Asymmetry in k-Center Variants ⋆

Inge Li Gørtz ∗,[1]

Department of Theoretical Computer Science, The IT University of Copenhagen, Denmark.

Anthony Wirth [2]

Department of Computer Science, Princeton University.

Abstract

This paper explores three concepts: the k-center problem, some of its variants, and asymmetry. The k-center problem is fundamental in location theory. 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 give 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, we show the following three versions of the asymmetric k-center problem to be inapproximable: *priority* k-center, *k-supplier*, and *outliers with forbidden centers*.

Key words: k-center, asymmetric, approximation algorithm.

⋆ A preliminary version of this paper appeared in the proceedings of APPROX 2003: 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, LNCS 2764, Springer, 2003, pp 59–70.

∗ Corresponding author. IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark, +45 72 18 52 77, fax +45 72 18 50 01.

Email addresses: `inge@itu.dk` (Inge Li Gørtz), `awirth@unimelb.edu.au` (Anthony Wirth).

[1] Part of this work was performed while visiting Princeton University.

[2] Supported by a Gordon Wu Fellowship, a DIMACS Summer Research Fellowship, and NSF ITR grant CCR-0205594. Present address: Depart-

1 Introduction

Imagine you have a delivery service. You want to place your k 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 [1] and k-median [2] 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. Recently, the asymmetric k-center problem was shown to be $\Omega(\log^* n)$ hard to approximate [3–5], even though the symmetric version has a factor 2 approximation. Facility location and k-median both have constant factor algorithms in the symmetric case, but are provably $\Omega(\log n)$ hard to approximate without symmetry [6]. The traveling salesman problem is a little better, in that no super-constant hardness is known, but without symmetry no algorithm better than $O(\log n)$ [7] 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 [8] showed that the k-center problem is NP-hard. Without the triangle inequality the problem is NP-hard to approximate within any factor (there is a straightforward reduction from the dominating set problem). We henceforth assume that the edge costs satisfy the triangle inequality. Hsu and Nemhauser [9], using the same reduction, showed that the metric k-center problem cannot be approximated within a factor of $(2 - \epsilon)$ unless P = NP. In 1985 Hochbaum and Shmoys [10] provided a (best possible) factor 2 algorithm for the metric k-center problem. In 1996 Panigrahy and Vishwanathan [11,12] gave the first approximation algorithm for the asymmetric problem, with factor $O(\log^* n)$. Archer [13] proposed two $O(\log^* k)$ algorithms based on many of the ideas in [12]. The complementary $\Omega(\log^* n)$ hardness result [3–5] shows that these approximation algorithms are asymptotically optimal.

ment of Computer Science and Software Engineering, The University of Melbourne, Australia.

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 [14] produced a factor 3 algorithm for this *weighted k-center* problem, which has recently been shown to be tight [3,5].

Hochbaum and Shmoys [14] 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 it is the best possible.

Khuller et al. [15] investigated the *p-neighbor k-center* problem where each vertex must have p centers nearby. This problem is motivated by the 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".

Maybe some demand points are more important than others. Plesnik [16] studied the *priority k-center* problem, in which the effective distance to a demand point is enlarged in proportion to its specified priority. Plesnik approximates the symmetric version within a factor of 2.

Charikar et al. [17] 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 may 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. [18] considered a network model, such as a city street network, in which the traversal times 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.

Problem	Symmetric		Asymmetric	
k-center	2	[10]	$O(\log^* k)$	[13]
k-center with dynamic distances	$1+\beta$ †	[18]	$O(\log^* n + \nu)$ ‡	[18]
weighted k-center	3	[14]	$\mathbf{O(\log^* n)}$	
p-neighbor k-center	3 (2 §)	[19]	$\mathbf{O(\log^* k)}$ ¶	
priority k-center	2	[16]	**Inapproximable**	
k-center with outliers and forbidden centers	3	[17]	**Inapproximable**	
k-suppliers	3	[14]	**Inapproximable**	

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

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 were not yet in the literature.

Section 2 contains the definitions and notation required to develop the results. In Section 3 we briefly review the algorithms of Panigrahy and Vishwanathan [12], and Archer [13]. 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. [15], the case where p is small is the most interesting case in practice. This a bicriteria algorithm, allowing $2k$ centers to be used rather than just k. It can, however, be converted to 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 within any factor (Section 6).

