Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 417–426
computers & operations research
www.elsevier.com/locate/cor

# BEAMR: An exact and approximate model for the $p$-median problem

Richard L. Church*

*Department of Geography and The National Center for Geographic Information and Analysis, University of California at Santa Barbara, Santa Barbara, CA 93106-4060, USA*

Available online 5 May 2006

**Abstract**

The $p$-median problem is perhaps one of the most well-known location–allocation models in the location science literature. It was originally defined by Hakimi in 1964 and 1965 and involves the location of $p$ facilities on a network in such a manner that the total weighted distance of serving all demand is minimized. This problem has since been the subject of considerable research involving the development of specialized solution approaches as well as the development of many different types of extended model formats. One element of past research that has remained almost constant is the original ReVelle–Swain formulation [ReVelle CS, Swain R. Central facilities location. Geographical Analysis 1970;2:30–42]. With few exceptions as detailed in the paper, virtually no new formulations have been proposed for general use in solving the classic $p$-median problem. This paper proposes a new model formulation for the $p$-median problem that contains both exact and approximate features. This new $p$-median formulation is called Both Exact and Approximate Model Representation (BEAMR). We show that BEAMR can result in a substantially smaller integer-linear formulation for a given application of the $p$-median problem and can be used to solve for either an exact optimum or a bounded, close to optimal solution. We also present a methodological framework in which the BEAMR model can be used. Computational results for problems found in the OR_library of Beasley [A note on solving large $p$-median problems. European Journal of Operational Research 1985;21:270–3] indicate that BEAMR not only extends the application frontier for the $p$-median problem using general-purpose software, but for many problems represents an efficient, competitive solution approach.


*Keywords:* $p$-median; Integer programming; Variable reduction; Approximation approach

## 1. Introduction

Given a connected network of nodes and arcs, where each node represents some non-negative amount of demand and where each arc has an associated distance, the $p$-median problem involves selecting $p$ locations on the network, either at nodes or along arcs, so as to minimize total weighted distance. Weighted distance for a specific node given a facility set is defined as the amount of demand (i.e. demand weight) times the distance of the shortest path from that node to the closest facility. Total weighted distance is defined as the sum of all weighted distances associated with each demand node assigning to its closest facility. This problem was originally defined within the context of a network by Hakimi [1] for the expressed purpose of locating communication-switching centers on a network. Hakimi [1] proved

* Tel.: +1 805 893 4217; fax: +1 805 893 3146.
*E-mail address:* church@geog.ucsb.edu.

0305-0548/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2006.03.006

that there would always be a nodal location that was optimal to the network 1-median problem. Hakimi [2] further proved that there would always be at least one optimal solution to the network $p$-median problem that was comprised entirely of nodes. Teitz and Bart [3] presented the first heuristic solution procedure for the network $p$-median problem that involved searching for the best solution among the nodes of a network. ReVelle and Swain [4] presented the first integer-linear programming (ILP) formulation for the $p$-median problem in *Geographical Analysis*. Kariv and Hakimi [5] demonstrated that the $p$-median problem belongs to the class of problems defined as NP-hard. Over the past 30 years the $p$-median location model has been applied in a number of different problem settings.

More than thirty years have passed since ReVelle and Swain presented a formal mathematical formulation for this central facilities location problem. During these three decades the original ReVelle and Swain formulation has played a key role in solving the $p$-median problem using exact procedures like linear programming with branch and bound and bounded procedures like Lagrangean relaxation [6], in developing extended model forms of the $p$-median problem (e.g. [7]), and model development in related location problems. There are several reasons why this model structure has survived the test of time. This will be discussed in detail in the next section of this paper, when the original ILP model is presented.

The original model defined on an $n$-node network contains $n^2$ variables and $n^2 + 1$ constraints. For a 500 node problem, this adds up to 250,000 variables and 250,001 constraints. With few exceptions, most of the work after ReVelle and Swain [4] has concentrated on the development of special purpose solution methods, instead of the direct use of general-purpose software. The exceptions are few, because of the practical limit in using general-purpose software applied to a problem that has so many variables and constraints [8–11]. Further, given that the $p$-median problem is a member of the class of NP-hard problems, there exist problem instances that will not be solvable to provable optimality, regardless of approach. Unless there is a way in which a given problem can be formulated with significantly fewer variables and constraints than required by the ReVelle Swain model, use of general-purpose software will always be limited to small sized problems.

The objective of this paper is to propose a new method for formulating the $p$-median problem. This new $p$-median formulation is called the both exact/approximate model representation (BEAMR). BEAMR can result in a substantially smaller integer-linear formulation for a given application of the $p$-median problem and can be used to solve for either an exact optimum or a bounded, close to optimal solution. The impact of this approach will be to effectively increase the size of $p$-median problems that are solvable using off the shelf, general-purpose software. We will show that not only are some types of large problems solvable using commercial general-purpose software, but that the solution times are competitive with other state-of-the-art $p$-median solution procedures. Our tests are based on problems involving up to 900 nodes, found in the OR_library of Beasley [12]. The next two sections introduce the original ReVelle–Swain formulation along with the past refinements that have been developed to pare down problem size. We then present the BEAMR model and follow that with a description of a methodological framework in which the BEAMR model can be solved. This paper concludes with computational experience and a summary including suggestions for further research.

