Integrating Normative Location Models into GIS: Problems and Prospects with the p-median Model

by

Richard L. Church
Paul Sorensen
University of California, Santa Barbara

Technical Report 94-5

June 1994

Integrating Normative Location Models into GIS: problems and prospects with the p-median model

by

Richard L. Church
Paul Sorensen
National Center for Geographic Information and Analysis
Department of Geography
University of California at Santa Barbara
Santa Barbara, CA 93106-4060
(805) 893-4217
e-mail: church @geog.ucsb.edu

Abstract

There has been considerable interest in the development of analysis techniques for Geographical Information Systems (GIS). This includes such normative spatial models as vehicle routing models, districting and turfing models for dividing up land into territories for schools, sales/service, and voting, and location models for identifying sites or patterns of sites to provide service accessibility. The GIS system may provide a unique data base for application and analysis. This paper discusses several problems associated with the integration of a normative location model into a GIS. A number of specifics associated with the p-median model are given. These include: demand zone definition, facility site definition, and solution algorithm selection and development. Within the context of selecting a solution process, we show that such a selection is not necessarily an easy one to make. We demonstrate that the best such solution technique may have some potential drawbacks in application. We present the results of several test applications and a glimpse of the pathology of the final solutions. This discussion raises a major, but often ignored, issue involving the application of a location heuristic: namely, how often should we restart a heuristic to help ensure that the best solution found is optimal or near-optimal. We propose and test a dynamic stopping rule that can be used to help ensure that the best solution is identified. The results of this analysis can be of great, practical value in the successful application of location models within a GIS framework.

Introduction

All human activities involve the choice, either explicit or implicit, of location. Whether we attempt to describe why existing location patterns have arisen, gather and inventory location data, or attempt to search for the optimal locations for an economic function like warehousing and distribution, a geographical information system (GIS) can serve an important role. At the bare minimum, the GIS can serve as a base for storing, retrieving, and mapping spatial data. Even though many GIS systems still lack a sophisticated modeling capability to address adequately many of the needs of Location Analysts and Researchers, research and development is underway to address many of these needs. Some of these Locational modeling needs might better be addressed with a special purpose Spatial Decision Support System (SDSS). Such systems may be loosely or tightly coupled with a GIS, which can provide an efficient form for storing and retrieving spatial data and provide a mapping capability for data and final results. In a closely coupled SDSS and GIS, the applications analyst may not even recognize when such functionality is provided by the GIS or SDSS. For example, it is very easy to produce a distance string data structure for a location problem in ARC/info 7.0 (beta2). This data structure can then be read by an "outside" modeling system and solved. Importing the results into ARC/info can be easily accomplished using the solution file formats that are generated by ARC/info. Thus, the results of the "outside model" can be easily imported back into ARC/info and displayed using one of several commands like spider. The whole application can be integrated within ARC/info in a user interface using the ARC Macro Language.

For this paper, we plan to focus on several fundamental issues associated with the integration of a normative location-allocation model with a GIS, whether it is integrated directly into a GIS or whether it is a part of a coupled GIS-SDSS. Further, we will concentrate our discussion on a classical location model called the p-median problem. Many of our comments and the issues which we raise concerning the integration of the p-median model with GIS can be generalized to other problems as well. Our selection of the p-median model is based on its relative popularity as a geographical analysis tool, the development of new SDSS systems like LADSS (Densham (1992)) and VISLOC (Figueroa, Sorensen, and Church (1994)) which use the p-median model, the recent integration of the p-median model in GIS systems (like ARC/info), and the significance that this model has in the Operations Research Literature.

To begin we will present an overview of the p-median location model in the next section. We will then discuss several inherent problems when applying this type of location-allocation problem. This will lead us to reviewing how such a model might be solved computationally and what possible alternative solution approaches might be considered in the design of a GIS-location model. We will give some insight into this problem by giving extended details on the application of two such solution techniques. We will demonstrate that this selection is not an easy one to make and can have significant ramifications on implementation in a GIS or SDSS. Further, results of a test are presented which was designed to better understand the pathology of location patterns and the p-median problem. We conclude with some guide rules in the application of such heuristic techniques to solve location-allocation problems in GIS systems.

The p-Median Problem

Location Models are usually classified into two basic types: 1) those that describe existing patterns of activities, and 2) those that prescribe a configuration of one or more facilities to accomplish some type of service or activity in an efficient manner (often called normative modeling). At times, models can serve both purposes. As an example the classical Weber model has been used to describe why existing facility locations were chosen as well as to prescribe how a new location could be selected. One of the classic normative models is the p-median problem. The p-median problem involves the location of a fixed number of facilities in such a manner that the total weighted distance of all users assigning to their closest facility is minimized. By minimizing weighted distance, accessibility is maximized. As the weighted distance is minimized, so is the average distance that demand is away from the closest facility. The p-median problem was first defined by Hakimi (1964). Hakimi defined the pmedian problem on a network of nodes and arcs. Each node was considered a place of demand as well as a potential facility location. The arcs represented transportation or accessibility linkages and could be used for facility locations as well. Hakimi (1965) proved that there existed at least one optimal solution to a network p-median problem that consisted entirely of nodes of the network. Because of this result, much of the research since that time has concentrated on identifying the optimal all nodal facility configuration to a given median problem.