2 Definitions

To avoid any uncertainty, we note that log stands for $\log_2$ by default, while ln stands for $\log_e$.

Definition 2 *For every integer $i > 1$, $\log^i x = \log(\log^{i-1} x)$, and $\log^1 x = \log x$. We let $\log^* x$ represent the smallest integer i such that $\log^i x \le 2$.*

The input to the asymmetric k-center problem is a distance function d on every *ordered* pair of vertices—distances are allowed to be infinite—and a bound k on the number of centers. Note that we assume that the edges are *directed*.

Definition 3 *Vertex c* covers *vertex v* within *r, or c r*-covers *v, if $d_{cv} \le r$. We extend the definition to sets so that a set C r-covers a set A if for every $a \in A$ there is some $c \in C$ such that c covers a within r. Often we abbreviate "1-covers" to "covers".*

Many of the algorithms for k-center and its variants do not, in fact, operate on graphs with edge costs. Rather, they consider bottleneck graphs [14], in which only those edges with distance lower than some threshold are included, and they appear in the bottleneck graph with unit cost. Since the optimal value of the covering radius must be one of the $n(n-1)$ distance values, many algorithms essentially run through a sequence of bottleneck 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 4 (Bottleneck Graph G_r) *For $r > 0$, define the* bottleneck graph *G_r of the graph $G = (V, E)$ to be the graph $G_r = (V, E_r)$, where $E_r = \{(i, j) : d_{ij} \le r\}$ and all edges have unit cost.*

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

Definition 5 (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 ordered pairs of distinct vertices that have a path of at most t edges between them in G.*

Definition 6 *For $i \in \mathbb{N}$ define*

$$\Gamma_i^+(v) = \{u \in V \mid (v, u) \in E^i\} \cup \{v\}, \qquad \Gamma_i^-(v) = \{u \in V \mid (u, v) \in E^i\} \cup \{v\} ,$$

i.e., in the bottleneck graph there is a path of length at most i from v to u, respectively from u to v.

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

Definition 7 *For $i \in \mathbb{N}$ define*

$$\Upsilon_i^+(v) = \Gamma_i^+(v) \setminus \Gamma_{i-1}^+(v), \qquad \Upsilon_i^-(v) = \Gamma_i^-(v) \setminus \Gamma_{i-1}^-(v) \ ,$$

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 $\Upsilon_1^-(v)$.

We call x a *parent* of y, and y a *child* of x, if $x \in \Upsilon^-(y)$. If $\Upsilon^-(y)$ is empty we call y an *orphan*.

Definition 8 (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.

The following three fundamental problems, related to k-center, are all NP-complete [20].

Definition 9 (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., $\Gamma^+(D) = V$.*

Definition 10 (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 $\mathcal{U}$.*

Definition 11 (Max Coverage) *Given $\langle \mathcal{U}, \mathcal{S}, k \rangle$, with $\mathcal{U}$ and $\mathcal{S}$ as above, find a sub-collection of k sets that includes the maximum number of elements of $\mathcal{U}$.*

3 Asymmetric k-Center Review

The $O(\log^* n)$ algorithm of Panigrahy and Vishwanathan [12] 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 bottleneck 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 and vertices are marked, the unmarked portion 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 bottleneck 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 within $O(\log^* n)$.

Archer [13] presents two $O(\log^* k)$ algorithms, both building on the work in [12]. 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 within radius 3. Secondly, he constructs a set cover-like integer 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. We now know that all of these approximation algorithms are asymptotically optimal [3–5].

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

Hochbaum and Shmoys [14] 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 complementary hardness result just announced [3–5], this algorithm is asymptotically the best possible. There is another variant that 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 bottleneck 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. We then mark everything in $\Gamma_3^+(u)$ as covered, and continue looking for CCVs. We can show that there exists a 49-cover of the unmarked vertices with total weight less than a quarter of the optimum. Finally, we recursively apply a greedy procedure for weighted sets and 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 concerning vertex-weighted digraphs is the key to our reduce phase and is analogous to Lemma 4 in [12] and Lemma 16 in [13].

Lemma 13 (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 $S \subseteq V$, $w(S) \leq w(V)/2$, 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 v with positive outdegree and if possible *select* one with indegree zero (that is, $\Upsilon^-(v)$ is empty). Compare sets $\{v\}$ and $\Upsilon^+(v)$: add the lighter set 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.

