Asymmetry in k-Center Variants

Inge Li Gørtz[1]⋆ and Anthony Wirth[2]⋆⋆

[1] Theory Department, The IT University of Copenhagen, Denmark.
inge@it-c.dk.

[2] Department of Computer Science, Princeton University.
awirth@cs.princeton.edu.

Abstract. This paper explores three concepts: the k-center problem, some of its variants, and asymmetry. The k-center problem is a fundamental clustering problem, similar to the k-median problem. Variants of k-center may more accurately model real-life problems than the original formulation. Asymmetry is a significant impediment to approximation in many graph problems, such as k-center, facility location, k-median and the TSP.

We demonstrate an $O(\log^* n)$-approximation algorithm for the asymmetric *weighted* k-center problem. Here, the vertices have weights and we are given a total budget for opening centers. In the *p-neighbor* variant each vertex must have p (unweighted) centers nearby: we give an $O(\log^* k)$-bicriteria algorithm using $2k$ centers, for small p.

Finally, the following three versions of the asymmetric k-center problem we show to be inapproximable: *priority* k-center, *k-supplier*, and *outliers with forbidden centers*.

1 Introduction

Imagine you have a delivery service. You want to place your delivery hubs at locations that minimize the maximum distance between customers and their nearest hubs. This is the *k-center problem*—a type of clustering problem that is similar to the facility location and k-median problems. The motivation for the *asymmetric* k-center problem, in our example, is that traffic patterns or one-way streets might cause the travel time from one point to another to differ depending on the direction of travel. Traditionally, the k-center problem was solved in the context of a metric; in this paper we retain the triangle inequality, but abandon the symmetry.

Symmetry is a vital concept in graph approximation algorithms. Very recently, the k-center problem was shown to be $\Omega(\log^* n)$ hard to approximate [6, 7], even though the symmetric version has a factor 2 approximation. Moreover, facility location and k-median both have constant factor algorithms in the

⋆ Part of this work was performed while visiting Princeton University.

⋆⋆ Supported by a Gordon Wu Fellowship, a DIMACS Summer Research Fellowship, and NSF ITR grant CCR-0205594.

S. Arora et al. (Eds.): APPROX 2003+RANDOM 2003, LNCS 2764, pp. 59–70, 2003.

symmetric case, but are provably $\Omega(\log n)$ hard to approximate without symmetry [1]. The traveling salesman problem is a little better, in that no $\Omega(\log n)$ hardness is known, but without symmetry no algorithm better than $O(\log n)$ has been found either.

Definition 1 (k-Center). *Given $G = (V, E)$, a complete graph with nonnegative (but possibly infinite) edge costs and a positive integer k, find a set S of k vertices, called* centers*, with minimum covering radius. The covering radius of a set S is the minimum distance R such that every vertex in V is within distance R of some vertex in S.*

Kariv and Hakimi [11] showed that the k-center problem is NP-hard. Without the triangle inequality the problem is NP-hard to approximate; we henceforth assume that the edge costs satisfy the triangle inequality.

The asymmetric k-center problem has proven to be much more difficult to understand than its symmetric counterpart. Hsu and Nemhauser [10] showed that the k-center problem cannot be approximated within a factor of $(2 - \epsilon)$ unless $P = NP$. In 1985 Hochbaum and Shmoys [8] provided a (best possible) factor 2 algorithm for the symmetric k-center problem. In 1996 Panigrahy and Vishwanathan [16, 13] gave the first approximation algorithm for the asymmetric problem, with factor $O(\log^* n)$. Archer [2] proposed two $O(\log^* k)$ algorithms based on many of the ideas in [13]. We now know [6, 7] that these algorithms are asympotitcally the best possible.

Variants of the k-Center Problem A number of variants of the k-center problem have been explored in the context of symmetric graphs. Perhaps some delivery hubs are more expensive to establish than others: instead of a restriction on the number of centers we can use, each vertex has a weight and we have a budget W, that limits the total weight of centers. Hochbaum and Shmoys [9] produced a factor 3 algorithm for this *weighted k-center problem*. This has recently been shown to be tight [6].

Hochbaum and Shmoys [9] also studied the *k-supplier problem* where the vertex set is segregated into suppliers and customers. Only supplier vertices can be centers and only the customer vertices need to be covered. Hochbaum and Shmoys gave a 3-approximation algorithm and showed that this is the best possible.