## 2. ReVelle and Swain's original model for the $p$-median problem

Consider a connected network comprised of $n$ nodes. For the rest of this paper we will consider the most general form of the $p$-median problem where each node represents both a place of demand and a potential facility site. The $p$-median model involves the location of $p$ facilities on the network so that total weighted distance is minimized. It is assumed that each demand is served by their closest facility. Given that Hakimi [2] proved that there is always at least one optimal solution to a $p$-median problem that consists entirely of nodes of a network, virtually all solution methods have been based on the search for the best solution comprised entirely of nodes.

Given any pair of nodes on the network, a shortest path exists between the pair of nodes, and the distance of that path is easily calculated by a variety of efficient shortest path algorithms [13]. Consider the following notation:

$i, j$ indices used to refer to a node, numbered as $1, 2, \ldots, n$,
$d_{ij}$ shortest distance from node $i$ to node $j$,
$a_i$ demand at node $i$,
$x_{ij}$ $\begin{cases} 1 & \text{if demand at } i \text{ assigns to facility at } j, \\ 0 & \text{otherwise,} \end{cases}$

$x_{jj}$ $\begin{cases} 1 & \text{if a facility is sited at site } j \text{ and demand at } j \text{ assigns to it as well,} \\ 0 & \text{otherwise,} \end{cases}$

$p$ the number of facilities that are to be located.

Using the notation defined above ReVelle and Swain [4] formulated the $p$-median problem as an ILP problem:

$$\text{Min } Z = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i d_{ij} x_{ij}, \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for each } i = 1, 2, \ldots, n, \tag{2}$$

$$\sum_{j=1}^{n} x_{jj} = p, \tag{3}$$

$$x_{ij} \leqslant x_{jj} \quad \text{for each } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, n \text{ where } i \neq j, \tag{4}$$

$$x_{ij} = 0, 1 \quad \text{for each } i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, n. \tag{5}$$

We will refer to this formulation as the RS model. The objective seeks to minimize the total weighted distance of all demand assignments. Constraints of type 2 ensure that each demand must assign exactly once. The fourth type of constraint maintains that a demand cannot assign to a node unless that node has been selected for a facility. Since the objective minimizes the distances of assignments, constraints of type 2 and 4 in concert with the objective ensure that at optimality, each demand assigns to their closest located facility. The third constraint fixes the number of located facilities to be equal to $p$. The integer restrictions on the variables are established in constraints of type 5. Note that in solving this model, it is only necessary to require the $x_{jj}$ variables to be zero-one. The above model can be solved directly using commercially available software that employ linear programming with branch and bound (LP/BB) techniques. In a relatively high percentage of problems, the optimal linear programming solution is integer optimal as well and a branch and bound routine is not needed. This property is called "integer friendly" [14]. Since branch and bound algorithms may take considerable computational time to resolve non-integer solutions, finding an optimal integer solution without branch and bound usually means relatively fast computational times for a given problem size.

There is a practical limit as to the size of $p$-median problem that can be solved using general-purpose software applied to the above formulation. This limit is based on the fact that the above formulation involves $n^2$ variables and $n^2 + 1$ constraints. From the outset, this limitation has encouraged the development of alternate solution procedures. The development of solution methods can be roughly classified into two general approaches: (1) heuristics and meta-heuristics; and (2) techniques based upon taking advantage of the structure of the above model formulation and providing bounded, if not optimal solutions. Heuristic-based approaches include: greedy [15], vertex substitution [3], Simulated annealing [16], genetic [17], Tabu search [18], variable neighborhood search [19] and grasp with path relinking [20], just to name a few. The second approach primarily involves the use of Lagrangean relaxation and dual ascent methods and includes the work of Narula et al. [6], Beasley [12,21], Hanjoul and Peters [22], Erlenkotter [23], and many others.

## 3. Model refinements

Four different research papers have discussed the possibility of trimming the size of the RS model. First, Rosing et al. [10] proposed using the following type of constraints to prevent demand assignments only to those nodes selected for a facility:

$$\sum_{i=1}^{n} x_{ij} \leqslant n x_{jj} \quad \text{for each } j = 1, 2, \ldots, n. \tag{6}$$

The constraints which restrict demand assignments to only those nodes selected for a facility have been aggregated into one constraint for each facility. By replacing the entire set of constraints (4) in the RS model with constraints (6), it is possible to formulate the $p$-median problem using only $2n + 1$ constraints rather than $n^2 + 1$ constraints. As the above constraints lack the integer friendly properties of the Balinski [24] style constraints (4), Rosing et al. [10]

reasoned that keeping constraints of type (4) for all demand assignment variables associated with the $k$-closest facilities to a given demand would help preserve the integer friendly nature found in the original model. Therefore, the number of constraints are reduced to $2n + kn + 1$ constraints from $n^2 + 1$ constraints. Rosing et al. [10] also eliminated the assignment variables associated with the $p - 1$ furthest assignments of a given demand, thereby reducing the number of variables to $n^2 - pn + n$ variables. Although there is a substantial reduction in constraints in the hybrid model, there is only a modest reduction in variables for most applications.

