A

On Uniform Capacitated k-Median Beyond the Natural LP Relaxation

Shi Li, Department of Computer Science and Engineering, University at Buffalo

In this paper, we study the uniform capacitated k-median problem. In the problem, we are given a set $\mathcal{F}$ of potential facility locations, a set $\mathcal{C}$ of clients, a metric d over $\mathcal{F} \cup \mathcal{C}$, an upper bound k on the number of facilities we can open and an upper bound u on the number of clients each facility can serve. We need to open a subset $\mathcal{S} \subseteq \mathcal{F}$ of k facilities and connect clients in $\mathcal{C}$ to facilities in $\mathcal{S}$ so that each facility is connected by at most u clients. The goal is to minimize the total connection cost over all clients. Obtaining a constant approximation algorithm for this problem is a notorious open problem; most previous works gave constant approximations by either violating the capacity constraints or the cardinality constraint. Notably, all these algorithms are based on the natural LP-relaxation for the problem. The LP-relaxation has unbounded integrality gap, even when we are allowed to violate the capacity constraints or the cardinality constraint by a factor of $2 - \epsilon$.

Our result is an $\exp(O(1/\epsilon^2))$-approximation algorithm for the problem that violates the cardinality constraint by a factor of $1 + \epsilon$. That is, we find a solution that opens at most $(1 + \epsilon)k$ facilities whose cost is at most $\exp(O(1/\epsilon^2))$ times the optimum solution when at most k facilities can be open. This is already beyond the capability of the natural LP relaxation, as it has unbounded integrality gap even if we are allowed to open $(2 - \epsilon)k$ facilities. Indeed, our result is based on a novel LP for this problem. We hope that this LP is the first step towards a constant approximation for capacitated k-median.

The version as we described is the hard-capacitated version of the problem, as we can only open one facility at each location. This is as opposed to the soft-capacitated version, in which we are allowed to open more than one facilities at each location. The hard-capacitated version is more general, since one can convert a soft-capacitated instance to a hard-capacitated instance by making enough copies of each facility location. We give a simple proof that in the uniform capacitated case, the soft-capacitated version and the hard-capacitated version are actually equivalent, up to a small constant loss in the approximation ratio.

CCS Concepts:•**Theory of computation** → **Design and analysis of algorithms; Approximation algorithms analysis; Facility location and clustering;**

Additional Key Words and Phrases: theory, approximation algorithms, capacitated k-median, pseudo-approximation

1. INTRODUCTION

In the uniform capacitated k-median (CKM) problem, we are given a set $\mathcal{F}$ of potential facility locations, a set $\mathcal{C}$ of clients, a metric d over $\mathcal{F} \cup \mathcal{C}$, an upper bound k on the number of facilities we can open and an upper bound u on the number of clients each facility can serve. The goal is to find a set $\mathcal{S} \subseteq \mathcal{F}$ of at most k open facilities and a connection assignment $\sigma : \mathcal{C} \to \mathcal{S}$ of clients to open facilities such that $\left|\sigma^{-1}(i)\right| \leq u$ for every facility $i \in \mathcal{S}$, so as to minimize the connection cost $\sum_{j \in \mathcal{C}} d(j, \sigma(j))$.

When $u = \infty$, the problem becomes the classical NP-hard k-median (KM) problem. There has been extensive work on approximation algorithms for k-median. The first constant approximation, due to Charikar et al. [Charikar et al. 1999], is an LP-based $6\frac{2}{3}$-approximation. This factor was improved by a sequence of papers [Jain and Vazirani 2001], [Charikar and Guha 1999], [Jain et al. 2002], [Arya et al. 2001], [Li and Svensson 2013]. In particular, Li and Svensson [Li and Svensson 2013] gave a $1 + \sqrt{3} + \epsilon \approx 2.732 + \epsilon$-approximation for k-median, improving the previous decade-old ratio of $3 + \epsilon$ due to [Arya et al. 2001]. Their algorithm is based on a psudo-approximation algorithm that opens $k + O(1)$ facilities, and a process that turns a pseudo-approximation into a true approximation. Based on this framework, Byrka et al. [Byrka et al.] improved the approximation ratio from $2.732 + \epsilon$ to the current best

$2.674+\epsilon$ very recently[1]. On the negative side, it is NP-hard to approximate the problem within a factor of $1 + 2/e - \epsilon \approx 1.736$ [Jain et al. 2002].

Little is known about the uniform CKM problem; all constant approximation algorithms are pseudo-approximation algorithms, which produce solutions that violate either the capacity constraints or the cardinality constraint (the constraint that at most k facilities are open). Charikar et al. [Charikar et al. 1999] obtained a 16-approximation for the problem, by violating the capacity constraint by a factor of 3. Later, Chuzhoy and Rabani [Chuzhoy and Rabani 2005] gave a 40-approximation with capacity violation 50, for the more general *non-uniform* capacitated k-median, where different facilities can have different capacities. Recently, Byrka et al. [Byrka et al.] improved the capacity violation constant 3 of [Charikar et al. 1999] for uniform CKM to $2 + \epsilon$ and achieved approximation ratio of $O(1/\epsilon^2)$. This factor was improved to $O(1/\epsilon)$ by Li [Li 2014]. Constant approximations for CKM can also be achieved by violating the cardinality constraint. Gijswijt and Li [Gijswijt and Li 2013] designed a $(7 + \epsilon)$-approximation algorithm for a more general version of CKM that opens $2k+1$ facilities.

There are two slightly different versions of the (uniform or non-uniform) CKM problem. In the version as we described, we can open at most one facility at each location. This is sometimes called hard CKM. This is as opposed to soft CKM, where we can open more than one facilities at each location. Notice that hard CKM is more general as one can convert a soft CKM instance to a hard CKM instance by making enough copies of each location. The result of Chuzhoy and Rabani [Chuzhoy and Rabani 2005] is for soft CKM while the other mentioned results are for (uniform or non-uniform) hard CKM.

Most previous approximation algorithms on CKM are based on the basic LP relaxation. A simple example shows that the LP has unbounded gap. This is the main barrier to a constant approximation for CKM. Moreover, the integrality gap is unbounded even if we are allowed to violate the cardinality constraint or the capacity constraint by a factor of $2 - \epsilon$. Thus, for algorithms based on the basic LP relaxation, [Li 2014] and [Gijswijt and Li 2013] almost gave the smallest capacity violation factor and cardinality violation factor, respectively.

Closely related to KM and CKM are the uncapacitated facility location (UFL) and capacitated facility location (CFL) problems. UFL has similar inputs as KM but instead of giving an upper bound k on the number of facilities we can open, it specifies an opening cost f_i for each facility $i \in \mathcal{F}$. The objective is the sum of the cost for opening facilities and the total connection cost. In CFL, every facility $i \in \mathcal{F}$ has a capacity u_i on the maximum number of clients it can serve. There has been a steady stream of papers giving constant approximations for UFL [Lin and Vitter 1992; Shmoys et al. 1997; Jain and Vazirani 2001; Chudak and Shmoys 2004; Korupolu et al. 1998; Charikar and Guha 1999; Jain et al. 2003; Jain et al. 2002; Mahdian et al. 2006; Byrka 2007]. The current best approximation ratio for UFL is 1.488 due to Li [Li 2011], while the hardness of approximation is 1.463 [Guha and Khuller 1998].

