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.

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

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

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

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.

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

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 to the current iteration's objective function value and stop the algorithm.

Algorithm 3 Stingy Algorithm

STEP 0: Initialisation

$$\text{Set } P^* = M;\ P = \emptyset \text{ and } u_i = 0 \text{ for all } i = 1,2,\ldots,n$$

STEP 1: For each node that belong to the median set, find the value of the objective function if this node were to be removed from the median set.

$$c_j^* = \sum_{i=1}^{n} \min_{j \in P^*, j \neq i} d_{ij}$$

STEP 2: Identify which node results in the smallest increase in the objective function. Find node r such that

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

STEP 3: Remove this node from the set of medians.

$$P^* = P^* \backslash r$$
$$P = P \in r$$

STEP 4: Update the iteration parameter.

$$k = k + 1$$

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 4 Interchange Algorithm of Teitz and Bart

INPUT: Cost Matrix, P^*, a set of medians, and S^*, objective function value that corresponds with P^*

STEP 0: Initialisation

$$\text{Set } q = a = 0;\ k = 1;\ b = m - p;\ S = S^* \text{ and } P = M - P$$

STEP 1: For all nodes, find the closest and second closest nodes that belong to the median set.

$$u_i^k = d_{ix} = \min_{j \in P^*}$$
$$w_i^k = d_{iy} = \min_{j \in P^*, j \neq x}$$

STEP 2: The value of q is updated until it is equal to the number of nodes in the network minus the number of nodes required for the median.
if $q = b$ **then**
go to STEP 5 [
else
Set $q = q + 1$ and go to STEP 3]
end if

STEP 3: Determine which node currently in the median set, P^*, should possibly be interchanged with the q^{th} node in the set of free nodes. The change in the objective function value if this interchange were to proceed is calculated.

$$r = P_q$$
$$S_{rt} = \min_{j \in P^*} \left[\sum_{i \in I} [\min(d_{ir}, u_i^k) - u_i^k] + \sum_{i \in J} \in I[\min(d_{ir}, w_i^k) - u_i^k] \right]$$

$$\text{where } I = \{\text{all } i \in G : d_{ij} > u_i^k\}$$
$$\text{and } J = \{\text{all } i \in G : d_{ij} = u_i^k\}$$

STEP 4: If the proposed node switch from STEP 3 results in a decrease of the objective function value, then proceed with the interchange and update the objective function value as well as the iteration parameter. If the interchange would result in an increase in the objective function value, then go back to STEP 2 and update q.
if $S_{rt} \geq 0$ **then**
go to STEP 2 [
else if $S_{rt} < 0$ **then**

$$k = k + 1$$

$$S^* = S * + S_{rt}$$
$$P_q = tP^* \qquad = P^* \backslash t \cup r$$

go to STEP 1]

Algorithm 5 Fast Interchange Algorithm

INPUT: Cost Matrix, P^*, a set of medians, and S^*, objective function value that corresponds with P^*

STEP 0: Initialisation

$$\text{Set } k = z = q = 1 \text{ and } b = m - p$$

$$\text{Define } P = M - P^*$$

For each node in the network, determine the distance to both the closest node in the median set as well as the second closest node in the median set.
For $i = 1,2,\ldots,n$, find x and y where x, $y \in P^*$ such that

$$u_i^1 = d_{ix} = \min_{j \in P^*} d_{ij}$$
$$w_i^1 = d_{iy} = \min_{j \in P^*, j \neq x} d_{ij}$$

STEP 1: Calculate the change in the objective function if one node from the median set where to be interchanged with a free node.
Set $r = P_q$ and find some node t, $t \in P^*$ such that

$$S_{rt} = \Big[\sum_{i \in I} d_{ir} - u_i^k\Big] + \min_{j \in P^*} \Big[\sum_{i \in K_j} [\min(d_{ir}, w_i^k) - u_i^k]\Big]$$
$$\text{where } I = \{\text{all } i \in G : d_{ir} < u_i^k\}$$
$$\text{and } K_j = \{\text{all } i \in G : d_{ir} \geq u_i^k \text{ and } d_{ij} - u_i^k\}$$

STEP 2: If the objective function value where to increase by the proposed interchange of nodes, then go to STEP 4. If the proposed change of nodes results in a decrease of the objective function value, then make the change, update iterative parameters and the objective function and go to STEP 3.
if $S_{rt} \geq 0$ **then**
go to STEP 4 [
else if $S_{rt} < 0$ **then**

$$k = k + 1$$
$$S^* = S * + S_{rt}$$
$$P_q = tP^* \qquad = P^* \backslash t \cup r$$

go to STEP 3]
end if

STEP 3: Update the cost of travel for each node in the network to the closest and second closest medians.
if $d_{it} > u_i^{k-1}$ **then**

$$u_i^k = \min(d_{ir}, u_k^{k-1})$$

let $s \in P^*$ be a median for node i such that $d_{is} = u_i^k$

Algorithm 6 Alternate Algorithm

INPUT: Cost Matrix, P^*, and S^*

STEP 0: Initialisation

$$\text{Set } k = 1$$

STEP 1: Assign each node in the network to its closest median.

$$K_j = \{\text{all } i \in G \text{ such that } \min(d_{ij})\} \text{ where } j \in P^*$$

STEP 2: Find the median node for each current median's set of associated nodes.

$$\text{Find } i \in K_j \text{ where } \min\left[\sum_{i \in K_j} d_{ij}\right]$$

STEP 3: Update P^* with the new set of medians.

$$P^* = \{\text{all } k_j\} \text{ for } j \in P^*$$

STEP 4: The algorithm must stop if either the updated P^* is identical to the new P^* or there has been no improvement in th value of S^* during the past five iterations.

if $P^*(k) = P^*(k-1)$ OR $S^{k-2} = S^{k-1} = S^k = 0$ **then**
STOP [
else

$$S^k = S^* \text{ and } k = k + 1$$

go to STEP 1]
end if