The second approach to "downsizing" the formulation was suggested by Hillsman [8]. Hillsman suggested two procedures that could be used to eliminate variables. If one or more facility locations already exist and are to be fixed into the model solution, then assignment variables associated with a specific demand that involve facility assignments further than the closest of these "fixed" facilities from that demand can be eliminated. Hillsman called this process "retrenchment." Hillsman [8] also suggested using a distance cutoff to help downsize a model. He argued that in many applications, there would be a service distance beyond which a solution would not be politically acceptable. Using this distance as a maximum assignment distance allows the elimination of variables associated with demand-facility assignments that exceed this distance cutoff. Such an approach would not, however, guarantee finding the optimal unconstrained $p$-median solution. Large-scale applications of this approach are used by Densham and Rushton [25,26]. Sorensen and Church [27] show how this approach can be used with some degree of confidence, but not complete certainty.

The third approach taken to downsize the $p$-median formulation is the heuristic concentration method of Rosing and Revelle [9]. Rosing and ReVelle argued that if one could reduce the possible site set to a fraction of the nodes on the network, then the resulting ILP model size would be considerably smaller. They suggested that such a reduced site set could be identified by repeatedly running a heuristic on a problem. They reasoned that the sites used in the solutions of the heuristic should, in general, be superior to the ones not used and could be used to form what they called a "concentration set." They used the concentration set as the set of possible sites, instead of including each and every node. They also identified cases where some sites were found in all heuristic generated solutions. They called such sites, core sites. They argued that core sites, by their very nature should be fixed into a solution which allowed them to use the approach of retrenchment suggested by Hillsman [8]. Although this approach does find many optimal solutions, it is a heuristic in that there is no guarantee that an optimal solution exists, which consists entirely of sites from the concentration set or which must use all of the identified "core" sites.

The fourth approach is the COBRA model of Church [11]. The COBRA model is based upon the existence of assignment variables that are equivalent. Church presented a methodology in which these redundant assignment variables can be identified in advance and combined without any loss of generality, resulting in problems with up to 80% fewer variables. This technique was found to be quite powerful in model size reduction, when the number of sites was a fraction of the number of demand nodes.

## 4. A new model: BEAMR

Suppose that we know in advance that any competitive solution (i.e. optimal or near optimal solution) to a given $p$-median problem involves a specific demand node g assigning to a facility site no further than its $h$th closest facility. If we knew this for certain, we could modify the model before solving it, whereby assignment variables associated with node $g$ are eliminated when they involve an assignment beyond the $h$th closest facility from $g$. Now, we do not know in general, such a certainty. But we can test for such a possibility by considering the following model construct:

$$\text{Min } Z = \sum_{i=1}^{n} \sum_{j \in H_i} a_i d_{i,j} x_{i,j} + \sum_{i=1}^{n} a_i d_{i,s_i} f_i, \tag{7}$$

$$\sum_{j \in H_i}^{n} x_{ij} + f_i = 1 \quad \text{for each } i = 1, 2, \ldots, n, \tag{8}$$

$$\sum_{j=1}^{n} x_{jj} = p, \tag{9}$$

$$x_{ij} \leqslant x_{jj} \quad \text{for each } i = 1, 2, \ldots, n \text{ and } j \in H_i \text{ and where } i \neq j, \tag{10}$$

$$x_{ij} = 0, 1 \quad \text{for each } i = 1, 2, \ldots, n \text{ and } j \in H_i, \tag{11}$$

where the additional notation is defined below:

$H_i$ the set of the $h_i$ closest sites to demand $i$,
$s_i$ the index of the $h_i + 1$ closest site to demand $i$,
$f_i$ $\begin{cases} 1 & \text{if demand } i \text{ must assign to a facility further than its } h_i \text{ closest facility,} \\ 0 & \text{otherwise.} \end{cases}$

The above model is similar to the RS model except that it has fewer assignment variables for each demand $i$ and fewer Balinski-type constraints (10). Basically, any assignment variable that involves a facility further than the $h_i$ closest facilities to a given demand $i$ has been eliminated along with its associated Balinski-type constraint (10). In its place is a variable, $f_i$, that represents the case that demand node $i$ is not served by a facility that is within the set of $h_i$ closest facilities to $i$. The distance of this assignment is set equal to the distance from demand node $i$ to the $h_i + 1$ closest facility. What the model allows for is an assignment beyond the set of $h_i$ closest facilities from node $i$ without restriction. Nothing restricts such an assignment to be made (i.e. $f_i = 1$), except that a closer assignment is made (i.e. $x_{ij} = 1$ where $j \in H_i$). The underlying idea is that either an assignment is made to one of $h_i$ closest sites or a cost of assigning to the $h_i + 1$ closest site is incurred, regardless as to whether the $h_i + 1$ closest site has been selected for a facility. This means that the cost of facility assignment is exact for the $h_i$ closest facilities or is estimated with a lower bound for any assignment beyond the $h_i$ closest facility. Essentially, the optimal solution to the above model will yield a valid lower bound on any feasible (or optimal) solution to the associated $p$-median problem, regardless of the values of $h_i$. However, if all $h_i$ values were set to a value of $n - 1$, then the model is exactly the same as the original ReVelle–Swain model.

