# $P$-Median Problems and Solution Strategies

ZACH SIEGEL (ZACHARY.EDMUND.SIEGEL@GMAIL.COM #805435913)
MGMTPHD242A - *Models for Operations Planning, Scheduling, and Control*
Prof. Kumar Rajaram (kumar.rajaram@anderson.ucla.edu)
Winter 2021

## 1 Introduction

Facility location problems have been widely and closely studied throughout the 20th century in theoretical and practical settings. Numerous reviews have been published. These problems tend to have a solution space that grows exponentially in the number of desired facilities and to be NP-hard. Some variants of facility location admit natural heuristic approximations. Here we focus on one such variant, *$P$-Median* problems, and review solution methods.

In general, a facility location problem seeks to select from a finite set of facilities to optimally serve demand sites. Usually the facilities and demand sites are distinct, though in some variants they are from the same set. There is always a pairwise cost of each facility serving each demand site, which often corresponds to the distance between the two physical entities, hence the "facility location" framing. There may be a fixed cost associated with selecting each facility, and there may be a limit on the number of allowable facilities. Finally, there may be a limit on the demand that can be satisfied by any single facility, which is known as *capacitated* facility location, as opposed to *uncapacitated*. Other concerns related to the demand-serving process may be captured in a "facility location" context, such as costs and capacities associated with facility echelons, such as warehouses; these variants allude to lot-sizing problems and are not the focus of this review.

In all cases, the problem seeks a many-to-one matching between facilities and demand sites. The objective may be to minimize cost, maximize some other notion of utility, or to maximize "coverage" of demand sites by the selected facilities.

When the number of facilities is fixed to $P$, the problem is referred to as a $P$-median problem, which is articulated in Section 2 and is the focus of this report.

### 1.1 Other Reviews of Facility Location

Facility location problems have been studied widely. In fact, not only have countless papers been published on the topic, but a sizeable body of *reviews* on the topic have been published.

ReVelle's [13] from 1970 is possibly the earliest review on the topic; that author's 2008 paper [14] reviews the state of the art. Cornuejols, Fisher, and Nemhauser's 1977 paper [3] describes heuristics, rigorously examines the performance of a greedy heuristic, and proposes a convincing but nonstandard method to evaluate heuristic performance. Kariv and Hakimi's 1979 [8] mostly reviews and extends Hakimi's seminal 1964 and 1965 papers [6] and [7]. Aikens' 1985 review [2] covers dynamic, multi-echelon, and multi-product variants of facility location problems, as well as several side constraints, focusing on mathematical programming articulations. Hesse Owen and Daskin's 1998 review [12] covers several problem variants and discusses both mathematical programming and queueing perspectives. Klose and Drexl's 2005 review is almost a reprise of Aikens' [2] in that it articulates several complicating variants of facility location problems.

## 2 $P$-Median Problem

The $P$-median can be formulated as follows.

$$\underset{x,y}{\text{Minimize}} \qquad \sum_{i\in\mathscr{I}}\sum_{j\in\mathscr{J}} d_j c_{ij} x_{ij} \tag{1a}$$

$$\text{s.t.} \qquad \sum_{i\in\mathscr{I}} x_{ij} = 1 \;\forall\; j \in \mathscr{J} \tag{1b}$$

$$x_{ij} \le y_i \;\forall\; i \in \mathscr{I}, j \in \mathscr{J} \tag{1c}$$

$$\sum_{i\in\mathscr{I}} y_i = P$$

$$0 \le x_{ij} \le 1 \;\forall\; i \in \mathscr{I}, j \in \mathscr{J}$$

$$y_i \in \{0,1\} \;\forall\; i, j.$$

with the system parameters

$$\begin{aligned} p &= \text{number of facilities to locate} \\ \mathscr{I} &= \text{set of indices of candidate facility locations} \\ \mathscr{J} &= \text{set of demand indices} \end{aligned}$$

and with the decision variables defined as

$$x_{ij} = \text{the fraction of the demand of customer } j \text{ supplied from facility } i$$

$$y_i = \begin{cases} 1 : \text{facility is located at candidate site } i \\ 0 : \text{otherwise.} \end{cases}$$

The $P$-median problem formulated above as (1) can also be cast as an optimization problem of a *set function*. In later sections, we will demonstrate properties of this set function that yield performance guarantees for some heuristics over a wide range of problem variants.

First, define the objective as a function of $X \subseteq \mathscr{I}$, the subset of facility indices indicating which facilities are selected:

$$z(\mathscr{J}, X) = \sum_{j \in \mathscr{J}} d_j \min_{i \in X} \{c_{ij}\}. \tag{2}$$

Then the $P$-median problem (1) can be formulated as

$$\begin{aligned} \underset{X \subseteq \mathscr{I}}{\text{Minimize}} \qquad & z(\mathscr{J}, X) = \sum_{j \in \mathscr{J}} d_j \min_{i \in X} \{c_{ij}\} \\ \text{s.t.} \qquad & |X| \qquad = P. \end{aligned} \tag{3}$$

## 3 Proof of NP-Hardness

A full proof that the $P$-Median problem is NP-hard is given in Kariv and Hakimi's 1979 reivew [8]. They show, in fact, that the problem is NP-hard even if the graph representing facilities and demand sites is

- planar
- has degree at most 3
- has edge weights (pairwise distances) all equal to 1
- have vertex weights (demands) of 1.

