Capacitated k-Center Problem with Vertex Weights

Aounon Kumar

Indian Institute of Technology Delhi, India
aounon.kumar.csy15@cse.iitd.ac.in

Abstract

We study the capacitated k-center problem with vertex weights. It is a generalization of the well known k-center problem. In this variant each vertex has a weight and a capacity. The assignment cost of a vertex to a center is given by the product of the weight of the vertex and its distance to the center. The distances are assumed to form a metric. Each center can only serve as many vertices as its capacity. We show an $n^{1-\epsilon}$-approximation hardness for this problem, for any $\epsilon > 0$, where n is the number of vertices in the input. Both the capacitated and the weighted versions of the k-center problem individually can be approximated within a constant factor. Yet the common extension of both the generalizations cannot be approximated efficiently within a constant factor, unless $\mathsf{P} = \mathsf{NP}$. This problem, to the best of our knowledge, is the first facility location problem with metric distances known to have a super-constant inapproximability result. The hardness result easily generalizes to versions of the problem that consider the p-norm of the assignment costs (weighted distances) as the objective function. We give $n^{1-1/p-\epsilon}$-approximation hardness for this problem, for $p > 1$.

We complement the hardness result by showing a simple n-approximation algorithm for this problem. We also give a bi-criteria constant factor approximation algorithm, for the case of uniform capacities, which opens at most $2k$ centers.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases approximation hardness, k-center, gadget reduction

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2016.8

1 Introduction

Resource location problems are a class of problems in which one is required to find a set of locations to open centers in order to serve clients (demands) placed in a metric space. The objective is to reduce the cost of opening the centers and/or the cost incurred to assign the clients to the centers. Various notions of distance/cost are used in different applications. The k-center problem is a very well known resource location problem in which a metric on n vertices is given. The objective is to open k centers and assign vertices (clients) to these centers such that the maximum distance between a vertex and its assigned center is minimized. This problem is NP-hard. It also has a $(2 - \epsilon)$-approximation hardness. [14] 2-approximation algorithms were given by Gonzalez [12] and Hochbaum and Shmoys [13].

Motivated by practical scenarios where each center has a limitation on the number of clients that it can serve, a generalization of this problem is the *capacitated* k-center problem. In this problem, each vertex has a capacity and a center opened at a vertex cannot serve more number of vertices than its capacity. Khuller and Sussmann [16] gave 5 and 6-approximation algorithms for uniform soft and hard capacities respectively. For non-uniform capacities, Cygan *et al.* [10] and An *et al.* [1] provide constant factor approximation algorithms using LP rounding.

36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2016).
Editors: Akash Lal, S. Akshay, Saket Saurabh, and Sandeep Sen; Article No. 8; pp. 8:1–8:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Another generalization of the k-center problem is one where vertices have *weights*. The assignment cost of a vertex to a center is given by the product of the weight of the vertex and its distance (*weighted distance*) to the center. This variant is motivated from scenarios where the clients are not treated equally. Some clients are more important than others and need to be kept closer to an open center. Weights can also be used to model the likelihood of clients demanding services. Wang and Cheng [20] provide a 2-approximation for the k-center problem with vertex weights. This is best possible as the k-center problem has $(2-\epsilon)$-approximation hardness.

A common extension of the above two generalizations is the *capacitated* k-center with *vertex weights*. In this variant each vertex has a capacity and a weight. Each center can serve no more vertices than its capacity. The assignment cost of a vertex to a center is given by its weighted distance to the center. In this paper we study the approximability of this problem. We show an $n^{1-\epsilon}$-approximation hardness and provide an n-approximation algorithm. The hardness result easily generalizes to variants of the problem that consider the p-norm of the assignment costs (weighted distances) as the objective function. We give $n^{1-\frac{1}{p}-\epsilon}$-approximation hardness for the general p-norm, for $p>1$. This immediately shows that for $p>1$, the problem is hard to approximate within a constant factor. Although this generalization does not immediately provide an inapproximability result for the 1-norm which is the corresponding variant of the k-median problem, it provides insights into the capacitated (unweighted) version of the problem. The capacitated k-median problem is interesting as not much is known about its approximability. Constant factor approximation algorithms by either violating the capacity constraints or the cardinality constraints up to a constant factor are studied in [9], [4], [18], [5].

A vast body of work is available on various facility location problems. A variety of techniques like local search [17], [2], [6], LP rounding [7] and primal-dual method [15] have been studied. The capacitated facility location problem is well studied in [17], [19], [8], [3] and constant factor approximations are known.

Our results and techniques

The main result of this paper is the approximation hardness of the capacitated k-center problem with vertex weights. We show that this problem cannot be efficiently approximated within a factor of $n^{1-\epsilon}$, unless $\mathsf{P}=\mathsf{NP}$, for any $\epsilon>0$, where n is the number of vertices in the input. We give a reduction from the Exact Cover by 3-Sets, which is an NP-complete problem. It requires one to find a set cover from a family of sets, where each set has exactly three elements, such that each element of the universe is in exactly one of the sets in the set cover. This set cover variant was used by Cygan *et al.* in [10] to show a $(3-\epsilon)$-approximation hardness for the capacitated k-center problem. The set gadget used in the reduction in [10] is designed for the unweighted case and does not generalize for the weighted case. In this paper, we introduce a novel set gadget that allows to create an polynomial factor gap between the solution cost of the *yes* and the *no* instances. It achieves this by allowing the vertices in a set gadget to be assigned to centers inside the gadget with *small* costs and making assignments to centers outside the gadget incur a *large* cost. Similarly, the vertices in an element gadget can only be assigned to centers in set gadgets corresponding to sets that it belongs to, with a small cost. Our reduction generates instances where the capacities are uniform and constant, showing that even this special case is hard to approximate within a constant factor.