Not only will the above model produce valid bounds on any optimal $p$-median solution for a given problem, it could also be an optimal solution as well. Consider the case when an optimal solution to the above formulation for a set of $h_i$ values where all $f_i$ variables are zero in value. Such a solution is not only a valid lower bound but it is an exact, optimal solution as none of the variables involve an assignment that represents a lower bound. Thus, solutions to the above model are optimal $p$-median solutions whenever all $f_i$ variables are zero in value. Therefore, this model is both an exact and approximate model representation for the $p$-median problem depending upon the values of the $f_i$ variables in the optimal solution.

It is important to point out that an optimal solution to BEAMR could be an optimal $p$-median solution even when some of the $f_i$ variables are positive. If, for any positive $f_i$ variables, the $h_i + 1$ closest facility site has been selected for a facility, then the model is also an exact, optimal solution to the $p$-median problem.

It is also important to consider the fact that when the BEAMR model does result in a solution where some $f_i$ variables are 1 in value, one can easily compute the true weighted distance associated with the selected set of $p$ facilities. If the weighted distance is within a small tolerance of the valid bound produced by the model, then the approximated model has produced a solution within a small known tolerance of optimality.

The value in BEAMR is threefold:

(1) with BEAMR, it might be possible to solve for exact, optimal solutions to general $p$-median problems without fully specifying all of the assignment variables and constraints needed in the RS model;
(2) any solution to BEAMR is a valid lower bound to the $p$-median problem and when some $f_i$ values are positive, the solution can be used to test how close to optimality the solution obtained is from optimality; and
(3) BEAMR may extend the frontier of $p$-median problems that can be optimally solved or solved within proven bounds using off-the-shelf general-purpose ILP software.

Some might argue in regards to (1) above that the Hillsman distance cutoff approach can also generate optimal solutions to a $p$-median problem efficiently. The major distinction here is that cutting assignments variables associated with long distances alone gives no guarantee of optimality, just ease of solving. Because BEAMR represents all assignments either explicitly or implicitly, the results of BEAMR can be interpreted in terms of a valid bound on optimality as well as whether an optimal solution has indeed been identified. The same argument is true within the context of point (3) above and Hillsman's approach to distance cutoff. The cutoff does make larger problems easier to solve, but not necessarily to optimality or within proven bounds.

To make BEAMR a valuable model it is necessary to have a workable methodology for estimating good values of $h_i$, small enough to keep the resulting model small but large enough that the solution is an exact $p$-median solution.

In the next section, we discuss two such approaches and then in the following section, we present some computational experience with BEAMR solving a set of problems found in the OR_library [12].

## 5. Methodological framework for using the BEAMR model

The size of a BEAMR model applied to a specific $p$-median problem is directly related to the values of $h_i$ . Essentially, the number of variables and constraints is approximately:

$$\sum_{i=1}^{n} h_i.$$

There are two relatively straightforward approaches for estimating the needed values for $h_i$ so that BEAMR produces an exact, optimal solution (or close to optimal within some desired tolerance gap). The first approach is called Process A and is defined as follows:

### Process A

*step* 1: Define $G =$ maximal acceptable convergence gap.
Define $\Delta =$ the increment size for modifying $h_i$ values.
Begin with an initial estimate of $h_i$ for each demand $i$.
*step* 2: Solve BEAMR using LP/BB software where $h_i$ values are as currently defined, resulting in objective $\hat{Z}$.
Define $\Gamma = \{i \mid f_i > 0\}$ for this BEAMR solution.
*step* 3a: If $\Gamma \neq \emptyset$, then objective $\hat{Z}$ is a valid lower bound on an optimal $p$-median solution.
Define $\bar{d_i} =$ the distance from $i$ to the closest open facility in the current BEAMR solution and calculate the actual weighted distance for this solution,

$$WD = \hat{Z} + \sum_{i \in \Gamma} a_i(\bar{d_i} - d_{i,s_i}).$$

If $WD - \hat{z}/\hat{z} > G$, then solution has not converged within an acceptable tolerance so go to step 4; otherwise, stop as the current BEAMR solution is within a maximal acceptable tolerance of optimality for the $p$-median problem.
*step* 3b: If $\Gamma = \emptyset$, then stop, the current BEAMR solution is an optimal $p$-median solution.
*step* 4: Increase $h_i$ by $\Delta$ for each $i \in \Gamma$ and return to step 2.

Process A begins with an initial estimate of $h_i$ values and then uses the BEAMR model to generate an optimal solution given those values. If the BEAMR solution is an optimal solution to the associated $p$-median problem, then the process stops. If some $f_i$ values are positive, then either the solution is close enough to optimality to stop, or the $h_i$ values are increased (by $\Delta$) for each positive $f_i$ variable and a revised BEAMR model is solved. This process is repeated until it converges. The only problem with this approach is that it might require repeated application of the BEAMR model for a given problem. Clearly this might not be a very efficient approach as it relies on repeated use of LP/BB software, unless one started with high estimates of $h_i$ and a high value of $\Delta$. Obviously, the power of the BEAMR model rests in being able to start with good estimates of $h_i$. Hence, the reason for Process B:

### Process B

*step* 1: Define $\Delta =$ the increment size for modifying $h_i$ values.
Begin with an initial estimate of $h_i$ for each demand $i$.
*step* 2: Solve BEAMR using a heuristic like the Teitz and Bart heuristic where $h_i$ values are as currently defined. When solving the heuristic, assignments for demand $i$ for sites that are further than the $h_i$ closest sites to $i$ have a set weighted distance of $a_i d_{i,s_i}$. For the resulting solution, define $\Gamma = \{i \mid f_i > 0\}$. $\Gamma$ is the set of demands $i$ where each demand $i$ is served beyond its $h_i$ closest sites.
*step* 3a: If $\Gamma \neq \emptyset$, then increase $h_i$ by $\Delta$ for each $i \in \Gamma$ and return to step 2.
*step* 3b: If $\Gamma = \emptyset$, then proceed to Process A with heuristically estimated values of $h_i$.