This very conservative reduction makes clear that even highly regular and limited special cases elude efficient solutions. Note that the formulation in this review does not mention a graph, but the natural translation is to set a distance $c_{ij}$ equal to $\infty$ if an edge does not exist in the graph. Most modern treatments eschew the graph setting entirely or lets pairwise distances denote shortest path distances; the NP-hardness despite the special graph structure described above is evidence that a graph structure does not yield significant computational difference. Their reduction proof is outlined below.

Note that their definition of a $P$-median is a subset of nodes that minimizes the vertex-weighted (i.e. demand-weighted) sum of distances between nodes in and out of the subset. In particular, the "candidate facility locations" are the same set as the demand nodes, i.e. $\mathscr{I} = \mathscr{J}$. To translate an instance of this form to an instance of the $P$-median facility location problem described in earlier sections, in which demand nodes and candidate facility locations are separate sets, one need only set distances between each pair of candidate locations equal to $\infty$, ensuring that an optimal solution will only serve demand nodes via candidate facility nodes.

*Proof.* First, note Garey and Jonshon's 1979 proof in [5] that the *dominating set problem* is NP-complete even on a planar graph of maximum degree 3 ([8] reproduces a proof of this lemma due to Garey and Johnson, then goes on to their reduction to the $P$-median problem):

> Given a graph $G(V, E)$ and a positive integer $p$, $(1 < p < n)$, does there exist a subset $V_p^*$ of $p$ or less vertices such that each vertex of $G$ is either in $V_p^*$ or is adjacent to a vertex of $V_p^*$. Garey and Johnson have proved the following result [5]:
> Lemma. Let $G(V, E)$ be a planar graph of maximum vertex degree 3 and let $p$ be an integer $1 < p < n$. The problem of finding if there exists in $G$ a dominating set of cardinality $p$ is NP-complete.

Finally, the reduction result: "The problem of finding a $P$-median is NP-hard even in the case when the network is a planar graph of maximum vertex degree 3 all whose edges are of length 1 and all whose vertices have weight 1":

> Let $G(V, E)$ be a planar graph of maximum vertex degree 3, all of whose edges are of length 1 and all of whose vertices have weight 1. We need only to show that the problem of whether there exists a dominating set of cardinality $p$ in $G$ is polynomial time reducible to the problem of finding a $P$-median of $G$. For, let $V_P$ be an arbitrary subset of $P$ vertices of $G$. Then, by the special structure of $G$, we have $H(V_P) = \sum_{v \in V} d(v, V_P) \geq n - P$. Thus, if there exists any subset $V_P^*$ for which $H(V_P^*) = n - P$ holds, then $V_P^*$ is a $P$-median of $G$. On the other hand, the equation $H(V_P^*) = n - P$ is satisfied if and only if $d(v, V_P^*) = 1$ for each of the $n - P$ vertices not in $V_P^*$, namely if and only if $V_P^*$ is a dominating set of cardinality $P$ in $G$. Therefore, there exists a dominating set of cardinality $P$ in $G$ if and only if the distance-sum $H(V_P^*)$ of a $P$-median $V_P^*$ of $G$ is $n - P$. This shows that the problem of finding a dominating set in $G$ is polynomial time reducible to the problem of finding a $P$-median of $G$, and thus the latter problem is NP-hard. Q.E.D.

□

# 4 Algorithms

The following algorithms are reproduced from [4]. Several of them are compared in Figure 6, which is a table from [8], in which these algorithms were implemented. We did not reproduce the entire battery of experiments, but implemented some of the algorithms and evaluated their performance.

## 4.1 LP Relaxation

Each of the algorithms to follow is compared to a MILP implemented exactly as (1) above in Gurobi. Of course the problem is NP-hard, as proved in the previous section, which would wholly contradict the correctness of a continuous relaxation of the MILP formulation. Nonetheless, Revelle's 1970 article [13] claims

> The formulation presented here makes use of linear programming to optimally locate central facilities on the road network. In the unlikely event of a non-integer solution, a branch-and-bound scheme is recommended to resolve the problem with integers. With the linear programming formulation, one can take any heuristic solution and tell whether it is optimal.

While LP relaxations are a common and nearly universal optimization strategy, it is extremely bold to refer to "the unlikely event of a non-integer solution" in the context of an NP-hard problem. No analytical probabalistic argument is made regarding the actual likelihood of fractional variables nor on the performance of a thresholded LP relaxation solution. In 1970, even an LP formulation was not necessarily tractible, and the authors offer a row-generation scheme, which to an extent dates their priorities.

I was curious whether numerical examples bore out their claim that integer solutions are likely even without enforcement, and was surprised that it was often the case. Anecdotally, it seems that very small problems (of the scale ReVelle was likely able to quickly solve on computers at the time), integer solutions are quite common. On larger instances (both with more nodes in total, and more nodes selected), they are extremely unlikely, as can be seen in Figure 1.

