Accepted Manuscript

Title: The Tradeoff Between the Median and Range of Assigned Demand in Facility Location Models

Authors: Mark S. Daskin and Emily L. Tucker


DOI: 10.1080/00207543.2017.1401751

This article can be cited as:

The Tradeoff Between the Median and Range of Assigned Demand in Facility Location Models

Mark S. Daskin and Emily L. Tucker
Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, MI 48109

Phone: 1-734-764-9410 FAX: 1-734-764-3451
Email: msdaskin@umich.edu

Keywords: Facility location; optimization; multi-objective; genetic algorithm

ABSTRACT

In this paper we present an extension of the classic $p$-median facility location model. The new formulation allows the user to trace the tradeoff between the demand-weighted average distance (the traditional $p$-median objective) and the range in assigned demand. We extend the model to incorporate additional constraints that significantly reduce the computation time associated with the model. We also outline a genetic algorithm-based approach for solving the problem. The paper shows that significant reductions in the range in assigned demand are possible with relatively minor degradations in the average distance metric. The paper also shows that the genetic algorithm does very well at identifying the approximate tradeoff curve. The model and algorithms were tested on real life datasets ranging in size from 33 nodes to 880 nodes.

1 INTRODUCTION

The $p$-median problem is a classic facility location problem in which the objective is to locate $p$ facilities to minimize the demand-weighted total (or average) distance between each demand node and one of the $p$ facilities (Hakimi, 1964). The problem is arguably at the heart of many, if not all, facility location problems (Laporte, 2017). The fixed charge location problem is a straightforward extension of the $p$-median problem in which the model endogenously determines the number of facilities to locate as well as their locations to minimize the combined site-specific facility location costs and the transport costs. The maximum covering problem
(Church and ReVelle, 1974) can also be readily recast as a \(p\)-median problem through an appropriate transformation of the distance matrix.

The \(p\)-median model typically assigns demands to the closest open facility. In doing so, the model treats each facility as being uncapacitated. The model also ignores differences in the assigned demand. Such differences can be exceptionally important, however. For example, in determining districts for salespeople, it is important to minimize the total distance traveled by salespeople – often represented as the average distance – and the range in assigned demands. The range is a proxy for equity, ensuring that the difference between the total demand assigned to the salesperson with the most assigned demand is not too different from the total demand assigned to the salesperson with the least assigned demand. Note that we are not necessarily talking about problems in which a salesman would visit customers on a traveling salesman tour. Similarly, in political districting, it is often important that districts be compact – again often measured by the average distance from a central node – and that the range in assigned demands be as small as possible so that each voter has the same level of representation.

To illustrate the problem with the \(p\)-median model, Table 1 gives the optimal \(p\)-median results for five values of \(p\) and four datasets. The 33 node Ann Arbor dataset represents the 33 census tracts in Ann Arbor, MI. The 83 node Michigan dataset represents the 83 counties in Michigan. The 150 city USA dataset includes the 150 largest (most populous) cities in the contiguous 48 states of the United States, and the 880 node USA dataset includes 880 zip-3 locations throughout the contiguous United States. In all cases, demand was proportional to population.
Longitude and latitude data were used for the node coordinates, and distances were computed as great circle distances rounded to the nearest mile in all cases except for the 33-node Ann Arbor data set in which distances were computed to the nearest $1/1000^{th}$ of a mile.

The first column of the table gives the dataset. The next column gives the total demand in the dataset. The third column gives the number of facilities. This is followed by the demand-weighted average distance, the total demand assigned to the facility with the most assigned demand, the total demand assigned to the facility with the least assigned demand, and the range in assigned demand. The eighth column gives the range as a percent of the total demand while the final column gives the range as a percent of the average demand assigned to a facility (which is the total demand divided by the number of facilities). The range can exceed 50 percent of the total demand as shown in the penultimate column. For the 20 results shown in the table, the range averages over 25 percent of the total demand (penultimate column), and the range is frequently over 75 percent of average demand (final column).

The purpose of this paper is to present a mathematical model that enables analysts to explore the tradeoff between minimizing the average distance and minimizing the range in assigned demand across all facilities. Following a review of the relevant literature in Section 2, the basic model is presented in Section 3. Section 4 outlines two additional constraints that can be added to the model and that significantly reduce the computation time associated with the basic model. Despite the addition of these constraints, solution times remain large, particularly
for small values of the allowable range. Section 5 outlines a genetic algorithm for the problem. A secondary objective of the paper is to argue that using genetic algorithms is an inherently natural way to attack multi-objective optimization problems since any population of solutions automatically provides a first-order approximation to the tradeoff curve. Computational results are given in Section 6, followed in Section 7 by conclusions and recommendations for future work.
Table 1: Average Distance and Range for 4 Different Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total Demand</th>
<th>Number of Facilities (p)</th>
<th>Average Demand-weighted Distance</th>
<th>Maximum Assigned Demand</th>
<th>Minimum Assigned Demand</th>
<th>Range</th>
<th>Range as % of Total Demand</th>
<th>Range as % of Average Assigned Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>33 node Ann Arbor</td>
<td>115,103</td>
<td>2</td>
<td>1.396</td>
<td>80,331</td>
<td>34,772</td>
<td>45,559</td>
<td>39.6%</td>
<td>79.2%</td>
</tr>
<tr>
<td></td>
<td>115,103</td>
<td>3</td>
<td>1.081</td>
<td>57,724</td>
<td>22,607</td>
<td>35,117</td>
<td>30.5%</td>
<td>91.5%</td>
</tr>
<tr>
<td></td>
<td>115,103</td>
<td>4</td>
<td>0.884</td>
<td>39,230</td>
<td>17,858</td>
<td>21,372</td>
<td>18.6%</td>
<td>74.3%</td>
</tr>
<tr>
<td></td>
<td>115,103</td>
<td>5</td>
<td>0.780</td>
<td>39,230</td>
<td>14,442</td>
<td>24,788</td>
<td>21.5%</td>
<td>107.7%</td>
</tr>
<tr>
<td>83 node Michigan</td>
<td>9,883,640</td>
<td>2</td>
<td>42.52</td>
<td>6,399,450</td>
<td>3,484,190</td>
<td>2,915,260</td>
<td>29.5%</td>
<td>59.0%</td>
</tr>
<tr>
<td></td>
<td>9,883,640</td>
<td>3</td>
<td>34.54</td>
<td>6,317,124</td>
<td>914,064</td>
<td>5,403,060</td>
<td>54.7%</td>
<td>164.0%</td>
</tr>
<tr>
<td></td>
<td>9,883,640</td>
<td>4</td>
<td>29.56</td>
<td>4,830,604</td>
<td>855,904</td>
<td>3,974,700</td>
<td>40.2%</td>
<td>160.9%</td>
</tr>
<tr>
<td></td>
<td>9,883,640</td>
<td>5</td>
<td>26.37</td>
<td>4,623,668</td>
<td>855,904</td>
<td>3,767,764</td>
<td>38.1%</td>
<td>190.6%</td>
</tr>
<tr>
<td></td>
<td>9,883,640</td>
<td>6</td>
<td>23.36</td>
<td>4,623,668</td>
<td>261,728</td>
<td>4,361,940</td>
<td>44.1%</td>
<td>264.8%</td>
</tr>
<tr>
<td>150 node USA</td>
<td>58,196,530</td>
<td>2</td>
<td>454.01</td>
<td>40,657,099</td>
<td>17,539,431</td>
<td>23,117,668</td>
<td>39.7%</td>
<td>79.4%</td>
</tr>
<tr>
<td></td>
<td>58,196,530</td>
<td>3</td>
<td>315.57</td>
<td>22,181,803</td>
<td>17,072,711</td>
<td>5,109,092</td>
<td>8.8%</td>
<td>26.3%</td>
</tr>
<tr>
<td></td>
<td>58,196,530</td>
<td>4</td>
<td>241.94</td>
<td>16,684,295</td>
<td>10,957,334</td>
<td>5,726,961</td>
<td>9.8%</td>
<td>39.4%</td>
</tr>
<tr>
<td></td>
<td>58,196,530</td>
<td>5</td>
<td>206.06</td>
<td>15,663,123</td>
<td>6,619,935</td>
<td>9,043,188</td>
<td>15.5%</td>
<td>77.7%</td>
</tr>
<tr>
<td></td>
<td>58,196,530</td>
<td>6</td>
<td>184.63</td>
<td>15,663,123</td>
<td>4,877,990</td>
<td>10,785,133</td>
<td>18.5%</td>
<td>111.2%</td>
</tr>
<tr>
<td>880 node zip3 USA</td>
<td>306,669,700</td>
<td>2</td>
<td>477.42</td>
<td>233,917,264</td>
<td>72,752,436</td>
<td>161,164,828</td>
<td>52.6%</td>
<td>105.1%</td>
</tr>
<tr>
<td></td>
<td>306,669,700</td>
<td>3</td>
<td>367.54</td>
<td>140,265,943</td>
<td>70,555,734</td>
<td>69,710,209</td>
<td>22.7%</td>
<td>68.2%</td>
</tr>
<tr>
<td></td>
<td>306,669,700</td>
<td>4</td>
<td>306.44</td>
<td>84,594,919</td>
<td>69,267,277</td>
<td>15,327,642</td>
<td>5.0%</td>
<td>20.0%</td>
</tr>
<tr>
<td></td>
<td>306,669,700</td>
<td>5</td>
<td>250.64</td>
<td>71,919,149</td>
<td>45,402,229</td>
<td>26,516,920</td>
<td>8.6%</td>
<td>43.2%</td>
</tr>
<tr>
<td></td>
<td>306,669,700</td>
<td>6</td>
<td>223.13</td>
<td>71,919,149</td>
<td>14,435,155</td>
<td>57,483,994</td>
<td>18.7%</td>
<td>112.5%</td>
</tr>
<tr>
<td>Minimum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.0%</td>
<td>20.0%</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26.5%</td>
<td>97.5%</td>
</tr>
<tr>
<td>Maximum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>54.7%</td>
<td>264.8%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 Range in assigned demand between the facility with the most assigned demand and the one with the least assigned demand
2 LITERATURE REVIEW

