Chapter 3

Center Problems and Median Problems

This chapter focuses on discrete optimization problems within a spatial context. Two types of objective functions are of interest: (i) minimizing the maximum cost of serving one of several clients, the so-called center problem, and (ii) minimizing the sum of the costs of serving several clients, the so-called median problem. The first few sections discuss well-known results and algorithms from the area. This part is based on Christofides [2, Chapters 5 & 6], Larson & Odoni [6, Chapter 6.5], and on two surveys from Mirchandani & Francis [5], namely, [5, Chapter 1] by J. Krarup and P. M. Pruzan, and [5, Chapter 2] by P. B. Mirchandani. We do not give original references for this part, and instead refer the reader to the above texts. The last section discusses a seminal paper of Lin & Vitter [3] that gives a bicriteria approximation algorithm for the k-median problem. Lin & Vitter introduce their filtering-and-rounding method. This method first solves an LP relaxation, then uses the LP solution to construct a filtered problem by fixing some variables to zero in the integer programming formulation. A near-optimal solution to the filtered problem gives a solution to the original problem such that the objective value is nearly optimal but some of the original constraints may be violated. Lin & Vitter's analysis shows that the objective value is within a guaranteed factor of optimal, and also the "constraint violation ratio" is within a guaranteed factor. Lin & Vitter's work is based on earlier results due to Raghavan and Thompson [8], and Raghavan [7].

3.1 Introduction

In this chapter, we would like to find optimal (or approximately optimal) locations for routine services, such as airports, factories, warehouses, schools, garbage incinerators, prisons, etc., as well as for emergency services such as ambulances, fire stations and snow plows. Locational decisions are based on many factors; some of these factors are physical, economic, social, environmental or political. Our focus will be on the quantitative analysis of discrete problems. That is, the locations must be chosen from a finite number of potential sites selected in a preliminary stage.

We focus on three types of problems:

1. Center problems (also called minimax problems):

A given number of facilities (or services) must be located so as to minimize the maximum travel cost (or maximum travel time) of the users to/from the facilities. These problems usually arise in applications involving the location of emergency services.

2. Median problems:
   A given number of facilities (or services) must be located so as to minimize the average travel cost (or average travel time) of the users to/from the facilities. These problems usually come from applications involving the location of routine, nonemergency services.

3. Uncapacitated facility location problems:
   Given a number of potential sites, fixed costs for opening a facility at each site, and profits made by serving certain clients from certain sites, the problem is to choose sites where facilities should be located and to assign clients to facilities such that the net earnings are maximized.

The uncapacitated facility location problem is studied in the next chapter.

3.2 The basic model for median problems and center problems

A weighted undirected graph (or network) $G = (V, E)$ is used to model the application. Demands for service originate at the nodes of G. Each node $v \in V$ has a real-valued demand $h(v)$; this indicates the rate (or intensity) at which demands for service originate from node v. Each edge $vw \in E$ has a real-valued cost $c(vw)$; this represents the cost of providing a service (or sending a commodity) from node v to node w (or from node w to node v). The goal is to locate one or more facilities to serve the node demands such that a specified objective function is maximized or minimized. A facility can be located at any *point* of the network G, where a point x is either a node or a location on an edge vw, and in the latter case the point is specified by giving $d(x, v)$ (or $d(x, w)$) such that $d(x, v) + d(x, w) = c(vw)$ (i.e., we add a "new node" x and "new edges" xv, xw with costs $d(xv), d(xw)$ such that $d(xv) + d(xw) = c(vw)$). For a point x and a node v, $d(x, v)$ denotes the minimum cost of a path between x and v. In other words, if x is a node, then $d(x, v)$ is the minimum over all paths $v_0v_1v_2\cdots v_{\ell-1}v_\ell$ such that $v_0 = x$ and $v_\ell = v$ of $\sum_{i=0}^{\ell-1} c(v_i, v_{i+1})$, otherwise, if x is a point on an edge $v'w'$ then $d(x, v)$ is the minimum over all paths $v_0v_1v_2\cdots v_{\ell-1}v_\ell$ such that $v_0 = x$, $v_1 = v'$ or $v_1 = w'$, and $v_\ell = v$ of $d(x, v_1) + \sum_{i=1}^{\ell-1} c(v_i, v_{i+1})$. For a set of points $x = \{x_1, \cdots, x_k\}$ and a node $v \in V$, $d(X, v)$ denotes $\min_{x_i \in X} d(x_i, v)$, that is, $d(X, v)$ is the minimum cost of a path between any one of the points x_i in X and v.

3.3 Center problems (minimax problems)

The problem is to locate k facilities on the points of a given network G so as to minimize the maximum cost of a path between a demand node and its "closest" facility. The main applications are to the location of emergency services such as fire stations, ambulances, etc.

For a point x of the network G let $m(x)$ denote

$$\max_{v \in V} d(x, v),$$

the cost of a "shortest" path between x and a "farthest" demand node. Recall that a point x is either a node of G or a location on an edge of G.

The *vertex center* (or node center) is defined to be a node $x^* \in V$ such that for every node $y \in V$,

$$m(x^*) \leq m(y).$$

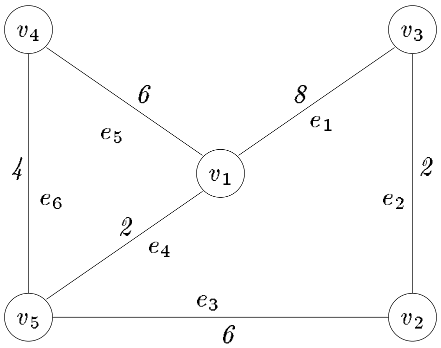
The *local center* of an edge pq is a point x_ℓ on pq such that for every point y on pq,

$$m(x_\ell) \leq m(y).$$

The *absolute center* of the network G is a point x^* in G such that for every point y in G (y may be on an edge of G)

$$m(x^*) \leq m(y).$$

To find a single node center, we compute the matrix of shortest paths costs for all pairs of nodes, and then choose a node such that the maximum entry in its row in the matrix is smallest among the maximum entries of all rows. See Figure 3.1 for an example.



$d(v, w)$

	v_1	v_2	v_3	v_4	v_5
v_1	0	8	8	6	2
v_2	8	0	2	10	6
v_3	8	2	0	12	8
v_4	6	10	12	0	4
v_5	2	6	8	4	0

Figure 3.1: Finding a single node center on an example (from Chapter 5, "Graph Theory: An Algorithmic Approach", by Christofides). The matrix d of shortest paths costs is shown. Either v_1 or v_5 may be taken as the node center.

Finding a single absolute center is more involved. We start with a simple method, then develop a more efficient method. Figure 3.2 shows the working on an example.