In contrast to CKM, constant approximations are known for CFL. Mahdian et al. [Mahdian et al. 2006] gave a 2-approximation for soft CFL. For uniform hard CFL, Korupolu et al. [Korupolu et al. 1998] gave an $(8+\epsilon)$-approximation, which was improved to $6 + \epsilon$ by Chudak and Williamson [Chudak and Williamson 2005] and to 3 by Aggarwal et al. [Aggarwal et al. 2010]. For (non-uniform) hard CFL, the best approximation ratio is 5 due to Bansal et al. [Bansal et al. 2012], which improves the ratio of $3 + 2\sqrt{2}$ by Zhang et al. [Zhang et al. 2005]. All these algorithms for hard CFL are based on local search. Recently, An et al. gave an LP-based constant approximation algorithm for hard CFL [An et al.], solving a long-standing open problem [Williamson and D 2011].

[1][Byrka et al.] claims an approximation ratio of 2.611, but the authors mentioned in [Byrka et al. 2015] that the result of [Byrka et al.] was incorrect.

Our contributions. In this paper, we introduce a novel LP for uniform CKM, that we call the *rectangle LP*. We give a rounding algorithm that achieves constant approximation for the problem, by only violating the cardinality constraint by a factor of $1+\epsilon$, for any constant $\epsilon > 0$. This is already beyond the approximability of the basic LP relaxation, as it has unbounded integrality gap even if we are allowed to violate the cardinality constraint by $2-\epsilon$. To be more specific, we prove

THEOREM 1.1. *Given a uniform capacitated k-median instance and a constant $\epsilon > 0$, we can find in polynomial time a solution with at most $\lceil (1+\epsilon)k \rceil$ open facilities and total connection cost at most $\exp(O(1/\epsilon^2))$ times the cost of the optimum solution with k open facilities.*

The running time of our algorithm is $n^{O(1)}$, where the constant in the exponent does not depend on ϵ. If we allow the running time to be $n^{O(1/\epsilon)}$, we can remove the ceiling in the number of open facilities: we can handle the case when $k \leq O(1/\epsilon)$ by enumerating the k open facilities. As our LP overcomes the gap instance for the basic LP relaxation, we hope it is the first step towards a constant approximation for capacitated k-median.[2]

Our algorithm is for the hard capacitated version of the problem; namely, we open at most one facility at each location. Indeed, we show that with uniform capacities, the hard capacitated version is equivalent to the soft capacitated version, up to a constant loss in the approximation ratio. Moreover, we can assume $\mathcal{F} = \mathcal{C}$; this was implicitly proved in [Li 2014].

THEOREM 1.2. *Let $(k, u, \mathcal{F}, \mathcal{C}, d)$ be a hard uniform CKM instance, and C be the minimum connection cost of the instance when all facilities in $\mathcal{F}$ are open.*[3] *Then, given any solution of cost C' to the soft uniform CKM instance $(k, u, \mathcal{C}, \mathcal{C}, d)$, we can find a solution of cost at most $C + 2C'$ to the hard uniform CKM instance $(k, u, \mathcal{F}, \mathcal{C}, d)$.*

C is a trivial lower bound on the cost of the hard uniform CKM instance $(k, u, \mathcal{F}, \mathcal{C}, d)$. Moreover, the optimum cost of the soft uniform CKM instance $(k, u, \mathcal{C}, \mathcal{C}, d)$ is at most twice the optimum cost of the hard uniform CKM instance $(k, u, \mathcal{F}, \mathcal{C}, d)$. Thus, any α-approximation for the soft instance $(k, u, \mathcal{C}, \mathcal{C}, d)$ implies a $1 + 2(2\alpha) = (1 + 4\alpha)$-approximation for the hard instance $(k, u, \mathcal{F}, \mathcal{C}, d)$. The reduction works even if we are considering pseudo-approximation algorithms by allowing violating the cardinality constraint by $\beta \geq 1$ and the capacity constraint by $\gamma \geq 1$; we can simply apply the above theorem to the instance $(\lfloor \beta k \rfloor, \lfloor \gamma u \rfloor, \mathcal{F}, \mathcal{C}, d)$. Thus, we only focus on soft uniform CKM instances with $\mathcal{F} = \mathcal{C}$ in the paper. Though we have $\mathcal{F} = \mathcal{C}$, we keep both notions to indicate whether the points are treated as facilities or clients. Most parts of our algorithm work without assuming $\mathcal{F} = \mathcal{C}$; only a single step uses this assumption.

The remaining part of the paper is organized as follows. In Section 2, we introduce some useful notations, the basic LP relaxation for uniform CKM, the gap instance and the proof of Theorem 1.2. In Section 3, we describe our rectangle LP. Then in Section 4, we show how to round a fractional solution obtained from the rectangle LP. Then in Section 5, we show that the rectangle LP is not sufficient to obtain a true $O(1)$-approximation for the uniform CKM problem, by giving an $\Omega(\log n)$-integrality gap for the LP.

[2]Very recently, the same author in [Li 2016] has extended this result from uniform CKM to non-uniform CKM.

[3]Given the set of open facilities, finding the best connection assignment is a minimum cost bipartite matching problem and thus can be solved efficiently.

2. PRELIMINARIES

Let $\mathbb{Z}, \mathbb{Z}_{\geq 0}, \mathbb{R}$ and $\mathbb{R}_{\geq 0}$ denote the set of integers, non-negative integers, real numbers and non-negative real numbers respectively. For any $x \in \mathbb{R}_{\geq 0}$, let $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the floor and ceiling of x respectively. Let $\lfloor x \rangle = x - \lfloor x \rfloor$ and $\lceil x \rangle = \lceil x \rceil - x$.

Given two sets $\mathcal{C}', \mathcal{C}'' \subseteq \mathcal{C}$ of points, define $d(\mathcal{C}', \mathcal{C}'') = \min_{j \in \mathcal{C}', j' \in \mathcal{C}''} d(j, j')$ be the minimum distance from points in $\mathcal{C}'$ to points in $\mathcal{C}''$. We simply use $d(j, \mathcal{C}'')$ for $d(\{j\}, \mathcal{C}'')$.

Following is the basic LP for the uniform CKM problem:

$$\min \quad \sum_{i \in \mathcal{F}, j \in \mathcal{C}} d(i,j) x_{i,j} \quad \text{s.t.} \qquad \text{(Basic LP)}$$

$$\sum_{i \in \mathcal{F}} y_i \leq k, \qquad (1)$$

$$\sum_{i \in \mathcal{F}} x_{i,j} = 1, \qquad \forall j \in \mathcal{C}, \qquad (2)$$

$$x_{i,j} \leq y_i, \qquad \forall i \in \mathcal{F}, j \in \mathcal{C}, \qquad (3)$$

$$\sum_{j \in \mathcal{C}} x_{i,j} \leq u y_i, \qquad \forall i \in \mathcal{F}, \qquad (4)$$

$$x_{i,j}, y_i \geq 0, \qquad \forall i \in \mathcal{F}, j \in \mathcal{C}. \qquad (5)$$

In the above LP, y_i is the number of open facilities at location i, and $x_{i,j}$ indicates whether a client j is connected to a facility at i. Constraint (1) says that we can open at most k facilities, Constraint (2) says that every client must be connected to a facility, Constraint (3) says that a client can only be connected to an open facility and Constraint (4) is the capacity constraint. In the integer programming capturing the problem, we require $y_i \in \mathbb{Z}_{\geq 0}$ and $x_{i,j} \in \{0,1\}$ for every $i \in \mathcal{F}, j \in \mathcal{C}$. In the LP relaxation, we relax the constraint to $x_{i,j} \geq 0, y_i \geq 0$.

