(2) For LR2 the test on the maximum allocation cost $\beta_j$.

If no optimal solution was found at the initial node, then the set of lagrange multipliers corresponding to $z_L^*$, the maximum lower bound, were recalled and the penalty tests for all possible allocations using this set of multipliers performed. Typically this removed a large percentage of the possible allocations.

### 6.2. The tree search nodes

Thirty subgradient iterations were performed at the tree search nodes with the initial set of lagrange multipliers being the best set associated with the previous node. A value of $\pi$ (eq. (46)) equal to one was used and the same penalty tests that were used at the initial tree node were performed at every node.

## 7. Results

The algorithm was programmed in FORTRAN and run on a CDC 7600 machine, (using the FTN

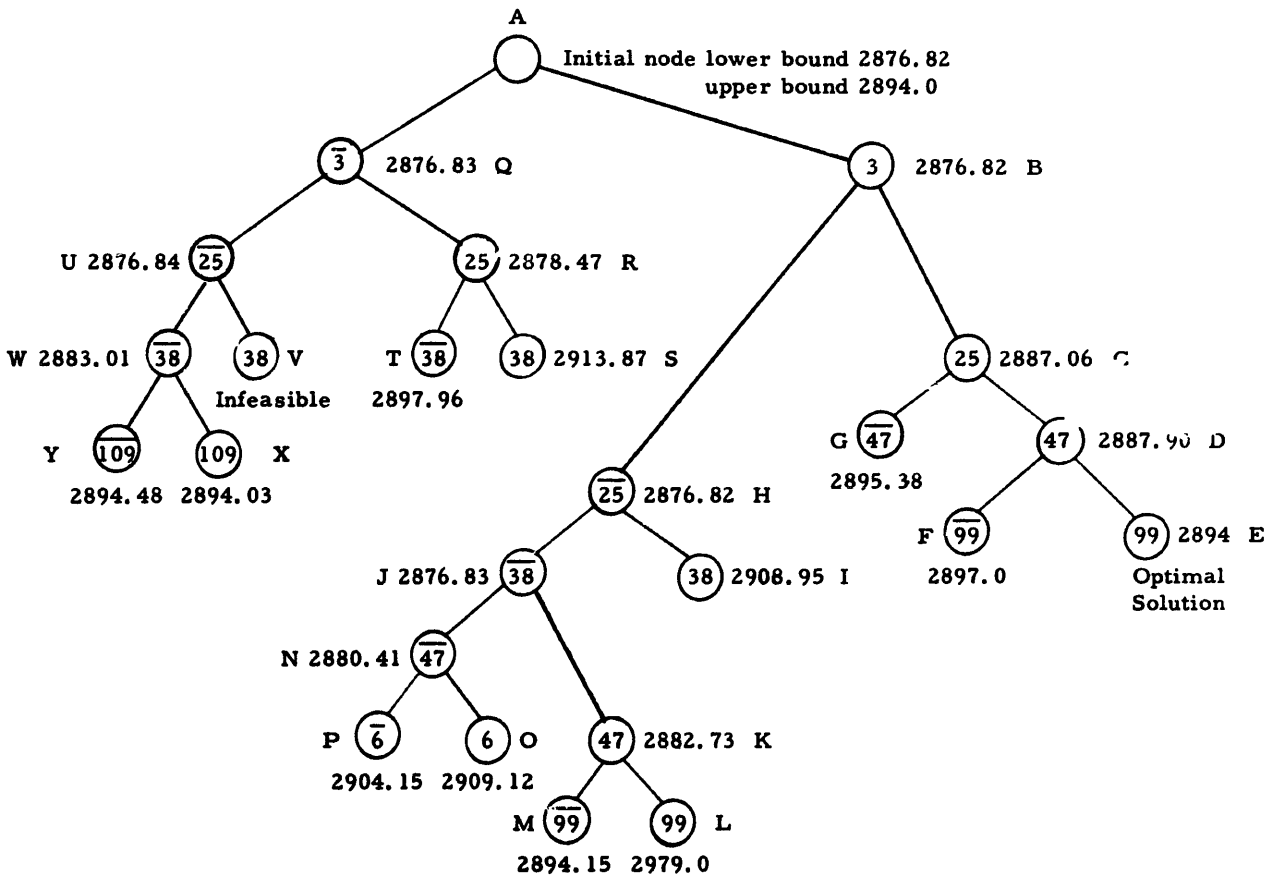


Fig. 1. The tree search for $p=5$, $n=150$, Relaxation LR1.