Algorithm Simple Algorithm for Absolute Center
input: Graph $G = (V, E)$, nonnegative edge costs c.
output: Absolute center of G, c.

for each edge pq of G, find the local center x_{pq} of the edge;

choose the local center x^* (among those found in the previous step) that minimizes $m(x^*)$;
output x^*, since it is an absolute center of G;

The absolute center of G may not coincide with the node center of G. Moreover, the absolute center of G may be located on an edge that is *not* incident to the node center of G.

To develop a more efficient method for finding absolute centers we state two results.

Proposition 3.1 *For the set of all points x on a fixed edge of G, the maximum distance function $m(x)$ is piecewise linear and its slope is always $+1$ or -1.*

Proposition 3.2 *For an edge pq, the local center x_ℓ satisfies*

$$m(x_\ell) \geq \frac{m(p) + m(q) - c(pq)}{2} ,$$

where $c(pq)$ denotes the cost of edge pq.

Proof: Consider any point on the edge pq. Let x, $0 \leq x \leq c(pq)$, specify the point, where we take the point with $x = 0$ to be p and the point with $x = c(pq)$ to be q. We take $d(x,p)$ to be x and $d(x,q)$ to be $c(p,q) - x$. The cost of a shortest path between x and a farthest demand node, $m(x)$, is piecewise linear with a slope of $+1$ or -1, its value at $x = 0$ is $m(p)$, and its value at $x = c(pq)$ is $m(q)$. Hence,

$$\begin{aligned} m(x) &\geq m(p) - x \qquad (\forall x:\ 0 \leq x \leq c(pq)), \quad \text{and} \\ m(x) &\geq m(q) - (c(pq) - x) \qquad (\forall x:\ 0 \leq x \leq c(pq)). \end{aligned}$$

Adding the two inequalities above and then dividing by 2, we obtain the result. ∎

Algorithm Improved Algorithm for Absolute Center
input: Graph $G = (V, E)$, nonnegative edge costs c.
output: Absolute center of G, c.

find a node center $x \in V$, and let $t = m(x)$;
for each edge pq in sequence **do**
  if $(m(p) + m(q) - c(pq))/2 < t$
  then find the local center x_{pq} of pq and let $t = \min\{t, m(x_{pq})\}$;
  end (if);
end (for);
output x^* and t, where x^* is the last point that assigned a value to t, and $t = m(x^*)$;

At termination of the algorithm, the absolute center is the point x^* (node center x or local center x_{pq}) that assigned the last value to t.

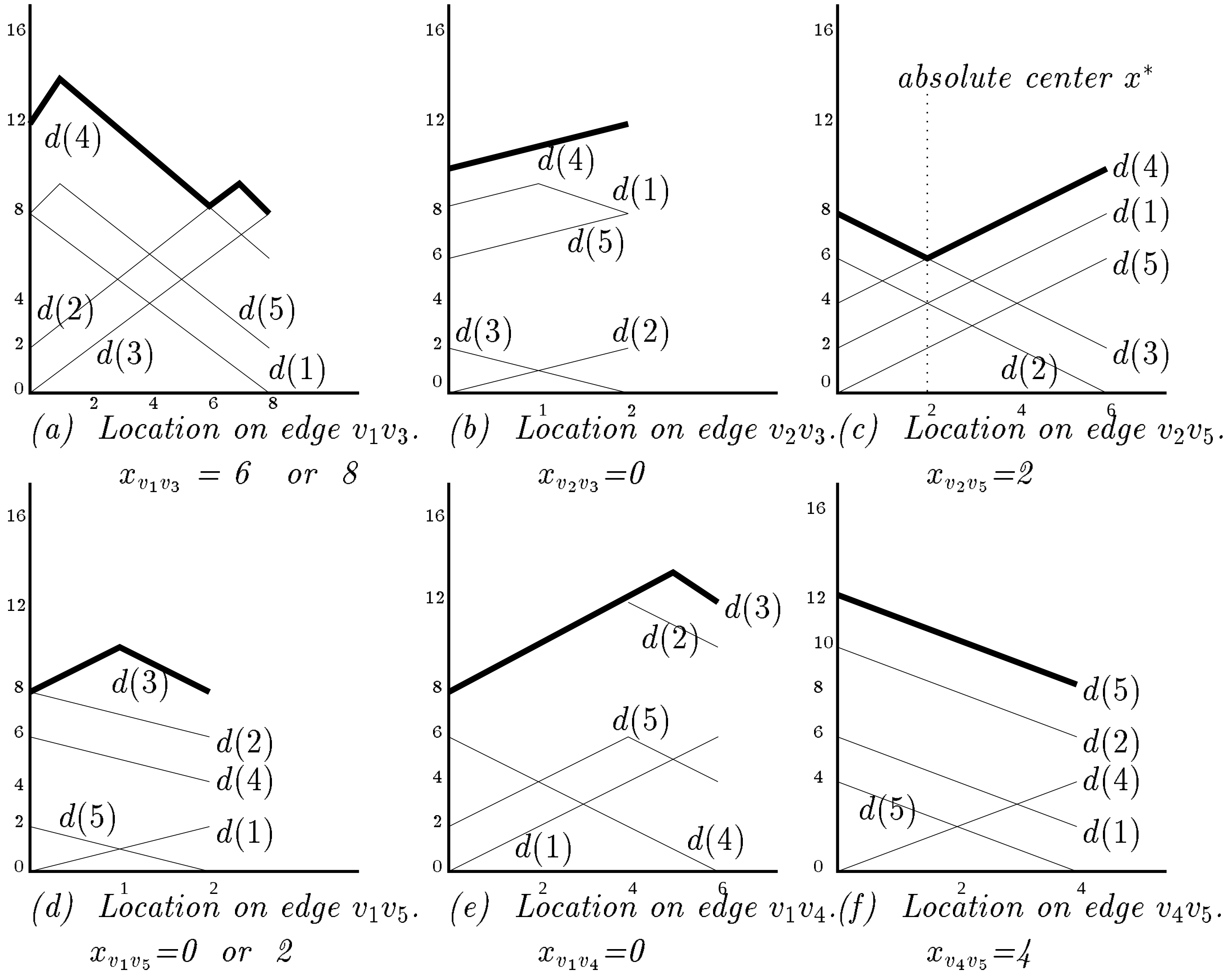


Figure 3.2: Finding a single absolute center for the example in Figure 3.1 (from Chapter 5, "Graph Theory: An Algorithmic Approach", by Christofides). For each edge pq, the local center is found by plotting $d(x, v_i)$ for each node $v_i \in V$, where $0 \leq x \leq c(pq)$ – in the figure, $d(x, v_i)$ is abbreviated to $d(i)$. The "upper envelope" of the five $d(x, v_i)$'s is indicated by thick lines: for each point x in pq, the upper envelope gives the cost of a shortest path from x to a farthest node. The local center x_{pq} is the point that minimizes the upper envelope. The absolute center is the point x^* on the edge v_5v_2 with $d(x^*, v_5) = 2$ and $d(x^*, v_2) = 4$; $m(x^*) = 6$.

