Approximation algorithms for the capacitated and uncapacitated k-center problems

1st Arsalan Firoozi
dept. Electrical Engineering
Sharif University of Technology
Tehran, Iran
arsalan.firoozi@gmail.com

2nd Sahel Vahedi Noori
dept. Electrical Engineering
Sharif University of Technology
Tehran, Iran
alvahedi379@gmail.com

***Abstract*—A k-center problem consists of choosing k nodes in an undirected weighted graph with weights that obey the triangle inequality so that the maximum distance between all nodes to the nearest chosen vertex is minimized. In this paper, all variations of k-center problem such as uncapacitated, general capacitated, and uniform capacitated k-center problems are discussed. For the uncapacitated type two 2-approximate algorithms and for the capacitated type 9-approximate and l-approximate solutions are presented.**

***Index Terms*—k-center problem, capacitated and uncapacitated, approximation algorithms**

I. Introduction

Basically, having a space where a metric of distance is defined, some points, and some probable locations is considered a facility location problem. The probable locations are called "facilities" and the points are called "clients". These elements and the problem have many interpretations in different fields. For instance, in order to protect wildlife, a geographer might have to locate k stations in a field. This problem can be modeled as a k-center problem.

This problem is divided into two categories: Discrete and Continuous. The discrete type of the problem is about deciding whether to locate the facility or not which is obviously an integer linear programming. Instead of deciding where to locate a facility, the continuous type specifies its coordinates.

The first known facility location problem is by Fermat with the question "Given three points in a plane, Find a fourth point such that the sum of its distances to the three given points is minimized.". A geometrical solution seems to be first suggested by Torricelli, his contemporaries. By assigning different weights to the known points, Weber generalized this problem. Weber's problem is more close to the reality in which different clients need different amounts of goods. In 1857 the problem of "Find the smallest circle that encloses a given set of points in the plane" was asked by Sylvester [1] and was answered in 1860 by himself and in 1885 by Chrystal. This problem is called "One-center Problem". Finally, Hakimi in 1964 presented the first modern location problem known as k-median problem [2, 3].

Facility location problem has many variations. The objective function can be sum of the distances to the facility which leads to Minisum problem, or it can be the maximum of the distances to the facility which is called Minimax problem. The single-facility problem, multiple-facility problem, Weber's problem and k-median problem make up the Minisum problem. The k-center problem is in the Minimax category.

Weber's problem can be formulated as:

$$Minimize \sum_{i=1}^{n} \omega_i.d(X, P_i) \tag{1}$$

Where the problem is choosing a point X in a way that the weighted sum of distances to n given points P_i is minimized. A solution for this problem was given by Kulin and Kuenne in 1962 [4].

K-median problem can be formulated as:

$$Minimize \sum_{j=1}^{k} \sum_{i \in S(j)} \omega_i.d(X_j, P_i) \tag{2}$$

The problem is choosing facilities X_j from $j = 1, \ldots, k$ in a way that the weighted sum of distances to allocated given points P_i is minimized, where $S(j)$ is the set of allocated clients. Since this problem separates nodes in k clusters, this problem is used in clustering. Similarly, in the k-mean problem we try to minimize the average distance. A solution for k-median problem was first given by Charikar, Guha, Tardos and Shymos in 1999 [5].

In the system model section, we will formulate the uncapacitated and capacitated k-center problems, and in the next sections, solutions to each of these problems are presented.

II. System Model

A graph $G = (V, E)$ is given. Each edge has its own weights by the $w : E \Rightarrow R^+$ function, where Weights satisfy the triangle inequality. The k-center problem is of choosing a subset of nodes $S \subseteq V$ as centers $|S| \leq k$. A simultaneous allocation problem should also be solved to determine which centers serve which clients. This assignment is called $\alpha : V \Rightarrow S$. The objective function is,

$$Minimize\ Sup_{v \in V}\ d(v, \alpha(v)) \tag{3}$$

The above problem is called uncapacitated k-center problem or metric k-center problem. Considering an additional constraint on the maximum number of clients each center can have, we will achieve another type of this problem, called capacitated k-center problem. The common feature of these two types is that both of them are NP-hard.