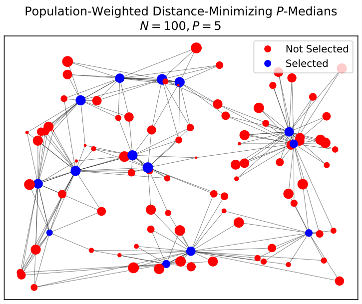
```
N=100,P=5
NON-INTEGER SOLUTION at trial 3/100. Nonzero values {0.5, 1.0}
NON-INTEGER SOLUTION at trial 11/100. Nonzero values {0.25, 0.5, 0.75}
NON-INTEGER SOLUTION at trial 16/100. Nonzero values {0.5, 1.0}
NON-INTEGER SOLUTION at trial 27/100. Nonzero values {0.5, 1.0}
NON-INTEGER SOLUTION at trial 44/100. Nonzero values {0.5, 1.0}
NON-INTEGER SOLUTION at trial 46/100. Nonzero values {0.5, 1.0}
NON-INTEGER SOLUTION at trial 77/100. Nonzero values {0.667, 1.0, 0.333}
NON-INTEGER SOLUTION at trial 90/100. Nonzero values {0.5, 1.0}
8/100 non-integer valued

N=1000,P=5
NON-INTEGER SOLUTION at trial 2/15. Nonzero values {0.125, 0.25, 0.188, 0.062,
0.312}
NON-INTEGER SOLUTION at trial 14/15. Nonzero values {0.333, 0.667}
2/5 non-integer valued

N=1000,P=50
NON-INTEGER SOLUTION at trial 0/5. Nonzero values {0.5, 1.0}
NON-INTEGER SOLUTION at trial 1/5. Nonzero values {0.5, 1.0}
NON-INTEGER SOLUTION at trial 3/5. Nonzero values {0.5, 1.0}
NON-INTEGER SOLUTION at trial 4/5. Nonzero values {0.5, 1.0}
4/5 non-integer valued
```

**Figure 1:** *The output from three trials of randomly-generated instances of the $P$-median problem LP relaxation. On the smalles instance, which selects $P = 5$ out of $N = 100$ facilities, fewer than 10% of solutions contain non-integer valued variables. On the largest instance, which selects $P = 50$ out of $N = 1000$ facilities, 80% of instances contained non-integer valued variables.*

Note that ReVelle addresses the $P$-Median problem in which candidate facility locations and demand nodes are the same (and selected nodes self-assign their demand), and that is the variant that was implemented. To get a sense of whether reasonably central facilities were assigned nonzero but non-integral demand (rather than spreading it out evenly) in instances with optimal non-integer valued variables, plots were generated in which demand at each node is represented by the size of the node. Some of those plots are included in Figure



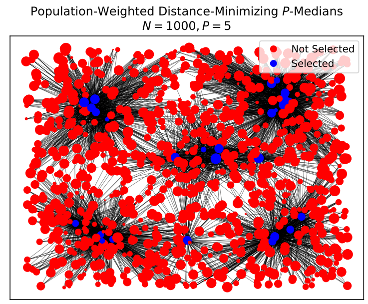




**Figure 2:** *In both cases, the objective was to select* 5 *nodes as facilities to serve* $N = 100$ *demand nodes on the left and* $N = 1000$ *demand nodes on the right. In both cases, the continuous relaxation did not result in an integral solution, though many nodes were assigned zero demand, alluding to the potential usefulness of this solution as a warm-start for any other heuristic method.*

## 4.2 Greedy/Myopic Algorithm

1 `Set` $X \leftarrow \emptyset$ `#` $X$ `is the set of locations to be used`

2 `Find` $i^* = \arg\min_{i \in \mathscr{I}} \{z(\mathscr{J}, X \cup \{i\})\}$

3 `Set` $X \leftarrow X \cup \{i^*\}$

4 `If` $|X| < p$`, go to Step 2; else stop.`

The greedy algorithm can perform quite poorly. Often the first "best" facility is far from the centers of any possible clusters of demand. The numerical experiment depicted in Figure 3 demonstrate this algorithm performing quite poorly in randomized examples.