Table 3
Results

| Problem number | $\|V\|$ | $p$ | Relaxation LR1 | | | | Relaxation LR2 | | | | Erlenkotter algorithm | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Number of sub-gradient iterations initial tree node | Duality gap (%) initial tree node | Number of tree nodes | Time CDC 7600 seconds | Number of sub-gradient iterations initial tree node | Duality gap (%) initial tree node | Number of tree nodes | Time CDC 7600 seconds | Number of cost iterations | Duality gap (%) final cost iteration initial tree node | Number of tree nodes final cost iteration | Time CDC 7600 seconds |
| 1 | 30 | 5 | 63 | – | 1 | 0.1 | 57 | – | 1 | 0.3 | 3 | – | 1 | 0.02 |
| 2 | | 10 | 67 | – | 1 | 0.3 | 68 | – | 1 | 0.4 | 1 | – | 1 | 0.01 |
| 3 | | 15 | 68 | – | 1 | 0.3 | 136 | 3.322 | 7 | 0.6 | 1 | – | 1 | 0.01 |
| 4 | 50 | 5 | 103 | – | 1 | 0.5 | 206 | 3.160 | 47 | 6.7 | 5 | – | 1 | 0.04 |
| 5 | | 10 | 113 | – | 1 | 0.6 | 206 | 4.251 | | (30) | 3 | – | 1 | 0.02 |
| 6 | | 15 | 110 | – | 1 | 0.6 | | | | | 2 | 0.424 | 3 | 0.02 |
| 7 | 75 | 5 | 14 | – | 1 | 0.3 | 296 | 2.837 | 27 | 11.0 | 6 | – | 1 | 1.53 |
| 8 | | 10 | 159 | – | 1 | 1.2 | 296 | 6.008 | | (60) | 5 | 0.251 | 5 | 0.21 |
| 9 | | 15 | 153 | – | 1 | 1.6 | | | | | 9 | – | 3 | 0.35 |
| 10 | | 25 | 172 | – | 1 | 2.2 | | | | | | | | * |
| 11 | 100 | 5 | 215 | – | 1 | 3.2 | 406 | 7.396 | | (60) | 4 | 0.090 | 3 | 0.78 |
| 12 | | 10 | 247 | – | 1 | 3.6 | | | | | 3 | – | 1 | 0.60 |
| 13 | | 15 | 294 | – | 1 | 4.3 | | | | | | | | * |
| 14 | | 33 | 222 | – | 1 | 4.2 | | | | | | | | * |
| 15 | 150 | 5 | 581 | 0.594 | 25 | 10.2 | | | | | 1 | 1.105 | 67 | 7.83 |
| 16 | | 10 | 581 | 0.275 | 89 | 27.8 | | | | | 2 | 1.088 | 27 | 4.48 |
| 17 | | 15 | 451 | – | 1 | 11.0 | | | | | 7 | 0.276 | 11 | 4.32 |
| 18 | | 50 | 406 | – | 1 | 16.1 | | | | | | | | * |
| 19 | 200 | 5 | 796 | 0.709 | 67 | 43.9 | | | | | 2 | 1.081 | 81 | 45.10 |

A time in brackets indicates that the algorithm did not finish in the time shown.
* means that no solution with this value of $p$ could be found.

compiler), for a number of randomly generated problems. For all of these problems the randomly generated allocation cost matrix $(d_{ij})$ was subjected to Floyd's [4] algorithm to ensure that the triangularity condition was satisfied.

Other authors have indicated that the $p$-median problem is hardest to solve for $p = |V|/3$ so for the larger problems with $|V| = 75$, 100, 150 we solved $p = 25$, 33 and 50 respectively.

Table 3 shows the results for the problems using both relaxations LR1 and LR2. The duality gap at the initial tree node is measured by

$$\frac{(\text{optimal value} - \text{lower bound at initial node})}{(\text{optimal value})} \times 100\%.$$

It can be seen from that table that relaxation LR1 is superior to relaxation LR2 and that for only 3 of the 19 problems considered was any branching required with relaxation LR1. The time quoted excludes the set-up time needed to generate the allocation cost matrix.