Table 1
Comparing running times for a set of problems reported in Table 2 of Senne et al. [28]

| $N$ | $p$ | Solution | Lagrangean[a] Run time | Lagrangean surrogate[a] Branch and Price Run time | BEAMR Process B heuristic[b] Run time (number of heuristic starts) | BEAMR model solved by CPLEX[b] Run time |
|---|---|---|---|---|---|---|
| 100 | 5 | 5819 | 12.98 | 13.61 | .15 (15) | .64 |
| 100 | 10 | 4093 | 52.57 | 42.80 | .11 (12) | .76 |
| 100 | 10 | 4250 | 78.65 | 84.30 | .21 (14) | .85 |
| 200 | 10 | 5631 | 123.56 | 94.75 | .63 (14) | 2.20 |
| 300 | 30 | 4374 | 458.56 | 601.73 | 2.03 (18) | 1.72 |
| 400 | 40 | 4809 | 21694.66 | 1742.25 | 3.17 (14) | 5.30 |
| 500 | 50 | 4619 | 19550.80 | 1921.23 | 3.14 (38) | 5.73 |

[a]Results reported by Senne et al. [28] on a 1.1 GHz Pentium 4 Intel Computer.
[b]Results on a 467 MHz Sun SPARC 10 workstation using Solaris 7.03 operating system.

The basic idea of Process B is to use a heuristic to test values of $h_i$, to see if assignments beyond the $h_i$ closest facility occurs. Since the speed of a heuristic like Teitz and Bart in solving a $p$-median structured problem is quite fast, one can easily test and search for good values of $h_i$ before using the general-purpose LP/BB software. It is easy to apply any $p$-median heuristic to test a BEAMR model as a BEAMR model can be set up by adjusting the values of the weighted distance matrix that is to be used by the heuristic. The adjustments are simple in that any weighted distance value associated with demand $i$ and a facility further than the $h_i$ closest sites to $i$ is set to $a_i d_{i,s_i}$.

Process B can also be defined where the heuristic is repeated $r$ times in step 2, to test if any heuristic solution generated within the repeated runs results in assignments beyond the closest $h_i$ sites for a given demand $i$. Once Process B has been completed for a given problem, then Process A can begin with these fine tuned estimates of $h_i$. In the next section we present computational results in using BEAMR to solve $p$-median problems involving both Processes A and B.

## 6. Computational testing of the BEAMR model for optimally solving $p$-median problems

In the last section, a framework was presented in which the BEAMR model could be used to solve for $p$-median solutions within a specified gap of optimality. If we set the gap to zero, the two Processes A and B will end with an optimal solution to the associated $p$-median problem. There are limits of course to such an approach, as general purpose LP/IP software is used to solve the BEAMR model in Process A. If the values of $h_i$ are too large then the size of the BEAMR model may be too large to solve efficiently, as compared to other approaches that have been developed to solve the $p$-median model optimally (see for example, Lagrangean relaxation with subgradient optimization [12,22,28]). In this section we present results in solving the $p$-median problem using the BEAMR model integrated within Process A and B. The computational experience was generated on a Sun UltraSparc 10 workstation with a speed slightly slower than 500 MHz using the SunOS 7.1 operating system. The BEAMR model was solved heuristically in Process B by a Teitz and Bart routine coded in FORTRAN. The BEAMR model was set up by a FORTRAN code in Process A and the BEAMR model was solved optimally in Process A by Cplex version 6.6. All FORTRAN code was compiled using the GNU 77 compiler without any code optimization.

The problems that were tested using the BEAMR model approach were taken from the OR_library of $p$-median problems found at http://mscmga.ms.ic.ac.uk/jeb/orlib [12]. This problem library contains 40 different $p$-median problems. Each problem involves a network of $n$ nodes and $n$ sites and a specific value of $p$. Networks range in size from 100 to 900 nodes. Distance matrices were generated in advance of the application of the BEAMR model by a shortest path routine. Computational times reported here include the time to read each distance matrix from a file by the Process B program and by the Process A program, as these two programs are linked only by an output file generated from the Process B program. Computational times could be reduced somewhat by integrating these two processes.

