

The p -Neighbor k -Center Problem

Shiva Chaudhuri*

Naveen Garg*

R. Ravi†

Abstract

The k -center problem with triangle inequality is that of placing k center nodes in a weighted undirected graph in which the edge weights obey the triangle inequality, so that the maximum distance of any node to its nearest center is minimized. In this paper, we consider a generalization of this problem where, given a number p , we wish to place k centers so as to minimize the maximum distance of any non-center node to its p^{th} closest center. We derive a best possible approximation algorithm for this problem.

1 Introduction

The k -center problem is a classical problem in facility location: given n cities and the distances between them, we wish to select k of these cities as centers so that the maximum distance of a city from its closest center is minimized. The problem is **NP**-hard and Hochbaum and Shmoys present a 2-approximation algorithm¹ for graphs with edge weights obeying triangle inequality [4]. Further they also show that no polynomial time algorithm for this problem can have a performance guarantee of $(2 - \epsilon)$ for any $\epsilon > 0$, unless **P=NP**. In this paper we consider a generalization of the k -center problem with triangle inequality in which we require that each city has some number (say p) of centers ‘close’ to it. We extend the techniques of Hochbaum and Shmoys and provide a best possible approximation algorithm.

Suppose that we wish to locate facilities at k out of n cities such that the maximum distance of a city to its p^{th} -closest facility is minimized. Considering ‘ p^{th} closest’ (as against closest in the k -center problem) is important when the facilities concerned are subject to failure and we wish to ensure that even if up to $p - 1$ facilities fail, every city has a functioning facility close to it. We refer to this generalization as the *p -neighbor k -center problem*. Formally, the problem is to find a subset S of at most k vertices which minimizes

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

Note that setting $p = 1$ reduces it to the k -center problem. We present a polynomial-time algorithm achieving an approximation ratio of 2 for the p -neighbor k -center problem. Since this problem is a generalization of the k -center problem, this approximation ratio is the best possible.

*Max-Planck-Institut für Informatik, Im Stadtwald, 66123 Saarbrücken, Germany. Work supported by the EU ESPRIT LTR Project No. 20244 (ALCOM-IT). Email: shiva,naveen@mpi-sb.mpg.de

†GSIA, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213. Email: ravi@cmu.edu. Work done while this author was visiting the Max-Planck-Institut. Supported in part by an NSF CAREER grant CCR-9625297.

¹An α -approximation algorithm for a minimization problem runs in polynomial time and always outputs a solution of value no more than α times the optimal.

1.1 Related work

Location problems including several versions of the k -center problem are surveyed in [3]. Kariv and Hakimi [6] describe exact solution methods for the k -center problem.

Turning to approximation algorithms, other than the work of Hochbaum and Shmoys mentioned above, Gonzalez [2] as well as Feder and Greene [1] also describe 2-approximation algorithms for the k -center problem. A generalization with vertex weights is addressed by Hochbaum and Shmoys in [5] which also describes a general paradigm for approximating bottleneck problems. In [9], Plesnik considered a generalization of the k -center problem where the distance to the center is multiplied by a vertex priority in the objective; He developed a 2-approximation algorithm. The paper by Wang and Cheng [10] also shows the same result.

The p -neighbor k -center problem was considered previously by Krumke [8] where he provided a 4-approximation algorithm. We use ideas from his work for deriving a lower bound for this problem but provide a different algorithm to achieve an approximation ratio of 2. Our techniques are graph-theoretic; we relate the size of a certain type of dominating set in a graph to the size of a certain type of independent sets. Khuller, Pless and Sussmann [7] have also considered this problem (among other variants) and provided an approximation with the same performance ratio of two using an entirely different approach.

2 The Basic Paradigm

The problem mentioned in the introduction falls into a general class of problems known in the literature as *bottleneck problems*. Roughly speaking a bottleneck problem is one in which we are trying to optimize a bottleneck, *i.e.* minimizing the maximum or maximizing the minimum value of some quantity. Thus for the k -center problem we wish to find from among all dominating sets of size k , the one in which the longest covering edge (we always use a shortest edge from a node to a neighbor in the dominating set as the covering edge for the node) is the shortest.