Khuller et al. [12] investigated the *p-neighbor k-center problem* where each vertex must have p centers nearby. This problem is motivated by need to account for facility failures: even if up to $p - 1$ facilities fail, every demand point has a functioning facility nearby. They gave a 3-approximation algorithm for all p, and a best possible 2-approximation algorithm when $p < 4$, noting that the case where p is small is "perhaps the practically interesting case".

Perhaps some demand points are more important than others. Plesnik [14] studied the *priority k-center problem*, in which the effective distance to a demand point is increased in proportion to its specified priority. Plesnik approximates the symmetric version within a factor of 2.

Charikar et al. [4] note that a disadvantage of the standard k-center formulation is that a few distant clients, *outliers*, can force centers to be located in isolated places. They suggest a variant of the problem, the k-center problem with *outliers and forbidden centers*, where a small subset of clients can be denied service, and some points are forbidden from being centers. Charikar et al. gave a (best possible) 3-approximation algorithm for the symmetric version of this problem.

Bhatia et al. [3] considered a network model, such as a city street network, in which the traversal time change as the day progresses. This is known as the *k-center problem with dynamic distances*: we wish to assign the centers such that the objective criteria are met at all times.

Results and Organization

Table 1 gives an overview of the best known results for the various k-center problems. In this paper we explore asymmetric variants that are not yet in the literature.

Table 1. An overview of the approximation results for k-center variants. †β is the maximum ratio of an edge's greatest length to its shortest length. ‡This is a bicriteria algorithm using $k(1 + 3/(\nu + 1))$ centers. §For $p < 4$. ¶This is a bicriteria algorithm using $2k$ centers, for $p \leq n/k$

Problem	Symmetric		Asymmetric	
k-center	2	[8]	$O(\log^* k)$	[2]
k-center with dynamic distances	$1 + \beta$ †	[3]	$O(\log^* n + \nu)$ ‡	[3]
weighted k-center	3	[9]	$\mathbf{O(\log^* n)}$	**Here**
p-neighbor k-center	3 (2 §)	[5]	$\mathbf{O(\log^* k)}$ ¶	**Here**
priority k-center	2	[14]	**Inapproximable**	**Here**
k-center with outliers and forbidden centers	3	[4]	**Inapproximable**	**Here**
k-suppliers	3	[9]	**Inapproximable**	**Here**

Section 2 contains the definitions and notation required to develop the results. In Section 3 we briefly review the algorithms of Panigrahy and Vishwanathan [13], and Archer [2]. The techniques used in the standard k-center problem are often applicable to the variants.

Our first result, in Section 4, is an $O(\log^* n)$-approximation for the asymmetric weighted k-center problem. In Section 5 we develop an $O(\log^* k)$ approximation for the asymmetric p-neighbor k-center problem, for $p \leq n/k$. As noted by Khuller et al. [12], the case where p is small is the most interesting case in practice. This a bicriteria algorithm, allowing an increase to $2k$ centers, but it can be turned into an $O(\log k)$-approximation algorithm using only k centers. Turning to hardness, we show that the asymmetric versions of the k-center problem with outliers (and forbidden centers), the priority k-center problem, and the k-supplier problem are NP-hard to approximate (Section 6).

2 Definitions

The input to the asymmetric k-center problem is a distance function d on *ordered* pairs of vertices—distances are allowed to be infinite—and a bound k on the number of centers.

Definition 2. *Vertex c* covers *vertex v within r, or c r*-covers *v, if $d_{cv} \leq r$. We extend this definition to a set C and a set A if for every $a \in A$ there is a $c \in C$ such that c covers a within r. Often we abbreviate "1-covers" to "covers".*

Most of the algorithms do not in fact operate on graphs with edge costs. Rather, they consider restricted graphs, in which only those edges with distance lower than some threshold are included, and the edges have unit cost. Hochbaum and Shmoys [9] refer to these as bottleneck graphs. Since the optimal value of the covering radius must be one of the n^2 distance values, many algorithms essentially run through a sequence of restricted graphs of every possible threshold radius in ascending order. This can be thought of as *guessing* the optimal radius R_{OPT}. The approach works because the algorithm either returns a solution, within the specified factor of the current threshold radius, or it fails, in which case R_{OPT} must be greater than the current radius.