In 1970 ReVelle and Swain formulated the p-median problem as an integer programming model. This formulation has played a major role in the development of optimal and near optimal approaches. The model formulation is based on the following notation:

i, I = index and set of demand areas, usually nodes of a network;
j, J = index and set of facility sites, usually nodes of a network;
d_{ij} = distance or time between demand area i and site j;
p = the number of facilities to be located;
a_i = the amount of demand at node i;

$$X_{ij} = \begin{cases} 1 \text{ if demand } i \text{ assigns to facility at } j \\ 0 \text{ if not} \end{cases}$$

$$Y_j = \begin{cases} 1 \text{ if site } j \text{ is selected for a facility} \\ 0 \text{ if not} \end{cases}$$

We can formulate the P-Median Problem in the following manner:

The p-Median Problem (PMP)

Minimize $Z = \sum_i \sum_j a_i d_{ij} X_{ij}$

Subject to:

1) Each demand node must assign to a facility:

$$\sum_j X_{ij} \geq 1 \quad \text{for each } i \in I$$

2) Locate exactly p-facilities:

$$\sum_j Y_j = p$$

3) Assignment can be made only to a facility:

$$X_{ij} \leq Y_j \quad \text{for each } i \text{ and } j$$

4) Integer requirements:

$$Y_j = 0 \text{ or } 1 \quad \text{for each } j \in J$$

The above model can be used when each node is a point of demand as well as a facility site or when the set of eligible sites is restricted to some predefined discrete set of network points. The objective is cast as minimizing the total weighted distance of assignment ($x_{ij} = 1$). When demand node i assigns to a facility located at j, then $x_{ij} = 1$ and the associated weighted distance $a_i d_{ij}$ is included in the weighted distance sum. The first constraint ensures that each demand point must assign to at least one facility site. Since weighted distance will only increase with multiple assignments, this constraint will hold as an equality in any optimal solution. Constraint 3 maintains that any such assignment must be made to only those sites that have been selected for a facility. Constraint 2 restricts the placement/allocation of facilities to exactly *p*. The final constraint lists the integer requirements for the y_j variables. Since at least one optimal solution to a network problem is comprised of a set of nodes, the above formulation can be used to identify the global optimal solution for a network-based application. Some simple but effective modifications have been proposed to reduced the size of the model for a given application by Rosing, ReVelle, and Rosing-Vogelaar (1979) but unfortunately the effective limit on the use of this model with LP and branch and bound appears to be limited to problems of 150 nodes or less.

It is worthwhile to note that this model has been extended and expanded in a number of ways. These include models such as the Hierarchical Median (Narula and Ogbu (1979)), the Stochastic Median (Mirchandani and Odoni (977)), the Temporal Median (Swain (1976)), the Transportation Median (Neebe (1978), the Zonal Constrained Median (Church(1990) and Berman, Einav, and Handler (1991)) and the Location and Scheduling Median Problems (Bloxham and Church (1991)). Further, it is easy to show that a number of location models are equivalent to one or more forms of the pmedian model (Hillsman (1984), Church and ReVelle (1976), Church and Weaver (1986)). Thus, a solution methodology for the p-median problem can be used for a number of related models. In fact, Hillsman (1984) proposed a Unified Linear Model where the basic model backbone was the p-median model and where a general pmedian solution technique is used to solve the family of ULM models. This general form has been one impetus for developing effective ways to solve the p-median model.

Two basic approaches have been taken to solve the p-median model: optimal and heuristic. Virtually all of the optimal techniques require at some point the application of a branch and bound algorithm. Because of this, there is the possibility that some problem may take an inordinate amount of computation time to solve. In practice, though, the most robust of these techniques is Lagrangian Relaxation with subgradient optimization (see Narula et al.(1977)). This technique has been used to solve problems on the order of 800-900 nodes (Beasley(1993)). More work in refining this approach is needed for solving larger problems.

The second basic approach to solving the p-median problem rests on the design of some heuristic process which in application tends to yield reasonable results. Comparisons of heuristics and optimal solution procedures have been made for only moderately sized problems. It is assumed that if a technique compares favorably with known optimal results for moderately sized problems, that such behavior extends to larger problems as well. Since researchers have been unable to prove such an assertion, one must always be cautious in extrapolating solution performance. Teitz and Bart (1968) developed the first heuristic for the p-median problem. This heuristic was based on the interchange or substitution process that was developed by Shin Lin (1965) for the Travelling Salesman Problem. Essentially, the process starts with a pattern or configuration of p locations. Then the process begins by selecting a candidate (but unused) site and considers swapping this candidate for each of the current p-facility locations. If any swap is encountered that improves weighted distance, then the best of those possible p swaps is made. The process then continues by selecting another candidate site and testing swaps. When no swap between candidate and a facility location exists which can improve the objective, the heuristic stops. When Rosing, Hillsman, and Rosing-Volegaar (1979) tested the Teitz and Bart process on a 49 node problem, the heuristic found virtually every optimal p -site pattern regardless of the starting solution. In 1964, Maranzana developed a warehouse location heuristic that was based on a strategy similar to that of Cooper's Alternate approach. This approach represented an alternate strategy for the p-median problem. The Maranzana approach starts with a pattern of p -locations just as Teitz and Bart procedure. The Maranzana heuristic works with two basic steps. The first step is to assign each demand to the closest facility location, thereby creating a series of partitions or service areas around each facility. The second step of this process involves taking each partition separately and relocating the facility within the partition to the place that best serves the demand within the partition. The heuristic repeats step 1 (partitioning) and step 2 (relocates within each partition) until no changes in partitions and locations take place. In 1979, Rosing, et al compared the Teitz and Bart approach with Maranzana and found that Maranzana was less robust in identifying the optimal solution and was also influenced by the starting solution. Many interpreted these results as showing that Teitz and Bart is superior to Maranzana and that because Teitz and Bart appeared to be somewhat insensitive to starting patterns, applications of Teitz and Bart for specific problems need to be started only once. Many examples of using Teitz and Bart results where the heuristic was started only once have appeared in the literature. Most recently, Fotheringham, Curtis, and Densham (1993) restarted Teitz and Bart only once for a set of 120 different problems, when a specific problem application appeared to have stopped with a noticeably high objective value.