Table 1 gives results recently presented by Senne et al. [28] along with results of the BEAMR model approach (see Table 2 of [28]). Senne et al. [28] compared a relatively straightforward Lagrangean relaxation approach with a newly developed "branch and price" based Lagrangean relaxation approach. Both processes employed a branch and bound

Table 2
Comparing model size of the RS formulation and the BEAMR formulation for selected OR_library problems given in Table 1

| Problem | | Classic RS *p*-median model | | BEAMR model with Processes A and B conditioning | | | |
|---|---|---|---|---|---|---|---|
| *N* | *p* | Rows | Columns | Rows | Columns | % Reduction | |
| | | | | | | Rows | Columns |
| 100 | 5 | 9601 | 9600 | 1801 | 1900 | 81 | 80 |
| 100 | 10 | 9101 | 9100 | 1301 | 1400 | 86 | 85 |
| 100 | 10 | 9101 | 9100 | 1321 | 1430 | 85 | 84 |
| 200 | 10 | 38,201 | 38,200 | 3646 | 3845 | 90 | 90 |
| 300 | 30 | 51,301 | 51,300 | 3861 | 4160 | 92 | 92 |
| 400 | 40 | 144,401 | 144,400 | 5371 | 5770 | 96 | 96 |
| 500 | 50 | 225,501 | 225,500 | 6556 | 7055 | 97 | 97 |

Size of classic RS model is $n^*(n-p+1)+1$ constraints and $n^*(n-p+1)$ variables.
Size of the BEAMR model is based upon the resulting model specification produced by Process B and implemented in Process A.

algorithm. Table 1 gives execution times associated with the Lagrangean relaxation, branch and price based Lagrangean relaxation, and BEAMR. As the times given by Senne et al. are repeated here, it is important to note that the Lagrangean results were generated on an Intel 1.1 GHz cpu, whereas the results of BEAMR were generated on a Sun Sparc chip running at half that speed.

In all instances, Process B was used to estimate values of $h_i$. For the results in this table, Process B was initiated with $h_i$ values set at 5, delta values were set at 5, and the heuristic (i.e. [3]) was stopped on the first instance of when no $f_i$ values were positive. An increment of 5 was added to all final $h_i$ values before being passed to Process A. For the results in Table 1, all sets of $h_i$ values resulted in an exact solution for the BEAMR model. Solution times then represented solving the BEAMR problem once using Cplex. This means that Process B was relatively robust at generating sets of $h_i$ values that were large enough to generate exact solutions to the associated *p*-median problems. Execution times for both Processes A and B are given in Table 1. Table 1 indicates, overall, that the branch and price Lagrangean relaxation solution process was faster than the more classic Lagrangean relaxation. Clearly, the fastest solution process was the BEAMR model using Cplex and the Teitz and Bart heuristic. All solutions generated by the BEAMR model were optimal. Given that the BEAMR results were generated on a machine with a clock speed less than half of the Lagrangean results, it is easy to see that the BEAMR model clearly outperformed the Lagrangean approach for this set of problems and results presented by Senne et al. [28].

The reason why the BEAMR approach outperformed the two Lagrangean approaches is that the resulting LP/IP models were quite small in comparison to the RS model formulation. That is, many of the variables in the RS model formulation are not necessary when actually solving a problem. The results of BEAMR clearly demonstrate this fact. Table 2 gives a comparison of sizes (in terms of constraints and variables) for this small set of problems between the original RS model formulation and the BEAMR model application. The problem sizes of the RS model do not include the furthest $p-1$ assignment variables for a given demand as well as any Balinski constraints associated with these chopped variables. Table 2 shows that the BEAMR model can be successful at solving an exact *p*-median problem with a problem size that ranged from 80% to 97% smaller than the "chopped" RS model. It is important to note that for a given network size *n*, the BEAMR model size tends to decrease as *p* increases.

Table 3 presents BEAMR results for all of the problems found in the OR_library. For each problem, BEAMR model sizes as determined by Processes A and B are presented. The majority of problems were relatively easy to solve via the BEAMR model, however, there are some notable exceptions. As long as *p* was greater than 10, the BEAMR model resulted in relatively short computation times. For such problems sizes, the BEAMR approach might well outperform most other optimal, branch and bound based solution approaches. For those cases where *p* was less than or equal to 10, computational times were larger. It is easy to reason that when *p* is low in value, the number of assignment variables needed for each demand node for exact specification should be larger than when the value of *p* is larger.

Table 3 demonstrates that all of the problems of the OR_library can be solved using BEAMR, off the shelf LP/BB software, and Processes A and B. This approach could be used as one standard in which to compare the efficacy of other solution approaches. Second, the results of Table 3 also demonstrate that problems of 900 nodes are not out of reach of general purpose software. Overall, this demonstrates the validity of testing off the shelf software in solving

