Approximability Results for the p-Center Problem

Stefan Buettcher

<stefan at buettcher dot org>

Course Project

Algorithm Design and Analysis

Prof. Timothy Chan

University of Waterloo, Spring 2004

The p-Center Problem

The p-Center problem, also known as the Min-Max Multicenter problem or the Facility Location problem, is a famous problem from operations research. Informally, it is the problem of placing fire stations in a city so that the maximum time to reach any point in the city is minimized. Like many other problems, it can be either formulated as a decision or as an optimization problem.

The p-Center Decision Problem

Definition (p-Center problem)
Given a complete graph $G = (V, E)$, a weight function $w : V \to \mathbb{N}$, a distance function $d : E \to \mathbb{N}$, a number $p \in \mathbb{N}$, and a number R; is there a vertex set $S \subseteq V$, $|S| = p$, such that for all vertices $v \in V$:

$$v \in S \ \vee \ \exists s \in S : d(s, v) \cdot w(v) \leq R. \tag{1}$$

For a given set S, the smallest R such that (1) holds is called the *radius* of the set S. The elements $s \in S$ are sometimes called *centers*.

There are many different definitions of the problem. Garey and Johnson [GJ79], for example, do not restrict the set S to being a subset of V. Instead, they allow a point $s \in S$ to be either a vertex v of the graph or any point on an edge e (they consider the edges as line segments between two points).

According to Garey and Johnson, Kariv and Hakimi [KH76] have shown that the general p-Center problem is **NP**-complete. It remains **NP**-complete if

- $w(v) = 1$ for all $v \in V$,
- $d(x, z) \leq d(x, y) + d(y, z)\ \forall x, y, z \in V$ (triangle inequality), and
- $d(x, y) = d(y, x)\ \forall x, y \in V$ (symmetric p-Center problem).

The **NP**-completeness of the restricted version of the problem can be shown by giving a polynomial-time reduction from Dominating Set to p-Center:

Given an an undirected graph $G = (V, E)$ (an instance of Dominating Set), we construct the complete graph $G' = (V, E')$, let $w(v) = 1\ \forall v \in V$, and define the distance function:

$$d(x, y) = \begin{cases} 1, \text{ if } (x, y) \in E, \\ 2, \text{ if } (x, y) \notin E. \end{cases} \tag{2}$$

The reduction is straightforward and can be done in polynomial time. Clearly, the distance function d is symmetric (because G is undirected), and d obeys the triangle inequality. Now, G' has a center set of size p and radius 1 if and only if G has a dominating set of size p. □

The p-Center Optimization Problem

The p-Center optimization problem can be derived from the decision problem in the natural way: Given a complete graph $G = (V, E)$, a weight function $w : V \to \mathbb{N}$, a distance function $d : E \to \mathbb{N}$, and a number $p \in \mathbb{N}$; what is the minimum R such that the answer to the associated decision problem is "yes"? For the optimization problem, it is usually assumed that $w(v) = 1\ \forall v \in V$.

There are 3 different types of the optimization problem, depending on which of the restrictions mentioned above are applied.

1. The general optimization problem in which the choice of the distance function d is not restricted in any way.
2. The metric problem in which d satisfies the triangle inequality.

3. The metric and symmetric problem in which $d(x, y) = d(y, x)$, and d satisfies the triangle inequality.

Using the same reduction as above, it can be shown there is no polynomial-time approximation algorithm with approximation factor $q < 2$ for the metric and symmetric p-Center optimization problem unless $\mathbf{P} = \mathbf{NP}$, since a solution with radius $r < 2$ would imply $r = 1$ and thus Dominating Set would be solvable in polynomial time. In fact, Hochbaum and Shmoys [HS86] gave a factor-2 approximation algorithm for this problem, so the bound is tight.

For the general problem with unrestricted distance function, it can be shown that, if $\mathbf{P} \neq \mathbf{NP}$, there is no polynomial-time approximation algorithm with approximation factor $f(|V|)$, where V is the set of vertices in the input instance and f is any (polynomial-time computable) function: Suppose we have an algorithm that can compute a solution with radius $r \leq r^* \cdot f(|V|)$ in polynomial time, where r^* is the radius of the optimal solution. Then, applying the same reduction as above, we have a distance function

$$d(x, y) = \begin{cases} 1, \text{ if } (x, y) \in E, \\ f(|V|) + 1, \text{ if } (x, y) \notin E. \end{cases} \tag{3}$$

Thus, the algorithm could be used to solve Dominating Set in polynomial time, which implies $\mathbf{P} = \mathbf{NP}$.