The children of a selected vertex v, $\Upsilon^+(v)$, 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, but not initially, then at some previous stage in the procedure some parent of v must have been removed by the selection of a grandparent (a vertex in $\Upsilon_2^-(v)$), so v is 2-covered. Note that if one of v's parents had been selected then v would already have been removed from G.

Now assume v has at least one parent when it is selected. Consequently, at that state in the procedure, there are no vertices that have children, but are orphans, otherwise on of them would have been selected instead of v. We conclude that the sets of parents of v, $S_1 = \Upsilon^-(v)$, parents of S_1, $S_2 = \Upsilon^-(S_1)$, and parents of S_2, $S_3 = \Upsilon^-(S_2)$, 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. $\square$

Henceforth call the vertices that have not yet been covered/marked *active*. Using Lemma 13 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 14 (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-covers A.*

PROOF. Let U be a subset of the optimal centers that covers 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 13 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: note that $x \in A$. 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. $\square$

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 15 (Greedy Algorithm in Weighted Dominating Set) *Let*

$$\left\langle G = (V, E), w : V \to \mathbb{Q}^+, A \subseteq V \right\rangle$$

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 $2 + \ln(w(A)/w^*)$.

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

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

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

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

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

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

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) has just dropped below w^*. That is to say,

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

Before we picked the vertex v_j we had

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

for (1) tells us that $w(v_j)$ is no greater than w^*. To cover the remainder, A^j, we just use A^j itself, at a cost less than 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 to the instance of weight at most $w^* \ln(w(A^0)/w^*)$. $\square$

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

Lemma 16 (Recursive Set Cover) *Given $A \subseteq V$, such that A has a cover of weight W, and a set $C_1 \subseteq V$ that covers $V \setminus A$, we can find in polynomial time a set of vertices of total weight at most $4W$ that, together with C_1, covers A (and hence V) within a radius of $O(\log^* n)$.*

PROOF. We will be applying the greedy algorithm of Lemma 15. The approximation guarantee is $2 + \ln(w(A)/W)$, which is less than $\log_{1.5}(w(A)/W)$ when $w(A) \geq 4W$.

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 and 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 15 shows that the greedy algorithm results in a set S_1 that covers A_0 and has weight

$$w(S_1) \leq W \log_{1.5}(\frac{Wn}{W}) = W \log_{1.5} n \ ,$$

assuming $n \geq 4$. The set C_1 covers $S_1 \setminus A$, so we need only consider $A_1 = S_1 \cap A$. We continue this procedure and note that at the ith iteration we have $w(S_i) \leq W \log_{1.5}(w(S_{i-1})/W)$. By induction, after $O(\log^* n)$ iterations the weight of our solution set, S_i, is at most $4W$. $\square$

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

Theorem 17 (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 bottleneck graph $G_{R_{\mathsf{OPT}}}$. Initially, the active set A is V. Repeat the following as many times as possible: Pick a 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. Moreover, since the optimum center lies in $\Gamma^+(v)$, $\Gamma_3^+(u)$ includes everything covered by it.

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 14 shows that we can cover the remaining uncovered vertices with weight no more than $W'/2$ if we use covering radius 7. Applying the lemma again, we can cover the remaining vertices with weight $W'/4$ centers if we allow radius 49. So let the active set A be $V \setminus \Gamma_{49}^{+}(C_1)$, and recursively apply the greedy algorithm as described in the proof of Lemma 16 on the graph $G_{R_{\mathsf{OPT}}}^{49}$. As a result, we have a set of size W' that covers A within radius $O(\log^* n)$. $\square$

5 Asymmetric p-Neighbor k-Center

Imagine that we wish to place k facilities so 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 cause severe network performance loss.

Definition 18 (Asymmetric p-Neighbor k-Center Problem) *For every subset S and vertex v in V, 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 [15].

We use the same techniques as before, including bottleneck graphs, but in the augment phase we use the greedy algorithm for the *constrained set multicover* problem [21]. 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 *active* if it occurs in fewer than p sets chosen so far. The greedy heuristic is to pick the set that covers the most active elements. It can be shown that this algorithm achieves an approximation factor of $H_n = O(\log n)$ [21, Section 13.2]. However the following result is more appropriate to our application.