Since the original developments of Maranzana(1964) and Teitz and Bart(1968), a number of other heuristic procedures have been developed to solve the p -median problem. They include the following types: genetic algorithm. simulated annealing., TABU search, GRASP, and hybrids. Genetic Algorithms were first tested for the median problem and discarded by Hosage and Goodchild (1986) since the performance was slower than Teitz and Bart and solution quality was no better than Teitz and Bart. Subsequent testing has shown that the process is very sensitive to the genetic encoding technique. With a good genetic encoding, the genetic algorithm process can identify extremely good solutions if not optimal. However, solution times are usually greater than for other techniques (Bianchi and Church(1993)). Simulated Annealing was originally discarded as an approach for the median problem by Golden and Skiscism (1986). Subsequent testing has shown that Simulated Annealing can do a good job for larger p-median problems (Murray and Church(1993)). Simulated annealing has been applied with success to a number of integer programming problems. It is based on an analogy to tempering/annealing materials originally developed by Metropolis et al.(1953) and represents a stochastic process of atoms (facilities) moving (substituting candidates for facilities) about in a structure (solution) of a cooling glass or metal (landscape). Such parameters as the cooling rate and the number of simulated moves before making a temperature decrease usually need fine tuning for a given application. Tabu Search is an advanced interchange/substitution heuristic that has the capability to arrive at a local optimum, back away from a local optimum, and attempt to reoptimize using interchange (see Glover(1990)). This process can use both short term memory and long term memory: 1) short term memory (TABU List) to prevent reversing local steps and falling back into a local optimum that the process had just backed away from, and 2) to help identify either productive areas for search or to identify untried areas of search. Some applications in location modeling have been very encouraging. There are questions to resolve in terms of the size of the TABU list. Perhaps this technique is one of the best performers found to date for the p-median problem. However, the number of times to test a local minimum and back away leaves this approach always limited by a time/iterations criterion.

GRASP has been tried for location problems including Hub and median problems (Klincewicz (1991)). GRASP is a randomized greedy process where the object is to develop a solution by selecting one facility at a time unti a p -facility pattern has been generated. At each selection, the next facility is chosen randomly from a small set of the k best alternative sites that can be added to improve the current pattern. If k=l, then the GRASP procedure is equivalent to the Greedy heuristic. Beyond k=l, GRASP chooses a randomized greedy solution. By repeating this simple process a large number of times, the best of those patterns that is generated is likely to be a very good configuration. Large scale applications have not been tested to determine running parameters (e.g. the number of top best sites used for the next greedy addition, the number of times the sampling process needs to be repeated) to guarantee finding a good solution. The approach is fast, but users may not like to see a sampling of solutions that can range in quality.

A very fruitful approach is the development of hybrids, hybrids in the sense that they represent a amalgamation of two or more approaches. A recent hybrid that appears to shine is the GRIA procedure of Densham and Rushton (1992). GRIA (Global-Regional Interchange Approach) is based on two processes 1) a local/regional substitution search which is similar to the relocation step in Maranzana, and 2) A Global drop and add similar to the Shannon and Ignizio warehouse location heuristic. This heuristic will be described in more detail in a later section of this paper. The appealing aspect of this approach is that the size of the search neighborhood is small enough to keep computational work to a minimum, but large enough that the process is not trapped from getting to a good local if not global optimum.

New Problem Setting: Integrating the p-median model into a GIS

There are a number of fundamental spatial analysis and design problems that involve some form of optimization. They include: 1) vehicle routing and delivery, 2) designing districts for schools, voting, sales, and turfing, and 3) locating service, sales, manufacturing, and warehousing locations. One of the location models that has been applied in a number of different types of applications, ranging from optimizing the placement of salt piles for road de-icing to the location of libraries, is the p-median problem. If such a model is to be integrated with a GIS system like ARC/info several issues need to be addressed. These issues include the following:

- A methodology is needed to define the nodes to represent demand areas. Most urban networks are large, including thousands of nodes. At what level do we represent or aggregate demand for an application. Demand aggregation for the pmedian problem has been studied by a number of researchers (see Fotheringham et al. (1993)).

- A methodology is needed to select feasible sites. In the OR literature, it is often assumed that each network node is a potential site. This is not a realistic assumption for many applications especially involving a complete network, which is usually a part of the GIS data base. Although some work has involved reducing a potential set of sites to a smaller subset with the objective of not eliminating the potential for the optimal pattern to be identified, more work needs to focus on the issue of identifying prospective sites for a given purpose (at the land parcel level). A classic problem of this type can be found in the power plant siting literature.

- A solution methodology is needed that can solve the median problem.

- A database structure is needed to facilitate integration of the model and solution process.

It is impossible to address each of the above problems in this single paper. Our objective is to focus on the importance of selecting a solution approach to use in a GIS for solving a p-median problem. A major parameter in selecting an approach to solve p-median problems in a GIS is the maximum design size of the problems to be solved. For a median problem, we generally measure the size of a problem in terms of the number of sites and the number of demand points. If we want to solve median problems optimally, we are constrained by problem size. Even though optimal solutions have been generated to problems where the numbers of demands and numbers of sites reached 800-900, the resulting solution times are relatively large and greater than what is considered acceptable by many to be of value in a good interactive decision making environment. On the other hand, some of the problems that industry has been interested in solving have exceeded 1000 nodes. Consequently, there is a good reason why GIS companies like ESRI have selected a heuristic solution approach for their location- allocation module. If this choice has to be made, then a simple question is which heuristic should be chosen for such a location-allocation module?

Making the Choice of the Location Heuristic for a GIS

The selection of a heuristic procedure for the solution of a location problem in a GIS system must be based on a number of criteria. They include:

- Robustness of heuristic process: that is, does the process consistently result in extremely good if not optimal solutions.

- Ease of Understanding: is this process easy to describe and understand. Most users will want to have some level of understanding of how the process works as well as what may prevent the process from finding the best possible solution.

- Speed or efficiency of technique: is the solution process fast enough to solve large problems in a reasonable amount of solution time?

- Ease of development and integration: is the process easy to develop and integrate into existing data structures and software, or does another data structure need to be supported?