While this is bad news, it is not as bad as it looks at the first sight. Most real-world instances of the problem either obey the triangle inequality, or they can be made metric by setting $d(x, y)$ to the length of the shortest path between x and y without contradicting the original meaning of the problem.

Until recently, not much has been known about the approximability of the metric, but asymmetric p-Center problem. In fact, Shmoys [Sh95] says about this problem:

A natural generalization of the problem is to relax the restriction that the distance matrix be symmetric. This turns out to be a non-trivial generalization and essentially nothing is known about the performance guarantee for this extension.

This is bad because asymmetry is inherently present in many scenarios (like one-way streets and mountain roads in the firestation problem). However, the situation completely changed when Panigrahy and Vishwanathan presented their algorithm for the problem in 1998.

An Approximation Algorithm for the Asymmetric p-Center Problem

In 1998, Panigrahy and Vishwanathan [PV98] presented the first non-trivial approximation algorithm for the asymmetric p-Center optimization problem. Their algorithm achieves an amazing approximation factor of $O(\log^*(n))$, where n is the number of vertices in the input graph. While this is obviously not a constant factor, $\log^*(n)$ is in fact very close to constant.

Before we can present the actual algorithm found by Panigrahy and Vishwanathan, we first need to give a few definitions, explain the connection between the p-Center problem and Set Cover and introduce and analyze two helper functions that are employed by the final algorithm.

Definition (R-cover)
We say that a vertex x R-covers a vertex y iff $d(x, y) \leq R$ for some number R. The notation is extended to sets in the natural way: A set X R-covers a set Y iff

$$\forall y \in Y \; \exists x \in X : \; x \; R\text{−covers } y. \tag{4}$$

We also say that X is an R-cover for Y.

We let $\Gamma_R^{out}(v)$ denote the endpoints of all outgoing edges e of a vertex v, such that $d(e) \leq R$, i.e. the set of vertices that are R-covered by v. In particular, we have $v \in \Gamma_R^{out}(v)$. Analogously, $\Gamma_R^{in}(v)$ is the set of all vertices that are R-covering the vertex v.

Definition (SET COVER problem)
Given a set S of objects, a collection of subsets $\mathcal{F} \subseteq 2^S$, and $k \in \mathbb{N}$; is there a collection $\mathcal{F}' \subseteq \mathcal{F}$, $|\mathcal{F}'| \leq k$, such that

$$\bigcup_{F \in \mathcal{F}'} F = S, \tag{5}$$

i.e., every element of S appears at least in one of the sets in $\mathcal{F}'$? The SET COVER optimization problem is defined accordingly.

Consider a slight change to the p-CENTER problem. Instead of finding a center set of size p with minimum radius, we want to know: What is the smallest p such that there is a center set of radius $\leq R$, for a given number R?

This problem is equivalent to the problem of finding a minimal SET COVER for the following problem instance:

$$S = V, \ \mathcal{F} = \{\Gamma_R^{out}(v) \mid v \in V\},$$

where V is the vertex set of the modified p-CENTER instance. This allows for the application of known SET COVER algorithm for the solution of the modified p-CENTER problem.

Algorithm (*Greedy-Set-Cover*)

```
// Input: $(S, \mathcal{F})$, as described above.
// Output: $\mathcal{F}' \subseteq \mathcal{F}$.

$\mathcal{F}' := \emptyset$
while ($S \neq \emptyset$) do
   pick $F \in \mathcal{F}$ such that $F \cap S$ is maximal
   $S := S \setminus F$
   $\mathcal{F} := \mathcal{F} \setminus \{F\}$
   $\mathcal{F}' := \mathcal{F}' \cup \{F\}$
return $\mathcal{F}'$
```

Analysis

If p is the cardinality of an optimal solution to the problem instance, and $n = |S|$, then the algorithm *Greedy-Set-Cover* returns a set cover of size at most $p \cdot (1 + \ln(\frac{n}{p}))$.

Proof In each iteration of the algorithm, we know that, for the residual set S, there is a set cover of size at most p. Therefore, the set $\mathcal{F}$ must contain at least one element F such that $|F \cap S| \geq \frac{|S|}{p}$. Hence, in each iteration, the size of S drops at least by factor $1 - \frac{1}{p}$.