An immediate consequence of the hardness result is an $n^{1-\frac{1}{p}-\epsilon}$-approximation hardness for the case where the objective function is a general p-norm of the assignment costs, for $p>1$. The k-center problem is a special case where the objective function is the ∞-norm of the assignment costs. This shows that interesting variants of the problem which consider a p-norm for $p > 1$ are hard to approximate within a constant factor.

We complement the hardness result by showing a simple n-approximation algorithm. For this algorithm, we use the standard *thresholding* technique modified to handle weights. We create threshold graphs corresponding to each distinct weight in decreasing order and open as many centers, in decreasing order of capacities, in each connected component as required to cover all the vertices in it.

Next, we relax the cardinality constraint on the set of centers. We consider the variant with uniform capacities where we show that if we are allowed to open twice the number of centers then we can output a solution with cost within a constant factor of the optimum cost. This simply modifies the 2-approximation by Wang and Cheng [20] by opening as many capacitated centers required in place of each uncapacitated center to serve all the vertices assigned to it.

2 Problem statement

The input for the capacitated k-center problem with vertex weights (CkCW) is a set of vertices V, a metric distance $d : V \times V \rightarrow \mathbb{R}_{\geq 0}$ on V, an integer k, a capacity function $L : V \rightarrow \mathbb{Z}_{\geq 0}$ and a weight function $W : V \rightarrow \mathbb{R}_{\geq 0}$. The output is a set $S \subseteq V$ of k vertices called *centers* and an *assignment map* $h : V \rightarrow S$ such that $|\{j \in V \mid h(j) = i\}| \leq L(i), \forall i \in S$. The *assignment cost* of a vertex $j \in V$ to a center $i \in S$ is given by $W(j)d(i, j)$. The goal is to minimize the maximum assignment cost of a vertex to its assigned center. Formally, the cost of the solution is given by $\max_{j \in V} W(j)d(h(j), j)$. Let $|V| = n$.

The metric distance d satisfies the following properties for $i, j, u \in V$:

1. $d(i, j) \geq 0$
2. $d(j, j) = 0$
3. $d(i, j) = d(j, i)$
4. $d(i, j) \leq d(i, u) + d(u, j)$

3 Hardness of approximation

In this section we show that the above problem cannot be approximated within a constant factor. We give a reduction from the Exact Cover by 3-Sets (EC3S), which is an NP-complete problem. This problem is used in [10] to show a $(3 - \epsilon)$-approximation hardness for the k-center problem (unweighted) with non-uniform capacities. The input of the problem is a set system $(\mathcal{F}, \mathcal{U})$, where each set in $\mathcal{F}$ has exactly 3 elements. The goal is to decide whether there exists a subset $\mathcal{F}' \subseteq \mathcal{F}$, such that each element of $\mathcal{U}$ belongs to exactly one set in $\mathcal{F}'$. For such a set cover to exist, $|\mathcal{U}|$ must be a multiple of three.

An instance of the EC3S problem can be viewed as a bipartite graph $(\mathcal{F} \cup \mathcal{U}, \mathcal{E})$ where the edge set $\mathcal{E}$ encodes the membership of the elements of $\mathcal{U}$ in the elements of $\mathcal{F}$. In our reduction, we encode this bipartite graph into an instance $\mathcal{I}$ of CkCW, with each vertex having a uniform capacity of L. We replace each vertex of $\mathcal{F}$ with the corresponding *set gadget* and that of $\mathcal{U}$ with the corresponding *element gadget*.

Figure 1 illustrates these *gadgets*. The *set gadget* consists of three *long* arms, one for each of the three elements in the set, joined together with a clique at the top. Each arm is divided into *integral* levels 0, 1, ..., t and *fractional* levels 0.5, 1.5, ..., $t + 0.5$, where t is an odd integer which we will fix later in this construction. An integral level l consists of a vertex of weight W_l. A fractional level $l + 0.5$ contains $\frac{L}{3}$ vertices of weight W_l if l is odd and $\frac{2L}{3} - 2$