Our study contributes to the broad literature on location modeling. To limit our scope, we specifically focus on discrete location problems. We briefly discuss the literature on the \( p \)-median problem and its variants; multi-objective models; equity objectives, including facility assignments; and the contributions of this paper. Thorough reviews of the facility location literature are available in Owen and Daskin (1998) and Melo et al. (2009). Textbook treatments include Daskin (2013) and Drezner and Hamacher (2002).

2.1 \( p \)-median problems

The \( p \)-median problem was first introduced in Hakimi (1964). While it is \( NP \)-hard on a general graph, an optimal solution can be found in polynomial time on a tree (Kariv and Hakimi, 1979; Tamir, 1996; Daskin and Maass, 2015). Numerous papers have explored algorithms and properties of the model. Church and ReVelle (1976) develop connections with the set-covering and maximal covering problems, and ReVelle et al. (2008) review major contributions to the \( p \)-median literature. Reese (2006) surveys both exact and heuristic solution methods. Mladenović et al. (2007) reviews metaheuristics. Hosage and Goodchild (1986) were the first to apply a genetic algorithm (Mitchell, 1998 and Whitley, 1994) to the \( p \)-median problem, and Bozkaya, Zhang, and Erkut (2002) present three crossover operations
that reduce the likelihood of being trapped in a local optimum. Alp et al. (2003) develop a genetic algorithm that uses unions of parent chromosomes and a deletion heuristic rather than a traditional crossover approach. Other metaheuristics include Tabu search (Glover, 1990 and Glover and Laguna, 1997) applied by Rolland et al. (1996) and Variable Neighborhood Search (Mladenović and Hansen, 1997) applied by Hansen and Mladenović (1997).

2.2 Multi-objective models

Current, Min, and Schilling (1990) and Farahani et al. (2010) provide comprehensive reviews on multi-objective discrete location modeling. Both papers find that bi-objective models tend to include a cost minimization objective and another demand-oriented objective such as coverage or distance. Common objectives also include environmental risk, customer equity, service levels, and profit. Papers that focus on specific applications may include a variety of other objectives. In the biomass supply chain literature (Atashbar, Labadie, and Prins, 2017), the trade-offs may include job creation, social footprint, and energy consumption.

Ross and Soland (1980) encourage the use of multi-objective models in public facility location contexts. These problems tend to have a diverse group of stakeholders with conflicting objectives, and multi-criteria models can begin to capture the necessary compromises. When two objectives are considered, modelers generally produce trade-off curves to determine Pareto-efficient solutions. This has the added benefit of providing the decision-maker with additional information
about their choices. Minor deviations from the optimal solution of a single objective can lead to major improvements in the second (Cohon, 1978). For example, Shen and Daskin (2005) consider the trade-off between customers located within a threshold distance of a facility and costs that include facility costs, transportation, and inventory costs. They find that consideration of both the distance and cost objectives can lead to dramatic improvements in service with limited effects on cost.

Early work on multi-objective problems also includes Lee, Green, and Kim (1981) who use goal programming to locate plants with ranked weights on several objectives, including demand satisfaction, cost, number of facilities, and quality of life. Badri, Mortagy, and Alsayed (1998) use goal programming to locate fire stations to attain goals, including fixed and operating costs; demand satisfaction; average and maximum travel time; desired number of facilities; political acceptability; and water availability. Belardo et al. (1984) present a partial covering model to locate equipment to protect against oil spills when there is uncertainty in the spill location. They identify Pareto-optimal solutions for the trade-offs between protecting against spills with different levels of potential damage.

Jayaraman (1999) presents a multi-objective capacitated location model for multiple products. The author studies the tradeoffs between three objectives: minimization of fixed costs; minimization of variable costs; and minimization of average distance. Nozick and Turnquist (2001) locate distribution centers by considering the tradeoff between cost and coverage (as defined as demand within a threshold distance). Costs include fixed charge, inventory, and transportation. Zhou, Min, and Gen (2003) consider facilities with different capacities to minimize
shipping cost and transit time, where each link has both a shipping cost and transit time. They use a genetic algorithm to generate Pareto optimal solutions. Fernández and Puerto (2003) develop an uncapacitated facility location problem for different, general, objectives. They find all of the non-dominated solutions and present both an exact method and an approximation. Mitropoulos et al. (2006) present a facility location problem arising in a healthcare system. Specifically, the authors consider the problem of locating an exogenous number of uncapacitated hospitals and capacitated health centers. Patients have preferences but can be assigned to either type of facility. They minimize the total weighted service distance and the maximum distance for care. Alçada-Almedia, Coutinho-Rodrigues, and Current (2009) locate incinerators for hazardous waste facilities by minimizing the fixed and processing costs, impact on population, maximum average impact per district, and maximum individual impact. Tang et al., 2016 locate radio frequency identification devices (RFID) and consider the two objectives of minimizing network cost and minimizing poor coverage performance. The authors develop a new genetic algorithm that incorporates a divide-and-conquer greedy heuristic.