Lemma 19 (Greedy Constrained Set Multicover) *Let k be the value of the optimum solution to the Constrained Set Multicover problem. The greedy algorithm gives approximation guarantee of $\log_{1.5}(np/k)$.*

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 $np(1-1/k)^i < np(e^{-1/k})^i$. Hence after $k \ln(np/k)$ steps the number of uncovered copies of elements is less than k. A naive cover of these last k element copies leads the total number of sets in the solution to be at most $k + k\ln(np/k)$. Since $p \geq 2$, this greedy algorithm has an approximation factor less than $\log_{1.5}(np/k)$. □

If $p \leq n/k$ the approximation guarantee above is less than $\log_{1.2}(n/k)$. We can now apply the standard recursive approach from [12]. Recall that Panigrahy and Vishwanathan use $O(\log^* n)$ iterations to get down to $2k$ centers, which gives them a $O(\log^* n)$ approximation because of the halve phase. They also state that using $O(\log n)$ iterations instead they would get down to k centers without the halve phase. Since we do not have anything similar to the halve phase, for the p-neighbor k-center problem we need $O(\log n)$ iterations to get down to k centers. There is no analogy to Lemma 4 [12], in which Panigrahy and Vishwanathan show that all vertices with positive indegree can be 2-covered by half the number of centers.

We can lower the approximation guarantee to $O(\log^* k)$, with $2k$ centers, using Archer's LP-based priming, which we describe now in detail.

We first solve the LP for the constrained set multicover problem. Let y_v be the (fractional) extent to which a vertex is a center:

$$\begin{array}{lrll} \text{minimize} & \displaystyle\sum_{v \in V} y_v & & \\ \text{subject to} & \displaystyle\sum_{u \in \Gamma^-(v)} y_u \geq p, & v \in A \\ & -y_v \geq -1, & v \in V \\ & y_v \geq 0, & v \in V\,. \end{array}$$

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 method to obtain an initial set of $k^2 \ln k$ centers that 2-covers every vertex in the active set with at least p centers.

Let A be the active vertices (the vertices that are covered fewer than p times) and let $A(v) = \Gamma^+(v) \cap A$. Let $y'(v) = y_v \cdot a_v$, where a_v is the number of times v still needs to be covered, and let $y'(S) = \sum_{v \in S} y'(v)$ for all $S \subseteq V$. Note that $v \in A \Leftrightarrow a_v > 0$ and thus $y'(A) = y'(V)$. The function y' indicates the extent to which an optimal fractional center is not yet covered. We will see that when the value of $y'(V)$ is low, we can be sure that we have found a reasonable cover of all the vertices.

Start with an empty set S and repeat the following until $y'(V) < p$: Choose the vertex v from $T = V - S$ maximizing $y'(\Gamma^+(v))$, add it to S, and set $a_u = a_u - 1$ for all vertices $u \in A(v)$.

Lemma 20 *Once $y'(V) < p$, the set S 2-covers every vertex with at least p centers.*

PROOF. For every v, let $\alpha(v)$ be its active parents, $\alpha(v) = \{u : u \in \Gamma^-(v), a_u \geq 1\}$, and let $\beta(v)$ be its inactive parents, $\beta(v) = \{u : u \in \Gamma^-(v), a_u = 0\}$.

Since $y'(V) < p$ we have

$$\sum_{u \in \alpha(v)} y_u \leq \sum_{u \in \alpha(v)} y'_u < p \ .$$

By the first LP constraint we have

$$\sum_{u \in \alpha(v)} y_u + \sum_{u \in \beta(v)} y_u = \sum_{u \in \Gamma^-(v)} y_u \geq p \ ,$$

and thus $\sum_{u \in \beta(v)} y_u > 0$. We conclude that there must be at least one vertex in $\beta(v)$. The p vertices covering this vertex 2-cover v. $\square$

The following lemma corresponds to Archer's Lemma 4 [13].