Figure 1 shows the tree search for problem 15 (relaxation LR1). A number $x$ ($\bar{x}$) inside the circle representing a node implies that vertex $x$ is chosen to be (not to be) a median. The number beside each node is the value of the lower bound at the end of the lagrangean ascent for that node. The tree search was generated in a depth-first manner. The letters next to the nodes indicate the order in which they were generated (A first, B second, etc.).

The difficulty in using the Erlenkotter [3] algorithm for the $p$-median problem lies in choosing the fixed cost to be associated with each median vertex, as the Erlenkotter algorithm balances the fixed costs of the median vertices against the variable costs of allocation in choosing an optimal solution. A fixed cost must be chosen that will produce exactly $p$ median vertices in the optimal solution (note that there may not exist a set of fixed costs which will give exactly $p$ median vertices in the optimal solution). Consequently in the results for the Erlenkotter algorithm shown in Table 3 we present the number of cost iterations we required before finding a fixed cost that gave exactly $p$ median vertices and the time quoted is the time over all cost iterations (again this time excludes the set-up time needed to create a sorted cost matrix). For 4 of the problems no integer fixed costs could be found that would lead to the appropriate value of $p$. The computational strategy used in the dual ascent phase at the initial tree node and in the tree search was identical to the strategy adopted by Erlenkotter in reporting his results.

It can be seen from Table 3 that relaxation LR1 always produced a lower duality gap at the initial tree node than either LR2 or Erlenkotter's algorithm. For the problems with $|V| = 30$ the Erlenkotter algorithm is an order of magnitude faster than relaxation LR1 but for the larger problems the two algorithms become competitive.

## References


[1] G. Cornuejols, M.L. Fisher and G.L. Nemhauser, Location of bank accounts to optimize float: An analytical study of exact and approximate algorithms, *Management Sci. 23* (1977) 789–810.

[2] M.A. Efroymson and T.L. Ray, A branch-bound algorithm for plant location, *Operations Res. 14* (1966) 361–368.

[3] D. Erlenkotter, A dual-based procedure for uncapacitated facility location, *Operations Res. 26* (1978) 992–1009.

[4] R.W. Floyd, Algorithm 97 – shortest path, *Comm. ACM 5* (1962) 345.

[5] R.S. Garfinkel, A.W. Neebe and M.R. Rao, An algorithm for the $m$-median plant location problem, *Transportation Sci. 8* (1974) 217–236.

[6] M. Held, P. Wolfe and H.P. Crowder, Validation of subgradient optimisation, *Math. Programming 6* (1974) 62–88.

[7] B.M. Khumawala, An efficient branch and bound algorithm for the warehouse location problem, *Management Sci. 18* (1972) 718–731.

[8] F.E. Maranzana, On the location of supply points to minimise transport costs, *Operational Res. Quart. 15* (1964) 261–270.

[9] S.C. Narula, U.I. Ogbu and H.M. Samuelson, An algorithm for the $p$-median problem, *Operations Res. 25* (1977) 709–713.

[10] R.M. Nauss, An improved algorithm for the capacitated facility location problem *J. Operational Res. Soc. 29* (1978) 1195–1201.

[11] A.W. Neebe and M.R. Rao, A subgradient approach to the $m$-median problem, Technical report no. 75-12, University of North Carolina at Chapel Hill (1975).

[12] A.W. Neebe, A branch and bound algorithm for the $p$-median transportation problem, *J. Operational Res Soc. 29* (1978) 989–995.

[13] R.E. Rosing and C.S. Revelle, A method for optimal solutions to large $p$-median problems, Economic Geography Institute, Erasmus University, Rotterdam (1978).

[14] L. Schrage, Implicit representation of variable upper bounds in linear programming, *Math. Programming Study 4* (1975) 118–132.

[15] K. Spielberg, Algorithms for the simple plant-location problem with some side conditions, *Operations Res. 17* (1969) 85–111.

[16] M.B. Teitz and P. Bart, Heuristic methods for estimating the generalised vertex median of a weighted graph, *Operations Res. 16* (1968) 955–961.

[17] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, (Freeman, San Francisco, 1979) 220.