How many iterations x do we need until $|S| = p$?

$$\begin{aligned}
& p = n \cdot (1 - \frac{1}{p})^x \\
\Leftrightarrow \quad & \ln(\frac{p}{n}) = x \cdot \ln(1 - \frac{1}{p} \\
\Leftrightarrow \quad & x = \ln(\frac{p}{n}) \cdot \frac{1}{\ln(p) - \ln(p-1)} \\
\Rightarrow \quad & x \leq \ln(\frac{p}{n}) \cdot \frac{1}{\frac{1}{p}} \quad \text{(because } \ln(p) - \ln(p-1) \geq \frac{\mathrm{d}}{\mathrm{d}p} \ln(p) = \frac{1}{p}) \\
\Rightarrow \quad & x \leq p \cdot \ln(\frac{p}{n}).
\end{aligned}$$

So, after $p \cdot \ln(\frac{n}{p})$ iterations, there are at most p elements left. These elements can trivially be covered by p sets (otherwise, no solution exists, which contradicts the assumption that we know a solution of size p), giving a solution size $|\mathcal{F}'| \leq p \cdot (1 + \ln(\frac{n}{p}))$. $\square$

Please note that, since $p \cdot \ln(\frac{n}{p}) \in O(n^2)$, we have a polynomial number of iterations. All operations in the loop body can be executed in polynomial time, so the algorithm *Greedy-Set-Cover* is a polynomial-time algorithm.

Using the algorithm *Greedy-Set-Cover* as a helper function, a second algorithm can be designed. This second algorithm can compute a cover for a subset $A \subseteq V$, where V is the set of vertices in the p-Center input instance. Assuming that we know the set A has an R-cover P of size p, and further assuming that we know a set Q that R-covers $(V \setminus A)$, the following algorithm computes a set $\mathcal{C}$ of size $\leq 2p$ that, together with Q, r-covers V, where $r \in O(R \cdot \log^*(|A|))$.

Algorithm (*Recursive-Cover*)

```
// Input: $(V, A \subseteq V, p, R)$.
// Output: $\mathcal{C} \in V$, $|\mathcal{C}| \leq 2p$.
// Precondition: $P$ $(|P| = p)$ $R$-covers $A$, $Q$ $R$-covers $(V \setminus A)$.
// Postcondition: $(Q \cup \mathcal{C})$ $r$-covers $V$, $r \in O(R \cdot \log^*(|A|))$.

$A_0 := A$
$i := 0$
while $(|A_i| > 2p)$ do
  construct the set cover instance $\mathcal{SC} := (A_i, \{\Gamma_R^{out}(x) \cap A_i \mid x \in V\})$
  $A'_{i+1} :=$Greedy-Set-Cover$(\mathcal{SC})$
  $A_{i+1} := A'_{i+1} \cap A$   // restrict to $A$; rest is already $R$-covered by $Q$
  $i := i + 1$
return $A_i$
```

Analysis

Assuming that the algorithm terminates, the condition $|\mathcal{C}| = |A_i| \leq 2p$ is obviously fulfilled. Furthermore, we can prove the following

Lemma For every i, the set $(A_i \cup \mathcal{C})$ $2iR$-covers A.

Proof At the start of the algorithm, we have $A_0 = A$, so A_0 clearly 0-covers A. After the execution of *Greedy-Set-Cover*, we know that

$$\begin{aligned} & A'_{i+1}\ R\text{−covers } A_i \\ \Rightarrow\ & ((A'_{i+1} \cap A) \cup (A'_{i+1} \cap (V \setminus A)))\ R\text{−covers } A_i \\ \Rightarrow\ & (A_{i+1} \cup (V \setminus A))\ R\text{−covers } A_i \\ \Rightarrow\ & (A_{i+1} \cup Q)\ 2R\text{−covers } A_i \end{aligned}$$

because, by the algorithm's precondition, the set Q R-covers $(V \setminus A)$. Hence, by induction, A_i $2iR$-covers A $\forall i \in \mathbb{N}$. $\square$

In order to prove that finally V is $O(R \cdot \log^*(|A|))$-covered, it is sufficient to show that the number of iterations of the algorithm is in $O(\log^*(|A|))$.

Proof For the algorithm *Greedy-Set-Cover*, we have proven that the size of the set returned is bounded by $p \cdot (1 + \ln(\frac{n}{p}))$, if the optimal solution set to the input set S, $n = |S|$, is of size at most p. By the precondition of *Recursive-Cover*, we know that A has an R-cover of size p. Hence, every A_i has an R-cover of size p, because $A_i \subseteq A$ $\forall i$.