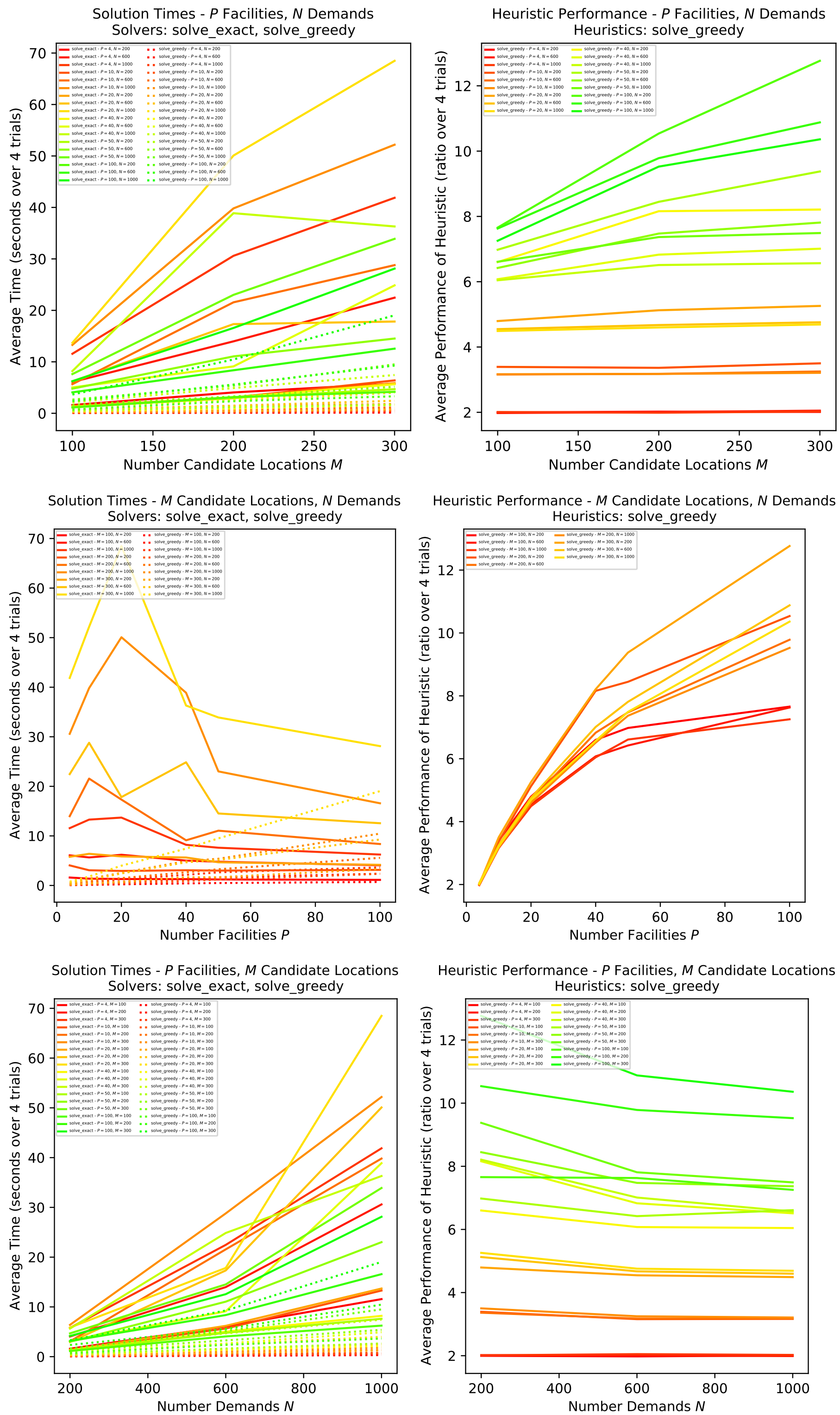



**Figure 3:** *Average time-to-solve alongside performance ratio for the greedy heuristic. In the three panes, the number of candidate locations $M$, the number of selected facilities $P$, and the nubmer of demands $N$ are varied as the other two are held constant. These are all representations of the same trials and show how these three (the only three) system size parameters affect the difficulty of solution and performance of this heuristic.*

Note in all the panes of Figure 3 that the time-to-solve for the greedy solution is less than half that of the MILP solver (Gurobi). Also notice that the minimum objective obtained by the greedy solution is as high as 12 times the optimal! The greedy solution is not bounded in its performance ratio, as is discussed below and demonstrated in Figure 4.

Note in the second pane of Figure 3 that as the number of facilities increases, the time to solve the $P$-Median problem using the greedy heuristic naturally increases linearly (more greedy solutions need to be made). By contrast, solving using a branch-and-bound MILP solver can result in reduced times for higher $P$. Indeed, the greatest combinatorial search space for this problem is when $P = \frac{M}{2}$, as there are $\binom{M}{P}$ possible choices for facilities. The Gurobi solver used to generate these plots does exhibit the expected bimodal behavior, but even over the average of many trials, it does not reach its maximum at $P = \frac{M}{2}$ nor is it exactly bimodal.

Note in the third pane of Figure 3 that as the number of demands $N$ increases for a fixed number of facilities $P$ out of $M$ candidate locations, the performance improves.

In Figure 4, the greedy algorithm can at first equivalently select any of the candidate locations, including the middle facility. If $P = 2$ facilities are to be selected, then the solution with objective $z = 0$ (the left and right facilities) may not be chosen, meaning the ratio of the objective to the best-possible objective is infinite.

Perhaps motivated in part by examples like this, Cornuejols, Fisher, and Nemhauser's 1977 paper [3] suggests evaluating heuristic performance with an alternative method. Rather than dividing performance of the heuristic solution by the optimal solution, they propose to subtract from both a "baseline" objective value, obtained using some obvious naive method, or even using a "worst possible" solution. The example in Figure 4 may have an infinitely bad "performance ratio", but this obscures the fact that the greedy solution selected the worst possible solution and the triviality of improvement.

It is notable that the local search *exchange algorithm* listed below in Section 4.3 would of course obtain the optimal solution in this simple example.

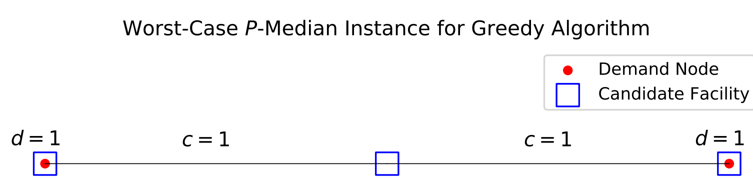



**Figure 4:** *A worst-case instance for selecting $P = 2$ facilities, in which the greedy algorithm performs arbitrarily poorly.*

## 4.3 Local Search Heuristics

A natural extension of any heuristic solution is local search, which seeks improvement to a given solution. The input to a local search is a solution, which here is a set $X$ of selected facilities. Two such local search heuristics are presented.