Definition 3 (Restricted Graph G_r). *For $r > 0$, define the restricted graph G_r of the graph $G = (V, E)$ to be the graph $G_r = (V, E_r)$, where $E_r = \{(i, j) : d_{ij} \leq r\}$ and all edges have unit cost.*

Most of the following definitions apply to *restricted* graphs.

Definition 4 (Power of Graphs). *The t^{th} power of a graph $G = (V, E)$ is the graph $G^t = (V, E^t)$, $t > 1$, where E^t is the set of edges between distinct vertices that have a path of at most t edges between them in G.*

Definition 5. *For $i \in \mathbb{N}$ define $\Gamma_i^+(v) = \{u \in G \mid (v, u) \in E^i\} \cup \{v\}$, and $\Gamma_i^-(v) = \{u \in G \mid (u, v) \in E^i\} \cup \{v\}$, i.e. in the restricted graph there is a path of length at most i from v to u, respectively u to v.*

Notice that in a symmetric graph $\Gamma_i^+(v) = \Gamma_i^-(v)$. We extend this notation to sets so that $\Gamma_i^+(S) = \{u \in G \mid u \in \Gamma_i^+(v) \text{ for some } v \in S\}$, with $\Gamma_i^-(S)$ defined similarly. We use $\Gamma^+(v)$ and $\Gamma^-(v)$ instead of $\Gamma_1^+(v)$ and $\Gamma_1^-(v)$.

Definition 6. *For $i \in \mathbb{N}$ define $\Upsilon_i^+(v) = \Gamma_i^+(v) \setminus \Gamma_{i-1}^+(v)$, and $\Upsilon_i^-(v) = \Gamma_i^-(v) \setminus \Gamma_{i-1}^-(v)$, i.e., the nodes for which the path distance from v is exactly i, and the nodes for which the path distance to v is exactly i, respectively.*

For a set S, the extension follows the pattern $\Upsilon_i^+(S) = \Gamma_i^+(S) \setminus \Gamma_{i-1}^+(S)$. We use $\Upsilon^+(v)$ and $\Upsilon^-(v)$ instead of $\Upsilon_1^+(v)$ and $\Upsilon_1^-(v)$.

Definition 7 (Center Capturing Vertex (CCV)). *A vertex v is a* center capturing vertex *(CCV) if $\Gamma^-(v) \subseteq \Gamma^+(v)$, i.e., v covers every vertex that covers v.*

In the graph $G_{R_{\mathsf{OPT}}}$ the optimum center that covers v must lie in $\Gamma^-(v)$. For a CCV v, it lies in $\Gamma^+(v)$, hence the name. In symmetric graphs all vertices are CCVs and this property leads to the standard 2-approximation.

Definition 8 (Dominating Set). *Given a graph $G = (V, E)$, and a weight function $w : V \to \mathbb{Q}^+$ on the vertices, find a minimum weight subset $D \subseteq V$ such that every vertex $v \in V$ is covered by D, i.e., $v \in \Gamma^+(D)$ for all $v \in V$.*

Definition 9 (Set Cover). *Given a universe $\mathcal{U}$ of n elements, a collection $\mathcal{S} = \{S_1, \ldots, S_k\}$ of subsets of $\mathcal{U}$, and a weight function $w : \mathcal{S} \to \mathbb{Q}^+$, find a minimum weight sub-collection of $\mathcal{S}$ that includes all elements of $\mathcal{U}$.*

The *Max Coverage* problem, on an instance $\langle \mathcal{U}, \mathcal{S}, k \rangle$, is similar to the Set Cover problem: instead of trying to *minimize* the number of sets used we have a *bound* on the number of sets we can use, and the problem is then to maximize the number of elements covered. The Dominating Set, Set Cover, and Max Coverage problems are all NP-complete.