### 2.3 Equity

A growing number of location papers have begun to include objectives that consider equity. However, most have focused on equity for customers rather than facility assignments. A review by Marsh and Schilling (1994) discusses various measures of equity, including the more common center and mean absolute deviation objectives as well as the variance, range, and the Gini coefficient. Ogryczak (2000) studies the tradeoff between minimizing the average customer distance and
minimizing the average upside deviation. Fernández et al. (2007) uses the concept of equity to locate one new grocery store in the midst of existing company-owned and competitor-owned stores. The authors present a lexicographic model in which the primary objective is to maximize profit, and the secondary objective is to minimize the difference in market share for company-owned stores before and after the new store is opened. Lejeune and Prasad (2013) study two tree networks to consider the tradeoffs between equity and effectiveness. They consider the bicriteria objectives of median vs. Gini coefficient and median vs. the sum of absolute weighted distance differences. A recent paper by Barbati and Piccolo (2016) presents and analyzes properties of equality measures including the principle of transfers, normalization, and transformation invariance.

However, even in single objective problems, there has been limited work on equity for facility assignments, sometimes referred to as balanced allocation. In a paper similar to our work, Marín (2011) locates \( p \) facilities and minimizes the difference in customer assignments between facilities with the most and least. The author includes the constraint that customers must be assigned to their closest facility and does not consider the median tradeoff. Similarly, customers are assigned to their closest facility in Berman et al. (2009) where the authors locate \( p \) facilities to minimize the maximum total demand assigned to a facility. Alternatively, some authors consider lower bounds on demand assignments rather than complete parity. Guha, Meyerson, and Mungala (2000) add threshold constraints to the uncapacitated facility location problem to require that each open facility must be assigned a minimum amount of demand. Similarly, Karger and
Minkoff (2000) require minimum demand assignments and minimize the transportation costs.

There is also a subset of papers that balance assignments when the facilities are fixed \textit{a priori}. Zhou, Min, and Gen (2002) seek to balance the weights of a star-spanning forest where the disjoint trees represent distribution centers and customers. To achieve balance, they minimize the maximum weight of a star tree (i.e., customer assignments at a facility) and solve the problem using a genetic algorithm. Min et al. (2005) consider a capacitated version of a similar assignment problem and minimize the maximum shipping costs for a facility. They define balance as equity in shipping costs rather than in assigned demand. They use a genetic algorithm to present the trade-off between balanced allocation and total shipping costs.

A problem closely related to facility assignment equity is that of assigning voters to electoral districts. This multi-objective problem often includes balancing the number of voters among districts and some measure of connectivity within a district. An early paper by Garfinkel and Nemhauser (1970) produces all of the optimal solutions to a two-stage problem: in the first stage, they determine all feasible districts (based on contiguity, compactness, and population balance) and in the second, they minimize the maximum deviation from the average number of voters. A review is available in Williams (1995). More recent papers include Bozkaya et al. (2003) who consider a variety of objectives: \textit{p}-median; population balance; compactness; socio-economic homogeneity; similarity to current districts; and similarity to community group. Ricca and Simeone (2008) minimize population
imbalance, noncompactness, and non-conformance to administrative boundaries. The former uses Tabu search, and the latter compares Descent, Tabu search, Simulated Annealing, and Old Bachelor Acceptance. Similar to setting voting boundaries, balance may also be considered in ambulance districting (e.g., Ansari, McLay, and Mayorga, 2017).

2.4 Research contribution

Realistic models of facility location often require multiple objectives. However, much of the literature to date has compared a median objective with measures of cost or centeredness. In this paper, we develop and discuss a new model that evaluates the trade-offs between median and facility assignment balance objectives.

3 FORMULATION OF THE MODEL

The classic $p$-median problem can be formulated using the following inputs:

**INPUTS:**

$I$ set of demand points  
$J$ set of candidate locations  
$h_i$ demand at node $i \in I$  
$d_{ij}$ distance between demand node $i \in I$ and candidate site $j \in J$  
$p$ number of facilities to locate

While we distinguish between the sets $I$ and $J$ in the formulation, throughout all of our testing we assumed they were the same sets; in other words, every demand
node was also a candidate facility site and there were no other facility sites. In addition, we define the following decision variables:

**DECISION VARIABLES:**

\[
X_j = \begin{cases} 
1 & \text{if a facility is located at candidate site } j \in J \\
0 & \text{if not} 
\end{cases}
\]

\[
Y_{ij} = \begin{cases} 
1 & \text{if demand node } i \in I \text{ is assigned to a facility at candidate site } j \in J \\
0 & \text{if not} 
\end{cases}
\]

With these inputs and decision variables, the \(p\)-median problem can be formulated as follows:

\[
\begin{align*}
\text{Minimize} & \quad \sum_{i \in I} \sum_{j \in J} h_{ij} d_{ij} Y_{ij} \\
\text{st.} & \quad \sum_{j \in J} Y_{ij} = 1 \quad \forall i \in I \\
& \quad Y_{ij} - X_j \leq 0 \quad \forall i \in I; j \in J \\
& \quad \sum_{j \in J} X_j = p \\
& \quad X_j \in \{0,1\} \quad \forall j \in J \\
& \quad Y_{ij} \in \{0,1\} \quad \forall i \in I; j \in J
\end{align*}
\]

The objective function (1) minimizes the demand-weighted total distance. Constraints (2) ensure that each demand node is assigned and constraints (3) limit the assignments to selected facilities. Constraint (4) states that we locate \(p\) facilities, while constraints (5) and (6) are integrality constraints. For the traditional \(p\)-median problem, constraints (6) can be relaxed to non-negativity constraints;
however, in the model that follows, we will generally insist on single-sourcing demands and so we retain constraints (6) as shown above.

To extend the model to consider the range in assigned demand, we define the following additional inputs and decision variables:

**ADDITIONAL INPUTS:**

- \( M \) a large number
- \( r^{\text{max}} \) maximum allowable range in the assigned demand

**ADDITIONAL DECISION VARIABLES:**

- \( L \) minimum total demand assigned to any open facility
- \( U \) maximum total demand assigned to any open facility

With this additional notation, we can add the following additional constraints to limit the range in the assigned demand to \( r^{\text{max}} \):

\[
U - L \leq r^{\text{max}} \quad (7)
\]

\[
U - \sum_{i \in I} h_i Y_{ij} \geq 0 \quad \forall j \in J \quad (8)
\]

\[
\sum_{i \in I} h_i Y_{ij} + M \left(1 - X_j \right) - L \geq 0 \quad \forall j \in J \quad (9)
\]

Constraint (7) limits the range in assigned demands to \( r^{\text{max}} \). Constraints (8) define the maximum assigned demand in terms of the assignment variables \( Y_{ij} \). Similarly, constraint (9) defines the minimum demand assigned to any facility. The term \( M \left(1 - X_j \right) \) is needed to ensure that the minimum is taken only over open facility sites. The value of \( M \) can be set to \( \frac{\sum_{i \in I} h_i}{p} \), since the minimum demand assigned to any facility cannot exceed the average demand assigned to the facilities.
To trace the tradeoff between the demand-weighted total (or average) distance and the range in assigned demands, we can use the following algorithm:

**Algorithm 1:**

1) Set $r_{\text{max}} = \sum_{i \in I} h_i$. Set $m=1$, where $m$ is a counter of the number of solutions found.