The basic LP has unbounded integrality gap, even if we are allowed to open $(2-\epsilon)k$ facilities. The gap instance is the following. Let $k = u+1$ and $n = |\mathcal{F}| = |\mathcal{C}| = u(u+1)$. The n points are partitioned into u groups, each containing $u+1$ points. Two points in the same group have distance 0 and two points in different groups have distance 1. The following LP solution has cost 0: $y_i = 1/u$ for every $i \in \mathcal{F}$ and $x_{i,j}$ is $1/(u+1)$ if i is co-located with j and 0 otherwise. The optimum solution is non-zero even if we are allowed to open $2u - 1 = 2k - 3$ facilities: there must be a group in which we open at most 1 facility and some client in the group must connect to a facility outside the group.[4]

2.1. Reduction to Soft Capacitated Case: Proof of Theorem 1.2

PROOF PROOF OF THEOREM 1.2. We construct two matchings. First, there is a matching of cost C between $\mathcal{F}$ and $\mathcal{C}$ (the cost of matching $i \in \mathcal{F}$ to $j \in \mathcal{C}$ is $d(i,j)$), where each facility in $\mathcal{F}$ is matched at most u times and each client in $\mathcal{C}$ is matched exactly once. The second matching is from the solution for the soft uniform CKM instance $(k, u, \mathcal{C}, \mathcal{C}, d)$. We construct a set $\mathcal{S}$ of size at most k as follows. Suppose we opened s facilities at some location $j \in \mathcal{C}$, we add s facility locations collocated with j to $\mathcal{S}$. So there is a matching of cost C' between $\mathcal{C}$ and $\mathcal{S}$, where each client in $\mathcal{C}$ is matched exactly once and each facility $i \in \mathcal{S}$ is matched t_i times, for some $t_i \leq u$.

By concatenating the two matchings and by triangle inequalities, we obtain a matching M between $\mathcal{F}$ and $\mathcal{S}$ of cost at most $C + C'$, such that every facility in $\mathcal{F}$ is matched at most u times and every facility in $i \in \mathcal{S}$ is matched t_i times. We can apply two operations to M repeatedly, which can only decrease the cost of the matching. When no operations can be performed, we are guaranteed that at most $|\mathcal{S}| \leq k$ facilities in $\mathcal{F}$ are matched. See Figure 1 for illustrations.

[4]Note that this gap instance is not bad when we are allowed to violate the capacity constraints by $1 + \epsilon$. However, if we are only allowed to violate the capacity constraints, there is a different bad instance: each group has $2u - 1$ clients and $k = 2u - 1$. Fractionally, we open $2 - 1/u$ facilities in each group and the cost is 0. But if we want to open $2u - 1$ facilities integrally, some group contains at most 1 facility and thus the capacity violation factor has to be $2 - 1/u$.

The first operation tries to break cycles in M. Since M is a bipartite matching, a cycle has an even length. We color the edges in the cycle alternatively in black and white. Assume w.l.o.g the total length of black edges is at most that of white edges. Then, we can increase the multiplicities of black edges by one and decrease the multiplicities of white edges by one. This does not increase the cost of M. We can apply this operation until the multiplicity of some white edge becomes 0. By applying the operation repeatedly, we can assume the edges in M form a forest, when we ignore multiplicities.

If there is a path of edges M, connecting two vertices $i, i' \in \mathcal{F}$, both of which are matched less than u times, then we can apply the second operation. We color the edges in the path alternatively in black and white. Assume the total length of black edges is at most that of white edges. We increase the multiplicities of black edges and decrease the multiplicities of white edges. We can apply this operation until either the multiplicity of some white edge becomes 0, or either i or i' is matched u times. Thus, by applying the second operation repeatedly, we can assume that in any tree of the forest formed by edges in M, at most one facility in $\mathcal{F}$ is matched less than u times.

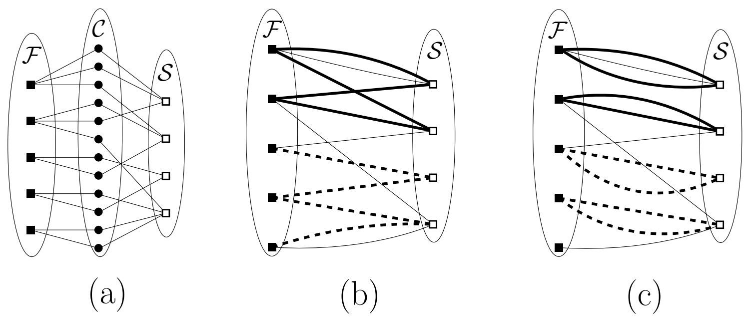

Fig. 1. Construction of matching between $\mathcal{F}$ and $\mathcal{S}$. (a) shows two matchings, one between $\mathcal{F}$ and $\mathcal{C}$ and the other between $\mathcal{C}$ and $\mathcal{S}$. (b) shows the concatenated matching M between $\mathcal{F}$ and $\mathcal{S}$. The thick lines form a cycle. The thick dashed lines form a path, whose two end points are vertices in $\mathcal{F}$ with degree less than u. (c) shows the matching M after we apply the first operation to the cycle and the second opertaion to the path.

Now we claim that at most k facilities in $\mathcal{F}$ are matched. To see this, focus on each tree in the forest containing at least one edge. If facilities in $\mathcal{S}$ in the tree are matched t times in total, so are the facilities in $\mathcal{F}$ in the tree. Thus, there are exactly $\lceil t/u \rceil$ facilities in $\mathcal{F}$ in this tree, since at most one facility in $\mathcal{F}$ in the tree is matched less than u times. The number of facilities in $\mathcal{S}$ in this tree is at least $\lceil t/u \rceil$ since each facility in $\mathcal{S}$ is matched $t_i \le u$ times. This proves the claim.