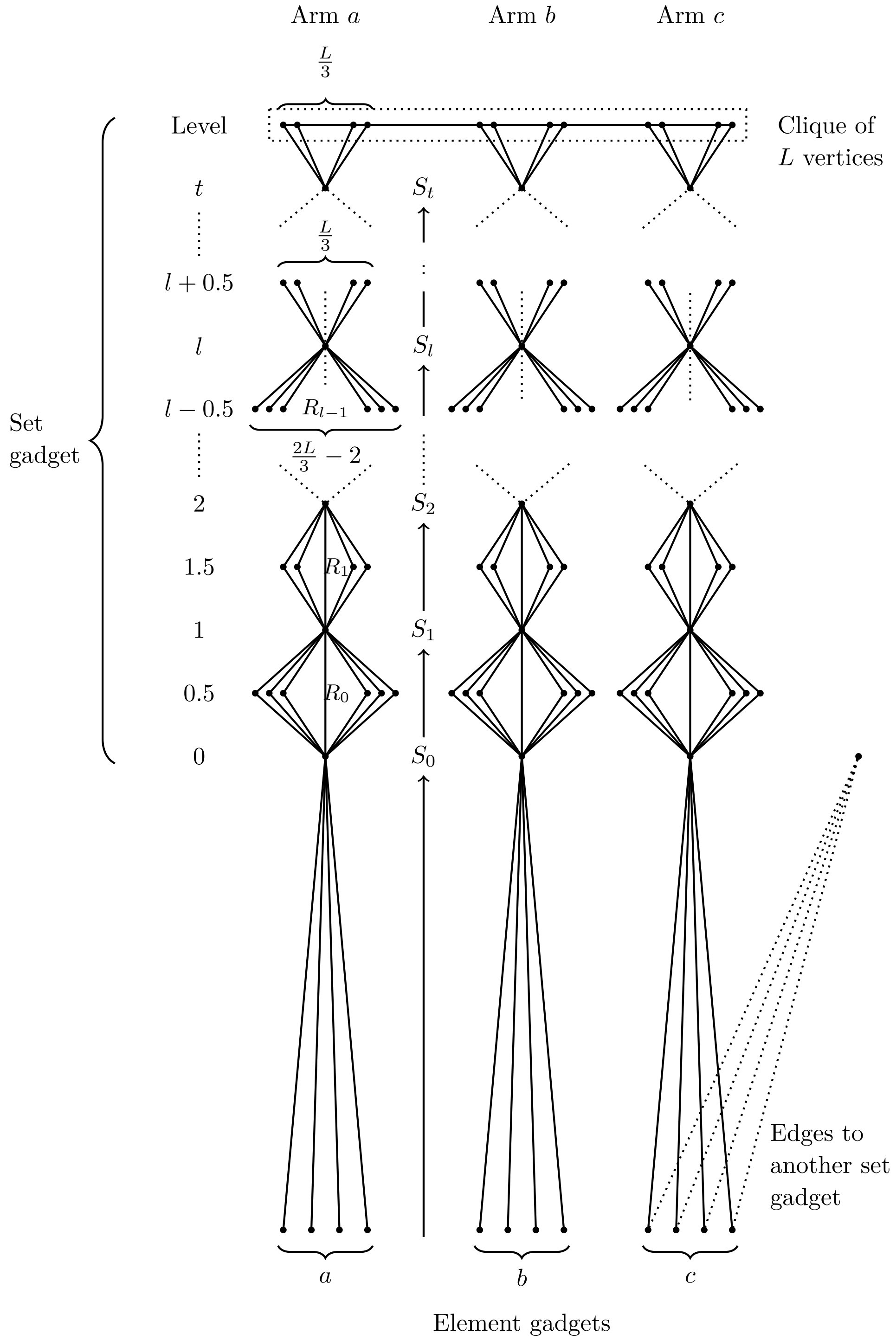

Figure 1 Gadgets for reduction.

vertices of weight W_l if l is even. The two vertices in levels l and $l+1$ are connected to each other and to the vertices in level $l+0.5$ by edges off length R_l. The $\frac{L}{3}$ vertices in level $t+0.5$ (the highest level) from each of the arms are all connected to each other by edges of length R_t, forming a clique of size L. The *element gadget* is a collection of $\frac{L}{3}$ vertices of unit weight connected to the level 0 vertex of the corresponding arm of the set gadget of each of the sets that it belongs to. The length of each of these connecting edges is S_0.

Let S_l denote the shortest distance of a level l vertex from the vertices in the element gadget connected to the corresponding arm (refer to Figure 1). Then we have the following relation:

$$S_l = R_{l-1} + S_{l-1} \tag{1}$$

We would like to set the parameters of the construction in such a way that any solution with cost $< w^2$ must assign the vertices in a set gadget to centers in the same gadget. It must also assign the vertices of an element gadget to centers in set gadgets corresponding to the sets it belongs to. So, we want the following relations to hold:

$$W_0 = w \tag{2}$$

$$W_l R_l = w \tag{3}$$

$$W_l S_l = w^2 \tag{4}$$

where w is some parameter. From equations 1, 3 and 4 we get:

$$\begin{aligned} S_l &= S_{l-1}\left(\frac{1}{w}+1\right) \\ &= S_0\left(\frac{1}{w}+1\right)^l = w\left(\frac{1}{w}+1\right)^l && \text{(from equations 2 and 4)} \end{aligned}$$

We fix t such that:

$$\begin{aligned} S_t &= w\left(\frac{1}{w}+1\right)^t \geq w^2 \\ t &\geq \frac{\log(w)}{\log\left(1+\frac{1}{w}\right)} \leq 2w\log w \end{aligned}$$

We set t to be an odd integer just greater than $2w\log w$ and k to be $3\left(\frac{t+1}{2}\right)|\mathcal{F}| + \frac{|\mathcal{U}|}{3}$. The distance metric d is given by the shortest distance metric.

▶ **Lemma 1.** *If there exists a solution to the* EC3S *instance, then there exists a solution to the instance $\mathcal{I}$ with cost w.*

Proof. Let $\mathcal{F}' \subseteq \mathcal{F}$ be the solution of the EC3S instance. Note that, $|\mathcal{F}'| = \frac{|\mathcal{U}|}{3}$. For a sets $A \in \mathcal{F}'$ place a center on each of the three vertices at even levels 0, 2, …, t-1 in the corresponding set gadget and one center on a vertex at level $t+0.5$. Assign the vertices of the element gadget corresponding to the elements in A and the vertices in levels 0, 0.5 and 1 to the centers at level 0. For a level $l \in \{2, 4, \ldots, t-1\}$, assign all the vertices at levels $l-0.5$, l, $l+0.5$ and $l+1$ to the centers at level l. Assign all the vertices in the clique at level $t+0.5$ to the center opened at this level. For all sets not in $\mathcal{F}'$, place centers similarly at odd levels 1,3, …, t. For a level $l \in \{1, 3, \ldots, t\}$, assign all the vertices at levels $l-1$, $l-0.5$, l and $l+0.5$ to the centers at level l. This is an assignment with cost w. The total number of centers opened is $\sum_{A\in\mathcal{F}'}\left(3\left(\frac{t+1}{2}\right)+1\right) + \sum_{A\notin\mathcal{F}'} 3\left(\frac{t+1}{2}\right) = 3\left(\frac{t+1}{2}\right)|\mathcal{F}| + \frac{|\mathcal{U}|}{3} = k$. ◀

Now consider a solution $\mathcal{S}$ to the instance $\mathcal{I}$ with maximum assignment cost $< w^2$. Note that each center must serve L vertices as $|V| = 3\binom{t+1}{2}|\mathcal{F}|L + |\mathcal{U}|\frac{L}{3} = kL$.