At this time it is reasonable to select the heuristic from the following approaches: 1) Maranzana (1964), 2) Teitz and Bart (1968), 3) Genetic, 4) GRASP, 5) Simulated Annealing, 6) TABU, and 7) GRIA (Densham and Rushton (1992). Of these methodologies, we can easily eliminate some of these prospective choices. Since the Teitz and Bart heuristic tends to outperform the Maranzana heuristic, we can easily eliminate the Maranzana from further consideration. Even though the genetic algorithm can produce extremely good results, solution times are usually much larger than other techniques. Such a process might be a candidate when computational resources are very large, but this is impractical for a general GIS solution engine. The same can be said for the simulated annealing algorithm. GRASP can be ruled out at this time due to the fact that not enough experience has been generated in location applications. Also, fine tuning may be required in using GRASP for individual problems. This is a disadvantage for users who are not willing to experiment.

Thus, the final selection must be based on selecting either Teitz and Bart, GRIA, or TABU. Even though TABU is a technique that we believe. ultimately, will be integrated into a GIS for location analysis, it might be somewhat preliminary to rely on TABU, as research is needed to fine tune this process for location models. Also, it is possible to use a TABU process that is built on either GRIA or Teitz and Bart. That is, it is logical to pick Teitz and Bart or GRIA for integration into GIS now, and then as development on TABU for solving location models is improved and fine-tuned, then make the integration. Consequently, it is understandable why companies such as ESRI have selected one of these two heuristics for their new Network module (as developed for ARC/info 7.0.). The final choice is discussed in the next section.

A Final Choice Of Location Heuristics

In the last section we reduced the choice of heuristics to selecting either Teitz and Bart or GRlA. In doing so, we must acknowledge that as some of the other processes evolve in their development (especially TABU), their use will become justified and there will be a need to integrate such processes into GIS analysis modules over time. Thus, the selection is based upon current state-of-the-art. The final choice between the two heuristics, GRIA and Teitz and Bart, is somewhat difficult to make without more details. First, Teitz and Bart is a proven approach, easy to program, relatively fast, easy to explain, and produces good results. GRIA is a little more complex as it is a hybrid approach, is faster than Teitz and Bart, has not been extensively tested, and has some intuitive drawbacks. On a first initial comparison, most would probably select Teitz and Bart. But this decision should not be made so lightly, so it is important to give a greater level of details and a comparison of these two approaches. Consider then the Teitz and Bart heuristic process:

Step 1: Start with an initial pattern of p sites selected among the n nodes. Call this the current best configuration.

Step 2: Define all candidate sites that are not used in the current best configuration as the set α, if α is *null*, then go to step 6. Otherwise, go to step 3.

Step 3: Select a candidate site $k \varepsilon \alpha$, remove site k from α and go to step 4.

Step 4: Calculate Δ_j for $j = 1, 2, ...p$ as the change in the objective function associated with moving the j^{th} facility to candidate node k.

Step 5: Define $\Delta_{\min} = \frac{\min}{j} \Delta_j$. If $\Delta_{\min} \geq 0$, go to step 6. Otherwise, create a new current configuration by replacing the location associated with the minimum Δ_j with site k and go to step 6.

Step 6: If set α is *null* and configuration changes have been made since α was last defined, go to step 2: If set α is *null* and no changes in configuration have been made since α was defined, stop, current solution is locally optimal. If α is not the *null* set, then go to step 3:

The process of search for a stable local if not global optimum is based on the local search neighborhood. The local search neighborhood is defined by the number of possible substitutions that could be tested without finding an improved solution. That is, if no suitable exchange can be made within the local search neighborhood, the configuration is locally optimal, and the Teitz and Bart heuristic stops. For a substitution heuristic, like Teitz and Bart, the neighborhood search space includes all possible swaps of any facility location with any unused candidate site. We call this a λ=1 "opt" neighborhood using the terminology originally used by Shin Lin in developing a substitution heuristic for the traveling salesman problem. The value of λ represents the number of unused candidates that we are considering simultaneously for substituting for a like number of facility sites. If we defined a λ-opt as p -opt, where the neighborhood search space involves any potential swap of q ($q \leq p$) facility locations with any q unused candidate sites, then the local search neighborhood is a global search neighborhood and would produce a global optimal location configuration.

In most problems, the size of the local search neighborhood is small compared to the size of the global search neighborhood. For a 5-median problem on a 50 node problem, the 1-opt local search neighborhood is:

$$5 \times 45 = 225$$

$$\text{OR} \quad p \times \left[n - p \right]$$

If the problem space doubles to 100 nodes, the local search space more than doubles:

5 x 95 = 475

This means that a thousand node problem will have a 1-opt search space, more than 10 times larger than a hundred node problem. The computational work by the Teitz and Bart heuristic is directly proportional to the size of the search neighborhood times the number of complete cycles of searching through such a neighborhood before stopping at a local or global optimum.

Now let us turn our attention to GRIA (global-regional interchange algorithm/approach). First, we begin with an algorithmic description of the GRIA process.

Step 1: Start with an initial pattern of p sites selected among the n nodes. Call this the current best configuration.

Step 2: Find the marginal site in the current configuration; that is the best site to drop. Call this location k and remove that site from the current configuration.

Step 3: Find the best site to add to the current configuration, that is, the site which when added to the current configuration improves the objective function the most. Call this site l.

Step 4: If $l = k$ place site l in the configuration and go to step 5. Otherwise, place site l in the configuration and go to step 2.

Step 5: Define Γ as the set of sites in the current configuration. For each $j \varepsilon \Gamma$ define a set Φ_j as a local search set or a set of nearby sites to site j. This set of sites may be comprised of the nodes that are served by site j or based on some other notion of

Step 6: Select a facility site $j \varepsilon \Gamma$. Remove site j from Γ. Of the local search set Φ_j, identify the site in Φ_j that is the best replacement site for site j. If the best replacement site yields an improvement in the objective, then swap j for that site in the current configuration.