**Neighborhood Search Algorithm**

Input $X$ # $X$ is a set of $P$ facility locations
Set $N_i \leftarrow \emptyset,\ \forall\, i \in X$ # $N_i$ will be the set of demand nodes for which candidate site $i$ is the closest open facility
For $j \in \mathscr{J}$ do
    Set $i^* \leftarrow \arg\min_{i \in X}\{c_{ij}\}$
    Set $N_{i^*} \leftarrow N_{i^*} \cup \{j\}$
End For
Set $X^{new} \leftarrow \emptyset$ # $X^{new}$ will be the set of new facility locations
For $i \in X$ do
    If $|N_i| > 0$ then
        Find $k^* = \arg\min_{k \in N_i} z(N_i, \{k\})$
        Set $X^{new} \leftarrow X^{new} \cup \{k^*\}$
    End If
End For
If $X \neq X^{new}$ then set $X \leftarrow X^{new}$ and go to Step 2; else stop.

This *neighborhood search* algorithm addresses the variant of the $P$-median problem in which the demand nodes and the candidate facility locations are the same set of nodes, and the demand at each node is the same: the aim is to choose a set of central demand nodes to serve as facility locations. A *neighborhood* consists of a candidate location along with all the demand nodes for which that candidate location is the closest. On each iteration, within each neighborhood, the best possible (most central) demand node is assigned to be the candidate location. Between iterations, as the candidate locations change, so may the neighborhoods. The algorithm stops when it converges, which it is guaranteed to do, as it improves its objective at each non-stationary iteration by a non-vanishing amount.

According to [4], the neighborhood search variant of local search was first proposed in [10].

**Exchange Algorithm**

Input: $X$ # $X$ is a set of $P$ facility locations
For $i \in X$ do
    For $k \in \mathscr{I} \backslash X$ do
        If $z(\mathscr{J}, X) > Z(\mathscr{J}, X \cup \{k\} \backslash \{i\})$ then
            Set $X \leftarrow X \cup \{k\} \backslash \{i\}$ and stop
        End If

```
    End For
End For
```

This algorithm is the most obvious and common local search heuristic: pairwise interchange. This algorithm applies to the variant of the $P$-median problem defined in formula (1) (or (3) with set notation), in which demand nodes and candidate facilities may be distinct, though this heuristic may also apply to the variant in which they are the same. When the candidate facilities and demand nodes are the same and demand is constant, this algorithm can be compared to the *Neighborhood Search Algorithm* defined above: according to [4], the *Exchange Algorithm* tends to perform better.

As written above, the *first* possible improving interchange is made to the solution; however, it is possible to make the *best* possible interchange. The *Best Exchange Algorithm* is written below. It is also possible to define the *exiting* and *entering* facility location using any combination of *first* and *best*, e.g. adding the candidate location that most improves the current solution as a $P+1$-Median, then removing the first node encountered whose removal results in an improvement over the previous $P$-Median.

**Best Exchange Algorithm**

1 `Input:` $X$ `#` $X$ `is a set of` $P$ `facility locations`

2 `Set` $(i^*, k^*) \leftarrow \arg\min_{i \in X, k \in \mathscr{I} \setminus X} z(\mathscr{J}, X \cup \{k\} \setminus \{i\})$ `#` $(i^*, k^*)$ `is the best pair of` *`exiting`* `and` *`entering`* `candidate locations`

3 `Set` $X^{new} \leftarrow X \cup \{k^*\} \setminus \{i^*\}$

4 `If` $X \neq X^{new}$ `then set` $X \leftarrow X^{new}$ `and go to Step 2; else stop.`

Note that the $\arg\min$ in Step 2 of the best exchange algorithm somewhat obscures the $|X| \cdot |\mathscr{I}|$ comparisons required to generate the best pair of exiting and entering facilities, $(i^*, k^*)$.

## 4.4 Lagrangian Heuristic

Daskin's 2013 book [1] describes two different Lagrangian heuristic to solve the $P$-Median problem. In one of these, they describe that when relaxing constraint (1c) ($x_{ij} \leq y_i \ \forall \ i, j$), the problem decomposes by facility. In the other, which we shall focus on, they relax the constraint (1b) requiring that every demand be assigned a facility in formulation (1), to obtain the following problem:

$$
\begin{aligned}
\underset{\lambda \geq 0}{\text{Maximize}}\ \underset{x,y}{\text{Minimize}} \quad & \sum_{i \in \mathscr{I}} \sum_{j \in \mathscr{J}} d_j c_{ij} x_{ij} + \sum_{j \in \mathscr{J}} \lambda_j (1 - \sum_{i \in \mathscr{I}} x_{ij}) && \text{(4a)} \\
& = \sum_{i \in \mathscr{I}} \sum_{j \in \mathscr{J}} (d_j c_{ij} - \lambda_j) x_{ij} + \sum_{j \in \mathscr{J}} \lambda_j && \text{(4b)} \\
\text{s.t.} \quad & x_{ij} \leq y_i \ \forall\ i \in \mathscr{I}, j \in \mathscr{J} \\
& \sum_{i \in \mathscr{I}} y_i = P \\
& 0 \leq x_{ij} \leq 1 \ \forall\ i \in \mathscr{I}, j \in \mathscr{J} \\
& y_i \in \{0, 1\} \ \forall\ i, j.
\end{aligned}
$$