III. Uncapacitated K-center Problem

In this section, two approximate solutions will be presented for the metric k-center problem.

A. Greedy algorithm

A simple solution to metric k-center problem can be a greedy algorithm given by Gonzalez [6] that selects a vertex from V and adds it to S in each iteration as a center:

Algorithm 1 Greedy Algorithm

Procedure ChooseS(V, ω)
  Initialize S=ϕ;
  While $|S| < $ k do
    Find argmax d(v, S) over all v $\in V$
    Add v to S
  End while
End procedure

To find out the approximation, assume the inequality,

$$Sup_{v\in V}\ d(v,S) > 2OPT \tag{4}$$

It can be inferred that each two centers in S have distance greater than $2OPT$, since the distance of centers in S are decreasing and are always greater than the maximum distance of a client v to the assigned center. By considering k centers in S and the client v in V which has the greatest distance to its center, we have $k+1$ nodes having distance greater than $2OPT$. In the optimal solution in these $k+1$ nodes, there are two nodes with a shared center based on the pigeonhole. Let's call these two nodes $v1$ and $v2$, which have a distance less than OPT to their shared center called u. Therefore,

$$d(v_1,u) \leq OPT \tag{5}$$

$$d(v_2,u) \leq OPT \tag{6}$$

By triangle inequality,

$$d(v_1,v_2) \leq 2OPT \tag{7}$$

This is a contradiction, which leads to the proof of (8),

$$Sup_{v\in V}\ d(v,S) \leq 2OPT \tag{8}$$

This means we have 2-aprroximation greedy algorithm for metric k-center problem.

B. Parametric Pruning algorithm

Another approximation algorithm for the k-center problem is parametric pruning. Suppose that the optimum solution is achieved by objective function being equal to t. In this algorithm we prune the graph by removing edges with weights higher that $2t$. When two nodes have a distance longer than t, and one of them becomes the center, the other one will not be assigned due to being further to the center. Therefore, the optimal solution will not change. Now the problem becomes finding a dominate set in the pruned graph. Finding a dominate set can be achieved in polynomial time by the following lemma,

Lemma 1: For any independent set I in G^2 and dominating set D in G, we have $|I| < |D|$, Where $G^2 = (V, E')$ is the square graph of G, and satisfies,

$$(u,v) \in E^{'} \Leftrightarrow d(u,v) < 2t,\ \ u \neq v \tag{9}$$

Proof: If we have a dominate set in G, these centers will make $|D|$ cliques in $G^{'}$. An independent set in G^2 can have at most one node from each cliques, since they should not have any edges connected to them. So,

$$|I| < |D| \tag{10}$$

The m in this algorithm denotes the number of edges $e_1, e_2, ..., e_m$ sorted by weights. To find the approximation ratio, any independent set would be a dominating set. Because if a node u is not covered, there can be an independent set containing u which contradicts with the assumption of having maximal independent set. By choosing $S = M_{i'}$, we have $Sup_{v\in V}\ d(v,S) \leq 2\omega(e_{i'})$. Since we have $|D_i| \leq k$, $|\omega(e_{i'})| < OPT$ and $Sup_{v\in V}\ d(v,S) \leq 2OPT$. So, this algorithm is a 2-approximation solution.