Step 7: If set Γ is *null* and configuration changes have been made since Γ was last initialized in Step 5, go to step 5: If set Γ is *null* and no changes in configuration have been made since Γ was last initialized in Step 5, go to step 8. If Γ is not the *null* set, then go to step 6:

Step 8: If a local/regional search (steps 5-7) produced a change in the configuration since the last global search (steps 2, 3 and 4), then return to step 2. Otherwise stop with a local optimal configuration.

To compare GRIA and Teitz and Bart, it makes since to compare the size of the search neighborhoods. The search space in GRIA for the global swaps is small and is based on the size of the best site to drop out of p and the best site to add out of $n - p$ thus the search space is related to checking p sites and then evaluating $n - p$ replacements. This is the same as $n - p$ tested swaps. The search space for the local/regional swaps is equal to the membership of the Φ_j sets. If the members of Φ_j are the partitions of closest sites to j, then the total membership of all Φ_j equals n elements. The total local search neighborhood of the GRIA heuristic is equal to $n - p + n$ or $2n - p$

Thus, a 10 median, thousand node problem for GRIA has a local search space of 2000 -10 or 1990. Whereas, the Teitz and Bart local search space is 9990. Thus, if the number of basic cycles is the same, GRIA would identify a local optimum almost five times faster. Consequently, if GRIA performed as well as Teitz and Bart, in terms of the quality of the configurations, but is substantially faster than Teitz and Bart, then the choice of selecting GRIA would be a simple one to make.

An obvious thought: IS the GRIA neighborhood Search Space Too Small? Does it allow enough flexibility for the GRIA process to find a path of improving solutions/configurations to a very good if not optimal solution. Obviously, the size of the neighborhood search space can dictate the quality of the results. Has the definition of the local neighborhood search space been defined too narrowly to produce good results? This requires testing: Teitz and Bart vs. GRIA.

In order to test both heuristics, we selected four data sets. They included the 55 node problem of Swain (1976), the reversed version of Swain where the nodal weights are reversed, the 150 node London, Ontario data set of Waters (1977), and the 377 node data of the Basin of Mexico of Ruggles (1992). We solved a number of p -median problems for each data set We used alternate software in order to determine the optimal solutions of all of the problems solved. For the Swain and Reversed Swain data sets we varied p from 4 to 9. For the London data set we used p values of 6, 9, 12, 15, 18, 21, and 24. On the basin of Mexico data set, we applied the heuristics for values of p equal to 10, 20, 30, and 40. This represents 23 different problems involving four different data sets. We restarted each heuristic 5 times as a test. From this we found that Teitz and Bart identified the optimal solution 54 times out of 115 trials and GRIA found the optimal solution 56 times out of 115 trials. Using 5 starts for each problem yielded 15 optimal solutions for 23 problems using Teitz and Bart whereas GRIA identified 18 optimal solutions for 23 problems. For just the larger data sets (i.e. London and Mexico) GRIA found 6 out of 11 optimal solutions and Teitz and Bart found 4 out of 11 optimal solutions. For these 11 larger problems, GRIA found optimal solutions 11 out of 55 trials and Teitz and Bart identified 8 optimal solutions out of 55 trials. The average solutions and the best and worst solutions for each problem set were very comparable. In short, the performance of GRIA was very comparable to the Teitz and Bart heuristic. Computationally, we found that the number of swaps tested by GRIA varied in comparison to Teitz and Bart, but were nearly always less than that tested by Teitz and Bart. For the 23 problems tested, the effort expended by GRIA as a function of that expended by Teitz and Bart ranged from .218 to 1.018. From our tests we conclude that GRIA's performance is just as robust as Teitz and Bart. Further, such a GRIA "defined" search neighborhood does reduce computational effort. Therefore, at this juncture, it does make sense to use a GRIA type process as a location heuristic in a location-GIS module.

Using a Heuristic: an operative approach

There are some remaining issues that should be addressed in the use of a heuristic as an analytic engine in a GIS location module. We know that Teitz and Bart and GRIA do not always identify the optimal solution. It is fairly safe to state that TAB or GRIA should **not** be applied once, given one initial starting configuration, but restarted with different initial seeds a number of times. But this raises a very basic issue:

1) How many times should such a heuristic be repeated using different starting solutions to ensure that the best configuration found among the various final local optima is either close-to-optimal or optimal?

or maybe

Is there some method that can be used to help identify when to stop, report the best configuration found up to that point, and not attempt further restarts ---- with some confidence that the optimal solution has been identified?

It is unfortunate that few have addressed this problem with location heuristics. Perhaps the single sentinel is the paper by Rosing, Hillsman and Rosing-Volegaar(1979). Unfortunately, they found that for the problem they addressed, restarting Teitz and Bart a number of times seemed a waste of time. Perhaps most took this statement as a final analysis, for many subsequent applications of Teitz and Bart have restarted the process on a given problem once or occasionally twice (see Fotheringham, Curtis and Densham (1993) as an example). This is a critical application issue and must be addressed if we can have some confidence that our solution process has achieved the best possible solution or if we are to know that the problem is so difficult that it confounds this type of search/substitution process.

To better understand this type of issue, we developed an experiment for several problems for which we had the computer resources and software to identify the optimal solutions. We used the 150 node London(Ontario) data set that was originally used by Nigel Waters (1977). Since we knew the optimal configurations in advance, we could restart either heuristic a number of times, each time with a randomly generated starting configuration, until it had identified an optimal solution a given number of times. Since we found the GRIA heuristic as good as the TAB process, we used the GRIA heuristic in the following analysis. For this analysis we continued to restart GRIA until the heuristic had identified the optimal solution 10 times. The results of that application are given in the following Table.

Results of Repeated Trials of GRIA for the London Data Set					
Number of facilities	# trials to find optimal value 10 times	Frequency of optimal Value	ρ	q	n (for C=.95)
6	16	1 every 1.6	.625	.375	4
9	42	1 every 4.2	.238	.761	12
12	53	1 every 5.3	.1886	.8113	15
15	39	1 every 3.9	.2564	.7435	11
18	391	1 every 39.1	.0255	.9744	116
21	2515	1 every 251.1	.00397	.9960	752
24	373	1 every 37.3	.0268	.97319	111

Where: ρ = the probability that the heuristic finds an optimal solution on any one run.