Hochbaum and Shmoys [5] developed a general paradigm for approximating **NP**-hard bottleneck problems; we illustrate this paradigm with the k -center problem. Let $w_1, w_2, w_3 \dots$ be the edge weights in increasing order and let G_i be the subgraph induced by edges of weight at most w_i . First observe that the optimum value for the k -center problem is equal to one of the edge weights; in particular it is the minimum edge weight w_i such that G_i has a dominating set of size at most k . While it is easy to generate the subgraphs $G_1, G_2, G_3 \dots$, the problem of checking if these subgraphs have a dominating set of size at most k is **NP**-complete. However, suppose that in the subgraph G_i we can find an independent set I of size more than k such that no vertex in G_i is adjacent to two vertices of I . Then any dominating set in G_i has a unique vertex dominating each vertex of I and therefore cannot be of size k or less.

Given a graph $G = (V, E)$ the x^{th} power of G , denoted by $G^x = (V, E^x)$ is a graph with the same vertex set as G and an edge between two vertices if they are connected by a path of at most x edges in G . Then I is an independent set of vertices in G_i^2 . Thus to argue that G_i has no dominating set of size at most k , it suffices to find an independent set in G_i^2 of size larger than k . What if the largest independent set we can find in G_i^2 is of size no more than k ? While we cannot say anything for sure about the size of a dominating set in G_i , we claim that G_i^2 has a dominating set of size at

most k .

To prove this claim we only need to assume that the independent set in G_i^2 that we find (say I) is maximal, *i.e.* the addition of any other vertex to I yields a set which is not independent. But this implies that every vertex not in I has a neighbor in I which means that I is a dominating set in G_i^2 .

Let G_j be the first subgraph in the sequence G_1, G_2, G_3, \dots such that the maximal independent set found in G_j^2 is of size no more than k . Since G_{j-1}^2 has an independent set of size larger than k , every dominating set in G_{j-1} is of size more than k and hence the optimum value is at least w_j . Further, G_j^2 has a dominating set (the maximal independent set found) of size at most k . Since the edge weights satisfy triangle inequality, the longest edge in G_j^2 has weight at most $2w_j$. Thus we have a k -center in which the distance of any vertex to its closest center is at most twice the optimum.

Summarizing, we have the following two key ingredients in this 2-approximation for the k -center problem.

1. If G^2 has an independent set of size more than k , G has no dominating set of size k or less.
2. A maximal independent set is also a dominating set.

The first observation is useful in establishing a lower bound on the optimum value while the second gives a solution of value at most twice the lower bound.

3 The p -neighbor k -center problem

We first generalize the notion of independent and dominating sets following Krumke [8] and sketch his proof of a lower bound relating these sets. However, to obtain the upper bound we describe a different algorithm motivated by proving a stronger graph-theoretic lemma about these sets.

Definition 3.1 *A set of vertices $S \subseteq V$ is p -dominating if every vertex not in the set has at least p neighbors in it, *i.e.* $\forall v \in V - S : \deg_S(v) \geq p$. Thus, a 1-dominating set is the same as a dominating set.*

Definition 3.2 *A set of vertices $S \subseteq V$ is p -independent if every vertex in the set has at most $p - 1$ neighbors in it, *i.e.* $\forall v \in S : \deg_S(v) \leq p - 1$. Thus, a 1-independent set is the same as an independent set.*

The following lemmas relate the size of a p -dominating set in a graph G to the size of a p -independent set in G and G^2 . These can be viewed as extending the relationship between dominating sets and maximal independent sets. The first lemma appears in [8] as Proposition 5. We sketch the proof here for completeness.

Lemma 3.1 [8] *If G has a p -dominating set of size k then no p -independent set in G^2 has size more than k .*

Proof: Let D be a p -dominating set in G ($|D| = k$) and I a p -independent set in G^2 and let v be a vertex in $I - D$. Let S_1 be the vertices in D that are neighbors of v and S_2 the vertices in $V - D$ that are neighbors of the vertices in S_1 . Further, let $S = S_1 \cup S_2$. Since each vertex in S is a neighbor of v in G^2 , the set I contains at most p vertices from S . The set D on the other hand contains at least p vertices from S (the subset S_1). In fact, $D - S$ is a p -dominating set in the residual graph $G[V - S]$ and $I - S$ is a p -independent set in the graph $G^2[V - S]$. Continuing in this manner we will eventually reach a situation when there is no vertex in the residual graph that belongs to the p -independent set but not to the p -dominating set. Since at each step the number of vertices deleted from I was at most the number deleted from D , we have that $|I| \leq |D| = k$. ■

While Krumke showed that a maximal p -independent set in G is p -dominating in G^2 , we show below that there is a p -independent set in G that is also a p -dominating set in G (rather than G^2). This reduces the performance ratio of the resulting algorithm from 4 to 2.