The Lagrangian formulation in (4) with objective (4b) can be solved by inspection. Note that due to the relaxation, the demands *do not have to be assigned exactly one facility.* As a result, a demand may be assigned no facilities or multiple facilities; i.e., for each $j \in \mathscr{J}$, the optimal value of $x_{ij}$ is

$$
x_{ij} = \begin{cases} 1 : y_i = 1 \text{ and } (d_j c_{ij} - \lambda_j) < 0 \\ 0 : \text{otherwise} \end{cases}
$$

As a result, the impact on the objective of introducing facility $i$ (i.e. $y_i = 1$) does not depend on $y_{i'}$ for any other facility $i'$. So, the "*value*" of introducing facility $i$ can be written

$$
V_i = \sum_{j \in \mathscr{J}} \min\{0, d_j c_{ij} - \lambda_j\} \tag{5}
$$

and the $P$ candidate sites with the most negative $V_i$ values are optimally selected.

Note that the relaxed constraint (1b) is technically an equality constraint, but can be equivalently written as an inequality constraint in the primal problem $\sum_{i \in \mathscr{I}} x_{ij} \geq 1\ \forall\ j \in \mathscr{J}$ because the minimized objective strictly increases in each $x_{ij}$, and no optimal solution with the inequality constraint would permit $\sum_{i \in \mathscr{I}} x_{ij} > 1$. That is, equality is implicit. This is notable because the relaxation with dual terms of the form $\lambda_j(\sum_{i \in \mathscr{I}} x_{ij} - 1)$ (rather than $\lambda_j(1 - \sum_{i \in \mathscr{I}} x_{ij})$) would exclusively yield trivial solutions in which $x_{ij} = 0\ \forall\ i, j$. Dualizing an inequality constraint yields a natural lower bound; dualizing an equality constraint has a less obvious interpretation and in this case ignorance of the implicit inequality could produce a useless relaxation.

As with all Lagrangian relaxations, the optimal value of the objective of problem (4) provides a lower bound on the optimal value of the original problem (1). Furthermore, by using the facilities selected in problem (4) (i.e. the values of $y_i$) and assigning each demand to the nearest facility (setting the $x_{ij}$ optimally), a feasible solution is obtained, providing an upper bound to problem (1).

Adjusting $\lambda$ using subgradient descent between iterations, we can increase the lower bound. In [1], they suggest the following procedure.

1. Update the step size

$$t^n = \frac{\alpha^n (UB - \mathscr{L}^n)}{\sum_{j \in \mathscr{J}} \left( \sum_{i \in \mathscr{I}} x_{ij}^n - 1 \right)^2}$$

   where

$$\begin{aligned} t^n &= \text{stepsize at } n\text{th iteration} \\ \alpha^n &= \text{a constant on the } n\text{th iteration, with } \alpha^1 \text{ generally set to 2} \\ UB &= \text{best (smallest) upper bound on the } P\text{-median objective function (1a) so far} \\ \mathscr{L}^n &= \text{objective function of the Lagrangian (4a) or (4b) on the } n\text{th iteration} \\ x_{ij}^n &= \text{optimal value of the } x_{ij} \text{ variable (in the primal problem) on the } n\text{th iteration} \end{aligned}$$

2. Update the multipliers $\lambda$ as

$$\lambda_j^{n+1} = \max \left\{ 0, \lambda_j^n - t^n \left( \sum_{i \in I} x_{ij}^n - 1 \right) \right\}$$

Note that initializing $\lambda_j = 0 \; \forall \; i$ induces all $x_{ij} = 0$ and makes the "value" of each facility equivalent, forcing a random selection for the first iteration.

Unfortunately, while this heuristic provides an upper and lower bound at each iteration, it is not guaranteed to perform well. Indeed, on most randomly generated instances (with $M \approx 100$ candidate locations, $P \approx 50$ facilities, and $N \approx 1000$ demands), the Lagrangian solution obtained exactly the greedy solution on some iteration(s) but did no better. In Figure 5, the performance of the Lagrangian heuristic is illustrated. The upper bounds obtained by finding a feasible solution from a Lagrangian-optimal solution need not monotonically decrease, and indeed often do not improve. The lower bounds obtained as solutions to the Lagrangian problem do by nature improve at every iteration, but need not be tight, and are indeed often negative. Perhaps tuning of meta-parameters could have improved the performance in this case.

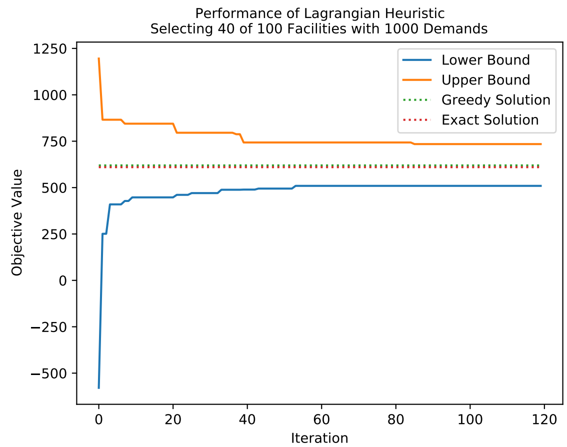



**Figure 5:** *Performance of the Lagrangian heuristic proposed in [4] is often worse than the greedy heuristic.*

Several of the algorithms in [4] were reproduced for this paper. The performance of the algorithms obtained by the authors of [4] themselves are listed below in Figure 6.