$q = 1 - \rho$ probability that heuristic does not find optimal value on any one run

C = the desired confidence interval

n = the number of runs necessary to identify the optimal solution within a confidence of C.

The number of restarts needed to identify the optimal solution 10 times varied from 16 to 2515. In the majority of cases, the GRIA needed 60 or less restarts, but in three cases GRIA needed nearly 400 restarts or more. Restarting the heuristic many times, enough to generate the optimal solution ten times, allows us to calculate the average frequency of finding the optimal solution. That is given as a probability p in the table. The probability of not finding an optimal solution on any given random start is given as q. If we solve the following equation for n:

$$1 - q^n \geq C$$

we can identify the number of restarts necessary to generate the optimal solution within a confidence interval C. This process indicates that for some cases it will take more than 100 restarts to be relatively confident that we had identified the optimal solution. It is important to note that this estimate of *n* is made in the case of complete information. We knew the optimal solution in advance. Such is not the case for most problems, for if we knew the optimal solution there is no need to apply a heuristic. If we had perfect information before we executed the GRIA or TAB heuristic as to the probability p, then we could determine in advance the number of trials that the heuristic should be repeated in order to produce within some confidence, say 95%, the optimal solution. Even though we do not have a crystal ball in terms of the probabilities p, this has given us some after-the-fact-estimate of what would have been necessary for the problems solved using the London data set.

The results of this experiment were thought provoking to say the least. An obvious question is: what stops a GRIA or a TAB from reaching an optimal configuration? That is, are there a few good local optima solutions that GRIA or TAB slip into, or is the landscape loaded with local optimal that represent a minefield that can frequently stop the heuristic in its tracks. Understanding this may help us in using such heuristics. One somewhat obvious question is how many different local optima were found, and were some more likely to be identified than others? We analyzed the results of the previous experiment and report them in the following table:

Number of Unique Local Optima found by GRIA for the London Data Set

Number of facilities	# trials to find global optimum 10 times	Number of unique local optima encountered in search	Maximum occurrence of any local optimum	Average occurrence of each optimum	
6	16	5	10	3.2	
9	42	8	11	5.25	
12	53	9	19	5.89	
15	39	17	10	2.29	
18	391	156	15	2.51	
21	2515	825	87	3.05	
24	373	239	10	1.34	

The above table presents the number of unique local optimal solutions that were identified, the maximum occurrence of any local optimum, and the average occurrence of any local optimum. It is interesting to note that the average occurrence of any local optima was less than the occurrence of the optimum (10 for this set of tests). Thus, the global optimum is generally found more frequently than the average local optima. The maximum occurrence of any local optimum varied from 10 (indicating that the global optimum was the most identified solution) to 87 (indicating that some local solutions had a much higher chance of being identified than the global optimum). Also note that in some problems, there was a very large number of local optima identified. For p = 21, there were 825 unique local optima. We can conclude that for some problems there are a large number of local optima. We can call it a minefield of local optimal solutions which can stop a λ = 1-opt process from arriving at the optimal solution. This means that enough restarts are needed to ensure that there is a high likelihood that the global solution has been found.

Most local optima did not occur with great frequency. Three of the seven test problems resulted in the optimal solution being the most frequent of the local optima found. Since the optimal configuration is found with greater frequency than most of the other local optima, restarting a GRIA or TAB should be successful at finding the optimal solution most of the time if it is restarted a sufficient number of times for a given problem. Now, when do we stop the heuristic, like TAB and GRIA, with some confidence that we have an optimal solution? Unfortunately, the above posterior analysis does not give us any absolute guidance as to how many times the TAB or GRIA process should be restarted for any specific problem. However, it does give us one major insight: the frequency of identifying the global optimum is relatively high compared to the frequency of identifying most other local optima. Based on this property, we propose the following dynamic stopping rule:

RAP test or stopping rule: Continue to restart the TAB or GRIA heuristic with additional random starting configurations until the lowest and best generated solution has been identified t times without finding a better configuration.

As the TAB or GRIA process is restarted and identifies local optima, track the best configuration found as well as the number of times that that particular solution has been identified. If the lowest or best configuration is found *t* times without finding a better configuration, then stop. This stopping criteria is dynamic, since you do not know how many restarts are required to meet the criteria of t in advance. Of the twenty-three different problems solved over four different data sets (four data sets of 55, 55, 150, and 377 nodes: values of p ranging from 4 to 40), we calculated how frequently global optimal solutions were found based on the proposed dynamic stopping rule. The results of this test are given in the following table:

Frequency of optimality based on *t* value (23 problems)				
Value of *t*	total # of heuristic starts	# of optimal solutions identified	Percent optimal found	Notes
1	23	11	48%	†
2	328	17	74%	
3	801	20	87%	
4	1233	21	91%	
5	1522	22	96%	
6	1707	22	96%	
7	2370	22	96%	
8	2749	23	100%	
9	3853	23	100%	
10	3990	23	100%	

† Note: This percentage tends to be higher for small problems and tends to be lower for larger problems.

From this table, it is easy to see that optimal solutions were generated 48% of the time with only one random start. This robustness (high level of optimality) can be attributed primarily to one of the four data sets. The heuristic found optimal solutions only 29% of the time on one random start for the remaining three data sets and problems. Often, the larger data sets required more restarts to identify the optimal solution. Notice that if we use a value of t = 3, the optimal solution can be identified 87% of the time, where an average number of restarts would be approximately 40. Beyond t = 3, the number of restarts necessary to identify additional optimal solutions increases significantly. It appears that if we use a value t = 3, we will have a reasonably good chance of identifying the optimal solution to a given problem without investing too much in any given application.