2) Solve (1)-(9). If no feasible solution is found, stop; all possible solutions have been found. If an optimal solution is found, record the objective function value, $\sum_{i \in I} \sum_{j \in J} h_i d_{ij} Y_{ij}$, as $D^m$ and the range in assigned demand, $U - L$, as $R^m$.

3) Increment $m$ and update $r_{\text{max}}$ in an appropriate manner. Return to step 2.

The only part of the algorithm that requires explication is the updating of $r_{\text{max}}$ in step 3. If all of the demands are integer-valued (as is generally the case), we can set $r_{\text{max}}$ to $R^m - 1$ if we want to trace the entire tradeoff curve. However, this may require an exceptionally large number of iterations. Instead, we can approximate the tradeoff curve by setting $r_{\text{max}} = \alpha R^m$ where $\alpha$ is a value between 0 and 1 (e.g., 0.98 or 0.95 or 0.9).

Before introducing two additional constraints that tighten the formulation above, we briefly discuss three properties of the model. First, relaxing the integer assignment constraints (6) can result in a better solution as shown using the network of Figure 1. The demands at the nodes are shown in boxes above the nodes. The solution to the 2-median problem is to locate at nodes A and C, assigning
demand node B to the facility at C. The demand weighted total distance is 40, and the range in assigned demand is 6 (or 16 minus 10). If instead we assign the demands at B to the facility at A, the demand-weighted total distance increases to 44, but the range is reduced to 2 (or 14 minus 12). With integer-assignments this is the best we can do. However, if we allow fractional assignments, we can assign 75% of the demand at B to the facility at A and 25% of the demand at B to the facility at C. This results in a demand-weighted total distance of 43 and a range in assigned demand of 0. In the computational results below, we insist on binary assignments (e.g., Li, Guo, and Zhang, 2017).

![Figure 1: Simple network showing the value of fractional assignments](image)

Second, we show that it may be optimal to locate a facility at a node but to assign the demand at that node to a different facility. Consider the tree network shown in Figure 2. If we want the minimum range, the optimal solution is to locate at nodes D and E but to assign the demand at node D to the facility at E. The demand-weighted total distance is 3020 and the range in assigned demand is 0. While the optimization model above allows this to occur, in the genetic algorithm outlined in Section 5, we force demands at facility nodes to assign to the facility at that node.
Finally, although the $p$-median problem is solvable in polynomial time for fixed $p$, the variant we are studying in this paper remains NP-Hard even for fixed $p$. This can be observed from a simple reduction from a generic instance of PARTITION with $n$ numbers, into an instance of the variant we propose with $n + 2$ locations. Set the demand of the first $n$ locations equal to the numbers from the PARTITION instance, and the demand of the last two locations to 0. In addition, set the distances from the first $n$ locations to the last two locations to 0, and assign positive values to the rest of the distances. Finally, set $p = 2$ and $r^{max} = 0$. Clearly, the PARTITION instance has a feasible solution if and only if this constructed instance has an optimal solution with a value of 0.
4 ADDITIONAL CONSTRAINTS

While formulation (1)-(9) and Algorithm 1 will allow us to trace out the tradeoff between the demand-weighted total distance and the range in assigned demand, the computation times can be very large, particularly for small values of $r^{\text{max}}$. In this Section, we outline two additional constraints that significantly tighten the formulation given above.

We begin by identifying a constraint on the maximum assigned demand, $U$, in terms of the total demand, $\sum_{i \in I} h_i$, the number of facilities to be located, $p$, and the maximum allowable range in assigned demand, $r^{\text{max}}$. Assume that a single facility is assigned the maximum possible demand, $U$, and that all other facilities are assigned the minimum possible demand, $L$. This represents the largest value for the maximum demand, $U$, because if any other facility were assigned demand greater than $L$, it would reduce $U$. In that case, we have:

$$U + (p-1)L = \sum_{i \in I} h_i$$  \hspace{1cm} (10)

In addition, from (7) we know that

$$U - L \leq r^{\text{max}}$$  \hspace{1cm} (7)

Multiplying the left and right hand sides of (7) by $p-1$, we have

$$(p-1)U - (p-1)L \leq (p-1)r^{\text{max}}$$  \hspace{1cm} (11)

Adding (10) and(11) and dividing by $p$, we obtain:

$$U \leq \frac{\sum_{i \in I} h_i}{p} + \frac{(p-1)r^{\text{max}}}{p}$$  \hspace{1cm} (12)
Constraint (12) states that the maximum assigned demand cannot exceed the average demand assigned to the facilities, \( \frac{\sum_{i \in I} h_i}{p} \), plus the maximum allowable range in assigned demand, \( r_{\text{max}} \), multiplied by \( \frac{(p-1)}{p} \).

Using similar arguments, we can derive the following constraint on the minimum demand assigned to any facility:

\[
L \geq \frac{\sum_{i \in I} h_i}{p} - \frac{(p-1)r_{\text{max}}}{p}
\]

(13)

As shown in the computational results (Section 6), constraints (12) and (13) significantly tighten the original formulation.

Finally, we note that constraint (12) can be used along with the maximum demand at any node, \( h_{\text{max}} \), to derive a lower bound on the allowable range if we have integer assignments. The upper bound must be at least as large as \( h_{\text{max}} \). Thus, we have

\[
h_{\text{max}} \leq U \leq \frac{\sum_{i \in I} h_i}{p} + \frac{(p-1)r_{\text{max}}}{p}
\]

which we can rearrange to derive

\[
r_{\text{max}} \geq \frac{p h_{\text{max}} - \sum_{i \in I} h_i}{p-1} = \left( \frac{p}{p-1} \right) (h_{\text{max}} - h_{\text{avg}})
\]

(14).

Thus, if the largest demand is significantly greater than the average demand assigned to a facility, \( h_{\text{avg}} = \frac{\sum_{i \in I} h_i}{p} \), the right hand side of (14) may significantly
exceed 0. As the number of facilities increases, the right hand side of (14) becomes larger since $h^{\text{avg}}$ decreases. Thus, for large values of $p$, the minimum possible range is likely to exceed 0 by a significant amount. This is most likely to occur in datasets with skewed demands, as often happen in real-world instances. For this reason, we limit our computational tests to cases with relatively small values of $p$.

5 A GENETIC ALGORITHM

Even with the constraints identified in Section 4, the optimization model can take a long time to find even one solution, particularly for small values of the allowable range, $r^{\text{max}}$, and for large datasets. Therefore, we also designed a heuristic procedure that has a genetic algorithm at its heart. The increase in solution times with the size of the problem is to be expected. The fact that solution times tend to increase with decreases in the allowable range may be less expected. Small values of the allowable range make the range-related problem close to an extension of the partition problem which is another NP-hard problem in its own right. (See the proof of NP-hardness at the end of Section 3.) This may explain the observed increase in solution time with smaller values of the allowable range.

Genetic algorithms are heuristic methods that iteratively produce solutions based on combinations of existing solutions and random changes. Solutions are represented as chromosomes (Section 5.2), and new solutions are produced by merging or crossing two existing (parent) solutions that are probabilistically selected from a population of solutions based on their quality (Section 5.3).
Mutations of these child solutions are changes that occur randomly and are not based on the parent solutions.