▶ **Lemma 2.** *$\mathcal{S}$ does not have a center in any of the element gadgets.*

Proof. Consider the vertex set g_a of the element gadget for an element $a \in \mathcal{U}$. From the construction of the gadget we can say that any $j \notin g_a$ and $i \in g_a, W(j)d(i,j) \geq w^2$. Therefore, only the vertices in g_a can be assigned to a center in g_a. But, $|g_a| = \frac{L}{3} < L$. ◀

▶ **Lemma 3.** *Each set gadget in $\mathcal{S}$ has at least $3\binom{t+1}{2}$ and at most $3\binom{t+1}{2}+1$ open centers in it.*

Proof. Consider the vertex set g_A of the set gadget for a set $A \in \mathcal{F}$. For any $j \in g_A$ and $i \notin g_A, W(j)d(i,j) \geq w^2$. Thus, all the vertices in g_A must be assigned to centers in g_A. Therefore, the number of centers in $g_A \geq \lceil |g_A|/L \rceil = 3\binom{t+1}{2}$.

Assume, for contradiction, that the number of centers in $g_A > 3\binom{t+1}{2}+1$. Let a, b and c be the elements of set A and let g_a, g_b and g_c be the vertex sets of their respective gadgets. For any $j \notin g_A \cup g_a \cup g_b \cup g_c$ and $i \in g_A, W(j)d(i,j) \geq w^2$. Thus the number of vertices that the centers in g_A can serve $\leq |g_A| + |g_a| + |g_b| + |g_c| = \left(3\binom{t+1}{2}+1\right)L$. Therefore, at least one of the centers in g_A must be serving less than L vertices. ◀

▶ **Lemma 4.** *In $\mathcal{S}$, gadgets corresponding to any two sets in $\mathcal{F}$ sharing a common element cannot have $3\binom{t+1}{2}+1$ open centers in each one of them.*

Proof. Assume, for contradiction, that there exist two sets $A, B \in \mathcal{F}$ having at least one element in common such that the vertex sets g_A and g_B of the corresponding gadgets each have $3\binom{t+1}{2}+1$ centers. As shown in the proof of Lemma 3, the vertices that can be assigned to a center in the gadget of a set $C = \{d, e, f\}$ are only those in $g_C \cup g_d \cup g_e \cup g_f$. Thus, the number of vertices that can be assigned to centers in g_A and $g_B \leq 2 \times \left(3\binom{t+1}{2}+1\right)L - \frac{L}{3}$ (since at least one element is common in A and B). Therefore, at least one of the centers in g_A or g_B must be serving less than L vertices. ◀

▶ **Lemma 5.** *If there exists a weighted k-center solution with cost $R < w^2$, then there exists a solution to the* EC3S *instance.*

Proof. From Lemmas 2, 3 and 4, there are $\frac{|U|}{3}$ set gadgets each of which have $3\binom{t+1}{2}+1$ centers and the corresponding sets are all disjoint. These $\frac{|U|}{3}$ sets form the solution set $\mathcal{F}'$. ◀

▶ **Theorem 6.** *The weighted k-center solution cannot be approximated within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$, unless* $\mathsf{P} = \mathsf{NP}$.

Proof. From Lemmas 1 and 5, an α-approximation is not possible for $\alpha < w$, unless $\mathsf{P} = \mathsf{NP}$.

Now, we show a lower bound on w in terms of n. In the construction, the number of vertices is given by:

$$
\begin{aligned}
n = kL &= \left(3\binom{t+1}{2}|\mathcal{F}| + \frac{|\mathcal{U}|}{3}\right)L \\
&\leq \text{constant} \times w \log w |\mathcal{F}| && \left(|\mathcal{F}| \geq \tfrac{|\mathcal{U}|}{3}, t \sim 2w\log w \text{ and } L \text{ is constant}\right) \\
&\leq \text{constant} \times w^{1+\frac{1}{q}} \log w && (\text{setting } w = |\mathcal{F}|^q, q > 0) \\
&\leq \text{constant} \times w^{1+\frac{2}{q}} \\
w &\geq \text{constant} \times n^{\frac{1}{1+\frac{2}{q}}} > n^{1-\epsilon} && \left(\text{for sufficiently large } n \text{ and } q > \tfrac{2}{\epsilon}\right)
\end{aligned}
$$

◀

▶ Remark. The capacity L of each vertex does not depend on the input of the reduction. Thus, L can be fixed to be a sufficiently large constant. In Appendix A, we show that the known hardness results of $(3-\epsilon)$ for the $\{0, L\}$ capacitated version [10] and $(2-\epsilon)$ for the uniform L capacitated version (which follows from the $(2-\epsilon)$-approximation hardness of the uncapacitated problem [14]) of k-center problem hold even when L is a constant. Note, that for $L = 1$ the problem can be solved trivially.

Generalizing to other cost functions

In the k-center problem the goal is to minimize the maximum assignment cost, that is, to minimize the infinity norm of the assignment costs. Now we generalize the hardness result for any p-norm as the objective function. The objective function is given by:

$$\left(\sum_{j \in V} (W(j)d(h(j), j))^p\right)^{\frac{1}{p}}.$$

Consider the instance $\mathcal{I}$ generated by the reduction. If there exists a solution to the EC3S instance, there exists a solution to the instance $\mathcal{I}$ with cost at most $n^{\frac{1}{p}}w$ and if there is no solution to the EC3S instance then any solution to $\mathcal{I}$ must have a cost at least w^2. Thus an approximation factor of $w/n^{\frac{1}{p}}$ or, $n^{1-\frac{1}{p}-\epsilon}$ cannot be achieved, unless P = NP. This gives a super-constant inapproximability result for, $p > 1$.