Algorithm 2 Parametric pruning algorithm

Procedure ChooseS(V, ω)
  Construct $G_1^2, G_2^2, ..., G_m^2$;
  Find the maximal independent set M_i in each G_i^2
  Find the Smallest i for which $|Mi| < k$ and call it $i^{'}$
  Return $M_{i'}$
End procedure

IV. Capacitated K-center Problem

Now we will analyze two methods for the capacitated k-center problem. The capacitated k-center problem, unlike the uncapacitated case, has a set of capacity constraints $L : V \rightarrow \mathbb{N}$. Considering these constraints, if we choose $v \in V$ as a center, it can not serve more than $L(v)$ clients. But in the uncapacitated case, without any restrictions, we assigned every vertex to the center which was closest to it. In the capacitated k-center problem, not only we should choose the set S, but also we have to solve an allocation problem to decide which open center a vertex should be allocated to.

This additional capacity constraint makes the problem much harder than the uncapacitated case. The first constant factor approximation for a special case, where the capacity constraints are same for all vertices, was given by Barilan, Kortsarz and Peleg [7] in 1993. The first constant factor approximation for general capacities was introduced in 2012 by Cygan et al. [8] for an unspecified constant.

The 9-approximation algorithm given by An, Bhaskara and Svensson [9] in 2013, is the best known algorithm for the capacitated k-center problem in terms of approximation ratio. The idea behind this algorithm was relatively simple and it was the first algorithm for the case with general capacities which had a reasonable approximation ratio. This algorithm is described below.

A. 9-approximation algorithm

This algorithm is an LP-rounding algorithm, that transfers the opening variables, given by the standard LP relaxation of the problem. The algorithm begins by reducing the problem from a weighted graph to an unweighted one. The idea of this reduction is to make a guess at an optimal value τ and check the feasibility of a standard LP for this optimal value. This determines whether it is possible to fractionally open k vertices, while assigning every vertex to a center. The algorithm gives a 9-approximation, if feasible.

The LP solution specifies a set of opening variables that show the fraction to which each vertex should be opened by the optimal fractional solution. We refer to the unweighted graph, made for the LP rounding, as $G = (V, E)$. The core of this algorithm is to round the opening variables by transferring openings between vertices to make them integral.

The algorithm first partitions V. Then introduces one representative vertex for each cluster, which is called the auxiliary vertex of the cluster (Fig. 1). The algorithm now proceeds to transfer openings to make the opening variable of every auxiliary vertex to be equal to one (Fig. 2). This transfer happens within the cluster itself with respect to the capacities limit. This method works because each cluster has a total opening variable of at least one. Now we have an auxiliary vertex for each cluster which has opening variable one. Now

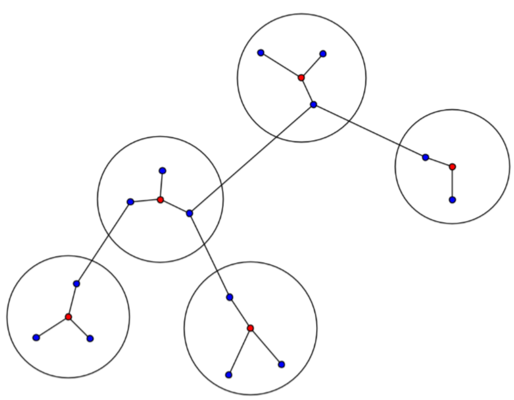

Fig. 1: Modified tree instance after Adding auxiliary vertices, red vertices, to each cluster [11]

we use the auxiliary vertices to define a tree instance, where the internal nodes are all the auxiliary vertices. Considering each auxiliary vertex, the leaf nodes for each vertex are the nodes that are in the same cluster as that auxiliary vertex, and still have a non-zero opening variable. Finally, we obtain a tree that each of its internal nodes has opening variable one. This tree contains a subset of vertices that are chosen as "candidates" to be opened. A 9-approximation algorithm will be developed by carefully selecting non-leaf nodes [9]. Next we look at the uniform capacity k-center problem.