3.3.1 Multiple centers

For a set $X = \{x_1, \cdots, x_k\}$ of points of G, let $m(x)$ denote $\max_{v \in V} d(X, v)$, where $d(X, v)$ as before denotes the minimum cost of a path between v and any one of the points x_i in X. A set X^* of points (nodes) of G with $|X| = k$ is called a set of absolute (node) k-centers if $m(X^*) \leq m(X)$ for every set X of k points (nodes) of G.

Proposition 3.3 *The problem of finding absolute (or, node) k-centers is NP-hard.*

See "p-Center Problems" by G.Y. Handler in "Discrete Location Theory", Ed., Mirchandani and Francis, and Christofides and Viola (1971) for approximation algorithms for multiple center problems.

3.4 Median problems (minisum problems)

Consider the single facility location problem of Figure 3.3. The network represents an urban area; the nodes represent locations where demands for services are generated, and the edges represent the roads between these locations. A single facility is to be located in the area (possibly on one of the roads), and the users need to travel to the facility. The daily demand for the service at each node v is shown next to the node; this number is denoted $h(v)$. The length of each road vw is also given; this number is the cost of vw, $c(vw)$. The problem is to choose a location for the facility such that the average travel distance to it is minimized. The key result in this section, Hakimi's theorem (Theorem 3.4), assures us that rather than considering the infinite number of points in the network, we need to focus only on the $|V|$ nodes: one of the nodes is an optimal location.

For a set of points X (the points in X may be nodes or locations on edges), let $J(X)$ denote $\sum_{v \in V} h(v) d(X, v)$. In other words, $J(X)$ is the total cost of serving all the node demands, assuming that a facility is located at each point $x_i \in X$ and that each node $v \in V$ is served from the "closest" point $x_i \in X$, i.e., all the demand of v is served by a point $x_i \in X$ with $d(x_i, v) = d(X, v)$.

A *median* of network G is defined to be a point $x^* \in G$ such that for every point $x \in G$,

$$J(x^*) \leq J(x),$$

i.e., a median is a point that minimizes the total cost of serving all node demands.

A *k-median* is defined to be a set X_k^* of k points $x_1^*, x_2^*, \cdots, x_k^* \in G$ such that for every set X_k of k points,

$$J(X_k^*) \leq J(X_k).$$

If k facilities are to be located at the k points in X_k, then finding the k-median X_k^* amounts to finding the set of k locations that minimizes the total cost of serving all node demands. If each $h(v)$ is fixed to be the fraction of all calls for service that originate from node v, then $J(X_k)$ gives the average cost of providing the service to all users, and finding the k-median X_k^* amounts to finding the set of k locations that minimizes the average cost of serving all node demands. The proof of Hakimi's theorem for a single median follows; the general case is left as an exercise for the reader.

Theorem 3.4 (Hakimi's theorem) *At least one set of k-medians exists solely on the nodes of G.*

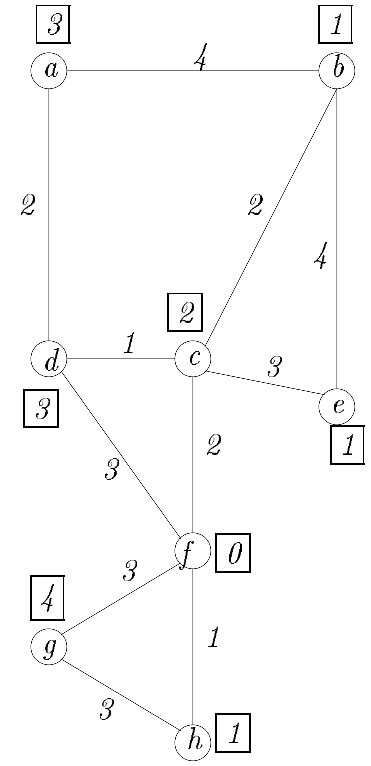


Figure 3.3: An example of a single median problem (from Chapter 6, "Urban Operations Research", by Larson and Odoni). Node c is a median of the network.

Proof: (for a single median) Let x^* be a median of G. If x^* is a node, then the proof is done, so suppose that x^* lies strictly inside an edge uw. Partition the node set V into sets U and W (so $U \cap W = \emptyset$, and $U \cup W = V$), where node v is in U if and only if there is a minimum-cost path between v and x^* that contains u. Without loss of generality, suppose that $\sum_{v \in U} h(v) \geq \sum_{v \in W} h(v)$. Now,

$$
\begin{aligned}
J(x^*) &= \sum_{v \in V} h(v) d(v, x^*) \\
&= \sum_{v \in U} h(v)[d(v,u) + d(u,x^*)] \quad + \quad \sum_{v \in W} h(v) d(v, x^*) \\
&= \sum_{v \in U} h(v) d(v,u) \quad + \quad \sum_{v \in U} h(v) d(u, x^*) \quad + \quad \sum_{v \in W} h(v) d(v, x^*) \\
&\geq \sum_{v \in U} h(v) d(v,u) \quad + \quad \sum_{v \in W} h(v)[d(u,x^*) + d(v,x^*)] \\
&\qquad \left(\text{because } \sum_{v \in U} h(v) \geq \sum_{v \in W} h(v)\right) \\
&\geq \sum_{v \in U} h(v) d(v,u) \quad + \quad \sum_{v \in W} h(v) d(v,u) \\
&\qquad (\text{ using the triangle inequality: } d(v,x^*) + d(x^*,u) \geq d(v,u)) \\
&= \sum_{v \in V} h(v) d(v,u) \\
&= J(u).
\end{aligned}
$$

This completes the proof: the 1-median can always be moved from a point strictly inside an edge to an end node of the edge without increasing the objective value. ∎

3.4.1 A single-median algorithm

To find a single median in a network, which consists of a graph $G = (V, E)$ and edge costs $c : E \rightarrow \Re_+$, we compute the matrix d of shortest paths costs between all pairs of nodes. This computation is done either using inspection or one of the all-pairs shortest-paths algorithms such as the Floyd-Warshall algorithm or $|V|$ applications of Dijkstra's algorithm. Next, we compute the terms $h(v)d_{wv}$ by multiplying each column of the matrix d by the demand of node v. Each of these terms gives the cost of satisfying the demands originating at node v assuming that the facility is located at node w. The optimum location for the facility can now be found by summing across the entries for each row w of the $[h(v)d_{wv}]$ matrix; this gives the total cost of serving all the node demands if the facility is located at node w. Normalizing the node demands $h(v)$ by dividing by the total demand $\sum_{v \in V} h(v)$, we can find the average cost associated with each of the $|V|$ candidate locations. See the top part of Figure 3.4 for an example.

