Department of Decision Sciences

HONPR2C
Project II

# A statistical analysis of heuristics for the $p$-median problem

Assignment 2: Preliminary Report

Assignment Unique Number: 650898

**Gemma Dawson**

**50223909**

**Study Leader: Ms J L le Roux**

30 August 2017

## Signed Declaration

I hereby declare that this project is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

Signed:

Gemma Dawson
Student Number: 50223909

TO DO:

1. extend literature study to include more details on algorithms used
2. ALTERNATE algorithm - code (Classic - LS)
3. check the rules of citations and full stops - done?
4. add a section discussing the diff between classical and meta heuristics - needs to be completed
5. add explanation around how choices between identical improvements are made (i.e. randomly)
6. add complexity for each algorithm
7. define "free node" in greedy (or first algorithm)

# Contents

## Introduction

The network facility location problem of finding a location for facilities or public services within a given space by optimising some predefined objective, is not a simple one. Within the single-objective branch of network location problems, there is a plethora of papers exploring minisum, minimax, and covering problems.

If the number of facilities needed is known and these facilities need to be located in the manner of minimising the total distance between demand nodes and their assigned facility, then the decision-maker is facing a $p$-median problem.

The $p$-median problem is well studied and many different approaches to finding both exact and approximate solutions have been proposed, but determining which method would be best for a particular problem is not clear.

The study proposed will analyse and compare several heuristics applied to the $p$-median problem in the hope that this decision may be made more clear.

## Description of the Problem

The uncapacitated $p$-median problem has been widely studied and many different approaches to finding an acceptable, if not optimal, solution have been presented over the past fifty years.

The focus of this study is to statistically analyse and compare the results obtained from various heuristic and metaheuristic algorithms applied to the $p$-median problem.

Different heuristics applied to $p$-median problems can produce varying quality of results dependant on the size of the original problem. As algorithms take a considerable amount of time and research to be suitably set-up and run, it may be of benefit to have some prior knowledge of which heuristics may be suitable for the problem on hand.

As such, the main objective of this proposed study would be to provide an indication of which method would be best suited to a given problem.

This study will also investigate the computational time each algorithm requires to generate a solution, taking into account the size of the problem being solved.

Since parametrisation of heuristics play a crucial part of the efficiency of the algorithm and the quality of the results it produces, this will be explained in detail in the study. Given time, different approaches to the same heuristic may be explored as well.

## Methodology

All algorithms will be coded in R [1], an open source programming language, due to its ease of use, extensive availability of online resources as well as the functionality provided by downloadable packages that provide useful data handling and visualisation.

As the results of the algorithms under investigation are to be compared, identical problems need to be presented to each algorithm to solve. Beasley [2] offers a forty-instance dataset which comprises of problems ranging from 100 to 900 demand nodes that require the locations of between five and 200 medians to be determined. These datasets ae given in a text format presenting the pairs of nodes along with their associated cost. These text files were converted to cost matrices using the R package igraph [3]. These matrices provide the cost of traversing from one node to any other node in each network of the forty instances of the dataset.

Beasley's library also provides the optimal solution's value of the weighted total distance objective function for each of the forty instances in the dataset. This will allow the results obtained from the algorithms to be objectively assessed against the optimal solution. The deviation of each algorithm's solution from the optimal solution will be analysed. In addition, the computational time required for each algorithm will be logged and compared.

## Literature Study

While the $p$-median problem may have been tackled by academics before 1964, it was first formally defined and formulated by Hakimi [4]. It was also in this paper that Hakimi proved that the optimal location of a facility in a connected discrete network would always be found to correspond to the location of a demand node. This considerably reduces the solution search space.

A few years later, ReVelle and Swain [5] formulated the $p$-median problem as an integer programme which provided the answers to both the question of where the optimal medians were located as well as which demand nodes should be allocated to each median. In this paper, the solution properties of solving this problem with linear programming using the branch-and-bound technique was also discussed.

Later that decade, it was shown that the $p$-median problem is NP-hard and as such there exists no algorithm that can find the optimal solution within polynomial time. Since most real-world problems are very large in nature, this poses a great problem. Well developed heuristics can provide satisfactory solutions and many such approaches have been proposed. Mladenović et al. [6] produced a comprehensive survey of the heuristics that have been used to solve the $p$-median problem. This survey divides heuristics into two groups, classical heuristics and metaheuristics. The classical heuristics group consists of constructive heuristics, such as the greedy algorithm, local search, and mathematical programming. The metaheuristics group lists tabu search, variable neighbourhood search, and simulated annealing among others.

There do exist studies that analysis the efficiency of heuristics in solving the $p$-median. Simulated annealing was analysed by Chiyoshi and Galvão [7] while Alp et al. [8] investigated their proposed genetic algorithm. Hansen and Mladenović [9] applied the variable neighbourhood search algorithm to the $p$-median problem as well as heuristic concentration, tabu search, and the greedy interchange algorithm. These studies used Beasley's test problem dataset although Hansen and Mladenović's review only used a portion of the available test problems.