5.1 Genetic algorithms for multi-objective problems

Any population of solutions provides an approximation to the set of non-inferior solutions. Thus, obtaining an approximation to the tradeoff curve is relatively easy using a genetic algorithm. Numerous other authors have employed genetic algorithms in finding multi-objective tradeoff surfaces including: Bozkaya, Zhang and Erkut, 2002; Gupta, 2012; Hosage and Goodchild, 1986; Rabiee, Zandieh and Ramerzani, 2012; Shen and Daskin, 2005; and Tang et al., 2016.

In our implementation, we use the genetic algorithm to search over the space of locations and employ improvement algorithms to (1) assign demands to facilities and (2) to improve the initial set of locations in any solution within the population.

Consider the set of 25 initial solutions for an instance of the 83-node Michigan problem shown in Figure 3. Four solutions, shown as squares, are on the non-dominated frontier; one or more of these four solutions dominates the remaining 21 solutions. Dominated solutions are plotted as circles.
5.2 Encoding a chromosome

Encoding a solution to the model formulated in Section 3 would entail representing both the locations of the facilities and the assignment of demand nodes to the facilities. This would create a very large encoding. Furthermore, since most demand nodes are assigned to the closest facility, many randomly generated encodings would not represent good solutions. Therefore, we encode only the facility locations; assignments will be determined for each solution using a different heuristic procedure as outlined below.

In our implementation, a solution is encoded as a string of bits. The encoding, or chromosome, gives the values of $X_j$, the location variables.
5.3 Evaluating the objective function

In evaluating the quality of a solution, we assigned a value of 0 to any solution that is on the (emerging) tradeoff curve. Note that as the algorithm progresses, solutions can move off of the tradeoff curve as better solutions are found that may dominate solutions currently on the tradeoff curve. All other solutions are assigned a value equal to the Euclidean distance between the solution and the nearest solution on the tradeoff curve. (This value is given in the last column of Table 2 below.) Figure 4 shows an enlarged region of the tradeoff curve near the non-inferior solutions of Figure 3. Two inferior solutions are closest to the third non-inferior solution from the left; the remaining two solutions are closest to the fourth non-inferior solution from the left. Table 2 lists these four solutions, the coordinates of the closest solution on the tradeoff curve, and the objective function value assigned to these four inferior solutions. Again, note that this objective value differs from the objective function value given in (1).
Figure 4: Enlarged region of the tradeoff curve
Table 2: Sample objective function values for solutions not on the emerging tradeoff curve

<table>
<thead>
<tr>
<th>Dem Wtd Total Distance</th>
<th>Range in Assigned Demand</th>
<th>Dem Wtd Total Distance</th>
<th>Range in Assigned Demand</th>
<th>Objective Value of Dominated Solution (Section 4.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>343,644,501</td>
<td>4,395,066</td>
<td>343,164,597</td>
<td>4,149,267</td>
<td>539,189.20</td>
</tr>
<tr>
<td>349,570,306</td>
<td>4,777,473</td>
<td>343,164,597</td>
<td>4,149,267</td>
<td>6,436,439.28</td>
</tr>
<tr>
<td>364,988,119</td>
<td>4,769,154</td>
<td>369,916,031</td>
<td>3,633,030</td>
<td>5,057,182.46</td>
</tr>
<tr>
<td>392,881,776</td>
<td>4,962,960</td>
<td>369,916,031</td>
<td>3,633,030</td>
<td>23,004,220.42</td>
</tr>
</tbody>
</table>

5.4 Improvement algorithms employed in the heuristic

Three improvement algorithms are employed in the genetic algorithm outlined below and presented in Figure 5. These heuristics operate on each solution in the population. *Improve_median* performs single-node facility *exchanges* to improve the value of the median associated with an input solution. It ignores the impact of such improvements on the range. The improvement algorithm is analogous to that proposed by Teitz and Bart (1968).

*Improve_range* is similar to *Improve_median*, but it focuses on single demand-node *reassignments* of demand nodes to reduce the range in assigned demands. Once an improving reassignment is identified and performed, the algorithm executes a series of local searches within the neighborhoods formed by the facilities and their assigned demands to find the optimal 1-median locations within each neighborhood (Maranzana, 1965).
**Improve_median (solution)**
Repeat
  
  $Found \leftarrow true$
  Find best single facility exchange to reduce the value of the median
  If an exchange is found
    Perform the exchange
  Else $Found \leftarrow False$
  Until not $Found$

End *Improve_median*


**Improve_range (solution)**
Repeat
  
  $Found \leftarrow true$
  Find best single node reassignment to reduce the value of the range
  If a reassignment is found
    Perform the reassignment
    Within each new neighborhood, perform a Maranzana-like search to improve the median
  Else $Found \leftarrow False$
  Until not $Found$

End *Improve_range*


**Improve_both (solution, $\beta^{use}$)**
Repeat
  
  $Found \leftarrow true$
  Find best single node reassignment to reduce the value of
  $\beta^{use}\text{Solution.median} + (1 - \beta^{use})\text{Solution.range}$
  If a reassignment is found
    Perform the reassignment
    Within each new neighborhood, perform a Maranzana-like search to improve the median
  Else $Found \leftarrow False$
  Until not $Found$

End *Improve_both*

*Figure 5: Three key improvement heuristics used in the genetic algorithm*

Finally, *Improve_both* is similar to *Improve_range*, except that it tries to reduce a suitably weighted combination of the median and range objective function values. The choice of the input parameter, $\beta^{use}$, is outlined below in Section 5.6.
5.5 Generating an initial population of solutions

An initial population of solutions is generated with each solution representing a different set of chosen facility sites. Figure 3 is an example of one such population. Demand nodes are initially assigned to the nearest open facility in each solution. Next, each random solution is improved using Improve_median (Teitz and Bart, 1968). Then, node reassignments are considered using Improve_range. Finally, Improve_both is applied to each solution. Throughout the process, we ensure that there are no duplicate solutions (in terms of both facility locations and demand node assignments) in the initial population; in other words, each solution is unique.

5.6 Crossover and mutation

To generate a new solution, crossover and mutation operators are used. Two solutions are selected at random with a bias toward the better solutions. We let $V_{kt}$ be the objective function value of the $k$th solution at iteration $t$ (as outlined in Section 5.3) and $V_{t}^{\text{max}} = \max_{k \in P_t} \{V_{kt}\}$, where $P_t$ is the set containing the population of solutions at current iteration $t$. The probability, $q_{kt}$, of selecting solution $k$ at iteration $t$ as a parent solution is then given by

$$q_{kt} = \frac{\rho V_{t}^{\text{max}} - V_{kt}}{\rho |P_t| V_{t}^{\text{max}} - \sum_{k \in P_t} V_{kt}}$$

(15)
In our computational studies, we used $\rho = 1.1$. This value, just slightly larger than 1.0, ensures that all solutions (including the worst solution in the population at the given iteration) will have a non-zero probability of being selected.

Table 3 gives the selection probabilities for the eight solutions shown in Figure 4, assuming these represent the entire population of solutions. Shaded solutions are dominated. Note that their selection probabilities are lower than are those of the solutions on the tradeoff curve.