3 Asymmetric *k*-Center Review

The $O(\log^* n)$ algorithm of Panigrahy and Vishwanathan [13] has two phases, the *halve* phase, sometimes called the *reduce* phase, and the *augment* phase. As described above, the algorithm guesses R_{OPT}, and works in the restricted graph $G_{R_{\mathsf{OPT}}}$. In the halve phase we find a CCV v, include it in the set of centers, mark every vertex in $\Gamma_2^+(v)$ as covered, and repeat until no CCVs remain unmarked. The CCV property ensures that, as each CCV is found, the rest of the graph can be covered with one fewer center. Hence if k'' CCVs are obtained, the unmarked portion of the graph can be covered with $k' = k - k''$ centers. The authors then prove that this unmarked portion, CCV-free, can be covered with only $k'/2$ centers if we use radius 5 instead of 1. That is to say, $k'/2$ centers suffice in the graph $G^5_{R_{\mathsf{OPT}}}$.

The k-center problem in the restricted graph is identical to the dominating set problem. This is a special case of set cover in which the sets are the Γ^+ terms. In the augment phase, the algorithm recursively uses the greedy set cover procedure. Since the optimal cover uses at most $k'/2$ centers, the first cover has size at most $\frac{k'}{2} \log \frac{2n}{k'}$.

The centers in this first cover are themselves covered, using the greedy set cover procedure, then the centers in the second cover, and so forth. After $O(\log^* n)$ iterations the algorithm finds a set of at most k' vertices that, together with the CCVs, $O(\log^* n)$-covers the unmarked portion, since the optimal solution has $k'/2$ centers. Combining these with the k'' CCVs, we have k centers covering the whole graph.

Archer [2] presents two $O(\log^* k)$ algorithms, both building on the work in [13]. The algorithm more directly connected with the earlier work nevertheless has two fundamental differences. Firstly, in the reduce phase Archer shows that the CCV-free portion of the graph can be covered with $2k'/3$ centers and radius 3. Secondly, he constructs a set cover-like linear program and solves the relaxation to get a total of k' *fractional* centers that cover the unmarked vertices. From these fractional centers, he obtains a 2-cover of the unmarked vertices with $k' \log k'$ (integral) centers. These are the seed for the augment phase, which thus produces a solution with an $O(\log^* k')$ approximation to the optimum radius.

During the preparation of the final version of this manuscript, it was announced that the asymmetric k-center problem is hard to approximate better than $\Omega(\log^* n)$ [6, 7], closing the gap with the upper bound.

4 Asymmetric Weighted *k*-Center

Recall the application in which the costs of delivery hubs vary. In this situation, rather than having a restriction on the number of centers used, each vertex has a weight and we have a budget W that restricts the total weight of centers used.

Definition 10 (Weighted k-Center). *Given a weight function on the vertices, $w : V \to \mathbb{Q}^+$, and a bound $W \in \mathbb{Q}^+$, the problem is to find $S \subseteq V$ of total weight at most W, so that S covers V with minimum radius.*

Hochbaum and Shmoys [9] gave a 3-approximation algorithm for the symmetric weighted version, applying their approach for bottleneck problems. We propose an $O(\log^* n)$-approximation for the asymmetric version, based on Panigrahy and Vishwanathan's technique for the unweighted problem. Note that in light of the hardness result just announced [6, 7], this algorithm is asymptotically optimal. Another variant has both the k and the W restrictions, but we will not expand on that problem here.

First a brief sketch of the algorithm, which works with restricted graphs. In the reduce phase, having found a CCV, v, we pick the lightest vertex u in $\Gamma^-(v)$ (which might be v itself) as a center in our solution. Then mark everything in $\Gamma_3^+(u)$ as covered, and continue looking for CCVs. We can show that there exists a 7-cover of the unmarked vertices with total weight less than half optimum. Finally we recursively apply a greedy procedure for weighted elements $O(\log^* n)$ times, similar to the one used for Set Cover. The total weight of centers in our solution set is at most W.

The following lemma about digraphs is the key to our reduce phase and is analagous to Lemma 4 in [13] and Lemma 16 in [2].

Lemma 1 (Cover of Half the Graph's Weight). *Let $G = (V, E)$ be a digraph with weighted vertices, but unit edge costs. Then there is a subset $S \subseteq V$, $w(S) \le w(V)/2$, such that every vertex with positive indegree is reachable in at most 3 steps from some vertex in S.*

Proof. To construct the set S repeat the following, to the extent possible: *Select* a vertex with positive outdegree, but if possible *select* one with indegree zero. Let v be the selected vertex and compare sets $\{v\}$ and $\Gamma^+(v) \setminus \{v\}$: add the set of smaller weight to S and remove $\Gamma^+(v)$ from G.