Let $\mathcal{F}' \subseteq \mathcal{F}$ be the set of facilities that are matched. Then, $|\mathcal{F}'| \le |\mathcal{S}| \le k$, and we have a matching between $\mathcal{F}'$ and $\mathcal{S}$ of cost at most $C + C'$, where each facility in $\mathcal{F}'$ is matched at most u times and each facility in $\mathcal{S}$ is matched t_i times. By concatenating this matching with the matching between $\mathcal{S}$ and $\mathcal{C}$ of cost C', we obtain a solution of cost $C + 2C'$ with open facilities $\mathcal{F}'$ to the uniform hard CKM instance $(k, u, \mathcal{F}, \mathcal{C}, d)$. This finishes the proof. $\square$

3. RECTANGLE LP

Our rectangle LP is motivated by the gap instance described in Section 2. Focus on a group of $u + 1$ clients in the gap instance. The fractional solution opens $1 + 1/u$ facilities for this group and use them to serve the $u(1 + 1/u) = u + 1$ clients in the group. We interpret this fractional event as a convex combination of integral events: with probability $1 - 1/u$ we open 1 facility for the group and serve u clients; with probability $1/u$ we open 2 facilities and serve $2u$ clients. However, there are only $u + 1$

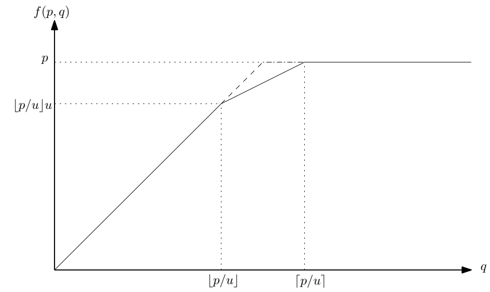

Fig. 2. The f function for a fixed p such that $p/u \notin \mathbb{Z}$. It contains three linear segments. The dashed line shows the function f defined by $f = \min\{p, qu\}$.

clients in this group; even if 2 facilities are open, we can only serve $u+1$ clients. Thus, we can only serve $(1-1/u)u + (1/u)(u+1) = u + 1/u < u+1$ clients using $1+1/u$ open facilities.

This motivates the following definition of $f(p,q)$ for any $p \in \mathbb{Z}_{\geq 0}, q \in \mathbb{R}_{\geq 0}$. When $q \in \mathbb{Z}_{\geq 0}$, let $f(p,q) = \min\{qu, p\}$ be the upper bound on the number of clients in a set of cardinality p that can be connected to a set of q facilities. We then extend the range of q from $\mathbb{Z}_{\geq 0}$ to $\mathbb{R}_{\geq 0}$ using linear interpolation(see Figure 2). Then the exact definition of $f(p,q)$ is the following:

$$f(p,q) = \begin{cases} qu & \text{if } q \leq \left\lfloor \frac{p}{u} \right\rfloor \\ u\left\lfloor \frac{p}{u} \right\rfloor + u\left\lfloor \frac{p}{u} \right\rangle \left(q - \left\lfloor \frac{p}{u} \right\rfloor\right) & \text{if } \left\lfloor \frac{p}{u} \right\rfloor < q < \left\lceil \frac{p}{u} \right\rceil \\ p & \text{if } q \geq \left\lceil \frac{p}{u} \right\rceil \end{cases}.$$

CLAIM 3.1. *Fixing $p \in \mathbb{Z}_{\geq 0}$, $f(p,\cdot)$ is a concave function on $\mathbb{R}_{\geq 0}$. Fixing $q \in \mathbb{R}_{\geq 0}$, $f(\cdot,q)$ is a concave function on $\mathbb{Z}_{\geq 0}$.*

PROOF. It is easy to see that $f(p,q) = \min\{p, uq, u\lfloor p/u \rfloor + u\lfloor p/u \rangle (q - \lfloor p/u \rfloor)\}$. Fix p, all the three terms are linear functions of q; thus the minimum of the three is concave.

Now we fix $q \in \mathbb{R}_{\geq 0}$. Then $f(p,q) = p$ if $p \leq u\lfloor q \rfloor$, $f(p,q) = u\lfloor q \rfloor + (p - u\lfloor q \rfloor)\lfloor q \rangle$ if $u\lfloor q \rfloor < p < u\lceil q \rceil$, and $f(p,q) = uq$ if $p \geq u\lceil q \rceil$. All three segments are linear on p and their gradients are $1, \lfloor q \rangle, 0$ respectively. The gradients are decreasing from left to right. Moreover, the first segment and the second segment agree on $p = u\lfloor q \rfloor$; the second segment and the third segment agree on $p = u\lceil q \rceil$. Thus, $f(\cdot,q)$ is a concave function on $\mathbb{Z}_{\geq 0}$. $\square$

For any subset $\mathcal{B} \subseteq \mathcal{F}$ of facility locations and subset $\mathcal{J} \subseteq \mathcal{C}$ of clients, define $y_{\mathcal{B}} := y(\mathcal{B}) := \sum_{i \in \mathcal{B}} y_i$ and $x_{\mathcal{B},\mathcal{J}} = \sum_{i \in \mathcal{B}, j \in \mathcal{J}} x_{i,j}$. We simply write $x_{i,\mathcal{J}}$ for $x_{\{i\},\mathcal{J}}$ and $x_{\mathcal{B},j}$ for $x_{\mathcal{B},\{j\}}$. By the definition of $f(p,q)$, $\sum_{j \in \mathcal{J}} x_{\mathcal{B},j} \leq f(|\mathcal{J}|, y_{\mathcal{B}})$ is valid for every $\mathcal{B} \subseteq \mathcal{F}$ and $\mathcal{J} \subseteq \mathcal{C}$. The constraint says that there can be at most $f(|\mathcal{J}|, y_{\mathcal{B}})$ clients in $\mathcal{J}$ connected to facilities in $\mathcal{B}$. Notice the constraint with $\mathcal{B} = \{i\}$ and $\mathcal{J} = \{j\}$ implies $x_{i,j} \leq f(1, y_i) \leq y_i$. The constraint with $\mathcal{B} = \{i\}$ and $\mathcal{J} = \mathcal{C}$ implies $\sum_{j \in \mathcal{C}} x_{i,j} \leq f(|\mathcal{C}|, y_i) \leq uy_i$. Thus, Constraint (3) and (4) are implied. The constraints of our rectangle LP are Constraint (1),(2),(5) and the new constraints:

$$\text{minimize} \sum_{i \in \mathcal{F}, j \in \mathcal{C}} x_{i,j} d(i,j) \text{ s.t.} \qquad \text{(Rectangle LP)}$$

$$x_{\mathcal{F},j} = 1,\ y_{\mathcal{F}} \leq k,\ x_{i,j} \geq 0,\ y_i \geq 0, \qquad \forall i \in \mathcal{F}, j \in \mathcal{C}, \tag{6}$$

$$x_{\mathcal{B},\mathcal{J}} \leq f(|\mathcal{J}|, y_{\mathcal{B}}), \qquad \forall \mathcal{B} \subseteq \mathcal{F}, \mathcal{J} \subseteq \mathcal{C}. \tag{7}$$

The LP is called the rectangle LP since we have a constraint for every "rectangle" $(\mathcal{B} \subseteq \mathcal{F}, \mathcal{J} \subseteq \mathcal{C})$. We use the concavity of $f(p,\cdot)$ to convert Constraint (7) to linear constraints. Since $f(p,q)$ is the minimum of p, uq and $u\lfloor p/u\rfloor + u\lfloor p/u\rangle(q - \lfloor p/u\rfloor)$, Constraint (7) is equivalent to a combination of three linear constraints.

For a fixed $\mathcal{B} \subseteq \mathcal{F}$, the separation oracle for Constraint (7) is simple: for every $p \in [|\mathcal{C}|]$, we take the sum of the p largest values in $\{x_{\mathcal{B},j} : j \in \mathcal{C}\}$; if it is larger than $f(p, y_{\mathcal{B}})$ we find a separation. Since there are exponential number of sets $\mathcal{B}$, we do not know how to find a separation oracle for the Constraint (7) efficiently. However, we can use the following standard trick: given $\{x_{i,j} : i \in \mathcal{F}, j \in \mathcal{C}\}$ and $\{y_i : i \in \mathcal{F}\}$ satisfying Constraint (6), we either find a rectangle $(\mathcal{B} \subseteq \mathcal{F}, \mathcal{J} \subseteq \mathcal{C})$ for which Constraint (7) is violated, or construct an integral solution with at most $\lceil(1+\epsilon)k\rceil$ facilities and the desired approximation ratio. This is sufficient for us to run the ellipsoid method.