The algorithm selects two (non-identical) solutions at random using the probabilities given in (15). In the crossover operation, we create a new child solution from the two parent solutions. First, all sites that are shared by the two parent solutions are added to the child solutions. Next, sites that are in one, but not both, of the two parent solutions are randomly selected for addition to the child solution until the number of sites in the child solution equals $p$. Finally, with probability $q_{\text{mutate}}$, the mutation operator is used on the child solution. If mutation is performed, we select at random one site that is currently in the child solution and exchange it for a randomly selected candidate location that is not in the solution. Demands are then assigned to the nearest facility. The process is repeated until the sites represented in the child solution do not correspond to any solution in the population.
Table 3: Selection probabilities for solutions shown in Figure 4

<table>
<thead>
<tr>
<th>Solution</th>
<th>Closest Non-Inferior Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dem Wtd Total Distance</td>
<td>Range in Assigned Demand</td>
</tr>
<tr>
<td>343,644,501</td>
<td>4,395,066</td>
</tr>
<tr>
<td>349,570,306</td>
<td>4,777,473</td>
</tr>
<tr>
<td>364,988,119</td>
<td>4,769,154</td>
</tr>
<tr>
<td>392,881,776</td>
<td>4,962,960</td>
</tr>
<tr>
<td>369,916,031</td>
<td>3,633,030</td>
</tr>
<tr>
<td>343,164,597</td>
<td>4,149,267</td>
</tr>
<tr>
<td>333,593,976</td>
<td>4,465,682</td>
</tr>
<tr>
<td>333,147,453</td>
<td>4,843,232</td>
</tr>
</tbody>
</table>

If the child solution is neither the solution with the best median value nor the solution with the best range value in the population, the child solution is then improved using *Improve_both*.

The value of $\beta_{use}$ is computed as follows. Using the current best median (with median and range values $M^a$ and $R^a$ respectively) and current best range solution with median and range values of $M^b$ and $R^b$, we compute a value of $\beta$ that makes the weighted values the same on a line connecting these two solutions as shown in Figure 4. In other words, we solve

$$\beta M^a + (1 - \beta) R^a = \beta M^b + (1 - \beta) R^b$$

(16)

for $\beta$ to obtain,

$$\beta = \frac{R^b - R^a}{M^a - M^b + R^b - R^a}$$

(17)
If we tried to improve all child solutions by driving all in a direction perpendicular to the line shown in Figure 6, all solutions would tend to be the same. Thus, the value of $\beta$ that is used in the $\text{Improve\_range}$ algorithm is the value shown in (17) multiplied by a random number between $\theta$ and $\theta + \phi$. In other words, we compute:

$$\beta^{\text{use}} = \beta(\theta + U\phi)$$

(18)

where $U$ is a uniform [0,1] random variable. In our computational testing, we used $\theta = 0.002$ and $\phi = 4.998$. Larger values of $\beta$ place more weight on improving the median values, and smaller values place more weight on minimizing the range.

![Figure 6: Computing the nominal beta value to improve a solution](image)

Figure 6: Computing the nominal beta value to improve a solution
5.7 Population dynamics

After an initial population is generated and improved as outlined above, the algorithm enters the main processing loop. In each iteration, two primary tasks are accomplished. First, as will be explained below, the population may need to be expanded. Second, the crossover/mutation/improvement procedure outlined above is executed once to generate a new child solution.

As the algorithm progresses, more non-dominated solutions will be found. Rather than trying to prune the population as is done in some multi-objective genetic algorithms (Rabiee, Zandieh, and Ramezani, 2012; Gupta, 2012; Tang et al., 2016), we allow the population to expand to include all non-dominated solutions found so far (up to a maximum population size). To ensure diversity of the population, at each iteration of the algorithm, we further expand the population by adding randomly generated solutions to ensure that the ratio of the number of dominated solutions to non-dominated solutions is at a certain level (or we are using the maximum possible population size). In our tests, the ratio was set to 3 for the largest dataset and 4 for the smaller datasets. A maximum population size of 500 was permitted in all runs. The randomly generated solutions are improved using Improve_both after the crossover procedure is executed.

5.8 Termination of the genetic algorithm

The genetic algorithm terminates when any of three conditions is satisfied:

1. The number of iterations exceeds a user specified value.
2. The number of iterations during which there was no improvement in the tradeoff curve exceeds a user-specified value.

3. The execution time of the algorithm exceeds a user-specified value.

Figure 7 presents a pseudo-code for the entire genetic algorithm.

```
Genetic Algorithm
Initialize all model parameters including initial_population_size and maximum_population_size
Generate an initial population
   Use Improve_median, Improve_range, and Improve_both to improve the solutions in the initial population ensuring that no duplicate solutions are stored
While termination criteria are not met
   Expand population as needed by generating new random solutions. Assign demands to nearest open facilities
   Generate a new solution using crossover, mutation, and improvement
   If current_population_size < maximum_population_size
      Add child solution to the population
      Set temp_population_size ← current_population_size + 1
   Else If child solution is better than worst solution in the population
      Replace the worst solution with the child solution
   Improve all un-improved solutions using Improve_both
   If temp_population_size > current_population_size, eliminate the worst solution (as measured by the objective value described in Section 5.3).
End While
```

Figure 7: Pseudo-code for the genetic algorithm

6  COMPUTATIONAL RESULTS

In this Section, we outline computational results we have obtained using both the formal optimization model outlined in Sections 3 and 4 and the genetic algorithm described in Section 5. The results were obtained using the four datasets outlined in Section 1. Table 4 summarizes key characteristics of the four datasets. As indicated above, great circle distances were used in all cases with distances
rounded to the nearest mile except for the Ann Arbor dataset for which distances were rounded to the nearest 1/1000th of a mile.

We begin by presenting optimal results for selected instances of the smaller three datasets in Section 6.1 followed by results using the genetic algorithm in Section 6.2.

Table 4: Summary of key characteristics of the datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Nodes</th>
<th>Total Demand</th>
<th>Maximum Demand</th>
<th>Minimum Demand</th>
<th>Maximum Distance Between Any Pair of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann Arbor, MI</td>
<td>33</td>
<td>115,103</td>
<td>7,098</td>
<td>1,551</td>
<td>6.181</td>
</tr>
<tr>
<td>Michigan</td>
<td>83</td>
<td>9,883,640</td>
<td>1,820,584</td>
<td>2,156</td>
<td>449</td>
</tr>
<tr>
<td>150 city U.S.</td>
<td>150</td>
<td>58,196,530</td>
<td>7,322,564</td>
<td>119,363</td>
<td>2,731</td>
</tr>
<tr>
<td>880 node zip-3 U.S.</td>
<td>880</td>
<td>306,669,700</td>
<td>2,906,700</td>
<td>8</td>
<td>2,817</td>
</tr>
</tbody>
</table>

6.1 Optimal results

We solved three instances of the model in Section 3: the 33-node Ann Arbor problem with $p=5$, the 83-node Michigan dataset with $p=5$, and the 150-node USA dataset with $p=4$. All models were solved using OpenSolver (version 2.8.4; http://OpenSolver.org; Mason, 2012) in Microsoft Excel for Mac (version 14.6.4) on a MacBook Air (running IOS 10.11.6) with a 2.2 GHz Intel Core i7 and 8 Gb of 1600 MHz RAM. To simplify the structure of the model in Excel, we used the aggregate version of constraints (3) given by

$$
\sum_{i \in I} Y_{ij} \leq |I| X_j \quad \forall j \in J
$$

(19)
While the use of constraints (19) instead of constraints (3) results in a weaker linear programming formulation, the use of (3) did not appreciably improve the solution times and significantly degraded the problem setup times in OpenSolver.