4 n-approximation algorithm

In this section, we present a simple n-approximation algorithm for the capacitated k-center problem with vertex weights. It guesses through all possible values R of the optimal solution cost in increasing order. The number of possible values can be at most $|V|^2$ as each value must be equal to $W(j)d(i,j)$ for some $i, j \in V$. For each R, consider the distinct values of the weights $w_1 > w_2 > \cdots > w_m$ in decreasing order, where m is the number of distinct weights. For each distinct weight w_i, it creates the undirected graph $G_{r_i} = (V, E_{r_i})$ where V is the input set of vertices and $E_{r_i} = \{(i,j) \mid d(i,j) \leq r_i = R/w_i\}$. Note that if R is the optimal solution cost then the optimal solution cannot assign a vertex j to a center i such that $d(i,j) > R/W(j)$. Let Γ_i be the set of connected components of G_{r_i} which have at least one vertex of weight at least w_i. For a component $\gamma \in \Gamma_i$, let $\mathcal{H}_i^\gamma = \{v \in \gamma \mid W(v) \geq w_i\}$ be the set of *heavy* vertices and $\mathcal{P}_\gamma$ be the set of open centers in γ. $\mathcal{P}_\gamma$ for $\gamma \in \Gamma_i$, initially consists of the centers opened at vertices in γ up till iteration $i-1$. We say a center in $\mathcal{P}_\gamma$ is *unsaturated* if the number of vertices assigned to it is less than its capacity. The algorithm, in iteration i, assigns vertices from $\mathcal{H}_i^\gamma$ for each component γ to unsaturated centers in $\mathcal{P}_\gamma$ till their capacities are exhausted and then adds new centers to serve all the remaining vertices in $\mathcal{H}_i^\gamma$. After m iterations, if the number of open centers is at most k it returns the set as the solution. Algorithm 1 illustrates this procedure.

▶ **Lemma 7.** *The assignment cost of each vertex is at most nR.*

Proof. Note that in each iteration i, the algorithm assigns all the vertices of weight w_i to some center. Also, each vertex in a component is assigned to some center in the same component. Thus the assignment cost is at most $w_i n r_i = w_i nR/w_i = nR$. ◀

Consider an optimal solution $\mathcal{S}^*$. Let R^* be the optimal solution cost. For a center u in the optimal solution, let $\sigma(u)$ be the number of vertices assigned to it. Now consider the iteration of Algorithm 1 when $R = R^*$.

Algorithm 1 n-approximation algorithm

```
for each guess R of the optimal solution cost in increasing order do
    Order the weights w_1 > w_2 > ··· > w_m
    for each w_i, i ∈ {1, 2, ..., m} do
        Construct G_{r_i}
        Construct the set Γ_i for G_{r_i}
        for each component γ ∈ Γ_i do
            Construct P_γ
            while ∃ unassigned vertex v ∈ H_i^γ do
                if ∃ unsaturated center u ∈ P_γ then
                    assign v to u
                else
                    P_γ → P_γ ∪ {u}, where u ∈ γ\P_γ, such that L(u) = max{L(v) | v ∈ γ\P_γ}
                end if
            end while
        end for
    end for
    if |⋃_{γ∈Γ_m} P_γ| ≤ k then
        return set of open centers and vertex assignment map
    end if
end for
```

▶ **Lemma 8.** *For a component $\gamma \in \Gamma_i$ of any G_{r_i}, there exists a set of centers χ_γ opened by $\mathcal{S}^*$ in component γ with the following properties:*

1. $|\chi_\gamma| = |\mathcal{P}_\gamma| = \kappa_\gamma$
2. *Order the elements u_i of χ_γ in decreasing order of the value of $\sigma(u_i)$ and the elements p_i of $\mathcal{P}_\gamma$ in decreasing order of their capacities $L(p_i)$. For $i \in \{1, 2, \ldots, \kappa_\gamma\}, \sigma(u_i) \leq \mathsf{L}(p_i)$.*

Proof. We prove this by induction on i. The lemma holds for Γ_1 since the algorithm opens centers in decreasing order of capacities in each of the components. Assume it holds for Γ_i for some i. Note that, from the construction of a component in Γ_i, we can say that each component in Γ_i is disjoint from other components in Γ_i and is a subset of some component in Γ_{i+1}. Now consider a component $\gamma \in \Gamma_{i+1}$. Let $\gamma_1, \gamma_2, \ldots, \gamma_z$ be the components in Γ_i which are subsets of γ. As long as there is an unsaturated center from iteration i, the algorithm assigns vertices to that center. If all the vertices in γ are assigned to some center from iteration i, then the lemma holds for Γ_{i+1}. The corresponding $\mathcal{P}_\gamma$ and χ_γ would be $\mathcal{P}_{\gamma_1} \cup \mathcal{P}_{\gamma_2} \cup \cdots \cup \mathcal{P}_{\gamma_z}$ and $\chi_{\gamma_1} \cup \chi_{\gamma_2} \cup \cdots \cup \chi_{\gamma_z}$ respectively.

Now consider the case when all the centers from iteration i are saturated. $\mathcal{P}_\gamma = \mathcal{P}_{\gamma_1} \cup \mathcal{P}_{\gamma_2} \cup \cdots \cup \mathcal{P}_{\gamma_z}$ and $\chi_\gamma = \chi_{\gamma_1} \cup \chi_{\gamma_2} \cup \cdots \cup \chi_{\gamma_z}$ satisfy the conditions of the lemma. Arrange all the vertices in γ in decreasing order of capacities. Let q be the smallest index in this ordering such that the algorithm has not opened a center at the q^{th} vertex. Replace the first $q-1$ centers in χ_γ with the highest $q-1$ centers opened in γ by $\mathcal{S}^*$, according to the number of vertices served. The new χ_γ and $\mathcal{P}_\gamma$ also satisfy both the conditions of the lemma. Now, if there are unassigned vertices even after all centers in $\mathcal{P}_\gamma$ are saturated, the algorithm opens center at the vertex at index q and adds it to $\mathcal{P}_\gamma$. The optimum solution must also have an open center $u \notin \chi_\gamma$ as the centers in χ_γ do not serve all the vertices in γ. $\sigma(u)$ can be at most the number of vertices served by the q^{th} maximum center in the optimum solution which is at most the capacity of the newly opened center. We compute q

again and replace the $q-1$ centers in χ_γ as previously. This shows that both the conditions of the lemma hold when each new center is added by the algorithm. Hence, the lemma holds for Γ_{i+1}. ◀