Algorithm Single-Median Algorithm

input: Graph $G = (V, E)$, nonnegative edge costs c, and node demands $h : V \to \Re$.
output: A median node.

obtain the matrix d of shortest paths costs;
multiply the vth column of d by the node demand $h(v)$ to obtain the matrix $[h(v)d_{wv}]$;
for each row w of the matrix $[h(v)d_{wv}]$, compute the sum of all terms in the row;
take the median x^* to be a node whose row sum is minimum;
output x^*;

3.4.2 A multimedian heuristic

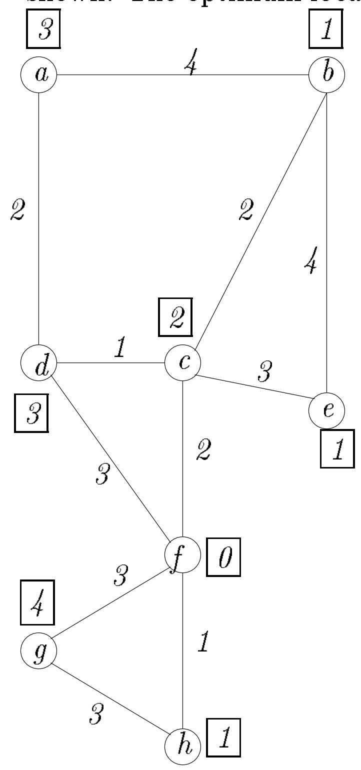
For small k (or small $|V| - k$), the k-median can be found by a straightforward extension of the single-median algorithm. By Hakimi's theorem, we only need to consider sets of points consisting of k nodes. With a total of $n = |V|$ nodes, the number of sets of k nodes to be examined is $\binom{n}{k} \approx \frac{n^n}{(n-k)^{n-k}k^k}$. If both k and $n - k$ are moderately large, then the number of sets to be examined becomes prohibitively large. For small k, total or average costs for each of the $\binom{n}{k}$ sets of locations can be obtained from the $[h(v)d_{wv}]$ matrix: all the demand from each node v is "assigned" to the facility "closest" to it, i.e., the facility w that minimizes d_{wv} among the k facilities.

The multimedian heuristic algorithm below finds a set S of k nodes that is not guaranteed to be optimal but is locally optimal in the sense that for any set X of k nodes if X and S have $k - 1$ nodes in common, then $J(S) \leq J(X)$, i.e., S has as good an objective value as any set X obtained by replacing one node of S. The algorithm begins by finding a single median and then increases the number of selected points in steps of one at a time until this becomes equal to the required number, k. By Hakimi's theorem, only the nodes need to be considered for inclusion in the set S. See Figure 3.4 for a worked example.

Algorithm Multimedian Heuristic Algorithm
input: Graph $G = (V, E)$, nonnegative edge costs c, node demands $h : V \to \Re$, and
a number $k \geq 1$.
output: "Locally optimal" set S with $|S| = k$; S may not be a k-median,
but for any set X with $|X| = k$ and $|S \cap X| = k - 1$, $J(S) \leq J(X)$.

find a single median of G, and suppose that it is node x;
$S := \{x\}$;
while $|S| < k$ **do**
(facility addition step)
find a node $y \in V \backslash S$ that maximizes $J(S) - J(S \cup \{y\})$;
(adding y to S gives the maximum improvement in the objective value)
$S := S \cup \{y\}$;

Figure 3.4: Finding a single median and an (approximate) 2-median on the example in Figure 3.3 (from Chapter 6, "Urban Operations Research", by Larson and Odoni). The matrix $[h(v)\ d_{wv}]$ is shown. The optimum location for the facility (single median) is at node c.



$[h(v)\ d_{wv}]$

	a	b	c	d	e	f	g	h	row sums
a	0	4	6	6	6	0	32	6	60
b	12	0	4	9	4	0	28	5	62
c	9	2	0	3	3	0	20	3	40
d	6	3	2	0	4	0	24	4	43
e	18	4	6	12	0	0	32	6	78
f	15	4	4	9	5	0	12	1	50
g	24	7	10	18	8	0	0	3	70
h	18	5	6	12	6	0	12	0	59

single median: node c, with $J(\{c\}) = 40$.

2-median heuristic:

facility addition: find $\min_{y \in V} J(\{y, c\})$.

$y \to$	a	b	c	d	e	f	g	h
J({y,c})	31	38	–	34	37	30	20	29

let $S = \{g, c\}$, with $J(S) = 20$.

solution improvement:

try swapping c with each of a, b, d, e, f, h in turn:

$S = \{d, g\}$ with $J(S) = 18$ gives improvement;

try swapping d with each of a, b, c, e, f, h in turn: no improvement;

try swapping g with each of a, b, c, e, f, h in turn: no improvement;

stop and **output** $S = \{d, g\}$ with $J(S) = 18$.

```
        (solution improvement step)
SWAP:   for each x ∈ S do
            for each y ∈ V\S do
                if J({y} ∪ (S\{x})) < J(S)
                then S := (S\{x}) ∪ {y};
                     go to SWAP;
                end (if);
            end (for);
        end (for);
end (while);
output S;
```

3.5 Bicriteria Approximations for the k-median Problem

In this section, we sketch a bicriteria approximation algorithm for an important facility location problem, the k-median problem. The performance guarantees are nearly best-possible.

3.5.1 Introduction

The k-median problem is an important problem that arises in a variety of contexts from facility location, clustering and data compression: In the most general version of the problem, we are given a complete directed graph with nonnegative costs c_{ij} in its arcs (ij). The objective is to choose k vertices as medians so that the sum of the costs from every vertex to its closest median is minimized. The *symmetric* version of the problem is the undirected counterpart of the problem obeying $c_{ij} = c_{ji}$. Another special case arises when the (symmetric or asymmetric) costs obey the triangle inequality.

The k-median problem can be viewed as the bicriteria problem of minimizing simultaneously the number of medians used and the sum of costs to the nearest medians. Call the latter quantity "median cost." In our terminology, this is the problem (Number of medians, Median cost, Empty subgraph) or simply (Number of medians, Median cost). Note that following our convention, we have imposed a budget k on the first objective in the above formulation.