Table 3
Solutions to all OR_library problems of Beasley [11] using BEAMR

| Problem name | N | P | Objective value | Number of constraints | Number of variables | Iterations (branches) | Run time |
|---|---|---|---|---|---|---|---|
| *BEAMR model results* | | | | | | | |
| Pmed1.txt | 100 | 5 | 5819 | 1801 | 1900 | 1133(0) | .64 |
| Pmed2.txt | 100 | 10 | 4093 | 1301 | 1400 | 748(2) | .76 |
| Pmed3.txt | 100 | 10 | 4250 | 1321 | 1430 | 728(2) | .85 |
| Pmed4.txt | 100 | 20 | 3034 | 1101 | 1200 | 274(0) | .12 |
| Pmed5.txt | 100 | 33 | 1355 | 1101 | 1200 | 237(0) | .11 |
| Pmed6.txt | 200 | 5 | 7824 | 5851 | 6050 | 7395(6) | 19.98 |
| Pmed7.txt | 200 | 10 | 5631 | 3646 | 3845 | 2443(0) | 2.20 |
| Pmed8.txt | 200 | 20 | 4445 | 2921 | 3120 | 1288(0) | .86 |
| Pmed9.txt | 200 | 40 | 2734 | 2211 | 2410 | 751(0) | .39 |
| Pmed10.txt | 200 | 67 | 1255 | 2201 | 2400 | 408(0) | .30 |
| Pmed11.txt | 300 | 5 | 7696 | 7936 | 8235 | 10,050(2) | 28.84 |
| Pmed12.txt | 300 | 10 | 6634 | 6761 | 7060 | 7845(4) | 21.98 |
| Pmed13.txt | 300 | 30 | 4374 | 3861 | 4160 | 1900(0) | 1.72 |
| Pmed14.txt | 300 | 60 | 2968 | 3336 | 3635 | 1230(1) | 1.99 |
| Pmed15.txt | 300 | 100 | 1729 | 3311 | 3610 | 741(0) | .78 |
| Pmed16.txt | 400 | 5 | 8162 | 15,071 | 1547 | 43,655(49) | 299.38 |
| Pmed17.txt | 400 | 10 | 6999 | 10,461 | 1086 | 25,059(42) | 123.08 |
| Pmed18.txt | 400 | 40 | 4809 | 5371 | 5770 | 4000(3) | 5.30 |
| Pmed19.txt | 400 | 80 | 2845 | 4491 | 4890 | 1445(0) | 1.65 |
| Pmed20.txt | 400 | 133 | 1789 | 4406 | 4805 | 902(0) | 1.47 |
| Pmed21.txt | 500 | 5 | 9138 | 19,211 | 19,710 | 21,480(0) | 100.97 |
| Pmed22.txt | 500 | 10 | 8579 | 14,931 | 15,430 | 36,768(17) | 347.96 |
| Pmed23.txt | 500 | 50 | 4619 | 6556 | 7055 | 4664(0) | 5.73 |
| Pmed24.txt | 500 | 100 | 2961 | 5606 | 6105 | 2766(0) | 2.73 |
| Pmed25.txt | 500 | 167 | 1928 | 5511 | 6010 | 1812(1) | 3.02 |
| *Pmed26.txt | 600 | 5 | 9917 | 18,597 | 53,280 | 112,037(130) | 5726.50 |
| *Pmed27.txt | 600 | 10 | 8307 | 18,588 | 40,680 | 52,866(11) | 988.35 |
| Pmed28.txt | 600 | 60 | 4498 | 8071 | 8670 | 5852(0) | 8.12 |
| Pmed29.txt | 600 | 120 | 3033 | 6771 | 7370 | 2907(0) | 2.61 |
| Pmed30.txt | 600 | 200 | 1989 | 6066 | 7205 | 1959(0) | 2.91 |
| *Pmed31.txt | 700 | 5 | 10,086 | 21,701 | 71,090 | 178,332(222) | 11958.51 |
| Pmed32.txt | 700 | 10 | 9297 | 24,111 | 24,810 | 40,392(6) | 437.51 |
| Pmed33.txt | 700 | 70 | 4700 | 9241 | 9940 | 6944(0) | 10.80 |
| Pmed34.txt | 700 | 140 | 3013 | 7931 | 8630 | 3847(0) | 5.59 |
| *Pmed35.txt | 800 | 5 | 10,401 | 24,792 | 78,415 | 415,483(356) | 28071.84 |
| *Pmed36.txt | 800 | 10 | 9933 | 34,196 | 34,995 | 2,388,770(966) | 40300.15 |
| Pmed37.txt | 800 | 80 | 5057 | 10,656 | 11,455 | 7064(2) | 10.81 |
| *Pmed38.txt | 900 | 5 | 11,060 | 27,895 | 101,790 | 1,156,801(528) | 94082.77 |
| *Pmed39.txt | 900 | 10 | 9423 | 27,891 | 74,655 | 617,290(413) | 26087.71 |
| Pmed40.txt | 900 | 90 | 5128 | 11,841 | 12,740 | 7944(0) | 10.66 |

All results unless noted with an (*): Process B set initial $h_i$ values at 5, $\Delta = 5$, and where final $h_i$ values were increased by 5 before setting up BEAMR model and applying Cplex in Process A.
*Process B set initial $h_i$ values at 15, $\Delta = 15$, and where final $h_i$ values were increased by 40. Process A utilized the hybrid of constraints (4) and (6) in order to reduce the total number of constraints. A minimum of 30 or 10% of the total variables were used in constraints of the form (4).

such problems as the $p$-median location problem. It should be noted that no attempt was made to fine tune starting parameters, i.e. $h_i = 5$ and $\Delta = 5$. The starting rules reported here were the first ones tested, with the noted exceptions for a few large problems which were started with $h_i = 15$. Further efficiencies in computational time and model size might be gained by thoroughly testing starting parameters.