Table 5 summarizes the results. For the 33-node Ann Arbor dataset, we set

\[ r_{\text{max},m+1} = \left\lfloor 0.99 (U^m - L^m) \right\rfloor, \]

where the superscript \( m \) indicates the iteration number. In other words, the value of \( r_{\text{max}} \) was set to 99% of the range found in the previous solution, i.e., \( \alpha = 0.99 \). For the 83-node Michigan dataset, we used

\[ r_{\text{max},m+1} = \left\lfloor 0.98 (U^m - L^m) \right\rfloor \]

for the first 52 solutions and then used

\[ r_{\text{max},m+1} = \left\lfloor 0.90 (U^m - L^m) \right\rfloor \]

for the next 22 solutions. For the final 11 solutions, \( r_{\text{max}} \) was set to a round number smaller than the previously found range. Details are available from the authors. Finally, for the 150-node USA dataset, we used

\[ r_{\text{max},m+1} = \left\lfloor 0.90 (U^m - L^m) \right\rfloor \]

for all iterations.

For these three instances, the smallest range we found was less than 0.04% of the range associated with the optimal \( p \)-median problem without restrictions on the range of the assigned demand. To achieve this reduction, the median objective function value increases by 40%, 29%, and 4.5% for the 33-node, 83-node, and 150-node problems respectively. In particular, for a 4.5% increase in the median objective function value for the 150-node problem, we can reduce the range in the assigned demand 99.97%. Many good compromise solutions exist that balance minimizing the demand-weighted average distance and the range in assigned demand. Table 6 presents one compromise solution for each of the three datasets.
For example, for the 150-node problem, we can reduce the range over 99% while incurring only a three percent degradation in the median objective function value.

Relatively few facility sites were selected. For example, in the 150-node problem, we found 39 solutions, each with 4 facilities. Only six different sites were used in at least one of the solutions: New York, NY; Los Angeles, CA; Dallas, TX; Garland, TX; Indianapolis, IN; and Fort Wayne, IN. Figures 8-10 plot the frequency with which various sites were used in these three runs. It is worth noting that for each of these datasets, there are three locations that appear in nearly every solution.

We recorded the solution times for the first 74 solutions for the 83-node Michigan problem both with and without the constraints identified in Section 4. Adding these constraints reduces the total solution time for these 74 problems by almost 90%, on average. In addition, the reduction seems to be the greatest for the problems with the smallest values of $r^{\text{max}}$. This makes sense since the right hand side values of constraints (12) and (13) depend linearly on $r^{\text{max}}$; smaller values of $r^{\text{max}}$ result in tighter constraints. The addition of these constraints clearly reduces the solution time. For the larger 150-node dataset, all runs used these constraints. Solution times were not recorded for the 33-node Ann Arbor dataset. The CPU time needed to find the 39 solutions for the 150 node dataset was large even when the added constraints were used. Therefore, we did not run the model for this problem instance without these constraints.
Table 5: Summary of optimal results for three datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Facilities</th>
<th>Number of solutions found</th>
<th>Demand-Weighted Average Distance</th>
<th>Range Value</th>
<th>Demand-Weighted Average Distance</th>
<th>Range Value</th>
<th>Number of different sites used</th>
<th>CPU time without Added Constraints</th>
<th>CPU time with Added Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>33 node Ann Arbor</td>
<td>5</td>
<td>48</td>
<td>0.78</td>
<td>24,788</td>
<td>1.09</td>
<td>8</td>
<td>12</td>
<td>not recorded</td>
<td>113,932.80</td>
</tr>
<tr>
<td>83 node Michigan</td>
<td>5</td>
<td>85</td>
<td>26.37</td>
<td>3,767,764</td>
<td>33.93</td>
<td>65</td>
<td>18</td>
<td>37,372.43</td>
<td></td>
</tr>
<tr>
<td>150 node US</td>
<td>4</td>
<td>39</td>
<td>241.94</td>
<td>5,726,961</td>
<td>252.90</td>
<td>1,551</td>
<td>6</td>
<td>not recorded</td>
<td>36,426.33</td>
</tr>
</tbody>
</table>

Notes: Times for 83-node Michigan dataset are for the 74 solutions with the largest range values. For tighter values of the range, the algorithm without the added constraints took excessively long to solve. The total solution time for all 85 solutions with the added constraints was 37,372.43 seconds.

Table 6: Possible compromise solutions for three datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Demand-Weighted Average Distance</th>
<th>Range Value</th>
<th>% Degradation in Median Value</th>
<th>% Improvement in Range Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>33 node Ann Arbor</td>
<td>0.86</td>
<td>861</td>
<td>10.58%</td>
<td>96.53%</td>
</tr>
<tr>
<td>83 node Michigan</td>
<td>27.04</td>
<td>1,399,404</td>
<td>2.52%</td>
<td>62.86%</td>
</tr>
<tr>
<td>150 node US</td>
<td>249.15</td>
<td>51,429</td>
<td>2.98%</td>
<td>99.10%</td>
</tr>
</tbody>
</table>

1 As compared with best median solution.
Figure 8: Distribution of the number of solutions using various sites for the Ann Arbor dataset

Figure 9: Distribution of the number of solutions using various sites for the Michigan dataset
Finally, Figures 11-13 plot the tradeoff curves for these three datasets. The graphs also plot the number of demand nodes that are assigned to the closest open facilities. This number generally goes down in all three cases. In general, the model reduces the range by reassigning demand nodes to facilities and, only secondarily, by changing the location of the facilities.
Figure 11: Tradeoff curve for Ann Arbor dataset with five facilities

Figure 12: Tradeoff curve for Michigan dataset with five facilities
6.2 Genetic algorithm results

The long computation times – even with the constraints of Section 4 – led us to develop the genetic algorithm outlined in Section 5. The genetic algorithm was coded in Delphi XE5. All results were run on a Dell Precision M4500 machine with an Intel Core i5 CPU running at 2.67 GHz and 4 Gb of RAM.

Table 7 lists the key input parameters for the genetic algorithm. Table 8 summarizes the results of using the genetic algorithm on the 20 problem instances shown in Table 1. On average, we found solutions for which the range was 0.63% of the range associated with the optimal median solution with approximately a 60% increase in the median value. In fact, increasing the average distance by less than 11 percent allows us to reduce the range to under 8 percent of the value associated
with the optimal median solution when we average over all 20 problem instances. *Significant reductions in the range can be achieved with only modest increases in the average distance.*

**Table 7: Input values for the genetic algorithm**

<table>
<thead>
<tr>
<th>Termination Criteria</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of iterations</td>
<td>25,000</td>
</tr>
<tr>
<td>Maximum number of iterations without a change in the tradeoff curve</td>
<td>1,000</td>
</tr>
<tr>
<td>Maximum solution time (sec.)</td>
<td>10,800</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Key Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial population size</td>
<td>25</td>
</tr>
<tr>
<td>Maximum population size</td>
<td>500</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.1</td>
</tr>
<tr>
<td>$\theta$</td>
<td>0.002</td>
</tr>
<tr>
<td>$\varphi$</td>
<td>4.998</td>
</tr>
<tr>
<td>Desired ratio of dominated to non-dominated solutions</td>
<td>All datasets except the largest 880-node dataset</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Random number seed</td>
<td>9753137</td>
</tr>
</tbody>
</table>

For the three smallest datasets, the algorithm generally terminated because the number of iterations without a change in the tradeoff curve was attained. For the 880-node dataset, all runs timed out at the 3-hour limit. Finally, as in the case of the optimal solutions, relatively few different sites were chosen in each of these problem instances.

Figure 14 compares the optimal and heuristic tradeoff curve for the 83-node Michigan case with $p=5$. The curves seem to lie on top of each other except for the very small range values shown in Figure 15. The heuristic seems to perform worse than the optimal algorithm for ranges below 20,000, which is about 0.5 percent of the range associated with the 5-median solution (3,767,764). It seems that the
genetic algorithm performs quite well, with the possible exception of cases with very small ranges.