| | | Exchange Heuristic | | Neighborhood Search | | Lagrangian Algorithm | |
|---|---|---|---|---|---|---|---|
| Number of Facilities | Myopic | Each Iteration | Last Iteration | Each Iteration | Last Iteration | Value | Iterations |
| 1 | 25.795 | 25.795 | 25.795 | 25.795 | 25.795 | 25.795 | 4 |
| 2 | 17.000 | 17.000 | 17.000 | 17.000 | 17.000 | 17.000 | 15 |
| 3 | 14.276 | 13.503 | 13.503 | 14.276 | 14.276 | 13.178 | 95 |
| 4 | 11.659 | 10.719 | 10.184 | 11.659 | 11.659 | 10.184 | 90 |
| 5 | 9.227 | 8.341 | 7.805 | 8.497 | 8.497 | 7.805 | 85 |
| 6 | 7.173 | 6.389 | 5.854 | 6.443 | 6.443 | 5.854 | 73 |
| 7 | 5.222 | 4.038 | 4.038 | 4.492 | 4.038 | 4.038 | 99 |
| 8 | 3.600 | 2.870 | 2.870 | 3.195 | 3.600 | 2.870 | 14 |
| 9 | 2.303 | 1.978 | 1.978 | 2.027 | 2.303 | 1.978 | 13 |
| 10 | 1.135 | 1.135 | 1.135 | 1.135 | 1.135 | 1.135 | 16 |
| 11 | 0.324 | 0.324 | 0.324 | 0.324 | 0.324 | 0.324 | 21 |
| 12 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1 |

**Figure 6:** *Table 6.8 from [1] compares the performance of four techniques on a problem instance selecting $P$ out of 12 candidate locations, for $P = 1, 2, \ldots, 12$.*

## 4.5 Other Meta-Heuristic Approaches

Both [4] and [1] note that most heuristic or iterative approaches permit improvement using either randomized or cache-based meta-heuristics. Randomized improvement strategies include

- *Simulated Annealing*, in which random modifications are introduced at each iteration to encourage terminating on global rather than local minima. The rate or magnitude of the random modifications begins high and is gradually lowered.

- *Concentration Algorithms*, proposed in [15], utilizes several solutions from any randomized algorithm (or the solutions from a suite of several different randomized or deterministic algorithms) to build a *concentration set* of facilities that have been selected in at least one solution, or in a sufficient number of iterations of a solution.

# 5 $P$-Median with Non-Negative Utility

As can be seen in Figure 3 and the discussion around Figure 4, the performance of the greedy heuristic on the $P$-median problem can be arbitrarily bad. However, there is a variant of this problem in which the performance of the greedy algorithm is guaranteed to be reasonable. Note that much of this section is taken from my first-year DOTM paper.

## 5.1 Non-Negative Utility and Submodular Set Functions

Consider the set $S \subset \mathscr{I}$ of selected facilities, which can be defined

$$S = \{i \in \mathscr{I} | y_i = 1\}.$$

Then let $g_j(S)$ represent the utility or disutility to a demand site $j \in \mathscr{J}$ of selecting set $S$. The objective function (1a) of the $P$-median problem (1) can be framed in this way as follows:

$$\underset{x,y}{\text{Minimize}} \sum_{i \in \mathscr{I}} \sum_{j \in \mathscr{J}} d_j c_{ij} x_{ij} = \sum_{j \in \mathscr{J}} g_j(S) \tag{6}$$

where $g_j(S) = \min_{i \in S} d_j c_{ij}$. Modifying slightly from the operational consideration of reducing demand-weighted distance, it is natural to consider $g_j(S)$ that represents utility rather than disutility. In particular, suppose $g_j : 2^{\mathscr{I}} \to \mathbb{R}_+$

- is non-negative (indicated by $\to \mathbb{R}_+$),
- depends only on the minimum distance beteen location $j$ and a facility in $S$, and
- is decreasing in the minimum distance between a facility in $S$ and location $j$,

then $g_j$ exhibits a property called *submodularity*.

**Definition 1.** A set function $f : 2^U \to \mathbb{R}_+$ is *submodular* if for any $S \subseteq T \subseteq U$ and $xinU \backslash T$,

$$f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T) \tag{7}$$

**Theorem 1.** *If the set functions $g_j : 2^{\mathscr{I}} \to \mathbb{R}_+$ are decreasing in the minimum distance between a facility in $S$ and a location $j$ (and depend only on this distance), then they are submodular for all $j \in \mathscr{J}$.*

*Proof.* Let $S \subseteq T \subseteq \mathscr{I}$, let $x \in \mathscr{I} \backslash T$ and consider two cases:

1. $g_j(S) = g_j(T)$.
   In this case, $g_j(S \cup \{x\}) = g_j(T \cup \{x\})$ because the closest facility to demand $j$ in $T$ is no closer than the closest facility to demand $j$ in $S$; considering a potentially closer facility will either improve both or neither. Then the two sides of definition (7) of submodular set functions are equal.

2. $g_j(S \cup \{x\}) < g_j(T)$.
   Here there are three sub-cases

   (a) $g_j(S \cup \{x\}) = g_j(S)$.
   This implies there is a facility in $S$ closer to demand $j$ than $x$. Since $S \subseteq T$, this implies $g_j(T \cup \{x\}) = g_j(T)$ as well. In this case, the condition in (7) reades "$0 \geq 0$" and thus holds.