Rolland et al. [10] explored the efficiency of tabu search using randomly generated datasets as test problems, and this was compared to results obtained from a two-exchange heuristic and an integer programming algorithm. A heuristic concentration was compared to tabu search by Rosing et al. [11]. Since datasets were independently and randomly generated for each of these studies, it makes it very difficult to compare results.

ADD MORE ABOUT THE ALGORITHMS USED IN THIS REPORT

## Classical Heuristics and Metaheuristics

### Overview

As previously mentioned, a review of approaches to solving the $p$-median problem was presented by Mladenović et al. [6] in 2007. This paper divided heuristics into one of two groups, Classical Heuristics and Metaheuristics. This classification has been utilised in this study.

A heuristic can be classified as a Classical Heuristic

XXXXXXXXXXXXXXXXXXXXXXXADD WHAT DIFFERENCE BETWEEN CLASSICAL & META ISXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

### Pseudocode Notation

For each algorithm examined in this study, a pseudocode will be presented using notation given below. This notation is the same as to that used by Whitaker [12].

$G$ : the problem network
$n$ : the number of nodes within the network
$m$ : the number of nodes nodes that are available to be assigned as a median
$p$ : the number of medians by required by the solution
$M$ : the set of nodes that are available to be assigned as a median
$P^*$ : the set of medians where $P^* \subset M$
$P$ : the set of nodes not in the solution median set where $P \subset M$ and $P \cup P^* = M$
$S^*$ : objective function value for supplying the network $G$
$k$ : an iteration parameter
$\infty$ : an arbitrarily large number
$d_{ij}$ : the cost of travelling from node $i$ to potential median $j$
$u_i^k$ : the cost of travelling from node $i$ to the closest median $j$ where $j \in P^*$ during iteration $k$
$w_i^k$ : the cost of travelling from node $i$ to the second closest median $j$ where $j \in P^*$ during iteration $k$
$c_j^k$ : the potential value of $S*$ if node $j$ is added to the median set $P*$ during iteration $k$
$S_r$ : the value of $S*$ after adding node $r$ to the median set $P*$
$S_{rt}$ : the potential increase (or decrease) to S* resulting from the interchange of node $r$ with median $t$
$I$, $J$, $K_j$ : subsets of the $n$ nodes as defined XXXXXDEFINE THESE!XXXXX
$a$, $b$, $z$, $q$ : indices and parameters as defined XXXXDEFINE!XXXX
$\emptyset$ : an empty set

## Classical Heuristics: Constructive

A constructive heuristic is a technique that starts with an empty solution and each iteration extends this solution until a complete solution has been found. Within the context of the unconstrained $p$-median problem, for each iteration of the algorithm, a median is identified by some deterministic method and added to the current solution. This process continues until the solution contains $p$ entries.

### The Greedy Algorithm

The greedy algorithm, as described by Whitaker [12], is a simple recursive algorithm that starts with an empty median set and, with each iteration of the algorithm, adds a node to the median set. The node that that is added to the median set is the one that decreases the objective function value the most when compared to the other non-median nodes.

#### Pseudocode

### The Fast Greedy Algorithm

Whitaker [12] proposed the fast greedy algorithm. This algorithm differs from the greedy algorithm in that only the nodes that were re-assigned to a new median in the preceding iteration are investigated. From these nodes, the new median is determined to be the one that will maximise the decrease in the objective function value.

#### Pseudocode

### The Stingy Algorithm

The stingy algorithm was suggested by Feldman XXXXXXADD REFERENCE HERE - feldman 1966XXXXXXXXXXXx Instead of starting with an empty median set and recursively adding a node until all $p$ medians are assigned, the stingy algorithm begins with all nodes in the network being designated as a median. A node is removed one at a time until only $p$ medians remain in the median set.

#### Pseudocode

### Computational Results

## Classical Heuristics: Local Search

A local search heuristic differs from a constructive heuristic in that it starts with a feasible solution and each iteration attempts to improve this solution by some deterministic method. This process continues until there is no better solution within reach of the current solution or some predetermined time-out is reached.

**Algorithm 1** Greedy Algorithm

STEP 0: Initialisation

$$\text{Set } P^* = \emptyset;\ k = 1;\ \text{and } u_i^1 = \infty \text{ for all } i = 1,2,\ldots,n$$

STEP 1: For each free node, the objective function value resulting from selecting this node as a median is determined. This objective function value is calculated by summing the cost of travel for each node in the network to either the closest median or the free node under investigation, whichever is smallest.

$$c_j^k = \sum_{i=1}^{n} \min(d_{ij},\, u_i^k) \quad \forall j \in M,\, j \notin P^*$$

STEP 2: The free node that will result in the smallest increase to the objective function value is identified, and the objective function value is updated.

$$S_r = \min_{j \notin P*}(c_j^k) \quad \text{for } r \in M,\, r \notin P^*$$

STEP 3: Add node $r$ to the set of medians and then determine if the algorithm has completed the number of required iterations.