We can conclude that

$$|A_{i+1}| \leq |A'_{i+1}| \leq p \cdot (1 + \ln(\frac{|A_i|}{p})),$$

and thus $|A_{i+1}| \in O(p \cdot \ln(\frac{|A_i|}{p}))$. If we apply the recursive formula several times, the p's annihilate each other, and we finally get

$$|A_i| \in O(p \cdot \ln^{(i)}(\frac{|A|}{p})).$$

If we want to know how many iterations are executed, we have to find an i such that $\ln^{(i)}(\frac{|A|}{p}) \leq 2$ (termination criterion of the algorithm). But this is just the definition of $\log^*(\frac{|A|}{p})$ (up to a constant factor).

Hence, the number of iterations is $O(\log^*(\frac{|A|}{p})) \subseteq O(\log^*(|A|))$, and by applying the Lemma we know that $(\mathcal{C} \cup Q)$ r-covers the set A, with $r \in O(\log^*(|A|))$. $\square$

Again, please note that, since $O(\log^*(|A|)) \subseteq O(|V|)$, and all the operations in the loop body can be executed in polynomial time (we have shown that *Greedy-Set-Cover* is a polynomial-time algorithm), the algorithm *Recursive-Cover* runs in polynomial time.

Right know, it is not clear what these results are good for. After all, we are interested in a center set of size p, not one of size $2p$, which *Recursive-Cover* produces. Surprisingly, it can be used nonetheless, as we will see.

The Final Algorithm

The final algorithm, *Approximate-p-Center*, takes $\mathcal{PC} = (V, d, p)$, i.e. an instance of the asymmetric p-Center optimization problem, and a number R, which is the radius R of an optimal solution to the problem. Of course, R is not known, but since there are at most $2 \cdot \binom{|V|}{2}$ different values $d(x, y)$, *Approximate-p-Center* can be executed $2 \cdot \binom{|V|}{2}$ times, each time with a different R value, and at the end the best feasible solution found is taken as the result. Because *Approximate-p-Center* runs in polynomial time, as we will show, the resulting algorithm would still be a polynomial-time algorithm.

Approximate-p-Center consists of two phases. In the first phase, the algorithm constructs a set $\mathcal{C}_1$ of vertices that have to belong to any optimal solution to the problem. In the second phase, using the function *Recursive-Cover*, it computes another set $\mathcal{C}_2$ that, together with $\mathcal{C}_1$, $O(R \cdot \log^*(|V|))$-covers the whole set V.

Algorithm (*Approximate-p-Center*)

```
// Input: $(V, d, p, R)$, as stated above.
// Output: A set $\mathcal{C} \subseteq V$.
// Precondition: The optimal solution to the problem instance $(V, d, P)$ has radius $R$.
// Postcondition: $\mathcal{C}$ $r$-covers $V$, where $r \in O(\log^*(|V|))$.

$\hat{p} := p$   // some initialization
$A := V$
$\mathcal{C}_1 := \emptyset$

// Phase I: find parts of the optimal solution

while there is a vertex $v \in A$, such that $\Gamma^{out}_R(v) \supseteq \Gamma^{in}_R(v)$, do
  $\mathcal{C}_1 := \mathcal{C}_1 \cup \{v\}$   // add $v$ to the solution set
  $A := A \setminus \Gamma^{out}_{2R}(v)$   // remove all vertices $2R$-covered by $v$
  $\hat{p} := \hat{p} - 1$   // we know that the remaining set $A$ can be $R$-covered by one less vertex
$A' := A \setminus \Gamma^{out}_{5R}(\mathcal{C}_1)$   // remove everything $5R$-covered by $\mathcal{C}_1$

// Phase I finished; we know that there is a $\hat{p}$-center that $R$-covers $A$

// Phase II: augment the set of vertices already found

if $(A' \neq \emptyset)$ then   // if there is anything left
  $\mathcal{C}_2 :=$Recursive-Cover$(V, A', \frac{\hat{p}}{2}, 5R)$   // $(\mathcal{C}_1 \cap \mathcal{C}_\in)$ is an $O(\log^*(|V|))$-cover for $A'$

return $(\mathcal{C}_1 \cup \mathcal{C}_2)$
```

Theorem

The algorithm *Approximate-p-Center* runs in polynomial time and returns a set $\mathcal{C} \subseteq V$ that $O(R \cdot \log^*(|V|))$-covers V.