We also remark that the definition of $f(p,q)$ for $\lfloor p/u\rfloor < q < \lceil p/u\rceil$ is what makes the rectangle LP powerful. If we change the definition of $f(p,q)$ to $f(p,q) = \min\{p, uq\}$(see Figure 2), then the rectangle LP is equivalent to the basic LP.

4. ROUNDING A FRACTIONAL SOLUTION OF THE RECTANGLE LP

Throughout this section, let $\big(\{x_{i,j} : i \in \mathcal{F}, j \in \mathcal{C}\}, \{y_i : i \in \mathcal{F}\}\big)$ be a fractional solution satisfying Constraints (6). Let $\mathsf{LP} := \sum_{i\in\mathcal{F}, j\in\mathcal{C}} x_{i,j} d(i,j)$ be the cost of the fractional solution. We then try to round the fractional solution to an integral one with at most $\lceil(1+\epsilon)k\rceil$ open facilities. We either claim the constructed integral solution has connection cost at most $\exp(O(1/\epsilon^2))\mathsf{LP}$, or output a rectangle $(\mathcal{B} \subseteq \mathcal{F}, \mathcal{J} \subseteq \mathcal{C})$ for which Constraint (7) is violated. We can assume Constraint (3) and (4) are satisfied by checking Constraint (7) for rectangles $(\{i\}, \{j\})$ and $(\{i\}, \mathcal{C})$ respectively.

Overall, the algorithm works as follows. Initially, we have 1 unit of demand at each client $j \in \mathcal{C}$. During the execution of the algorithm, we move demands fractionally between clients. We pay a cost of $\alpha d(j,j')$ for moving α units of demand from client j to client j'. Suppose finally our moving cost is C, and each client $j \in \mathcal{C}$ has α_j units of demand. Then we use the fact that $\mathcal{F} = \mathcal{C}$: we open $\lceil \alpha_j/u\rceil$ facilities at the location $j \in \mathcal{C} = \mathcal{F}$. By the integrality of matching, there is an integral matching between the $\mathcal{F}$ and $\mathcal{C}$, such that each $i \in \mathcal{F}$ is matched at most $u\lceil\alpha_i/u\rceil$ times and each $j \in \mathcal{C}$ is matched exactly once. The cost of the matching is at most C (cost of matching i and j is $d(i,j)$). Thus our goal is to bound C and $\sum_{j\in\mathcal{C}} \lceil\alpha_j/u\rceil$.

4.1. Moving Demands to Client Representatives

In this section, we define a subset of clients called *client representatives* (representatives for short) and move all demands to the representatives. The definition of client representatives is similar to that of Charikar and Shymoys [Chudak and Shmoys 2004].

Let $d_{\mathsf{av}}(j) = \sum_{i\in\mathcal{F}} x_{i,j} d(i,j)$ be the connection cost of j, for every client $j \in \mathcal{C}$. Then $\mathsf{LP} = \sum_{j\in\mathcal{C}} d_{\mathsf{av}}(j)$. Let $\ell = \Theta(1/\epsilon)$ be an integer whose value will be decided later. Let $\mathcal{C}^* = \emptyset$ initially. Repeat the following process until $\mathcal{C}$ becomes empty. We select the client $v \in \mathcal{C}$ with the smallest $d_{\mathsf{av}}(v)$ and add it to $\mathcal{C}^*$. We remove all clients j such that $d(j,v) \le 2\ell d_{\mathsf{av}}(j)$ from $\mathcal{C}$ (thus, v itself is removed). Then the final set $\mathcal{C}^*$ is the set of client representatives. We shall use v and its variants to index representatives, and j and its variants to index general clients.

We partition the set $\mathcal{F}$ of locations according to their nearest representatives in $\mathcal{C}^*$. Let $\mathcal{U}_v = \emptyset$ for every $v \in \mathcal{C}^*$ initially. For each location $i \in \mathcal{F}$, we add i to $\mathcal{U}_v$ for the $v \in \mathcal{C}^*$ that is closest to i. Thus, $\{\mathcal{U}_v : v \in \mathcal{C}^*\}$ forms a Voronoi diagram of $\mathcal{F}$ with centers being $\mathcal{C}^*$. For any subset $\mathcal{A} \subseteq \mathcal{C}^*$ of representatives, we use $\mathcal{U}_{\mathcal{A}} = \bigcup_{v\in\mathcal{A}} \mathcal{U}_v$ to denote the union of Voronoi regions with centers in $\mathcal{A}$.

CLAIM 4.1. *The following statements hold:*

(4.1a) *for all* $v, v' \in \mathcal{C}^*, v \neq v'$, *we have* $d(v, v') > 2\ell \max\{d_{\mathsf{av}}(v), d_{\mathsf{av}}(v')\}$;
(4.1b) *for all* $j \in \mathcal{C}$, *there exists* $v \in \mathcal{C}^*$, *such that* $d_{\mathsf{av}}(v) \leq d_{\mathsf{av}}(j)$ *and* $d(v, j) \leq 2\ell d_{\mathsf{av}}(j)$;
(4.1c) $y(\mathcal{U}_v) \geq 1 - 1/\ell$ *for every* $v \in \mathcal{C}^*$;
(4.1d) *for any* $v \in \mathcal{C}^*$, $i \in \mathcal{U}_v$ *and* $j \in \mathcal{C}$, *we have* $d(i, v) \leq d(i, j) + 2\ell d_{\mathsf{av}}(j)$.

PROOF. First consider Property (4.1a). Assume $d_{\mathsf{av}}(v) \leq d_{\mathsf{av}}(v')$. When we add v to $\mathcal{C}^*$, we remove all clients j satisfying $d(v, j) \leq 2\ell d_{\mathsf{av}}(j)$ from $\mathcal{C}$. Thus, v' can not be added to $\mathcal{C}^*$ later.

For Property (4.1b), just consider the iteration in which j is removed from $\mathcal{C}$. The representative v added to $\mathcal{C}^*$ in the iteration satisfy the property.

Then consider Property (4.1c). By Property (4.1a), we have $\mathcal{B} := \{i \in \mathcal{F} : d(i, v) \leq \ell d_{\mathsf{av}}(v)\} \subseteq \mathcal{U}_v$. Since $d_{\mathsf{av}}(v) = \sum_{i \in \mathcal{F}} x_{i,v} d(i, v)$ and $\sum_{i \in \mathcal{F}} x_{i,v} = 1$, we have $d_{\mathsf{av}}(v) \geq (1 - x_{\mathcal{B},v})\ell d_{\mathsf{av}}(v)$, implying $y(\mathcal{U}_v) \geq y_{\mathcal{B}} \geq x_{\mathcal{B},v} \geq 1 - \frac{1}{\ell}$, due to Constraint (3).