▶ **Theorem 9.** *Algorithm 1 is an n-approximation algorithm for the capacitated k-center problem with vertex weights.*

Proof. When $R = R^*$, consider $\mathcal{P}_\gamma$ for $\gamma \in \Gamma_m$ after the algorithm has iterated through all the distinct weights. At this point, each vertex is assigned to some open center. From Lemma 8, there exists a set of centers χ_γ opened by $\mathcal{S}^*$ in component γ such that $|\chi_\gamma| = |\mathcal{P}_\gamma| = \kappa_\gamma$. Since all the components are disjoint, the number of centers opened by the algorithm is $\sum_{\gamma\in\Gamma_m} |\mathcal{P}_\gamma| = \sum_{\gamma\in\Gamma_m} |\chi_\gamma| \leq k$. Also, from Lemma 7, each assignment cost is at most nR^*. ◀

5 Relaxing the number of centers

In this section we present a greedy $(2, 2)$-approximation algorithm[1] for the uniform soft capacitated k-center problem with vertex weights. In the soft capacitated version, the solution is allowed to have multiple centers at a vertex. All vertices have equal capacities of L. The algorithm uses the greedy clustering technique used by Wang and Cheng in [20] to produce a solution for the uncapacitated version of the problem. It then replaces the open uncapacitated centers with the required number of capacitated ones.

For an input instance $\mathcal{I}$ and a solution cost R, we can construct a digraph $G_R = (V, E_R)$, where V is the set of vertices in $\mathcal{I}$ and $E_R = \{(j, i) \mid W(j)d(i, j) \leq R\}$ is the set of edges that a solution with cost R can potentially use to assign vertices to centers. Thus, a directed edge $(j, i) \in E_R$ if j can be assigned to i within cost R. It is easy to verify that there exists a solution to $\mathcal{I}$ with cost R if and only if there exists a set $S \subseteq V, |S| = k$ and an assignment map $h : V \to S$ assigning vertices to centers respecting the capacity constraint and using only the edges in E_R, that is, $h(j) = i \implies (j, i) \in E_R$.

Given an instance $\mathcal{I}$ of the problem, the algorithm goes through all possible values R (which can be at most $|V|^2$) of the optimal solution cost, in increasing order. It constructs the graph G_R and for each vertex $v \in V$, computes its neighbourhood $N(v)$ as:

$$N(v) = \{v\} \cup \{u \mid (u, v) \in E_R\} \cup \{u \mid \exists x \in V, (v, x), (u, x) \in E_R\}$$

It then select a set of vertices S greedily according to weight and clusters (C_v) the vertices in the neighbourhood of each vertex $v \in S$. It opens sufficient number centers at the vertices in S (with multiple centers at a vertex if required) such that all the vertices can be assigned to some center with cost at most $2R$, respecting the capacity constraint. Algorithm 2 formally defines this greedy procedure.

▶ **Lemma 10.** *For any vertex in a cluster $C_v, \forall v \in S$, its cost of assignment to an open center at v is at most $2R$. Formally,*

$$d(v, j)W(j) \leq 2R, \forall j \in C_v\,.$$

Proof. The lemma holds trivially for v. All other vertices $j \in C_v$ are of the following two types:

[1] An (α, β)-approximation algorithm outputs a solution with cost at most αR by opening at most βk centers

Algorithm 2 Greedy algorithm

for each guess R of the solution cost in increasing order **do**
  Construct G_R.
  for each $v \in V$ **do**
    Construct $N(v)$
  end for
  $\mathcal{X} \leftarrow V$
  $S \leftarrow \phi$
  while $\mathcal{X}$ is not empty **do**
    select $v \in \mathcal{X}$ such that $W(v) = \max\{W(v) \mid v \in \mathcal{X}\}$
    $S \leftarrow S \cup \{v\}$
    Assign $\mathcal{X} \cap N(v)$ to cluster C_v
    Open $\lceil |C_v|/L \rceil$ centers at v
    $\mathcal{X} \leftarrow \mathcal{X} \setminus N(v)$
  end while
  if number of open centers $\leq 2k$ **then**
    return set of open centers
  end if
end for

Type 1: $(j, v) \in E_R$. In this case, by definition of E_R we have:

$$W(j)d(v, j) \leq R \leq 2R\,.$$

Type 2: $\exists x \in V, (v, x), (j, x) \in E_R$. Algorithm 2 in its while loop selects the maximum weight vertex v from the set $\mathcal{X}$ in a given iteration. Since, $C_v \subseteq \mathcal{X}$, therefore, $W(j) \leq W(v)$.

$$\begin{aligned}
W(j)d(v, j) &\leq W(j)\left(d(v, x) + d(x, j)\right) && \text{(using triangle inequality)}\\
&\leq W(j)\left(\frac{R}{W(v)} + d(x, j)\right) && ((v, x) \in E_R)\\
&\leq W(j)\left(\frac{R}{W(j)} + d(x, j)\right) && (W(j) \leq W(v))\\
&\leq R + W(j)d(x, j) \leq 2R && ((j, x) \in E_R)
\end{aligned}$$

◀

Let R^* be the optimal solution cost. Now consider the iteration of Algorithm 2 when $R = R^*$.

▶ **Lemma 11.** *Algorithm 2 opens at most $2k$ centers and every vertex in cluster C_v can be assigned to some open center at v.*

Proof. Algorithm 2 opens $\lceil |C_v|/L \rceil$ centers at vertex v in cluster C_v which is sufficient to serve all vertices in C_v. Also, no two vertices in S can be served by the same center in the optimal solution, otherwise one of them must be in the neighbourhood of the other. Thus, $k \geq |S|$. The total number of centers k' opened by Algorithm 2 follows,

$$k' = \sum_{v \in S} \lceil |C_v|/L \rceil = |S| + \sum_{v \in S} \lfloor |C_v|/L \rfloor \leq |S| + \lfloor |V|/L \rfloor \leq 2k \qquad \text{(all clusters are disjoint)}$$

◀

▶ **Theorem 12.** *Algorithm 2 is a* $(2, 2)$*-approximation algorithm for the uniform soft capacitated* k*-center problem with vertex weights.*

Proof. Follows from Lemmas 10 and 11. ◀

▶ Remark. Algorithm 2 can be modified to a $(4, 2)$-approximation for the uniform hard capacitated k-center problem with vertex weights. In the hard capacitated version, multiple centers are not allowed to be opened at the same location. So, instead of opening all the centers in a cluster at one vertex we open one center at each of the top $\lceil |C_v|/L \rceil$ vertices in C_v in decreasing order of weight. The cost of assigning a vertex with a lower weight to a center with higher weight is at most $4R$.