We shall study a rounding method due to Lin and Vitter [3] that works on the linear programming relaxation of a natural IP formulation for the problem. First, we review hardness results on approximating this problem.

3.5.2 Hardness of approximation

The k-median problem is very closely related to the dominating set problem in an undirected graph. In the latter problem, we are given an undirected graph $G = (V, E)$ and a positive integer k, and the task is to determine if there exist a dominating set of G of size at most k. In other words, does there exist k nodes in G such that every vertex is either one among these k or is adjacent to one among these k vertices.

Even when the cost matrix for the k-median problem is symmetric and obeys the triangle inequality (i.e., metric), it is hard to obtain a $(o(\log n), 1)$ approximation in an n-node problem.

Theorem 3.5 *Let $f(n) = o(\log n)$, and suppose there exists a $(f(n), 1)$ approximation algorithm for the k-median problem on an n-node graph with metric costs (i.e., if V_k is an optimal median set, the approximation algorithm outputs a set U of medians with $|U| \leq f(n)k$ and $\sum_{i\in V} \min_{j\in U}\{c_{ij}\} \leq \sum_{i\in V} \min_{j\in V_k}\{c_{ij}\}$). Then both the dominating set and set covering problems can be approximated within a factor of $O(f(n))$.*

Proof: We use a reduction from the dominating set problem. Given an unweighted graph G, we use its edges to define a metric on its nodes, namely the shortest path metric using edges in G. Thus c_{ij} is the number of edges in a shortest path between i and j in G. Suppose a minimum dominating set in G has size k. Then there is a solution to the corresponding k-median problem with median cost at most $n - k$.

Conversely, suppose the approximation algorithm for the median problem outputs a median set U with at most $f(n)k$ vertices and median cost at most $n-k$. We bound the number of vertices non-dominated by U, i.e., not in or adjacent to U. We claim there can be at most $f(n)k$ such vertices. For otherwise, the median cost of the solution U would be at least $2f(n)k + (n - f(n)k - f(n)k)$. The first term is a lower bound on the median cost of all the nondominated vertices, while the second term is a lower bound on the cost of all the remaining vertices other than those in U, and those that are nondominated. The resulting median cost is greater than $n - k$, a contradiction. Now, it is easy to form a dominating set out of U: simply include all the nondominated vertices in the set to obtain a dominating set of size at most $2f(n)k$. This implies the theorem. ∎

For the same problem, without the triangle inequality, it is easy to show that there is no $(1, \rho)$ approximation for any ρ.

Theorem 3.6 *Let $\rho > 1$ and suppose there exists a $(1, \rho)$ approximation algorithm for the k-median problem on an n-node graph with symmetric costs. Then there is a polynomial time algorithm for the dominating set problem.*

Proof: We use a similar reduction from the dominating set problem as before. Given an unweighted graph G, we now define c_{ij} to be one if the edge ij is in G, and to be $\rho(n - k) + 1$ if ij is not an edge. Suppose a minimum dominating set in G has size k. Then there is a solution to the corresponding k-median problem with median cost at most $n - k$.

Conversely, a $(1, \rho)$ approximation algorithm will return a median set of size at most k such that its median cost is at most $\rho(n - k)$. But by our cost function, every node in the graph must be adjacent to this median set since the cost of every non-edge is more than $\rho(n - k)$. Thus the returned median set is dominating, giving us an algorithm for the dominating set problem. ∎

In fact the proof of the above theorem is approximation preserving and gives the following stronger theorem.

Theorem 3.7 *Let $\rho > 1$ and suppose there exists a $(f(n), \rho)$ approximation algorithm for the k-median problem on an n-node graph with symmetric costs. Then there is a $f(n)$-approximation algorithm for the dominating set problem.*

The fact below is a consequence of a straightforward approximation-preserving reduction (do it!) of the set covering problem to a dominating set problem in an undirected graph.

Fact 3.8 *A $f(n)$-approximation algorithm for the dominating set problem implies an $O(f(n))$-approximation for the unweighted set cover problem.*

By the known hardness results on the set covering problem [3], we can infer that $f(n) = \Omega(\ln n)$.

3.5.3 Rounding by filtering

In this section, we present a $((1+\frac{1}{\epsilon})H(n), 1+\epsilon)$-approximation for the general k-median problem for any $\epsilon > 0$. To do this, we first formulate the problem as an integer program.

Formulation

The choice variables in our IP formulation correspond to deciding whether a median is located at any given node. Let y_j denote this choice, i.e., $y_j = 1$ exactly when vertex j is chosen as a median node. To formulate the objective function, it is convenient to have a variable that denotes for every vertex i, the closest median node to this vertex that it is assigned to. Let x_{ij} denote the assignment of node i to median j. thus $x_{ij} = 1$ exactly when $y_j = 1$ (node j is a median) and vertex i is assigned to j, where node j is the closest median node to i. Now we are ready to write down the integer programming formulation.

$$\boxed{\begin{array}{llrcll}
(IP) & \text{minimize} & \sum_{i\in V}\sum_{j\in V} c_{ij}x_{ij} & & & \\
& \text{subject to} & \sum_{j\in V} x_{ij} & \geq & 1 & \forall\, i\in V \\
& & \sum_{j\in V} y_j & \leq & k & \\
& & x_{ij} & \leq & y_j & \forall\, i,j\in V \\
& & x_{ij}, y_j & \in & \{0,1\} & \forall\, i,j\in V
\end{array}}$$

The LP relaxation of the above formulation relaxes the last set of integrality constraints to allow the variables x_{ij} and y_j to take rational values between 0 and 1.

A solution to the linear programming relaxation is completely characterized by a fractional assignment $\hat{y_j}$ to the variables y_j. The remaining fractional variables x_{ij} can be set "optimally" as follows [1].

We assign each vertex to all its nearest fractional medians whose $\hat{y_j}$-values sum to at least 1. Formally, sort the values c_{ij} for $j \in V$ so that $c_{ij_1(i)} \leq c_{ij_2(i)} \leq \ldots c_{ij_n(i)}$. Let p be the index such that $\sum_{l=1}^{p-1} \hat{y_l} < 1 \leq \sum_{l=1}^{p} \hat{y_l}$. Then set $\hat{x_{ij}} = \hat{y_j}$ for $j = j_1(i), \ldots, j_{p-1}(i)$, and $x_{i\hat{j_p}(i)} = 1 - \sum_{l=1}^{p-1} \hat{y_l}$. For all other j, set $\hat{x_{ij}} = 0$.