Up to his point it should be more than obvious that even though the selection of a heuristic was somewhat simple to make, the conditions under which the heuristic is applied may have a significant impact on the final results and any conclusions. Since we found for the majority of problems that a one start approach did not identify the optimal solution, then a restart strategy is absolutely necessary if there is any attempt to identify the optimal solution with some high degree of confidence. If we relied on only one start per problem, it is likely that the our interpretation of any solution could vary (even substantially) from that of what we might interpret from the best pattern (if we had identified it). For example, for a six facility problem on the London data set, Teitz and Bart generated 6 different configurations out of 25 starts where there was a maximum of 3.17% difference in objectives. The overlap between the six different solutions averaged about 2 sites. There were some patterns that used no sites in common. For GRIA applied to the same problem, there were five different solutions that were generated, where the objective values differed by slightly less than 3%. Two of the patterns that were generated are given in Figure 1. Note that there are no sites used in common. Both are local optima (as defined by the local neighborhood search) and both were identified by both heuristics (Teitz and Bart & GRIA). As another example, for a 15 facility problem (London data), Teitz and Bart generated 10 different final solutions and GRIA generated 13 different solutions out of 25 starts. The Teitz and Bart solutions varied by no more than .8% in weighted distance and the GRIA solutions varied by up to 1.6% in weighted distance. Comparing all of these solutions, the average overlap was approximately 10 sites out of 15. That is, each solution differed on the average from some other identified local optimum by about 5 sites. If there is such variance in solving a fixed problem, then there should be extreme caution when applying a heuristic to test a zone-definition aggregation approach like that of

Fotheringham, Curtis and Densham (1993) In the Zone-definition problem, Fotheringham et al. generated a number of spatial aggregation scenarios and for each of them they solved a p-median problem. Only when an objective value of a solution was suspiciously high did they restart the Teitz and Bart heuristic. If one uses a heuristic like TAB or GRIA to study a particular problem like the Zone-Definition problem, a number of restarts are necessary before there is a good probability of having results that can be compared.

Conclusions

A number of location problems can be viewed as equivalent p-median problems. Further, it is possible to analyze other problems at least partially using a p-median structure. For these two reasons, it is understandable that there is interest in providing the capability of solving a p-median problem as a part of GIS functionality. There are a number of issues that must be addressed in the integration of a p-median model into a GIS. One of these problems involves the design or selection of a solution approach. We have addressed this basic design question and narrowed the selection at this time to two heuristics: Teitz and Bart and GRIA. Future developments with TABU and Simulated Annealing may provide improved alternatives. We have provided a comparison of these two heuristics and found that the GRIA performance is just as robust as Teitz and Bart. Further, the reduced search neighborhood of GRIA does reduce computational effort, but does not eliminate a pathway to an optimal solution. Both appear to generate an optimal solution with about the same frequency from random starting solutions. Regardless of which heuristic approach is taken, there should be considerable caution exercised when using either approach with only one random start. We have identified that for some problems there is a minefield of local optima in which to stop Teitz and Bart or GRIA in its tracks. It is absolutely imperative that such heuristics be restarted a number of times. We found that the number of times that a Teitz and Bart or GRIA heuristic needs to be restarted in order to find the global optimum with some degree of confidence can be quite high. We have proposed a dynamic stopping rule that can be used to help judge when to stop restarting the Teitz and Bart or GRIA heuristic in application. When using this rule there appears to be a reasonably high probability that the global p -median optimum will be identified.

References

1. Beasley J, 1993, "Lagrangean heuristics for location problems" European Journal of Operational Research 65, 383-399

2. 0. Berman, D. Einav, and G. Handler, The zone-constrained location problem on a network. E. J. Qpl. Res. 53, 14-24 (1991).

3. G. Bianchi and R. L. Church, A non-binary encoded genetic algorithm for a facility location problem. in review (1993).

4. C. Bloxham and R.L. Church, The p-median scheduling and location problem. Papers in Regional Science 70, 21-35 (1991).

5. R.L. Church, The regionally constrained p-median problem. Geogr. Analysis 22, 22-32 (1990).

6. R.L. Church and C.S. ReVelle, Theoretical links between the p -median, location set covering, and the maximal covering location problem. Geographical Analysis 406-415 (1976).

7. R.L. Church and J.R. Weaver, Theoretical links between median and coverage location problems. Annals of Operations Research. 6, 1-19 (1986).

8. P.J. Densham, The location analysis decision support system (LADSS). NCGIA software series S-92-3, Santa Barbara, CA. (1992).

9. P. Densham and G. Rushton, A more efficient heuristic for solving large p-median problems. Papers in Regional Science 71, 307-329 (1992).

10. M. Figueroa, P. Sorensen, and R. Church, VISLOC: a visual and model based decision support system for Location Exploration. Manuscript in preparation, Dept. of Geography, UCSB, Santa Barbara, Cal. (1994).

11. A. S. Fotheringham, A. Curtis and P.J. Densham, The zone definition problem in location- allocation modeling. Unpublished Manuscript. NCGIA, State University of New York, Buffalo, NY (1993).

12. B.L. Golden and C. Skiscim, Using simulated annealing to solve routing and location problems. Naval Logistics Quaterly, 33, 261-279 (1986).

13. F. Glover, Tabu search: a tutorial. Interfaces. 20, 74-94 (1990).

14. S.L. Hakimi, Optimum locations of switching centers and the absolute centers and medians of a graph. Opns. Res. 11, 450-459 (1964).

15. S.L. Hakimi, Optimum distribution of switching centers in a communication network and some related graph theoretic problems. Opns. Res. 13, 462-475 (1965).

16. E.L.Hillsman, The p-median structure as a unified linear model for locationallocation analysis. Environment and Planning A 16, 305-318 (1984).

17. C.M. Hosaoe and M. F. Goodchild, Discrete space location-allocation solutions from genetic algorithms. Annals of Operations Research 6, 35-46 (1986).