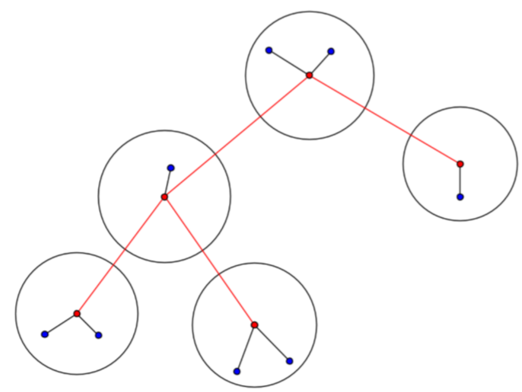

Fig. 2: A tree instance obtained from the auxiliary vertices [11]

B. Uniform Capacity k-center problem

The uniform capacity k-center problem is a special case of the capacitated k-center problem, where the capacity of all vertices are equal. Consider the capacity of every vertex as $L \in \mathbb{N}$. The first constant factor algorithm for the uniform capacity k-center problem was given by Barilan, Kortsarz and Peleg [7] in 1993. Khuller and Sussmann [10] in 1996 gave a 6-approximation algorithm for this problem. There was also another 6-approximation given by An et al. [9] in 2013.

In the following section, a novel approximation algorithm for the uniform capacity k-center problem is introduced.

V. L-APPROXIMATION ALGORITHM

Considering a spanning tree T of $G \leq \tau$, the goal is to give a recursive algorithm that assigns centers such that every client is at a distance of at most L from its assigned center in the graph $G \leq \tau$. We start by choosing an arbitrary vertex r as the root of T and mark an appropriate number of vertices in the sub-trees rooted at each of the children of r, where marking a vertex means that it has been assigned to a center. We then proceed to recursively solve sub-trees rooted at each of the children of r.

Given a weighted graph $G = (V, E)$ with weight function $w : E \rightarrow R^+$, which satisfies triangle inequality and an integer capacity L, the uniform capacity k-center problem is to choose k vertices to open as centers and assign every vertex to an open center, which minimizes the longest distance between a vertex and the assigned center, while respecting the capacity constraints. Consider $d(u, v)$ as the distance between u and v in the graph G.

We reduce the weighted problem to an unweighted one (An et al. [9]) by determining a lower bound τ^* on the optimal value of the solution, then we make a guess τ at OPT and try to decide whether $\tau < OPT$. We consider $G \leq \tau = (V, E \leq \tau)$ to be an unweighted graph in which two vertices are adjacent if and only if they are at a distance less than τ in G. That is, $E \leq \tau = \{(u, v) | d(u, v) \leq \tau\}$. A feasible solution of value τ assigns every vertex to a center that is adjacent in $G \leq \tau$ and conversely, if a solution assigns every vertex to a center that is adjacent in $G \leq \tau$, its value is at most τ. We check the feasibility of τ by considering a standard LP (11) that checks whether there is a fractional feasible solution that assigns vertices to centers that are adjacent in $G \leq \tau$.

Problem 1. Feasibility LP

$$\begin{aligned} &\sum_{u \in v} y_u = k \\ &x_{uv} \leq y_u && \forall u, v \in V \\ &\sum_{v:(u,v) \in E} x_{uv} \leq L.y_u && \forall u \in V \\ &\sum_{u:(u,v) \in E} x_{uv} = 1 && \forall v \in V \\ &0 \leq x, y \leq 1 \end{aligned} \tag{11}$$

x_{uv} indicates the fraction of the opening of v that has been given to the center u, and y_u indicates the fraction vertex u that has been opened as a center. This LP has an unbounded integrality gap in general, but assuming that it is connected, changes the situation. Without loss of generality, we can assume that $G \leq \tau$ is connected, as shown by Cygan et al. [8].