The proof of the theorem is divided into three parts. First, it is shown that the algorithm has indeed polynomial running time. Then, we show that the set A of vertices that are remaining uncovered after phase I has an R-cover of size $\hat{p}$. Finally, it is shown that, after the execution of phase I, a set of size $\frac{\hat{p}}{2}$ can be found that, together with $\mathcal{C}_1$ from phase I, $5R$-covers A'. After this proof, *Recursive-Cover* may be applied to the set A', and the theorem follows.

Proof (First part: The algorithm has polynomial running time.)

Each Γ set that is used in phase I can be computed in time $O(|V|)$ by a simple loop over all vertices. Thus, the loop condition can be checked in time $O(|V|^2)$ (creating the Γ set for each of the remaining vertices). The set $(A \setminus \Gamma_{2R}^{out}(v)$ can be computed in time $O(|V|)$, too. Since there are at most $O(|V|)$ iterations of the loop, the total running time of phase I is $O(n^3)$.

The reduction from A to A' can be done in time $O(|V|^2)$, applying the same argument as before. The function *Recursive-Cover* is a polynomial-time algorithm, as discussed after the presentation of the algorithm. Therefore, all parts of *Approximate-p-Center* can be executed in polynomial time.

Proof (Second part: After phase I has finished, there is an R-cover for A of size $\hat{p}$.)

We prove the loop invariant

$$A \text{ has an } R\text{−cover of size at most } \hat{p} \tag{6}$$

by induction: Before the first iteration, (6) holds by the algorithm's precondition. So, assume (6) holds before the ith iteration. Because the vertex v picked by the algorithm has the property

$$\Gamma_R^{out}(v) \supseteq \Gamma_R^{in}(v), \tag{7}$$

and any vertex R-covering v has to be $\in \Gamma_R^{in}(v)$, we know that there has to be *at least one* center vertex c in the set $\Gamma_R^{out}(v)$. For this vertex c, we know that the vertex set R-covered by c is a subset of the set $\Gamma_{2R}^{out}(v)$, since v R-covers c, and the distance matrix obeys the triangle inequality.

Because A can be R-covered by $\hat{p}$ vertices, we know that $A \setminus \Gamma_{2R}^{out}(v)$ can be R-covered by $\hat{p}-1$ vertices. These vertices, however, do not have to be elements of the resulting set A. They can be arbitrary vertices from V.

We conclude that there is an R-cover for A of size $\hat{p}$ after phase I has finished.

Proof (Third part: After phase I, there are $\frac{\hat{p}}{2}$ vertices that, together with $\mathcal{C}_1$, $5R$-cover A')

By the previous part of the proof, we know that the set A has an R-cover $X = \{x_1, ..., x_p\}$ of size $\hat{p}$. For each $x_j \in X$, there are three possibilities:

1. $x_j \in (V \setminus A)$: Since the set $\mathcal{C}_1$ $2R$-covers $(V \setminus A)$, all vertices R-covered by x_j are $3R$-covered by $\mathcal{C}_1$.
2. $x_j \in A$ and $\Gamma_{2R}^{in}(x_j) \cap (V \setminus A) \neq \emptyset$: This means that there is a vertex $v \in (V \setminus A)$ that $2R$-covers x_j, which implies that x_j is $4R$-covered by $\mathcal{C}_1$ (triangle inequality). Therefore, all vertices vertices R-covered by x_j are $5R$-covered by $\mathcal{C}_1$.
3. $x_j \in A$ and $\Gamma_{2R}^{in}(x_j) \cap (V \setminus A) = \emptyset$: The vertices R-covered by x_j may or may not be $5R$-covered by $\mathcal{C}_1$.

In order to show that there are $\frac{\hat{p}}{2}$ vertices that, together with $\mathcal{C}_1$, $5R$-cover A', it is sufficient to show that there is a set of size $\frac{\hat{p}}{2}$ that $5R$-covers the nodes R-covered by centers of type 3.