## 7. Summary and conclusions

This paper has presented a new formulation for the $p$-median problem, which allows much larger problems to be solved by general-purpose ILP software than what is possible using previous model formulations. This model, called

BEAMR, allows for the elimination of many of the variables and constraints that are found in the original formulation of the ReVelle and Swain [4] model formulation. The model serves as both an exact model form as well as an approximate model form, depending upon the extent of the variable and constraint reductions. For any application and model size reduction, the results of BEAMR can be easily interpreted as to the nature of the solution: i.e. exact-optimal or a known approximation bound and measure of optimality gap. Further research is needed to test for the best starting values of parameters as well as test the BEAMR model for its use in solving for approximate, bounded solutions. It is possible that solutions within a tight bound from optimality could be generated with model sizes that are even smaller than what is presented here.

We also presented a methodological framework in which the BEAMR model can be applied to solve for exact-optimal solutions to the $p$-median problem. The BEAMR framework is designed to identify the extent to which variable and constraint reductions can be made before solving with general-purpose software. This means that the framework allows for an estimate of problem solving difficulty, before actual model application. Results show that all of the OR_library problems of Beasley [12] can be solved using off the shelf general-purpose ILP software and the BEAMR model.

Computational results show that the BEAMR model is particularly efficient for large values of $p$, the number of facilities. The BEAMR model solved by general-purpose software outperformed the recent results of Senne et al. [28], for problems involving large p (i.e. $p > 10$). The BEAMR model approach might also lead to reduced model sizes for other location models like the fixed charge plant location problem and the capacitated plant location problem. This new model and application framework not only extends the capability of using general purpose software to solve larger $p$-median problems, but it also makes it possible to solve smaller problems more efficiently when using general-purpose software.

## References


[1] Hakimi SL. Optimum location of switching centers and the absolute centers and medians of a graph. Operations Research 1964;12:450–9.
[2] Hakimi SL. Optimum distribution of switching centers and some graph related theoretic problems. Operations Research 1965;13:462–75.
[3] Teitz MB, Bart P. Heuristic methods for estimating the generalized vertex median of a weighted graph. Operations Research 1968;16:955–61.
[4] ReVelle CS, Swain R. Central facilities location. Geographical Analysis 1970;2:30–42.
[5] Kariv O, Hakimi SL. An algorithmic approach for network location problems II: the $p$-medians. SIAM Journal on Applied Mathematics 1979;37:539–60.
[6] Narula SC, Ogubu UI, Samuelson HM. An algorithm for the $p$-median problem. Operations Research 1977;25:709–13.
[7] Weaver JR, Church RL. A median location model with nonclosest facility service. Transportation Science 1985;19:58–74.
[8] Hillsman EL. A system for location–allocation analysis. PhD dissertation, University of Iowa, Iowa City; 1979.
[9] Rosing KE, ReVelle CS. Heuristic concentration: two stage solution concentration. European Journal of Operational Research 1997;97:75–86.
[10] Rosing KE, ReVelle CS, Rosing-Vogelaar H. The $p$-median and its linear programming relaxation: an approach to large problems. Journal of the Operational Research Society 1979;30:815–22.
[11] Church RL. COBRA: a new formulation of the classic $p$-median location problem. Annals of Operations Research 2003;122:103–20.
[12] Beasley JE. A note on solving large $p$-median problems. European Journal of Operational Research 1985;21:270–3.
[13] Evans JR, Minieka E. Optimization algorithms for networks and graphs. 2nd ed., New York: Marcel Dekker, Inc.; 1992.
[14] ReVelle CS. Facility siting and integer friendly programming. European Journal of Operational Research 1993;65:147–58.
[15] Whitaker RA. A tight bound drop exchange algorithm for solving the $p$-median problem. Environment and Planning A 1981;13:669–80.
[16] Murray AT, Church RL. Applying simulated annealing to location-planning models. Journal of Heuristics 1996;2:31–53.
[17] Hosage C, Goodchild M. Discrete space location–allocation solutions from genetic algorithms. Annals of Operations Research 1986;6:35–46.
[18] Rolland E, Schilling D, Current JR. An efficient tabu search procedure for the $p$-median problem. European Journal of Operational Research 1996;96:329–42.
[19] Hansen P, Mladenovic N. Variable neighborhood search for the $p$-median problem. Location Science 1997;5:207–26.
[20] Resende MGC, Werneck RF. A GRASP with path-relinking for the $p$-median problem. AT&T Labs Technical Report TD-5E53XL, Florham Park, NJ, 2002.
[21] Beasley JE. Lagrangean heuristics for location problems. European Journal of Operational Research 1993;65:383–99.
[22] Hanjuol P, Peters D. A comparison of two dual-based procedures for solving the $p$-median problem. European Journal of Operational Research 1985;20:387–96.
[23] Erlenkotter D. A dual based procedure for uncapacitated facility location. Operations Research 1978;26:992–1009.
[24] Balinski M. Integer programming methods, uses and computation. Management Science (Series A) 1965;12:253–313.
[25] Densham PJ, Rushton G. Strategies for solving large location–allocation problems by heuristic methods. Environment and Planning A 1992;24:289–304.
[26] Densham PJ, Rushton G. A more efficient heuristic for solving large $p$-median problems. Papers in Regional Science 1992;71:307–29.
[27] Sorensen PA, Church RL. A comparison of strategies for data storage reduction in location–allocation problems. Geographical Systems 1996;3:221–42.
[28] Senne ELF, Lorena LAN, Pereira MA. A branch and price approach to $p$-median problems. Unpublished manuscript, 2003.