Finally, consider Property (4.1d). By Property (4.1b), there is a client $v' \in \mathcal{C}^*$ such that $d_{\mathsf{av}}(v') \leq d_{\mathsf{av}}(j)$ and $d(v', j) \leq 2\ell d_{\mathsf{av}}(j)$. Notice that $d(i, v) \leq d(i, v')$ since $v' \in \mathcal{C}^*$ and i was added to $\mathcal{U}_v$. Thus, $d(i, v) \leq d(i, v') \leq d(i, j) + d(j, v') \leq d(i, j) + 2\ell d_{\mathsf{av}}(j)$. □

Now, we move demands to $\mathcal{C}^*$. For every representative $v \in \mathcal{C}^*$, every location $i \in \mathcal{U}_v$ and every client $j \neq v$ such that $x_{i,j} > 0$, we move $x_{i,j}$ units of demand from j to v. We bound the moving cost:

LEMMA 4.2. *The total cost of moving demands in the above step is at most* $2(\ell+1)\mathsf{LP}$.

PROOF. The cost is bounded by

$$\sum_{v \in \mathcal{C}^*} \sum_{i \in \mathcal{U}_v} \sum_{j \in \mathcal{C}} x_{i,j}(d(j, i) + d(i, v)) \leq \sum_{v \in \mathcal{C}^*} \sum_{i \in \mathcal{U}_v} \sum_{j \in \mathcal{C}} x_{i,j}(2d(i, j) + 2\ell d_{\mathsf{av}}(j))$$

$$= 2 \sum_{j \in \mathcal{C}} \sum_{v \in \mathcal{C}^*, i \in \mathcal{U}_v} x_{i,j} \left(d(i, j) + \ell d_{\mathsf{av}}(j)\right) = 2 \sum_{j \in \mathcal{C}} \left(d_{\mathsf{av}}(j) + \ell d_{\mathsf{av}}(j)\right) = 2(\ell + 1)\mathsf{LP}.$$

The inequality is by Property (4.1d). The second equality used the fact that $\{\mathcal{U}_v : v \in \mathcal{C}^*\}$ form a partition of $\mathcal{F}$, $\sum_{i \in \mathcal{F}} x_{i,j} = 1$ and $\sum_{i \in \mathcal{F}} x_{i,j} d(i, j) = d_{\mathsf{av}}(j)$. □

After the moving operation, all demands are at the set $\mathcal{C}^*$ of representatives. Every representative $v \in \mathcal{C}^*$ has $\sum_{i \in \mathcal{U}_v} \sum_{j \in \mathcal{C}} x_{i,j} = x_{\mathcal{U}_v, \mathcal{C}}$ units of demand. Let $y'_i := x_{i,\mathcal{C}}/u$ for any facility location $i \in \mathcal{F}$. Since Constraint (4) holds, we have $y'_i \leq y_i$. Define $y'_{\mathcal{B}} := y'(\mathcal{B}) := \sum_{i \in \mathcal{B}} y'_i = x_{\mathcal{B},\mathcal{C}}/u$ for every $\mathcal{B} \subseteq \mathcal{F}$. Obviously $y'_{\mathcal{B}} \leq y_{\mathcal{B}}$. The amount of demand at $v \in \mathcal{C}^*$ is $x_{\mathcal{U}_v,\mathcal{C}} = uy'(\mathcal{U}_v)$.

So far we have obtained an $O(1)$ approximation with $2k$ open facilities if we set $\ell = 2$: we open $\lceil y'(\mathcal{U}_v) \rceil$ facilities at each location $v \in \mathcal{C}^* \subseteq \mathcal{C} = \mathcal{F}$. By Lemma 4.2, the connection cost is at most $2(\ell + 1)\mathsf{LP} = 6\mathsf{LP}$. The number of open facilities is at most $2k$, as $\max_{v \in \mathcal{C}^*} \frac{\lceil y'(\mathcal{U}_v) \rceil}{y(\mathcal{U}_v)} \leq \max_{y \geq 1 - 1/\ell} \frac{\lceil y \rceil}{y} \leq 2$. No matter how large ℓ is, the bound is tight as $\frac{\lceil 1+\epsilon \rceil}{1+\epsilon}$ approaches 2. This is as expected since we have not used Constraint (7). In order to improve the factor of 2, we shall further move demands among client representatives.

4.2. Bounding cost for moving demands out of a set

Suppose we are given a set $\mathcal{A} \subseteq \mathcal{C}^*$ of representatives such that $d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A})$ is large. If $\lceil y'(\mathcal{U}_{\mathcal{A}}) \rceil / y(\mathcal{U}_{\mathcal{A}})$ is large then we can not afford to open $\lceil y'(\mathcal{U}_{\mathcal{A}}) \rceil$ open facilities inside $\mathcal{A}$. (Recall that $\mathcal{U}_{\mathcal{A}} = \bigcup_{v \in \mathcal{A}} \mathcal{U}_v$ is the union of Voronoi regions with centers in $\mathcal{A}$.) Thus, we need to move demands between $\mathcal{A}$ and $\mathcal{C}^* \setminus \mathcal{A}$. The goal of this section is to bound $d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A})$; this requires Constraint (7) for $\mathcal{B} = \mathcal{U}_\mathcal{A}$.

To describe the main lemma, we need some notations. Let $D_i = \sum_{j \in \mathcal{C}} x_{i,j} d(i,j)$ and $D'_i = \sum_{j \in \mathcal{C}} x_{i,j} d_{\mathsf{av}}(j)$ for any location $i \in \mathcal{F}$. Let $D_{\mathcal{F}'} := D(\mathcal{F}') := \sum_{i \in \mathcal{F}'} D_i$ and $D'_{\mathcal{F}'} := D'(\mathcal{F}') := \sum_{i \in \mathcal{F}'} D'_i$ for every subset $\mathcal{F}' \subseteq \mathcal{F}$ of locations. It is easy to see that $\mathsf{LP} = D_\mathcal{F} = D'_\mathcal{F}$. With this fact, each facility i can afford to pay a cost that is comparable to $D_i + D'_i$.

LEMMA 4.3. *Let $\emptyset \subsetneq \mathcal{A} \subsetneq \mathcal{C}^*$ and $\mathcal{S} = \mathcal{U}_\mathcal{A}$. Suppose $y'_\mathcal{S} \geq \lfloor y_\mathcal{S} \rfloor$ and Constraint* (7) *holds for $\mathcal{B} = \mathcal{S}$ and every $\mathcal{J} \subseteq \mathcal{C}$. Then,*

$$u \lfloor y'_\mathcal{S} \rangle \lceil y_\mathcal{S} \rangle \, d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A}) \leq 4D_\mathcal{S} + (4\ell + 2) D'_\mathcal{S}.$$

Before proving the lemma, we explain why the bound is what we need. We can open $\lfloor y'_\mathcal{S} \rfloor$ facilities in $\mathcal{A}$ and move $u \lfloor y'_\mathcal{S} \rangle$ units of demand from $\mathcal{A}$ to representatives in $\mathcal{C}^* \setminus \mathcal{A}$. If we guarantee that the moving distance is comparable to $d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A})$, then the moving cost is comparable to $u \lfloor y'_\mathcal{S} \rangle \, d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A})$. When $\lceil y_\mathcal{S} \rangle$ is not too small, the cost is bounded in terms of $D_\mathcal{S} + D'_\mathcal{S}$. On the other hand, if $\lceil y_\mathcal{S} \rangle$ is very small, we can simply open $\lceil y_\mathcal{S} \rceil$ facilities in $\mathcal{A}$ as $\lceil y_\mathcal{S} \rceil / y_\mathcal{S}$ is close to 1.