It is clear that the weight of S is no more than half the weight of V. We must now show that S 3-covers all non-orphan vertices—we call x a parent of y if $x \in \Gamma^-(y)$.

The children of v are clearly 1-covered. Assume v is not in S (trivial otherwise): if v was an orphan initially then ignore it. If v is an orphan when selected, then some parent must have been removed by the selection of a grandparent, so it is 2-covered.

So v has at least one parent when it is selected, implying there are no orphan vertices at that time. Therefore the sets of parents of v, S_1, grandparents of v, S_2, and great-grandparents of v, S_3, are not empty. Although these sets might not be pairwise disjoint, if they contained any of v's children, then v would be 3-covered.

After v is removed, there are three possibilities for S_2: (i) Some vertex in S_3 is selected, removing part of S_2; (ii) Some vertex in S_2 is selected and removed; (iii) Some vertex in S_1 is selected, possibly making some S_2 vertices childless. One of these events *must* happen, since S_1 and S_2 are non-empty. As a consequence, v is 3-covered. □

Henceforth call the vertices that have not yet been covered/marked *active*. Using Lemma 1 we can show that after removing the CCVs from the graph, we can cover the active set with half the weight of an optimum cover if we are allowed to use distance 7 instead of 1.

Lemma 2 (Cover of Half Optimal Weight). *Consider a subset $A \subseteq V$ that has a cover consisting of vertices of total weight W, but no CCVs. Assume there exists a set C_1 that 3-covers exactly $V \setminus A$. Then there exists a set of vertices S of total weight $W/2$ that, together with C_1, 7-cover A.*

Proof. Let U be the subset of optimal centers that cover A. We call $u \in U$ a *near* center if it can be reached in 4 steps from C_1, and a *far* center otherwise. Since C_1 5-covers all of the nodes covered by near centers, it suffices to choose S to 6-cover the far centers, so that S will 7-cover all the nodes they cover.

Define an auxiliary graph H on the (optimal) centers U as follows. There is an edge from x to y in H if and only if x 2-covers y in G (and $x \neq y$). The idea is to show that any far center has positive indegree in H. As a result, Lemma 1 shows there exists a set $S \in U$ with $|S| \leq W/2$ such that S 3-covers the far centers in H, and thus 6-covers them in G.

Let x be any far center. Since A contains no CCVs, there exists y such that y covers x, but x does not cover y. Since $x \notin \Gamma_4^+(C_1)$, $y \notin \Gamma_3^+(C_1)$, and thus $y \in A$ (since everything not 3-covered by C_1 is in A). Thus there exists a center $z \in U$, which is not x, but might be y, that covers y and therefore 2-covers x. Hence x has positive indegree in the graph H. □

As we foreshadowed, we will use the greedy heuristic to complete the algorithm. We now analyze the performance of this heuristic in the context of the dominating set problem in node-weighted graphs. All vertices V are available as potential members of the dominating set (i.e. centers), but we need only dominate the active vertices A. The heuristic is to select the most *efficient* vertex: the one that maximizes $w(A(v))/w(v)$, where $A(v) \equiv A \cap \Gamma^+(v)$.

Lemma 3 (Greedy Algorithm in Weighted Dominating Set). *Let $G = (V, E)$, $w : V \to \mathbb{Q}^+$ be an instance of the dominating set problem in which a set A is to be dominated. Also, let w^* be the weight of an optimum solution for this instance. The greedy algorithm gives an approximation guarantee of*

$$2 + \ln \frac{w(A)}{w^*} = O\left(\log \frac{w(A)}{w^*}\right) .$$

Proof. In every application of the greedy selection there must be some vertex $v \in V$ for which

$$\frac{w(A(v))}{w(v)} \geq \frac{w(A)}{w^*} \quad \Rightarrow \quad \frac{w(A(v))}{w(A)} \geq \frac{w(v)}{w^*} \tag{1}$$

otherwise no optimum solution of weight w^* would exist. This is certainly true of the most efficient vertex v, so make it a center and mark all that it covers, leaving A' uncovered. Now,

$$w(A') = w(A) - w(A(v)) \leq w(A)\left(1 - \frac{w(v)}{w^*}\right) \leq w(A)\exp\left(-\frac{w(v)}{w^*}\right)$$

After j steps, the remaining active vertices, A^j, satisfy

$$w(A^j) \leq w(A^0) \prod_{i=1}^{j} \exp\left(-\frac{w(v_i)}{w^*}\right) , \tag{2}$$

where v_i is the ith center picked (greedily) and A^0 is the original active set.