Filtering

The key idea in rounding the LP solution $\hat{y_j}$ is to filter out (set to zero) some of the integral assignments x_{ij} by using information from $\hat{y_j}$. In this case, this corresponds to allowing every vertex i to be assigned only to the remaining (non-filtered) centers. We must do the filtering so that there is still a good median solution where every vertex i can be assigned to some non-filtered vertex that is chosen in the solution. In our case, we can view this assignment as a fractional set covering problem and hence derive a good filtering rule.

Denote by $\hat{C}_i$ the cost contribution of vertex i to the fractional median cost, i.e. $\hat{C}_i = \sum_{j \in V} c_{ij} \hat{x_{ij}}$. For a given $\epsilon > 0$, define the neighborhood of i to be

$$S_i = \{j \in V | c_{ij} \leq (1+\epsilon)\hat{C}_i\}.$$

The filtering now disallows the choices x_{ij} for those j that are not in the neighborhood S_i. Despite this, if we regard y_j as fractional choices for the medians, by our filtering rule, we ensure that every vertex i still has enough fractional medians that it can be assigned to.

Lemma 3.9 *For every $i \in V$ and $\epsilon > 0$,*

$$\sum_{j \in S_i} \hat{y}_j > \frac{\epsilon}{1+\epsilon}.$$

Proof: We prove by contradiction. Suppose $\sum_{j \in S_i} \hat{y}_j \leq \frac{\epsilon}{1+\epsilon}$. Then

$$\begin{aligned} \hat{C}_i &= \sum_{j \in V} c_{ij} \hat{x_{ij}} \\ &\geq \sum_{j \notin S_i} c_{ij} \hat{x_{ij}} \\ &> (1+\epsilon)\hat{C}_i \sum_{j \notin S_i} \hat{x_{ij}} \\ &\geq (1+\epsilon)\hat{C}_i (1 - \frac{\epsilon}{1+\epsilon}) \\ &= \hat{C}_i \end{aligned}$$

The contradiction proves the lemma. ∎

Transformation to set covering

The above lemma allows us to look upon the problem of assigning medians to cover the neighborhoods S_i of all the vertices in the graph as a set covering problem. We must now choose a small number of medians to cover all the sets S_i. Note that this restriction immediately implies a bound on the total median cost of our choice since every vertex i is assigned to some median within distance at most $(1+\epsilon)\hat{C}_i = (1+\epsilon)\sum j \in V \hat{x_{ij}}$.

Lemma 3.10 *Suppose every vertex i is assigned to a median within its neighborhood S_i. Then the median cost of the resulting solution is at most $(1+\epsilon)\sum_{i \in V} \sum_{j \in V} c_{ij} \hat{x_{ij}}$, which in turn is at most $(1+\epsilon)$ times the the optimal k-median cost.*

It remains to solve the resulting set cover problem to find an integral solution with a minimum number of medians. To this end, we can use the several algorithms we studied earlier for rounding a fractional solution to the set cover problem. However, we still do not have a fractional feasible solution for the set cover problem of assigning medians to cover all the neighborhoods S_i. However, Lemma 3.9 is useful in producing such a solution.

Set $\bar{y_j} = (1+\frac{1}{\epsilon})\hat{y_j}$ for every vertex $j \in V$. By lemma 3.9, for any vertex i, we have

$$\sum_{j\in S_i} \bar{y_j} \geq \sum_{j\in S_i}(1+\frac{1}{\epsilon})\hat{y_j} > 1.$$

In other words, $\bar{y_j}$ s a fractional solution to the set covering problem and the total fractional value of this solution is

$$\sum_{j\in V} \bar{y_j} \leq \sum_{j\in V}(1+\frac{1}{\epsilon})\hat{y_j} \leq (1+\frac{1}{\epsilon})k.$$

Using any of numerous approximation algorithms for the set covering problems we saw earlier, we can find an integral solution of size at most the fractional cover times $H(n)$ where n is the size of the largest set and $H(n) = 1+\frac{1}{2}+\ldots+\frac{1}{n} \leq \ln n + 1$, the n^{th} harmonic number. In our case n is at most the number of nodes in the graph since every set is a neighborhood of nodes. Thus the value of the approximate set cover we find is at most $(1+\frac{1}{\epsilon})H(n)k$. Thus, we have proved the following theorem.

Theorem 3.11 *For any $\epsilon > 0$, there is a $((1+\frac{1}{\epsilon})H(n), 1+\epsilon)$-approximation for the general k-median problem on an n-node graph.*

3.6 Exercises

1. Find (i) the node center, and (ii) the absolute center of the network G, c in Figure 3.5 using any method you like. First, give a precise explanation of your method.

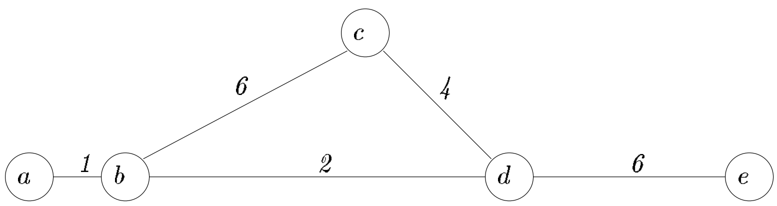


Figure 3.5: Network for exercise

2. Construct a network G, c such that the absolute center is located on an edge that is *not* incident to the node center.

3. The goal is to show that the following algorithm finds a single absolute center of a tree.

Algorithm Single Absolute Center of a Tree
input: Tree $G = (V, E)$, and nonnegative edge costs c.
output: Absolute center x^*.

choose an arbitrary point x in G;
find an (end) node v_s that is farthest from x, i.e., $d(x, v_s) = \max_{v \in V} d(x, v)$;
find an (end) node v_t that is farthest from v_s, i.e., $d(v_s, v_t) = \max_{v \in V} d(v_s, v)$;
let x^* be the midpoint of the path from v_s to v_t;
output x^*; (x^* is the absolute center of G, c)

Show that for all points $x \in G$ we must have

$$m(x) = m(x^*) + d(x, x^*).$$

Hence, argue that

$$m(v_s) = 2\, m(x^*)$$

and, therefore, that x^* must lie on the path associated with $m(v_s)$ and must be at the halfway point between v_s and v_t.

4. Explain whether or not the following method finds the absolute center of a graph $G = (V, E)$ with respect to edge costs $c : E \to \Re_+$:
Find a pair of nodes x, y such that the cost of a shortest path from x to y, $d(x, y)$, is **maximum** among all pairs of nodes. Take the absolute center x^* to be a point at the middle of a shortest path from x to y.

5. Either prove the following statement or give a counterexample: For every network G, c (G is any graph, c is any nonnegative cost function on its edges) and a single absolute center x^* of G, c there exists a pair of nodes v_s and v_t such that x^* is the absolute center of a shortest path from v_s to v_t (i.e., x^* is at the halfway point between v_s and v_t).