We prove the following lemma and then show that it implies Lemma 4.3.

LEMMA 4.4. *Suppose* $(\{x_{i,j} : i \in \mathcal{F}, j \in \mathcal{C}\}, \{y_i : i \in \mathcal{F}\})$ *satisfies Constraint* (7) *for some set $\mathcal{B} \subseteq \mathcal{F}$ and every $\mathcal{J} \subseteq \mathcal{C}$. Moreover, suppose $y'_\mathcal{B} \geq \lfloor y_\mathcal{B} \rfloor$. Then*

$$\sum_{j \in \mathcal{C}} x_{\mathcal{B},j}(1 - x_{\mathcal{B},j}) \geq u \lfloor y'_\mathcal{B} \rangle \lceil y_\mathcal{B} \rangle. \tag{8}$$

PROOF. We first give an intuition behind the lemma. Let us assume $y'_\mathcal{B} = y_\mathcal{B} \notin \mathbb{Z}$ and $uy_\mathcal{B} \in \mathbb{Z}$. Thus, $\mathcal{B}$ serves $uy'_\mathcal{B} = uy_\mathcal{B}$ fractional clients. Without Constraint (7), it can happen that B serves $uy_\mathcal{B}$ integral clients, in which case the left side of Inequality (8) is 0. So, Inequality (8) prevents this case from happening. Indeed, we show that the left side of (8) is minimized when the following happens: $\mathcal{B}$ serves $u \lfloor y_\mathcal{B} \rfloor$ integral clients, and u fractional clients, each with fraction $\lfloor y_\mathcal{B} \rangle$. In this case, Inequality (8) holds with equality.

For simplicity we let $y = y_\mathcal{B}, y' = y'_\mathcal{B}$ and $x_j = x_{\mathcal{B},j}$ for every $j \in \mathcal{C}$. Throughout the proof, y and y' are fixed. We assume $\mathcal{C} = [n]$ and $1 \geq x_1 \geq x_2 \geq \cdots \geq x_n \geq 0$. Let $\bar{f}(p) = \min\{f(p, y), uy'\}$ for every integer $p \in [0, n]$. Notice that $\bar{f}$ is a non-decreasing concave function as $f(\cdot, y)$ is concave and uy' is independent of p. The conjunction of Constraint (7) and $y' = \sum_{j=1}^n x_j / u$ is equivalent to $\sum_{j=1}^p x_j \leq \bar{f}(p)$ for every $p \in [n]$.

Let $g : [0,1] \to \mathbb{R}$ be any second-order differentiable concave function such that $g(0) = 0$. We shall show that $\sum_{j=1}^n g(x_j) \geq \sum_{j=1}^n g(x^*_j)$, where $x^*_j = \bar{f}(j) - \bar{f}(j-1)$ for every $j \in [n]$.

We use g' and g'' to denote the first-order and second-order derivative functions of g respectively. For any $x \in [0,1]$, let $\psi(x) = |\{j \in \mathcal{C} : x_j \geq x\}|$. Then

$$\begin{aligned}\sum_{j \in \mathcal{C}} g(x_j) &= \int_0^1 \psi(x) g'(x) dx = \int_0^1 \left(g'(0) + \int_0^x g''(t) dt \right) \psi(x) dx \\ &= g'(0) \int_0^1 \psi(x) dx + \int_0^1 \left[\int_t^1 \psi(x) dx \right] g''(t) dt.\end{aligned}$$

Notice that the first term is equal to $g'(0) \sum_{j \in \mathcal{C}} x_j = g'(0) u y'$, which is independent of $\vec{x} := (x_1, x_2, \cdots, x_n)$. Since g is concave, we have $g''(t) \leq 0$ for every $t \in [0,1]$. We show that $Q(t) := \int_t^1 \psi(x)dx$ is maximized when $\vec{x} = \vec{x}^* := (x_1^*, x_2^*, \cdots, x_n^*)$, for every $t \in [0,1]$.

We now fix $t \in [0,1]$. Notice that $Q(t) = \sum_{j=1}^{p_t}(x_j - t)$ where p_t is the largest integer p such that $x_p \geq t$. Then $Q(t) \leq \bar{f}(p_t) - tp_t \leq \max_{p=0}^n \left(\bar{f}(p) - tp\right)$.

We show that $Q(t) = \max_{p=0}^n \left(\bar{f}(p) - tp\right)$ when $\vec{x} = \vec{x}^*$. Consider the sequence $x_1 - t, x_2 - t, \cdots, x_n - t$. The sequence is non-increasing; $\bar{f}(p) - tp$ is the sum of the first p number in the sequence by the definition of $\{x_j^*\}$. Thus, the sum is maximized when $\bar{f}(p) - tp$ is the largest number such that $x_p \geq t$. This p is exactly the definition of p_t. Thus, $Q(t)$ is maximized when $\vec{x} = \vec{x}^*$. This proves that $\sum_{i=1}^n g(x_i) \geq \sum_{i=1}^n g(x_i^*)$.

Now we let $g(x) \equiv x(1-x)$. Then $g(0) = g(1) = 0$. Thus, $\sum_{j \in \mathcal{C}} g(x_j^*) = \left\lfloor \frac{u\lfloor y'\rangle}{\lfloor y\rangle} \right\rfloor g\left(\lfloor y\rangle\right) + g\left(\left\lfloor \frac{u\lfloor y'\rangle}{\lfloor y\rangle} \right\rangle \lfloor y\rangle\right)$. By the concavity of g and $g(0) = 0$, we have $g\left(\left\lfloor \frac{u\lfloor y'\rangle}{\lfloor y\rangle} \right\rangle \lfloor y\rangle\right) \geq \left\lfloor \frac{u\lfloor y'\rangle}{\lfloor y\rangle} \right\rangle g\left(\lfloor y\rangle\right)$. Thus $\sum_{j\in\mathcal{C}} g(x_j) \geq \left(\left\lfloor \frac{u\lfloor y'\rangle}{\lfloor y\rangle} \right\rfloor + \left\lfloor \frac{u\lfloor y'\rangle}{\lfloor y\rangle} \right\rangle\right) g\left(\lfloor y\rangle\right) = \frac{u\lfloor y'\rangle}{\lfloor y\rangle} g(\lfloor y\rangle) = \frac{u\lfloor y'\rangle}{\lfloor y\rangle} \lfloor y\rangle\left(1 - \lfloor y\rangle\right) = u\lfloor y'\rangle \lceil y\rangle$. The last equation used the fact that $\lfloor y\rangle + \lceil y\rangle = 1$ if y is fractional and $\lfloor y'\rangle = 0$ if y is integral. □

PROOF **Proof of Lemma 4.3.** Focus on some $i \in \mathcal{S}, i' \in \mathcal{F} \setminus \mathcal{S}, j \in \mathcal{C}$. Suppose $i \in \mathcal{U}_v$ for some $v \in \mathcal{A}$ and $i' \in \mathcal{U}_{v'}$ for some $v' \in \mathcal{C}^* \setminus \mathcal{A}$. Then

$$\begin{aligned} d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A}) &\leq d(v, v') \leq d(v, i') + d(v', i') \leq 2d(v, i') \leq 2(d(v, i) + d(i, j) + d(j, i')) \\ &\leq 2[2d(i, j) + 2\ell d_{\mathsf{av}}(j) + d(i', j)]. \end{aligned}$$