Our problem now can be reduced to an unweighted version, that finds an assignment of centers in $G \leq \tau^*$. Define C_v to be the set of children of v in a rooted undirected tree $\mathcal{T}$. Assume an arbitrary but fixed ordering of vertices $\{v1, v2...\}$ in C_v. Define $C_v = \phi$ if v is a leaf. Define $\mathcal{T}_v$ as the sub-tree rooted at v, given an undirected tree $\mathcal{T}$ with root r and a vertex $v \in \mathcal{T}$. Define $|\mathcal{T}_v|$ as the number of vertices in $\mathcal{T}_v$. with these definitions we have,

Algorithm 3 Constructing $\mathcal{I}_v$

```
procedure ConstructSOI(T_v, L)
initialize I_v = {v};
    for each node u ∈ C_v do
        t = |T_u| (mod L);
        U = Select(T_u, t);
        I_v = I_v ∪ U;
    end for
return I_v;
end procedure
```

where U selects and returns a set of t vertices from $\mathcal{T}_u$. The subroutine for choosing U can be seen in Algorithm 4.

VI. Conclusion

in this paper, first we gave an introduction to k-center problems and their history. After that, the system model was described. In the later sections, starting from the un-capacitated case, greedy and parametric pruning algorithms were explained. Then for the capacitated case, first the 9-approximation algorithm, which can be used when having unrestricted capacities (general form), was explained and finally, for the specific case of uniform capacity, where the capacities must be the same, a novel approach (l-approximation) was explained.

Algorithm 4 Subroutine for selecting vertices

```
procedure Select(T_u, t)
    if t == 0 then
        return φ;
    end if
initialize S = {u};
    i = 1;
    w = u_i;
    while (|S| < t and |T_w| (mod L) ≥ (t − |S|)) do
        S = S ∪ Select(T_w, |T_w| (mod L));
        i = i + 1;
        w = u_i;
    end while
    if |S| < t then
        S = S ∪ Select(T_w, t − |S|);
    end if
return S;
end procedure
```

References

[1] J. J. Sylvester, A Question in the Geometry of Situation, Quarterly Journal of Pure and Applied Mathematics, 1, 1857.

[2] S. L. Hakimi, Optimum locations of switching centers and the absolute centers and medians of a graph, Operations Research, 12(3):450–459, 1964. doi: 10.1287/opre.12.3.450.

[3] Vladamir Marianov and H. A. Eiselt, editors. Foundations of Location Analysis. Springer US, 2011. ISBN 978-1-4419-7571-3.

[4] Harold W. Kulin and Robert E. Kuenne, An efficient algorithm for the numerical solution of the generalized weber problem in spatial economics, Journal of Regional Science, 4 (2):21–33, 1962. doi: 10.1111/j.1467-9787.1962.tb00902.x.

[5] Moses Charikar, Sudipto Guha, va Tardos, and David B. Shmoys, A constant-factor approximation algorithm for the k-median problem, Journal of Computer and System Sciences, 65 (1):129 – 149, 2002. doi: http://dx.doi.org/10.1006/jcss.2002.1882.

[6] Teofilo F. Gonzalez, Clustering to minimize the maximum intercluster distance, Theoretical Computer Science, 38(0):293 – 306, 1985. doi: http://dx.doi.org/10. 1016/0304-3975(85)90224-5.

[7] J. Barilan, G. Kortsarz, and D. Peleg, How to allocate network centers, Journal of Algorithms, 15(3):385 – 415, 1993. doi: http://dx.doi.org/ 10.1006/jagm.1993.1047.

[8] Marek Cygan, Mohammad Taghi Hajiaghayi, and Samir Khuller, LP rounding for k-centers with non-uniform hard capacities, CoRR, abs/1208.3054, 2012.

[9] Hyung-Chan An, Aditya Bhaskara, and Ola Svensson, Centrality of trees for capacitated k-center, CoRR, abs/1304.2983, 2013.

[10] Samir Khuller and Yoram J. Sussmann, The capacitated k-center problem, In Proceedings of the 4th Annual European Symposium on Algorithms, Lecture Notes in Computer Science 1136, pages 152–166. Springer, 1996.

[11] K. Padh, The k-center problem, ResearchGate, 2015.