Assume that after some number of steps, say j, there are still some active elements, but the upper bound in (2) drops below w^*. That is to say,

$$\sum_{i=1}^{j} w(v_i) \geq w^* \ln(w(A^0)/w^*) .$$

Before we picked the vertex v_j we had

$$\sum_{i=1}^{j-1} w(v_i) \leq w^* \ln(w(A^0)/w^*) , \quad \text{and so,} \quad \sum_{i=1}^{j} w(v_i) \leq w^* + w^* \ln(w(A^0)/w^*) ,$$

because (1) tells us that $w(v_i)$ is no greater than w^*. To cover the remainder, A^j, we just use A^j itself, at a cost of at most w^*. Hence the total weight of the solution is at most $w^*(2 + \ln(w(A^0)/w^*))$.

On the other hand, if the upper bound on $w(A^j)$ never drops below w^* before A^j becomes empty, then we have a solution of weight at most $w^* \ln(w(A^0)/w^*)$. □

We now show that this tradeoff between covering radius and optimal cover size leads to an $O(\log^* n)$ approximation.

Lemma 4 (Recursive Set Cover). *Given $A \subseteq V$, such that A has a cover of weight W, and a set $C_1 \in V$ that covers $V \setminus A$, we can find in polynomial time a set of vertices of total weight at most $2W$ that, together with C_1, cover A (and hence V) within a radius of $O(\log^* n)$.*

Proof. Our first attempt at a solution, S_0, is all vertices of weight no more than W: only these vertices could be in the optimum center set. Their total weight is at most nW. Since C_1 covers $S_0 \setminus A$, consider $A_0 = S_0 \cap A$, which has a cover of size W. Lemma 3 shows that the greedy algorithm results in a set S_1 that covers A_0, and has weight

$$w(S_1) \leq O\left(W \log \frac{Wn}{W}\right) = O(W \log n) \ .$$

Set C_1 covers $S_1 \setminus A$, so we need only consider $A_1 = S_1 \cap A$, and so forth. At the ith iteration we have: $w(S_i) \leq O(W \log(w(S_{i-1})/W))$ and hence by induction at most $O(W \log^{(i)} n)$. Thus after $\log^* n$ iterations the weight of our solution set falls to $2W$. □

All the algorithmic tools can be assembled to form an approximation algorithm.

Theorem 1 (Approximation of Weighted k-Center). *We can approximate the weighted k-center problem within factor $O(\log^* n)$ in polynomial time.*

Proof. Guess the optimum radius, R_{OPT}, and work in the restricted graph $G_{R_{\mathsf{OPT}}}$. Initially, the active set A is V. Repeat the following as many times as possible: Pick CCV v in A, add the lightest vertex u in $\Gamma^-(v)$ to our solution set of centers and, remove the set $\Gamma_3^+(u)$ from A. Since v is covered by an optimum center in $\Gamma^+(v)$, u is no heavier than this optimum center, and $\Gamma_3^+(u)$ includes everything covered by the optimum center.

Let C_1 be the centers chosen in this first phase. We know the remainder of the graph, A, has a cover of total weight $W' = W - w(C_1)$, because of our choices based on CCV and weight.

Lemma 2 shows that we can cover the remaining uncovered vertices with weight no more than $W'/2$ if we use distance 7. So let the active set A be $V \setminus \Gamma_7^+(C_1)$, and recursively apply the greedy algorithm as described in the proof of Lemma 4 on the graph $G^7_{R_{\mathsf{OPT}}}$. As a result, we have a set of size W' that covers A within radius $O(\log^* n)$. □

5 Asymmetric p-Neighbor k-Center

Imagine that we wish to locate k facilities at such that the maximum distance of a demand point from its p^{th}-closest facility is minimized. As a consequence, failures in $p - 1$ facilities do not bring down the network.