6. (More improved algorithm for absolute center; from Chapter 6, "Urban Operations Research", by Larson and Odoni.) The goal is to obtain an algorithm for the absolute center that is better than the improved algorithm (page 28), by using a better lower bound $L_2(pq)$ given below for the value $m(x_\ell)$ of the local center x_ℓ of an edge pq. For the edge pq, let $v_p \in V$ and $v_q \in V$ denote the farthest nodes from p and q, respectively, i.e., $m(p) = d(p, v_p)$ and $m(q) = d(q, v_q)$. Clearly, the local center x_ℓ of pq satisfies $m(x_\ell) \geq \max(d(x_\ell, v_p),\ d(x_\ell, v_q))$. Prove:

   (a) For a point x in the interior of edge pq (i.e., $x \in pq$, $x \neq p$ and $x \neq q$), $\max(d(x, v_p),\ d(x, v_q))$ may attain at most a single local minimum.

   (b) Such a local minimum, if it exists, is attained at the point $x \in pq$ whose distance from p along edge pq is
   $$\frac{c(pq) + d(q, v_p) - d(p, v_q)}{2},$$
   and at this point $\max(d(x, v_p),\ d(x, v_q))$ attains the value
   $$L_2(pq) = \frac{d(p, v_q) + d(q, v_p) + c(pq)}{2}.$$

(c) $L_2(p,q) \geq \dfrac{m(p)+m(q)-c(pq)}{2}$, i.e., $L_2(p,q)$ provides a better lower bound on $m(x_\ell)$ than the one used in the algorithm on page 28.

7. (Location of a "supporting facility"; from Chapter 6, "Urban Operations Research", by Larson and Odoni.) Consider the network of Figure 3.6 and suppose that the nodes v represent five cities, the numbers next to the nodes give the demands $h(v)$ of the cities, and the numbers next to the links vw give the mile length c_{vw} of roadways connecting the cities. Cars travel on the roadways at an effective speed of 30 m.p.h. Assume that a major facility, say an airport, has been located at some point on this network. A regional planning group now wishes to install a high-speed transportation link to the airport with a single station. The high-speed vehicles will travel on the network at twice the speed of cars and their route will be the shortest route to the airport. It is assumed that travelers to the airport will choose the combination of transportation modes which minimizes their access time to the airport (ignoring transfer times). To clarify the description above, assume that the airport is at node b and that the single station of the high-speed link is located at node e. Then, the access time to the airport of travelers from node e is 25 minutes. However, the access time of travelers from node a is still 40 minutes. (It would take travelers from node a exactly 20 minutes to get to node e by car and then another 25 minutes to get from node e to node b, so that it is better to go directly to the airport by car.)

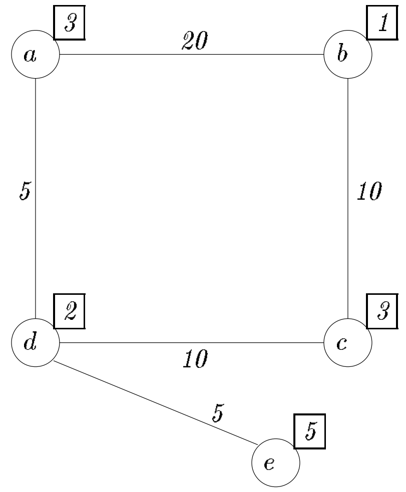


Figure 3.6: Network for exercise

(a) Show that no matter where the airport is located, an "optimal location" for the station of the high-speed vehicles must be on a node of the graph, where an optimal location is one that minimizes the total weighted travel distance to the airport for travelers from the five cities. Note that the airport is not restricted to be at one of the nodes of the network. Alternatively, show this for a general network rather than for this specific case only.

(b) Assuming that the airport is located at node b, where should the station be located? Devise an algorithm for solving this type of problem.

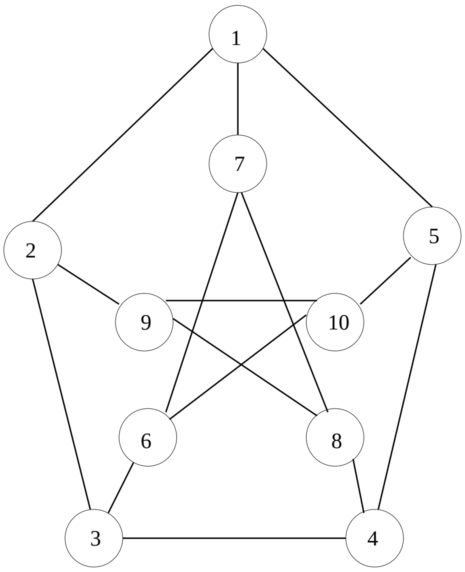
(c) Assume now that travel time, in minutes, by car between any two points x and y on the network is given by $f(d_{xy}) = \left(\frac{d_{xy}}{5}\right)^2$, where d_{xy} is the minimum cost of a path between the two points, and that travel time through the high-speed link between the same two points is given by $g(d_{xy}) = \frac{1}{2}\left(\frac{d_{xy}}{5}\right)^2$. The optimal location may not be on a node. How would you answer part (b) now?

(d) Show that the result you proved in part (a) holds as long as the functions $f(d_{xy})$ and $g(d_{xy})$ are both concave in d_{xy} (Mirchandani 1979).

8. Prove Hakimi's k-median theorem, Theorem 3.4.

   (Hint: Suppose X_k is a k-median, and $x \in X_k$ is not a node. Focus on the nodes $v \in V$ such that $d(x, v) < d(X_k \backslash \{x\}, v)$.)

9. Consider the k-median problem on the graph $G = (V, E)$ in Figure 3.7. This is the Petersen graph. Note that there are 10 nodes, 15 edges, and every node $v \in V$ has degree equal to 3. Take the cost of each edge to be 1. Take the node set V to be $\{1, 2, \ldots, 10\}$. The matrix of shortest paths costs $[c_{vw}]$ is given in Figure 3.7.



$$[c_{vw}] = \begin{bmatrix} 0 & 1 & 2 & 2 & 1 & 2 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 & 1 & 2 & 2 & 2 & 2 \\ 2 & 2 & 1 & 0 & 1 & 2 & 2 & 1 & 2 & 2 \\ 1 & 2 & 2 & 1 & 0 & 2 & 2 & 2 & 2 & 1 \\ 2 & 2 & 1 & 2 & 2 & 0 & 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 2 & 2 & 1 & 0 & 1 & 2 & 2 \\ 2 & 2 & 2 & 1 & 2 & 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 0 & 1 \\ 2 & 2 & 2 & 2 & 1 & 1 & 2 & 2 & 1 & 0 \end{bmatrix}$$

Figure 3.7: The Petersen graph $G = (V, E)$ and cost matrix $[c_{vw}]$ for Exercise 9

(a) Prove or disprove:
The objective value of the 2-median problem is at least 11, i.e., the optimal value of

(IP) is at least 11, where (IP) is the integer programming formulation of the 2-median problem. Recall that (IP) has a variable y_j for each node $j \in V$, and has a variable x_{ij} for each ordered node pair $i, j \in V \times V$.