Lemma 3.2 *Given a graph $G = (V, E)$ and $1 \leq p \leq n$, there exists a p -independent set $S \subseteq V$ that is also p -dominating.*

Proof: Let S be a p -independent set that is not p -dominating. In particular let $v \in V - S$ be such that $\deg_S(v) = q < p$. Let U be the neighbors of v in S that have exactly $p - 1$ neighbors in S and let $G[U]$ be the subgraph induced by U in G . Let I be a maximal independent set (and hence also a dominating set) in $G[U]$. Therefore the set $S - I \cup \{v\}$ is also p -independent.

The idea of the proof then is to define a potential function for a p -independent set in such a way that the above swap causes the new p -independent set $S - I \cup \{v\}$ to have strictly more potential than the original set S . This would then imply that the p -independent set with maximum potential must also be p -dominating.

With this motivation, define the *potential* of a p -independent set, S , as $\psi(S) = p \cdot |S| - |E(G[S])|$, where $E(G[S])$ denotes the edge set of the subgraph induced by S in G . Since

$$\begin{aligned} |S| - |S - I \cup \{v\}| &= |I| - 1 \\ |E(G[S])| - |E(G[S - I \cup \{v\}])| &= (p - 1)|I| - (q - |I|) \end{aligned}$$

we have

$$\psi(S) - \psi(S - I \cup \{v\}) = q - p < 0$$

As mentioned earlier, given any p -independent set that is not p -dominating we can obtain another p -independent set that has strictly larger potential. Therefore the p -independent set with maximum potential is also p -dominating. ■

The proof of the above lemma also yields a polynomial time procedure for computing a p -independent set that is also p -dominating. We start with some p -independent set and if this is not p -dominating we find a vertex that has less than p neighbors in the set. Then as in the proof we delete and add vertices to the set to obtain another p -independent set with strictly larger potential. Since the potential of a p -independent set is at least zero and at most $p|V|$, we will obtain a p -independent set that is also p -dominating in at most $p|V|$ steps.

Let G_i be the first subgraph in the sequence $G_1, G_2, G_3 \dots$ for which the p -independent set found in G_i^2 by using the above procedure is of cardinality at most k . By triangle inequality it follows

that the longest edge in G_i^2 is of length at most $2w_i$ and hence we have a p -neighbor k -center of value $2w_i$. Since in G_{i-1}^2 we found a p -independent set of cardinality more than k , G_{i-1} does not have a p -dominating set of size k or less by Lemma 3.1. Hence the optimum value is strictly larger than w_{i-1} (*i.e.* at least w_i) and this gives a 2-approximation algorithm for this problem.

References

- [1] T. Feder and D. H. Greene, "Optimal algorithms for approximate clusters," *ACM Symp. on Theory of Comp.* 434-444 (1988).
- [2] T. F. González, "Clustering to minimize the maximum intercluster distance," *Theoretical Computer Science* **38** 293-306 (1985).
- [3] G. Y. Handler and P. B. Mirchandani, *Location on Networks: Theory and Algorithms*, MIT Press, Cambridge MA, (1979).
- [4] D. S. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k -center problem," *Math. of Oper. res.*, Vol. 10, No. 2, 180-184 (1985).
- [5] D. S. Hochbaum and D. B. Shmoys, "A unified approach to approximation algorithms for bottleneck problems," *Journal of the ACM* **33** 533-550 (1986).
- [6] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. Part I: The p -centers problem," *SIAM J. of Appl. Math.* **37** 513-538 (1979).
- [7] S. Khuller, R. Pless and Y. Sussmann, "Fault-tolerant K -center problems," to appear in 3rd Italian Conference on Algorithms and Complexity (March 1997). Also appears as Technical Report UMIACS-TR-96-40.
- [8] S. O. Krumke, "On a generalization of the p -center problem," *Inf. Proc. Lett.*, 56, 67-71 (1995).
- [9] J. Plesnik, "A heuristic for the p -center problem in graphs", *Disc. Appl. Math.*, **17** 263-268 (1987).
- [10] Q. Wang and K. H. Cheng, "A heuristic algorithm for the k -center problem with vertex weight," *Proc. SIGAL '90*, LNCS 450, 388-396 (1990).