$$P^* = P^* \cup r$$

**if** $k = p$ **then**
go to STEP 5 [
**else**
go to STEP 4 ]
**end if**

STEP 4: Increase the value of iteration parameter and update the cost of travelling from any node to its closet median given that node $r$ has been added to the median set.

$$\begin{aligned} k &= k + 1 \\ u_i^k &= \min(d_{ir},\, u_i^{k-1}) \quad \text{for } i = 1,2,3\ldots,n \end{aligned}$$

Go to STEP 1

STEP 5: Once $p$ iterations are complete, set the final objective function value equal to the current iteration's objective function value and stop the algorithm.

$$S^* = S_r$$

STOP

**Algorithm 2** Fast Greedy Algorithm

STEP 0: Initialisation

$$\text{Set } P^* = \emptyset;\ I = \emptyset;\ k = 1$$
$$c_j^0 = c_j^1 = 0 \text{ for } j \in M$$
$$u_i^0 = 0;\ u_i^1 = \infty \text{ for all } i = 1,2,\ldots,n$$
$$I = I \cup i \text{ for } i = 1,2,\ldots,n$$

STEP 1: For each free node that was reassigned to a new median in the previous iteration, the objective function value resulting from selecting this node as a median is determined. This objective function value is calculated by summing the cost of travel for each node in the network to either the closest median or the free node under investigation, whichever is smallest. For the first iteration, all nodes in the network are considered part of the reassigned set.

$$c_j^k = \sum_{i\in I} \min(d_{ij},\, u_i^k) + c_j^{k-1} - \sum_{i\in I} \min(d_{ij},\, u_i^{k-1}) \quad \forall j \in M,\, j \notin P^*$$

STEP 2: The free node that will result in the smallest increase to the objective function value is identified, and the objective function value is updated.

$$S_r = \min_{j\notin P*}(c_j^k) \quad \text{for } r \in M,\, r \notin P^*$$

STEP 3: Add node $r$ to the set of medians and then determine if the algorithm has completed the number of required iterations.

$$P^* = P^* \cup r$$

**if** $k = p$ **then**
go to STEP 5 [
**else**
Set $I = \emptyset$ and go to STEP 4 ]
**end if**

STEP 4: Increase the value of iteration parameter and update the cost of travelling from any node to its closet median given that node $r$ has been added to the median set.

$$k = k + 1$$
$$u_i^k = \min(d_{ir},\, u_i^{k-1}) \quad \text{for } i = 1,2,3\ldots,n$$

**if** $d_{ir} < u_i^{k-1}$ **then**$I = I \cup i$
go to STEP 1
**end if**



STEP 5: Once $p$ iterations are complete, set the final objective function value equal

### The Interchange Algorithm

#### Pseudocode

### The Fast Interchange Algorithm

#### Pseudocode

### The Alternate Algorithm

#### Pseudocode

### Computational Results

## Conclusion

# References

[1] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013. [Online]. Available: http://www.R-project.org/

[2] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of the Operational Research Society*, vol. 41, no. 11, pp. 1069–1072, 1990.

[3] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, p. 1695, 2006. [Online]. Available: http://igraph.org

[4] S. L. Hakimi, "Optimum locations of switching centers and the absolute centers and medians of a graph," *Operations Research*, vol. 12, no. 3, pp. 450–459, 1964.

[5] C. S. ReVelle and R. W. Swain, "Central facilities location," *Geographical Analysis*, vol. 2, no. 1, pp. 30–42, 1970.

[6] N. Mladenović, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez, "The p-median problem: A survey of metaheuristic approaches," *European Journal of Operational Research*, vol. 179, no. 3, pp. 927–939, 2007.

[7] F. Chiyoshi and R. D. Galvão, "A statistical analysis of simulated annealing applied to the p-median problem," *Annals of Operations Research*, vol. 96, no. 1, pp. 61–74, 2000.

[8] O. Alp, E. Erkut, and Z. Drezner, "An efficient genetic algorithm for the p-median problem," *Annals of Operations Research*, vol. 122, no. 1, pp. 21–42, 2003.

[9] P. Hansen and N. Mladenović, "Variable neighborhood search: Principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, 2001.

[10] E. Rolland, D. A. Schilling, and J. R. Current, "An efficient tabu search procedure for the p-median problem," *European Journal of Operational Research*, vol. 96, no. 2, pp. 329–342, 1997.

[11] K. E. Rosing, C. S. ReVelle, E. Rolland, D. Schilling, and J. Current, "Heuristic concentration and tabu search: A head to head comparison," *European Journal of Operational Research*, vol. 104, no. 1, pp. 93–99, 1998.

[12] R. Whitaker, "A fast algorithm for the greedy interchange for large-scale clustering and median location problems," *INFOR: Information Systems and Operational Research*, vol. 21, no. 2, pp. 95–108, 1983.

# Index of Terms

# Index of Authors