Lemma 21 *There exists $v \in T$ such that*

$$y'(A(v)) \geq \frac{y'(A)}{y(T)} \ .$$

PROOF. We take a weighted average of $y'(A(v))$ over $v \in T$.

$$\begin{aligned}
\frac{1}{y(T)} \sum_{v \in T} y_v \cdot y'(A(v)) &= \frac{1}{y(T)} \sum_{v \in T} \sum_{u \in A(v)} y_v \cdot y'(u) \\
&= \frac{1}{y(T)} \sum_{u \in A} y'(u) \sum_{v \in \Gamma^-(u) \cap T} y_v \\
&\geq \frac{1}{y(T)} \sum_{u \in A} y'(u)
\end{aligned}$$

The inequality follows from the fact that for all $u \in A, y'(u) \geq 0$ and $y(\Gamma^-(u) \cap T) \geq 1$ (otherwise there would be more than $p-1$ members of $\Gamma^-(u)$ in S). Since some term is at least as large as the weighted average, the lemma follows. $\square$

Lemma 22

$$|\, S \,| \leq k^2 \ln k \; .$$

PROOF. Due to Lemma 21, the vertex v chosen in every application of the greedy method has $y'(\Gamma^+(v)) = y'(A(v)) \geq y'(A)/y(T)$. In this proof we focus on one iteration at a time and let A' stand for the active vertices *after* the iteration and A for those before. Now,

$$\begin{aligned} y'(A') &= y'(A) - y(A(v)) \\ &\leq y'(A) - y'(A(v))/p \\ &\leq y'(A) - \frac{y'(A)}{y(T) \cdot p} \\ &\leq y'(A) - \frac{y'(A)}{kp} \\ &= y'(V)(1 - \frac{1}{kp}) \end{aligned}$$

since $y(B) \geq y'(B)/p$ for any set B and $y(T) \leq k$. Initially, $y'(V) = kp$, so $y'(V) < p$ after at most $kp \ln k$ iterations. Since $p \leq k$—otherwise no solution exists—we have $|S| \leq k^2 \ln k$. $\square$

Repeatedly applying the greedy procedure for constrained set multicover, this time for $O(\log^* k)$ 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

A disadvantage of the standard k-center problem is that a few distant clients can force centers to be located in isolated places. This situation is averted in the following variant problem, in which a small subset of clients may be denied service, and some points are forbidden from being centers.

Definition 23 (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 24 *For any function $\alpha(n)$, the asymmetric k-center problem with outliers (and forbidden centers) cannot be approximated within a factor of $\alpha(n)$ in polynomial time, 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 each vertex $v_S \in A$, create an edge of unit length to vertex $v_e \in B$ if $e \in S$. Let $p = |B| + k$.

If the optimum value of the max coverage instance is $|\mathcal{U}|$, then the k nodes in A that correspond to some optimal sub-collection will cover p nodes within radius 1. Our $\alpha(n)$-approximation algorithm will thus return k centers that cover p nodes in some *finite* distance. If the maximum coverage with k sets is less than $|\mathcal{U}|$, then the optimum covering radius for p nodes, using k centers, is infinite. Since our approximation can distinguish between these two cases, the approximation problem must be NP-complete. $\square$

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

Perhaps some demand points have a greater need for centers to be closer to them than others. This situation is captured by the priority k-center problem, in which the distance to a demand vertex is effectively enlarged by its priority. Note that the triangle inequality still holds for the original distances.

Definition 25 (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 $p_v d_{cv} \leq R$.*

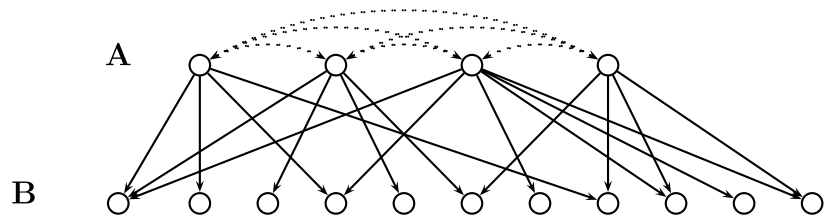

Fig. 1. k-center with priorities. Solid lines have length 1, dotted lines length ℓ.

Theorem 26 *For any polynomial time computable function $\alpha(n)$, the asymmetric k-center problem with priorities cannot be approximated within a factor of $\alpha(n)$ in polynomial time, unless $P = NP$.*

PROOF. The construction of the sets A and B is similar to the proof of Theorem 24. Again, we have the unit length set-element edges from A to B, but this time we make the set A a complete digraph, with edges of length ℓ, as in Figure 1. Give the nodes in set A priority 1 and the nodes in set B priority ℓ.