   (b) $g_j(S \cup \{x\}) > g_j(S)$ and $g_j(T \cup \{x\}) = g_j(T)$.
   This implies there is a facility in $T$ closer to demand $j$ than facility $x$, but none of the facilities in $S$ are closer to demand $j$ than facility $x$. In this case, the left-hand side of condition (7) is $g_j(T \cup \{x\}) - g_j(T) = 0$, and so the condition holds.

   (c) $g_j(T \cup \{x\}) > g_j(T)$.
   This implies that facility $x$ is closer to demand $j$ than any of those in $T$. Since $S \subseteq T$, $x$ is also necessarily closer than any in $S$, and $g_j(S \cup \{x\}) > g_j(S)$. In particular, however, it means that

   $$g_j(T \cup \{x\}) = g_j(\{x\}) = g_j(S \cup \{x\}).$$

   Then, condition (7) holds by the following:

   $$\begin{gathered} g_j(T) \geq g_j(S) \text{ (monotonicity)} \\ \Downarrow \\ g_j(\{x\}) - g_j(S) \geq g_j(\{x\}) - g_j(T) \\ \Downarrow \\ g_j(S \cup \{x\}) - g_j(S) \geq g_j(T \cup \{x\}) - g_j(T). \end{gathered}$$

So, in all cases, condition (7) holds, and so the set function $g_j$ is submodular.

□

**Corollary 1.** *The objective function of a variant of problem* (1) *in which utility to each demand site has the conditions outlined earlier in this section (non-negativity, depends only on and decreases in distance to closest facility), written as a submodular set function as in* (6)*, constitutes a submodular set function.*

*Proof.* The theorem 1 shows that the utility to each demand site is submodular, and this corollary follows from the fact that a sum of submodular set functions is itself a submodular set function. □

**Theorem 2.** *Every submodular set function for a set of size $P$ permits $\left(1-(1-\frac{1}{P})^P\right)$ performance of a greedy algorithm. This bound is decreasing in $P$, but it is always at least as strong as the limit*

$$\lim_{P\to\infty}\left(1-(1-\frac{1}{P})^P\right)=(1-\frac{1}{e})\approx 0.63.$$

*Proof.* The lecture notes [11] (or, again, my first-year DOTM paper) demonstrate this fact, which follows directly from the definition of submodularity. □

In summary, a greedy algorithm may perform arbitrarily poorly on the $P$-median problem, and earlier randomized examples exhibited performance many times worse than the exact solution. However, if the weighted distance minimization is replaced with a utility from the reasonably broad category of non-negative functions decreasing in the distance to the closest facility for each individual, then a greedy solution to this NP-hard problem attains at least 63% of an optimal objective value.

# 6 Conclusion

We begin by defining the $P$-median problem and situating it within the context of other facility location variants. We reproduce a proof of NP-hardness. We describe popular heuristic solution methods: LP relaxation, greedy algorithms, pairwise interchange, and a Lagrangian relaxation; we implement some of these for a close inspection of their performance. Finally, we note that when the operational objective is replaced with an economic utility of a very broad and natural class, a greedy heuristic has a strong performance guarantee.

# References


[1] *Introduction to Location Theory and Models*, chapter 1, pages 1–28. John Wiley & Sons, Ltd, 2013.

[2] C.H. Aikens. Facility location models for distribution planning. *European Journal of Operational Research*, 22(3):263–279, 1985.

[3] Gerard Cornuejols, Marshall Fisher, and George L. Nemhauser. On the uncapacitated location problem**this research was supported by nsf grants eng75-00568 and soc-7402516. sections 1–4 of this paper include a technical summary of some results given in [2]. some proofs are omitted and may be obtained in [2]. In P.L. Hammer, E.L. Johnson, B.H. Korte, and G.L. Nemhauser, editors, *Studies in Integer Programming*, volume 1 of *Annals of Discrete Mathematics*, pages 163–177. Elsevier, 1977.

[4] Mark S. Daskin and Kayse Lee Maass. Chapter 2 the p-median problem. 2017.

[5] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Mathematical Sciences Series. W. H. Freeman, 1979.

[6] S. L. Hakimi. Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459, 1964.

[7] S. L. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13(3):462–475, 1965.

[8] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.

[9] Andreas Klose and Andreas Drexl. Facility location models for distribution system design. *European Journal of Operational Research*, 162(1):4–29, 2005. Logistics: From Theory to Application.

[10] F. E. Maranzana. On the location of supply points to minimize transport costs. *Journal of the Operational Research Society*, 15(3):261–270, 1964.

[11] Sewoong Oh. Submodular function optimization. University Lecture, IE 512: Graphs, Networks, and Algorithms, 2013.

[12] Susan Hesse Owen and Mark S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, 1998.

[13] Charles S. ReVelle and Ralph W. Swain. Central facilities location. *Geographical Analysis*, 2(1):30–42, 1970.

[14] C.S. ReVelle, H.A. Eiselt, and M.S. Daskin. A bibliography for some fundamental problem categories in discrete location science. *European Journal of Operational Research*, 184(3):817–848, 2008.

[15] K.E. Rosing and C.S. ReVelle. Heuristic concentration: Two stage solution construction. *European Journal of Operational Research*, 97(1):75–86, 1997.