6 Conclusion and open problems

In this paper we make progress towards showing approximation hardness for capacitated facility location problems with vertex weights. To the best of our knowledge, this is the first facility location problem known to be hard to approximate within a constant factor. This provides insight into other variants, for many of which not much is known about their approximabilities. It would be interesting to extend our result for the k-median problem.

Other directions for future work would be to reduce the gap between the lower bound of $n^{1-\epsilon}$ and the upper bound of n presented in this paper and to design algorithms that achieve a constant factor on the solution cost by relaxing the cardinality or capacity constraints up to a constant smaller than 2.

Acknowledgements. We thank Naveen Garg and Amit Kumar for many helpful discussions. We also thank anonymous reviewers for their valuable comments.

References

1. Hyung-Chan An, Aditya Bhaskara, Chandra Chekuri, Shalmoli Gupta, Vivek Madan, and Ola Svensson. Centrality of trees for capacitated k-center. In *Integer Programming and Combinatorial Optimization*, pages 52–63. Springer, 2014.
2. Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC'01, pages 21–29, New York, NY, USA, 2001. ACM. `doi:10.1145/380752.380755`.
3. Manisha Bansal, Naveen Garg, and Neelima Gupta. A 5-approximation for capacitated facility location. In *Proceedings of the 20th Annual European Conference on Algorithms*, ESA'12, pages 133–144, Berlin, Heidelberg, 2012. Springer-Verlag. `doi:10.1007/978-3-642-33090-2_13`.
4. Jarosław Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor approximation algorithms for hard capacitated k-median problems. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'15, pages 722–736, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=2722129.2722178`.
5. Jarosław Byrka, Bartosz Rybicki, and Sumedha Uniyal. An approximation algorithm for uniform capacitated k-median problem with $(1 + \epsilon)$ capacity violation. In *Proceedings of the 18th International Conference on Integer Programming and Combinatorial Optimization (IPCO'16)*, volume 9682 of *LNCS*, pages 262–274, New York, NY, USA, 2016. Springer-Verlag New York, Inc. `doi:10.1007/978-3-319-33461-5_22`.

6 Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS'99, pages 378–, Washington, DC, USA, 1999. IEEE Computer Society. URL: `http://dl.acm.org/citation.cfm?id=795665.796483`.

7 Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem (extended abstract). In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, STOC'99, pages 1–10, New York, NY, USA, 1999. ACM. `doi:10.1145/301250.301257`.

8 A. Fabián Chudak and P. David Williamson. Improved approximation algorithms for capacitated facility location problems. *Mathematical Programming*, 102(2):207–222, 2005. `doi:10.1007/s10107-004-0524-9`.

9 Julia Chuzhoy and Yuval Rabani. Approximating k-median with non-uniform capacities. In *In SODA'05*, pages 952–958, 2005.

10 Marek Cygan, MohammadTaghi Hajiaghayi, and Samir Khuller. Lp rounding for k-centers with non-uniform hard capacities. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 273–282. IEEE, 2012.

11 Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

12 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. `doi:10.1016/0304-3975(85)90224-5`.

13 Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

14 Wen-Lian Hsu and George L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979. `doi:10.1016/0166-218X(79)90044-1`.

15 Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, March 2001. `doi:10.1145/375827.375845`.

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

17 Madhukar R. Korupolu, C.Greg Plaxton, and Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. *Journal of Algorithms*, 37(1):146–188, 2000. `doi:10.1006/jagm.2000.1100`.

18 Shi Li. On uniform capacitated k-median beyond the natural lp relaxation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'15, pages 696–707, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics. URL: `http://dl.acm.org/citation.cfm?id=2722129.2722176`.

19 Martin Pal, Eva Tardos, and Tom Wexler. Facility location with nonuniform hard capacities. In *Proceedings of the 42nd IEEE Symposium on the Foundations of Computer Science*, pages 329–338, 2001.

20 Qingzhou Wang and Kam Hoi Cheng. A heuristic algorithm for the k-center problem with vertex weight. In *Proceedings of the International Symposium on Algorithms*, SIGAL'90, pages 388–396, New York, NY, USA, 1990. Springer-Verlag New York, Inc. URL: `http://dl.acm.org/citation.cfm?id=88307.88459`.

A Hardness of unweighted capacitated k-center problem

The hardness result for the capacitated k-center problem by Cygan *et al.* in [10] also holds for bounded capacities. The reduction in [10] uses the EC3S problem in which there is no bound on the number of sets in $\mathcal{F}$ an element of $\mathcal{U}$ may belong to. This requires L to be

$\Theta(|\mathcal{F}|)$. A different version of the EC3S problem in which each element of $\mathcal{U}$ can belong to at most three sets in $\mathcal{F}$ is also NP-complete [11] [12]. We use the same reduction as in [10]. Figure 2 illustrates the gadgets used in the reduction. Each vertex has a uniform capacity of L and $k = |\mathcal{F}| + \frac{|\mathcal{U}|}{3}$. All edges are of unit length.