Definition 11 (Asymmetric p-Neighbor k-Center Problem). *Let $d_p(S, v)$ denote the distance from the p^{th} closest vertex in S to v. The problem is to find a subset S of at most k vertices that minimizes*

$$\max_{v \in V \setminus S} d_p(S, v) .$$

We show that we can approximate the asymmetric p-neighbor k-center problem within a factor of $O(\log^* k)$ if we allow ourselves to use $2k$ centers. Our algorithm is restricted to the case $p \leq n/k$, but this is reasonable as p should not be too large [12].

We use the same techniques as usual, including restricted graphs, but in the augment phase we use the greedy algorithm for the *Constrained Set Multicover* problem [15]. That is, each element, e, needs to be covered r_e times, but each set can be picked at most once. The p-neighbor k-center problem has $r_e = p$ for all e. We say that an element e is *alive* if it occurs in fewer than p sets chosen so far. The greedy heuristic is to pick the set that covers the most live elements. It can be shown that this algorithm achieves an approximation factor of $H_n = O(\log n)$ [15]. However the following result is more appropriate to our application.

Lemma 5 (Greedy Constrained Set Multicover). *Let k be the optimum solution to the Constrained Set Multicover problem. The greedy algorithm gives approximation guarantee $O(\log(np/k))$.*

Proof. The same kind of averaging argument used for standard Set Cover shows that the greedy choice of a set reduces the total number of unmarked element copies by a factor $1 - 1/k$. So after i steps the number of copies of elements yet to be covered is $np(1 - 1/k)^i \leq np(e^{-1/k})^i$. Hence after $k \ln(np/k)$ steps the number of uncovered copies of elements is at most k. A naive cover of these last k element copies leads to the total number of sets being $k + k \ln(np/k)$. □

If $p \leq n/k$ this greedy algorithm gives an approximation factor of $O(\log(n/k))$. Applying the standard recursive approach in [13], which works in the p-neighbor case, we can achieve an $O(\log n)$ approximation with k centers, or $O(\log^* n)$ with $2k$ centers. We can lower the approximation guarantee to $O(\log^* k)$, with $2k$ centers, using Archer's LP-based priming. First solve the LP for the constrained set multicover problem. In the solution each vertex is covered by an amount p of fractional centers, out of a total of k. We can now use the greedy set cover algorithm to get an initial set of $k^2 \ln k$ centers that 2-covers every vertex in the active set with at least p centers. Repeatedly applying the greedy procedure for constrained set multicover, this time for $(\log^* k + 1)$ iterations, we get $2k$ centers that cover all active vertices within $O(\log^* k)$. Alternatively, we could carry out $O(\log k)$ iterations and stick to just k centers.

6 Inapproximability Results

In this section we give inapproximability results for the asymmetric versions of the k-center problem with outliers, the priority k-center problem, and the

k-supplier problem. These problems all admit constant factor approximation algorithms in the symmetric case.

Asymmetric k-Center with Outliers

Definition 12 (k-Center with Outliers and Forbidden Centers). *Find a set $S \subseteq C$, where C is the set of vertices allowed to be centers, such that $|S| \leq k$ and S covers at least p nodes, with minimum radius.*

Theorem 2. *For any polynomial time computable function $\alpha(n)$, the asymmetric k-center problem with outliers (and forbidden centers) cannot be approximated within a factor of $\alpha(n)$, unless $P = NP$.*

Proof. We reduce instance $\langle U, \mathcal{S}, k\rangle$ of Max Coverage to our problem. Construct vertex sets A and B so that for each set $S \in \mathcal{S}$ there is $v_S \in A$, and for each element $e \in U$ there is $v_e \in B$. From every vertex $v_S \in A$, create an edge of unit length to vertex $v_e \in B$ if $e \in S$.

Let $p = |B| + k$, so that if we find k centers that cover p vertices within any finite distance, we *must* have found k vertices in A that cover all $|B|$ vertices. Hence we have solved the instance of Max Coverage which is an NP-complete problem. □

Note that the proof never relied on the fact that the B vertices were forbidden from being centers (setting p to $|B| + k$ ensured this).