18. J. G. Klincewicz, Avoiding local optima in the p -hub location problem using tabu search and grasp. Annals of Operations Research 14, (1991).

19. S. Lin, Computer Solutions of the travelling salesman Problem. The Bell System Technical Journal 44, (1964).

20. F. Maranzana, On the location of supply points to minimize transport costs. Operational Research Quaterly 15, 261-270 (1964).

21. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, Equation of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087-1093 (1953).

22. P.B. Mirchandani and A.R. Odoni, Locations of medians on stochastic networks. Transportn. Sci. 13, 86-97 (1979).

23. A. Murray and R.L. Church, Applying Simulated Annealing to Location Planning Models. FIpresented at the at the ORSA/TIMS meeting, Phoenix, Az. Nov. (1993)

24. S.C. Narula and U.I. Ogbu, An hierarchal location-allocation problem. Omega 7, 137-143 (1979).

25. S.C. Narula, U.I. Ogbu, and H.M. Samuelsson, An algorithm for the p-median problem. Opns. Res. 25, 709-713 (1977).

26. A. Neebe, "A branch and bound algorithm for the p-median transportation problem" Journal of the Operational Research Society 29, 989-995 (1978).

27. C.S. ReVelle and R.W. Swain, Central facilities location. Geogr. Analysis 2, 30-42 (1970).

28. K.E. Rosing, E.L. Flillsman, and H. Rosing-Vogelaar, A note comparing optimal and heuristic solutions to the p-median problem. Geogr. Analysis 11, 86-89 (1979).

29. K. Rosing, C. ReVelle, and H. Rosing-Vogelaar, "The p-median and its linear programming relaxation: an approach to large problems" Journal of the Operational Research Society 30 815-823 (1979).

30. R. Swain, "Central facilities location with multiple planning periods", research report, Department of Industrial and Systems Engineering, Ohio State University (1976).

31. M.B. Teitz and P. Bart, Heuristic methods for estimating the generalized vertex median of a weighted graph. Gpns. Res. 16, 955-961 (1968).

32. N. Waters, Methodology for servicing the geography of urban fire: an exploration with special reference to London, Ontario. PhD dissertation The University of Western Ontario, London Ontario (1977).

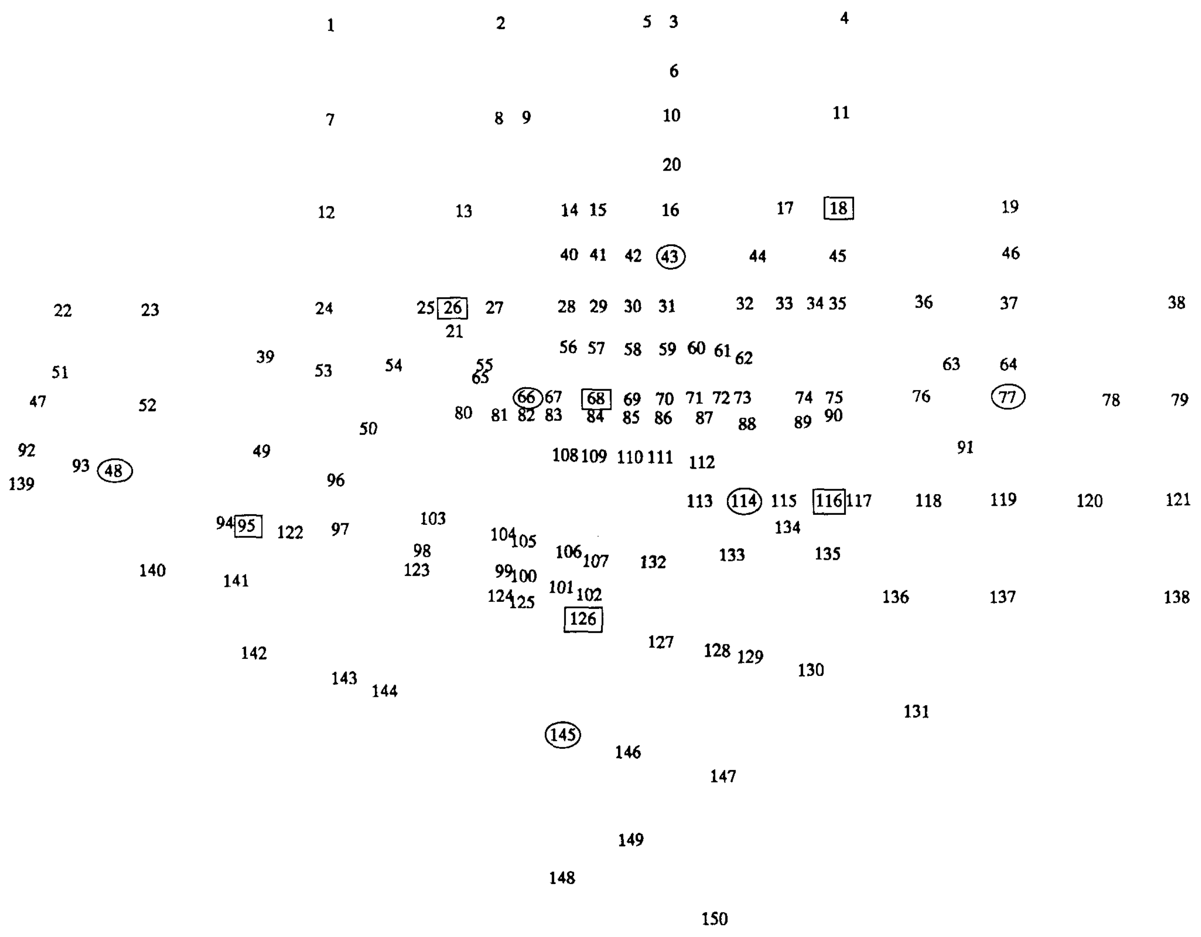

Figure 1. Two different locally optimal configurations to the 6-median problem (London data)