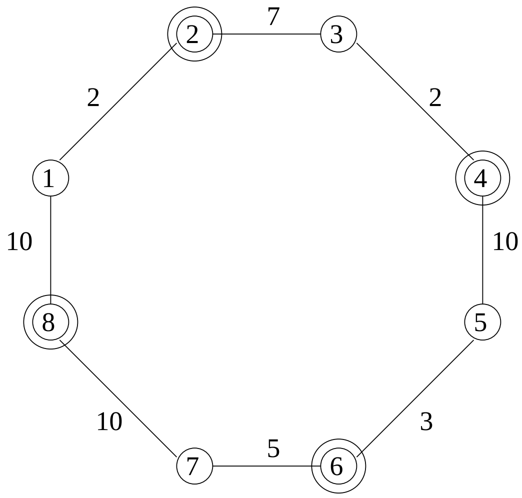
(b) Consider the LP relaxation (P) of (IP). Let $\widehat{y}$ be given by

$$\widehat{y}_j = 1/5, \ \forall j \in V.$$

Compute the optimal $\widehat{x}$ with respect to $\widehat{y}$, and find the objective value of (P) corresponding to the feasible solution $\widehat{x}, \widehat{y}$.

10. The aim is to apply Lin & Vitter's Filtering & Rounding method to an example of the k-median problem, for $k = 1$. Recall the integer linear programming formulation (IP) of the k-median problem, and the LP relaxation (P) of (IP).

Consider the k-median problem in Figure 3.8 below. Note that $k = 1$. The "cost matrix" $[c_{vw}]$ for all node pairs $v, w \in V \times V$ is given in Figure 3.8.



$$[c_{vw}] = \begin{bmatrix} 0 & 2 & 9 & 11 & 21 & 24 & 20 & 10 \\ 2 & 0 & 7 & 9 & 19 & 22 & 22 & 12 \\ 9 & 7 & 0 & 2 & 12 & 15 & 20 & 19 \\ 11 & 9 & 2 & 0 & 10 & 13 & 18 & 21 \\ 21 & 19 & 12 & 10 & 0 & 3 & 8 & 18 \\ 24 & 22 & 15 & 13 & 3 & 0 & 5 & 15 \\ 20 & 22 & 20 & 18 & 8 & 5 & 0 & 10 \\ 10 & 12 & 19 & 21 & 18 & 15 & 10 & 0 \end{bmatrix}$$

Figure 3.8: Graph $G = (V, E)$ and cost matrix $[c_{vw}]$ for Exercise 10.

Let $\widehat{y}$ give a feasible solution to (P), where $\widehat{y}$ is:

$$\widehat{y}_2 = 1/4, \quad \widehat{y}_4 = 1/4, \quad \widehat{y}_6 = 1/4, \quad \widehat{y}_8 = 1/4, \qquad \text{and} \qquad \widehat{y}_j = 0, \ \forall j \in \{1, 3, 5, 7\}.$$

- Find $\widehat{x}$ that minimizes the objective function with respect to $\widehat{y}$.
- Take $\epsilon = 1/2$ and apply the Filtering & Rounding method as follows.
- For each node i, compute the cost $\widehat{C}_i = \sum_{j \in V} c_{ij} \widehat{x}_{ij}$ and compute the neighborhood V_i.
- Write down the set covering problem for the filtered problem, by making use of the neighborhoods V_i.
- Solve the set covering problem, using <u>either</u> inspection <u>or</u> the greedy heuristic.

11. Recall the bicriteria k-median problem (number of medians, median cost). We discussed a $((1+\frac{1}{\epsilon})H(n), 1+\epsilon)$-approximation for any $\epsilon > 0$ in an n-node graph. The goal here is to give an $(O(1), O(1))$-approximation algorithm for the special case of metric costs. In detail:
Suppose that the cost matrix $[c_{vw}]$ $(\forall v, w \in V \times V)$ is symmetric and satisfies the triangle inequality. Let $\epsilon > 0$ be the parameter. Then modify the filtering and rounding method to find a set of columns J^* such that

$$|J^*| \leq (1 + \frac{1}{\epsilon})k,$$

and

$$\sum_{i \in V} \min_{j \in J^*} c_{ij} \leq 3(1+\epsilon) z_{LP},$$

where z_{LP} is the optimal value of the LP relaxation. (In other words, the modified method may increase the number of medians by a factor of $(1 + \frac{1}{\epsilon})$ and must achieve an objective value within a factor of $3(1+\epsilon)$ of z_{LP}.)

Bibliography

[1] S. Ahn, C. Cooper, G. Cornuejols, and A. Frieze, *Probabilistic analysis of a relaxation for the k-median problem*, Math. of Oper. Res. **13**, pp. 1–31 (1988).

[2] N. Christofides, *Graph Theory: An Algorithmic Approach.* Academic Press, London, 1975.

[3] U. Feige, *A threshold of ln n for approximating set cover*, Proc. 28th Annual ACM Symp. on Theory of Computing (1996), pp. 314–318.

[4] D. S. Hochbaum and D. B. Shmoys, *A unified approach to approximation algorithms for bottleneck problems*, J. Assoc. Comput. Mach. **33** (1986), pp. 533–550.

[5] R. C. Larson and A. R. Odoni, *Urban Operations Research.* Prentice-Hall, Englewood Cliffs, N. J., 1981.

[6] J.-H. Lin and J. S. Vitter, *ϵ-approximations with minimum packing constraint violation*, Proc. 24th Annual ACM Symp. on Theory of Computing (1992), pp. 771–782.

[7] J.-H. Lin and J. S. Vitter, *Approximation algorithms for geometric location problems*, Inf. Proc. Lett. **44** (1992), pp. 245–249.

[8] P. B. Mirchandani and R. L. Francis, *Discrete Location Theory.* John Wiley & Sons, New York, 1990.

[9] P. Raghavan, *Probabilistic construction of deterministic algorithms: approximating packing integer programs*, J. Comp. Sys. Sci. **37** (1988), pp. 130–143.

[10] P. Raghavan and C. D. Thompson, *Randomized rounding: a technique for provably good algorithms and algorithmic proofs*, Combinatorica **7** (1987), pp. 365–374.