Asymmetric Priority k-Center

Definition 13 (Priority k-Center). *Given a priority function $p : V \to \mathbb{Q}^+$ on the vertices, find $S \subseteq V$, $|S| \leq k$, that minimizes R so that for every $v \in V$ there exists a center $c \in S$ for which $p_v d_{cv} \leq R$.*

Theorem 3. *For any polynomial time computable function $\alpha(n)$, the asymmetric k-center problem with priorities cannot be approximated within a factor of $\alpha(n)$, unless $P = NP$.*

Proof. The construction of the sets A and B is the similar to the proof of Theorem 2, except that we reduce from Set Cover. This time make the set A a complete digraph, with edges of length ℓ, as well as the unit length set-element edges from A to B. Give the nodes in set A priority 1 and the nodes in set B priority ℓ. An optimal solution to the priority k-center problem is k centers in A and a radius of ℓ, which covers every vertex. This implies that the k centers cover (in the Set Cover sense) all the elements in B. If $k' < k$ centers were chosen from A and $k - k'$ centers were chosen from B instead, we could trivially convert this to a solution choosing k centers from A.

Any non-optimal solution requires a radius of at least $\ell^2 + \ell$, as this would involve covering some B vertex by stepping from an A center through another A vertex. Therefore any algorithm with approximation guarantee $\ell + 1 - \varepsilon$ or better would solve Set Cover. We can make ℓ any function we like and the result follows. □

Asymmetric k-Supplier

Definition 14 (k-Supplier). *Given a set of suppliers Σ and a set of customers C, find a subset $S \subseteq \Sigma$ that minimizes R such that S covers C within R.*

Theorem 4. *For any polynomial time computable function $\alpha(n)$, the asymmetric k-supplier problem cannot be approximated within a factor of $\alpha(n)$, unless $P = NP$.*

Proof. By a reduction from the Max Coverage problem similar to the proof of Theorem 2. □

Acknowledgements

The authors would like to thank Moses Charikar and the reviewers.

References

[1] A. Archer. Inapproximability of the asymmetric facility location and k-median problems. Unpublished manuscript available at `www.orie.cornell.edu/~aarcher/Research`, 2000.
[2] A. Archer. Two $O(\log^* k)$-approximation algorithms for the asymmetric k-center problem. In K. Aardal and B. Gerads, editors, *IPCO*, volume 2081 of *Lecture Notes in Computer Science*, pages 1–14. Springer-Verlag, 2001.
[3] R. Bhatia, S. Guha, S. Khuller, and Y. Sussmann. Facility location with dynamic distance function. In *Scand. Workshop on Alg. Th. (SWAT)*, pages 23–34, 1998.
[4] M. Charikar, S. Khuller, D. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proc. 12th SODA*, pages 642–51, 2001.
[5] S. Chaudhuri, N. Garg, and R. Ravi. The p-neighbor k-center problem. *Info. Proc. Lett.*, 65:131–4, 1998.
[6] J. Chuzhoy, S. Guha, S. Khanna, and S. Naor. Asymmetric k-center is $\log^* n$-hard to approximate. Technical Report 03-038, Elec. Coll. Comp. Complexity, 2003.
[7] E. Halperin, G. Kortsarz, and R. Krauthgamer. Tight lower bounds for the asymmetric k-center problem. Technical Report 03-035, Elec. Coll. Comp. Complexity, 2003.
[8] D. Hochbaum and D. Shmoys. A best possible approximation algorithm for the k-center problem. *Math. Oper. Res.*, 10:180–4, 1985.
[9] D. Hochbaum and D. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *JACM*, 33:533–50, 1986.
[10] W. Hsu and G. Nemhauser. Easy and hard bottelneck location problems. *Disc. Appl. Math.*, 1:209–16, 1979.
[11] O. Kariv and S. Hakimi. An algorithmic approach to network location problems. I. The p-centers. *SIAM J. Appl. Math.*, 37:513–38, 1979.
[12] S. Khuller, R. Pless, and Y. Sussmann. Fault tolerant k-center problems. *Theor. Comp. Sci. (TCS)*, 242:237–45, 2000.
[13] R. Panigrahy and S. Vishwanathan. An $O(\log^* n)$ approximation algorithm for the asymmetric p-center problem. *J. Algorithms*, 27:259–68, 1998.
[14] J. Plesnik. A heuristic for the p-center problem in graphs. *Disc. Appl. Math.*, 17:263–268, 1987.
[15] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.
[16] S. Vishwanathan. An $O(\log^* n)$ approximation algorithm for the asymmetric p-center problem. In *Proc. 7th SODA*, pages 1–5, 1996.