▶ **Lemma 13.** *If there exists a solution to the EC3S instance, then there exist a capacitated k-center solution with cost ≤ 1.*

Proof. Let $\mathcal{F}' \subseteq \mathcal{F}$ be the solution of the EC3S instance. Note that, $|\mathcal{F}'| = \frac{|\mathcal{U}|}{3}$. For each set $A \in \mathcal{F}$, place a center at the vertex x_A in the corresponding set gadget. For each set $A \in \mathcal{F}'$, place a center at the vertex A in the corresponding set gadget. Thus, the vertices in each set gadget g_A is served by the center at vertex x_A and the vertices in the element gadget of the elements in a set $A \in \mathcal{F}'$ are served by the center at A. The number of centers used is $|\mathcal{F}| + |\mathcal{F}'| = |\mathcal{F}| + \frac{|\mathcal{U}|}{3} = k$. ◀

Now consider a solution $\mathcal{S}$ of the capacitated k-center instance with cost < 2. Note that each center must serve L vertices as $|V| = kL$.

▶ **Lemma 14.** *$\mathcal{S}$ does not have a center in any of the element gadgets.*

Proof. Consider an element $a \in \mathcal{U}$. The vertices with distance < 2 to a vertex in the element gadget g_a are the vertex itself and the vertices x_A for each set $A \in \mathcal{F}$ that it belongs to. Since, each element can belong to at most three sets in $\mathcal{F}$, the number of vertices that can be assigned to a center in an element gadget is bounded by a constant. For sufficiently large but constant L, the center will not be able to serve L vertices. ◀

▶ **Lemma 15.** *Each set gadget in $\mathcal{S}$ has at least one and at most two open centers in it.*

Proof. Consider the vertex set g_A of the set gadget for a set $A \in \mathcal{F}$. The $L-2$ pendant vertices in g_a cannot be served by a center outside g_a. Thus, g_A has at least one center in it.

Assume, for contradiction, that the number of centers in $g_A > 2$. Let a, b and c be the elements in set A and let g_a, g_b and g_c be the vertex sets of their respective gadgets. The vertices that are at a distance < 2 from some vertex g_A are the ones in g_A, g_a, g_b, g_c. Thus, the number of vertices that the centers in g_A can serve $\leq |g_A| + |g_a| + |g_b| + |g_c| = 2L$. Therefore, at least one of the centers in g_A must be serving less than L vertices. ◀

▶ **Lemma 16.** *In $\mathcal{S}$, gadgets corresponding to any two sets in $\mathcal{F}$ sharing a common element cannot have two open centers in each one of them.*

Proof. Assume, for contradiction, that there exist two sets $A, B \in \mathcal{F}$ having at least one element in common such that the vertex sets g_A and g_B of the corresponding gadgets each have 2 centers. As shown in the proof of Lemma 15, the vertices that can be assigned to a center in the gadget of a set $C = \{d, e, f\}$ are those in $g_C \cup g_d \cup g_e \cup g_f$. Thus, the number of vertices that can be assigned to centers in g_A and $g_B \leq 4L - \frac{L}{3}$ (since at least one element is common in A and B). Therefore, at least one of the centers in g_A or g_B must be serving less than L vertices. ◀

▶ **Lemma 17.** *If there exists a capacitated k-center solution with cost $R < 2$, then there exists a solution to the EC3S instance.*

Proof. From Lemmas 14, 15 and 16, there are $\frac{|\mathcal{U}|}{3}$ set gadgets each of which have 2 centers and the corresponding sets are all disjoint. These $\frac{|\mathcal{U}|}{3}$ sets form the solution set $\mathcal{F}'$. ◀

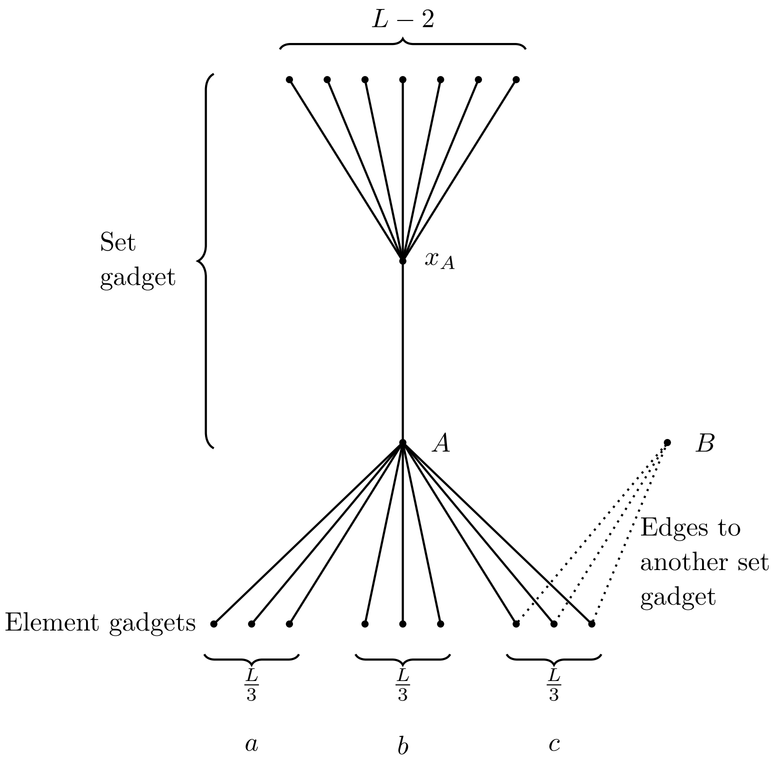

Figure 2 Gadgets for reduction (for the capacitated k-center problem).

▶ **Theorem 18.** *An α-approximation is not possible for the uniform capacitated k-center problem for $\alpha < 2$, unless* $\mathsf{P} = \mathsf{NP}$.

Proof. Follows from Lemmas 13 and 17. ◀

▶ Remark. Using the same reduction and allowing capacities of L at vertices x_A and A in g_A for each set $A \in \mathcal{F}$ and a capacity of zero at every other vertex, it can be shown that the $\{0, L\}$-capacitated k-center problem is hard to approximate within a factor of $(3 - \epsilon)$ for a constant L.