Finally, we plot six solutions for the 880-node 2-median problem. Figure 16 plots the tradeoff curve with six solutions highlighted. The maps associated with these six solutions are shown in Figure 17. There is generally a clear demarcation line between the demand nodes assigned to each of the locations, though there is a slight overlap in the regions for the last two solutions plotted, solutions 135 and 137. Perhaps the most egregious is 3-digit zip code 770 in TX which is assigned to the facility at node 151, PA instead of the closer facility at node 730, OK in map 137. Also, in map 135, there is some overlap in the Gulf coast states of Florida, Georgia, and Alabama. It is nevertheless worth noting that we can reduce the range in assigned demand from over 161 million to 0 in this problem instance. There is also a clear pattern to the facilities selected in this instance as we reduce the range. The western facility starts in California and moves eastward ending up in Oklahoma. The eastern facility begins in West Virginia and moves eastward to Virginia (in map 135) and then north to Pennsylvania (in map 137).
Table 8: Summary of the genetic algorithm results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Facilities</th>
<th>Number of solutions found</th>
<th>Best Median Solution</th>
<th>Best Range Solution</th>
<th>Number of different sites used</th>
<th>CPU Time</th>
<th>Iterations</th>
<th>Iterations Without Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>33 node Ann Arbor</td>
<td>2</td>
<td>18</td>
<td>1.396</td>
<td>45,559</td>
<td>1.578</td>
<td>9</td>
<td>9</td>
<td>10.0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>19</td>
<td>1.081</td>
<td>35,117</td>
<td>1.307</td>
<td>9</td>
<td>8</td>
<td>13.6</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>28</td>
<td>0.884</td>
<td>21,372</td>
<td>1.386</td>
<td>14</td>
<td>13</td>
<td>31.3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>28</td>
<td>0.780</td>
<td>24,788</td>
<td>1.131</td>
<td>14</td>
<td>14</td>
<td>22.6</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>23</td>
<td>0.703</td>
<td>14,503</td>
<td>1.052</td>
<td>14</td>
<td>14</td>
<td>36.0</td>
</tr>
<tr>
<td>83 node Michigan</td>
<td>2</td>
<td>14</td>
<td>42.518</td>
<td>2,915,260</td>
<td>57.897</td>
<td>2</td>
<td>5</td>
<td>23.1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>61</td>
<td>34.541</td>
<td>5,403,060</td>
<td>39.496</td>
<td>346</td>
<td>13</td>
<td>87.7</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>87</td>
<td>29.564</td>
<td>3,974,700</td>
<td>55.996</td>
<td>593</td>
<td>17</td>
<td>306.5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>70</td>
<td>26.375</td>
<td>3,767,764</td>
<td>51.295</td>
<td>1,871</td>
<td>23</td>
<td>343.1</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>67</td>
<td>23.361</td>
<td>4,361,940</td>
<td>41.843</td>
<td>209,210</td>
<td>21</td>
<td>284.0</td>
</tr>
<tr>
<td>150 node USA</td>
<td>2</td>
<td>39</td>
<td>454.014</td>
<td>23,117,668</td>
<td>852.781</td>
<td>68</td>
<td>12</td>
<td>88.4</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>29</td>
<td>315.571</td>
<td>5,109,092</td>
<td>615.675</td>
<td>584</td>
<td>14</td>
<td>283.9</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>47</td>
<td>241.944</td>
<td>5,726,961</td>
<td>339.584</td>
<td>10,626</td>
<td>10</td>
<td>726.4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>128</td>
<td>206.056</td>
<td>9,043,188</td>
<td>470.771</td>
<td>19,727</td>
<td>18</td>
<td>2,992.2</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>110</td>
<td>184.626</td>
<td>10,785,133</td>
<td>312.492</td>
<td>34,758</td>
<td>33</td>
<td>1,753.6</td>
</tr>
<tr>
<td>880 node zip3 USA</td>
<td>2</td>
<td>138</td>
<td>477.418</td>
<td>161,164,828</td>
<td>617.521</td>
<td>0</td>
<td>21</td>
<td>10,811.6</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>87</td>
<td>367.545</td>
<td>69,710,209</td>
<td>422.995</td>
<td>62</td>
<td>28</td>
<td>10,821.2</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>56</td>
<td>306.438</td>
<td>15,327,642</td>
<td>610.168</td>
<td>884</td>
<td>17</td>
<td>10,802.5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>35</td>
<td>250.638</td>
<td>26,516,920</td>
<td>348.371</td>
<td>1,169</td>
<td>26</td>
<td>10,810.0</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>40</td>
<td>223.127</td>
<td>57,483,994</td>
<td>466.514</td>
<td>6,908</td>
<td>79</td>
<td>10,921.0</td>
</tr>
</tbody>
</table>

44
Comparison of Optimal and Heuristic Tradeoffs: Michigan data, five facilities

Figure 14: Comparison of the optimal and heuristic tradeoff curves for the 83-node Michigan $p=5$ case
Figure 15: Comparison of the optimal and heuristic tradeoff curves for the 83-node $\rho=5$ Michigan case, focused on the very small range values
Figure 16: 880-node 2-median tradeoff curve with six solutions highlighted (whose maps are shown in Figure 17)

Figure 17a: Map of solution 1
Figure 17b: Map of solution 35

Figure 17c: Map of solution 72
Figure 17d: Map of solution 114

Figure 17e: Map of solution 135
7 CONCLUSIONS AND FUTURE WORK

In this paper, we have shown that the range in assigned demand associated with the $p$-median problem can, and often is, very large. This can be reduced significantly with very little degradation in the median value. When averaged over 20 instances of four datasets ranging in size from 33 to 880 nodes with between 2 and 6 facilities, increasing the median by as little as 11 percent results in over a 92 percent reduction in the range.

An integer-linear programming formulation of the model was given Section 3 and additional constraints that significantly tighten the formulation were outlined in Section 4. Even with these constraints, solution times can be large for large problem instances and for tight constraints on the range. For this reason, we introduced a genetic algorithm-based procedure in Section 5. Computational tests in Section 6
showed that the genetic algorithm performed well when compared to optimal solutions.

There are a number of areas for future work. First, the parameters of the genetic algorithm can be further tuned to identify values that might result in better solutions for very small range limits. In particular, the value of $\phi$ can be reduced so that the algorithm is more likely to focus on reducing the range.

More importantly, we can add constraints that force the model either to use closest assignments in all cases or to enforce contiguity constraints to ensure that the demand areas assigned to the selected facilities are contiguous. Finally, there are opportunities for finding optimization-based heuristics (e.g., Lagrangian relaxation) algorithms to solve the proposed formulation.

Finally, we believe that location modeling will move in three primary directions in the future. First, while many multi-objective location models have been developed, we believe that this trend will increase in the coming years. Second, we believe that location models will increasingly be viewed as a component of larger multi-problem models that encompass vehicle routing, facility scheduling, inventory management, and customer service. Third, we believe that location models will be and should be tested against real data and solution algorithms may exploit the structure of real data in the future.

8 REFERENCES


**Acknowledgments**

The authors would like to thank the referees for their helpful comments. The work of the second author was partially supported by the National Science Foundation Graduate Research Fellowship Program under Grant DGE 1256260. The opinions expressed in this paper do not necessarily reflect the views of the National Science Foundation.