Let $X_3 := \{x_1, ..., x_q\}$ be the set of centers of type 3. Now, look at the vertices in X_3. We know that

$$\Gamma_R^{out}(x) \not\supseteq \Gamma_R^{in}(x) \ \forall x \in X_1, \tag{8}$$

for otherwise x would have been collected by phase I of the algorithm. So, for every $x_j \in X_3$ there is at least one vertex y_j that R-covers x_j but is not R-covered by x_j. We know that $y_j \in A$ because otherwise x_j would not be of type 3 (by triangle inequality and definition of type 3). Thus, for each y_j there is at least one $x \in X$ that R-covers y_j (because X R-covers A). We now show how to construct a $5R$-cover from the sets X_3 and X:

```
// initially, all $x \in X$ are unmarked
for each $x_j \in X_3$ do
  if $x_j$ is not marked then
    find a $y_j$, as described above, and an $x_k \in X$ that $R$-covers $y_j$
    if ($x_k$ is marked as "level 0") then
      mark $x_j$ as "level 1"
    else if ($x_k$ is marked as "level 1") then
      mark $x_j$ as "level 2"
    else
      mark $x_j$ as "level 1"
      mark $x_k$ as "level 0"
return $\mathcal{L}_0 := \{x \in X \mid x$ is "level 0"$\}$
```

We see that every level-1 vertex is $2R$-covered by a level-0 vertex, and every level-2 vertex is $2R$-covered by a level-1 or a level-0 vertex. So, by using the triangle inequality, we know that at the end the set X_3 is $4R$-covered by the level-0 vertices. Thus, all vertices R-covered by X_3 are $5R$-covered by $\mathcal{L}_0$.

Since, by the construction rules, there is at least one level-1 vertex for every level-0 vertex (note that, if a vertex is marked as "level 1", this will never be changed again), and $|X| \leq \hat{p}$, we have $|\mathcal{L}_0| \leq \frac{\hat{p}}{2}$.

This result now allows us to apply *Recursive-Cover* to the set A' in phase II of the algorithm and obtain an $O(R \cdot \log^*(|V|))$-cover of size $\hat{p}$ for A'. Since $V \setminus A'$ is already $5R$-covered by $\mathcal{C}_1$, $|\mathcal{C}_1| = p - \hat{p}$, we can combine $\mathcal{C}_1$ and $\mathcal{C}_2$ and get a vertex set of size p that $O(R \cdot \log^*(|V|))$-covers V. □

Conclusion

The metric, asymmetric p-Center problem had remained unstudied even 10 years after its symmetric counterpart had been finally solved (by presenting an algorithm with optimal approximation factor) in 1986. In 1998, the $O(\log^*(n))$ approximation algorithm found by Panigrahy and Vishwanathan was published. Thus, it was clear that – in contrast to the general p-Center problem without any restrictions to the distance function – this problem could be approximated. And the approximation was a very good one, although the algorithm could not guarantee a constant-factor approximation.

A few years later, in 2003, it turned out that this is the best approximation ratio possible (unless $\mathbf{P} = \mathbf{NP}$) [HKK03]. So, the p-Center problem is one of the rare problems for which essentially nothing was known, then a first non-trivial algorithm was found that can approximate the problem, and it was already this very first algorithm that achieves the best approximation ratio possible. This alone is already very exciting.

What makes the p-Center problem even more fascinating is the approximability of the problem. $\log^*(n)$ is one of the functions where the discrepancy between theoretical results and practical consequences becomes very clear. $\log_2^*(n)$ can be assumed ≤ 6 for *all* practical purposes. Yet, from a theoretical point of view, there is a clear distinction between constant approximation factor and $\log^*(n)$. How do we deal with this situation?

References

[GJ79] M. Garey, D. Johnson. *Computers and intractability. A guide to the theory of NP-completeness.* Freeman, San Francisco, 1979.

[HKK03] E. Halperin, G. Kortsarz, R. Krauthgamer. *Tight lower bounds for the asymmetric k-center problem.* Electronic Colloquium on Computational Complexity 35, 2003.

[HS86] D. Hochbaum, D. Shmoys. *A unified approach to approximation algorithms for bottleneck problems.* Journal of the ACM, 33:533-550, July 1986.

[KH76] O. Kariv, S. Hakimi. *An algorithmic approach to network location problems – Part I: the p-centers.* Unpublished Manuscript, 1976.

[PV98] R. Panigrahy, S. Vishwanathan. *An $O(log^*(n))$ approximation algorithm for the asymmetric p-center problem.* Journal of Algorithms, 27:259-268, May 1998.

[Sh95] D. Shmoys. *Computing near-optimal solutions to combinatorial optimization problems.* Advances in Combinatorial Optimization, 355-397. American Mathematical Society, Providence, 1995.