If there exists a collection of k sets that cover all elements, then there exist k elements of A that cover every vertex in A and B within radius ℓ. If such sets do not exist, then the optimal covering radius using k centers is $\ell^2 + \ell$: Some vertex in B is at distance $\ell + 1$ from its nearest center and has priority ℓ. Since we can set ℓ equal to $\alpha(n)$, our algorithm can distinguish between the two types of max coverage instance. Therefore the approximation problem is NP-complete. $\square$

Asymmetric k-Supplier

In the k-supplier problem the vertex set is segregated into suppliers and customers. Only supplier vertices can be centers and only customer vertices need to be covered.

Definition 27 (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 28 *For any function $\alpha(n)$, the asymmetric k-supplier problem cannot be approximated within a factor of $\alpha(n)$ in polynomial time, unless $P = NP$.*

PROOF. By a reduction from the max coverage problem similar to the proof of Theorem 24. □

Acknowledgements

The authors would like to thank Moses Charikar and the reviewers of the various versions of this paper.

References

[1] M. Mahdian, Y. Ye, J. Zhang, Improved approximation algorithms for metric facility location problems, in: Proc. of 5th APPROX, 2002, pp. 229–42.

[2] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Mungala, V. Pandit, Local search heurisitcs for k-median and facility location problems, in: Proc. of 33rd STOC, 2001, pp. 21–9.

[3] J. Chuzhoy, S. Guha, S. Khanna, S. Naor, Asymmetric k-center is $\log^* n$-hard to approximate, Tech. Rep. 03-038, Elec. Coll. Comp. Complexity (2003).

[4] E. Halperin, G. Kortsarz, R. Krauthgamer, Tight lower bounds for the asymmetric k-center problem, Tech. Rep. 03-035, Elec. Coll. Comp. Complexity (2003).

[5] J. Chuzhoy, S. Guha, E. Halperin, G. Kortsarz, S. Khanna, S. Naor, Asymmetric k-center is $\log^* n$-hard to approximate, in: Proc. of 36th STOC, 2004, pp. 21–7.

[6] A. Archer, Inapproximability of the asymmetric facility location and k-median problems, Unpublished manuscript available at `www.orie.cornell.edu/~aarcher/Research` (2000).

[7] A. Frieze, G. Galbiati, F. Maffioli, On the worst-case performance of some algorithms for the asymmetric traveling salesman problem, Networks 12 (1982) 23–39.

[8] O. Kariv, S. Hakimi, An algorithmic approach to network location problems. I. The p-centers, SIAM J. Appl. Math. 37 (1979) 513–38.

[9] W. Hsu, G. Nemhauser, Easy and hard bottelneck location problems, Disc. Appl. Math. 1 (1979) 209–16.

[10] D. Hochbaum, D. Shmoys, A best possible approximation algorithm for the k-center problem, Math. Oper. Res. 10 (1985) 180–4.

[11] S. Vishwanathan, An $O(\log^* n)$ approximation algorithm for the asymmetric p-center problem, in: Proc. of 7th SODA, 1996, pp. 1–5.

[12] R. Panigrahy, S. Vishwanathan, An $O(\log^* n)$ approximation algorithm for the asymmetric p-center problem, J. Algorithms 27 (1998) 259–68.

[13] A. Archer, Two $O(\log^* k)$-approximation algorithms for the asymmetric k-center problem, in: IPCO, 2001, pp. 1–14.

[14] D. Hochbaum, D. Shmoys, A unified approach to approximation algorithms for bottleneck problems, JACM 33 (1986) 533–50.

[15] S. Khuller, R. Pless, Y. Sussmann, Fault tolerant k-center problems, Theor. Comp. Sci. 242 (2000) 237–45.

[16] J. Plesnik, A heuristic for the p-center problem in graphs, Disc. Appl. Math. 17 (1987) 263–268.

[17] M. Charikar, S. Khuller, D. Mount, G. Narasimhan, Algorithms for facility location problems with outliers, in: Proc. of 12th SODA, 2001, pp. 642–51.

[18] R. Bhatia, S. Guha, S. Khuller, Y. Sussmann, Facility location with dynamic distance function, in: Proc. of 6th Scand. Workshop on Alg. Th. (SWAT), 1998, pp. 23–34.

[19] S. Chaudhuri, N. Garg, R. Ravi, The p-neighbor k-center problem, Info. Proc. Lett. 65 (1998) 131–4.

[20] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, 1979.

[21] V. Vazirani, Approximation Algorithms, Springer, 2001.