In the above sequence, the third inequality used the fact that $i' \in \mathcal{U}_{v'}$ and the fifth inequality used Property (4.1d) in Claim 4.1. Thus,

$$\begin{aligned} & u\lfloor y'_{\mathcal{S}}\rangle \lceil y_{\mathcal{S}}\rangle\, d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A}) \leq \sum_{j \in \mathcal{C}} x_{\mathcal{S},j}(1 - x_{\mathcal{S},j}) d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A}) \\ = & \sum_{j\in\mathcal{C}, i\in\mathcal{S}, i'\in\mathcal{F}\setminus\mathcal{S}} x_{i,j} x_{i',j} d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A}) \leq 2\sum_{j,i,i'} x_{i,j} x_{i',j} \left[2d(i,j) + 2\ell d_{\mathsf{av}}(j) + d(i', j)\right] \\ = & \; 4\sum_{j,i} x_{i,j}(1 - x_{\mathcal{S},j}) \left[d(i,j) + \ell d_{\mathsf{av}}(j)\right] + 2\sum_{j,i'} x_{\mathcal{S},j} x_{i',j} d(i', j) \\ \leq & \; 4\sum_{i,j} x_{i,j} \left[d(i,j) + \ell d_{\mathsf{av}}(j)\right] + 2\sum_{j} x_{\mathcal{S},j} d_{\mathsf{av}}(j) = 4D_{\mathcal{S}} + (4\ell + 2)D'_{\mathcal{S}}. \end{aligned}$$

In above summations, j is over all clients in $\mathcal{C}$, i is over all locations in $\mathcal{S}$ and i' is over all locations in $\mathcal{F} \setminus \mathcal{S}$. The first inequality in the sequence used Lemma 4.4. All other inequalities and equations follow from the definitions of the notations used. □

4.3. Constructing family of neighborhood trees

Lemma 4.3 gives a necessary bound for our analysis. Still, we need to guarantee some other conditions when moving the demands. For example, when moving demands out of an "isolated" set $\mathcal{A}$, we should make sure that the distance is comparable to $d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A})$. If $y'(\mathcal{U}_{\mathcal{A}}) \leq \lfloor y(\mathcal{U}_{\mathcal{A}})\rfloor$, then we should not move demands out of $\mathcal{A}$, as $d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A})$ may not be bounded any more.

We guarantee these conditions by building a set of rooted trees over $\mathcal{C}^*$, called *neighborhood trees*. Roughly speaking, each neighborhood tree contains representatives that are nearby; moving demands within a neighborhood tree does not cost too much.

We use a triple $T = (\mathcal{V}, E, r)$ to denote a rooted tree, with vertex set $\mathcal{V} \subseteq \mathcal{C}^*$, edge set $E \subseteq \binom{\mathcal{V}}{2}$ and root $r \in \mathcal{V}$. Given a rooted tree $T = (\mathcal{V}, E, r)$ and a vertex $v \in \mathcal{V}$, we use $\Lambda_T(v)$ to denote the set of vertices in the sub-tree of T rooted at v. If $v \neq r$, we use $\rho_T(v)$ to denote the parent of v in T.

Definition 4.5. A rooted tree $T = (\mathcal{V} \subseteq \mathcal{C}^*, E, r)$ is called a *neighborhood tree* if for every vertex $v \in \mathcal{V} \setminus r$, $d(v, \mathcal{C}^* \setminus \Lambda_T(v)) = d(v, \rho_T(v))$.

In other words, $T = (\mathcal{V}, E, r)$ is a neighborhood tree if for every non-root vertex v of T, the parent $\rho_T(v)$ of v is the nearest vertex in $\mathcal{C}^*$ to v, except for vertices in $\Lambda_T(v)$. Our goal is to construct a set of neighborhood trees in $\mathcal{C}^*$. The next lemma is useful when constructing the trees:

LEMMA 4.6. *Let $T = (\mathcal{V} \subseteq \mathcal{C}^*, E, r)$ and $T' = (\mathcal{V}' \subseteq \mathcal{C}^*, E, r)$ be two disjoint neighborhood trees. Moreover assume $v^* := \arg\min_{v \in \mathcal{C}^* \setminus \mathcal{V}} d(r, v)$ is in $\mathcal{V}'$. Let T'' be the rooted tree $(\mathcal{V} \cup \mathcal{V}', E \cup E' \cup \{(r, v^*)\}, r')$. Then T'' is a neighborhood tree.*

PROOF. For every $v \in \mathcal{V} \setminus \{r\}$ we have $d(v, \mathcal{C}^* \setminus \Lambda_{T''}(v)) = d(v, \mathcal{C}^* \setminus \Lambda_T(v)) = d(v, \rho_T(v)) = d(v, \rho_{T''}(v))$ since $\Lambda_{T''}(v) = \Lambda_T(v)$, T is a neighborhood tree and $\rho_{T''}(v) = \rho_T(v)$. Also, we have $d(r, \mathcal{C}^* \setminus \Lambda_{T''}(r)) = d(r, \mathcal{C}^* \setminus \mathcal{V}) = d(r, v^*) = d(r, \rho_{T''}(r))$ since $\Lambda_{T''}(r) = \mathcal{V}$ and $v^* = \rho_{T''}(r)$ is the nearest neighbor of r in $\mathcal{C}^* \setminus \mathcal{V}$. Finally, for every $v \in \mathcal{V}' \setminus \{r'\}$, we have $d(v, \mathcal{C}^* \setminus \Lambda T''(v)) \geq d(v, \mathcal{C}^* \setminus T'_v) = d(v, \rho_{T'}(v)) = d(v, \rho_{T''}(v))$ since $\mathcal{C}^* \setminus \Lambda_{T''}(v) \subseteq \mathcal{C}^* \setminus \Lambda_{T'}(v)$, T' is a neighborhood tree and $\rho_{T'}(v) = \rho_{T''}(v)$. As $\rho_{T''}(v) \in \mathcal{C}^* \setminus T''_v$, we have $d(v, \mathcal{C}^* \setminus T''_v) \leq d(v, \rho_{T''}(v))$, implying $d(v, \mathcal{C}^* \setminus T''_v) = d(v, \rho_{T''}(v))$. □

Then we show how to construct the neighborhood trees, by proving the following lemma:

LEMMA 4.7. *Given any positive integer $\ell \leq |\mathcal{C}^*|$, we can find a set $\mathbb{T}$ of neighborhood trees such that*

(4.7a) *$\ell \leq |\mathcal{V}| \leq \ell^2$ for every neighborhood-tree $(\mathcal{V}, E, r) \in \mathbb{T}$;*
(4.7b) *$\bigcup_{(\mathcal{V}, E, r) \in \mathbb{T}} \mathcal{V} = \mathcal{C}^*$;*
(4.7c) *For two distinct trees $(\mathcal{V}, E, r), (\mathcal{V}', E', r') \in \mathbb{T}$, $\mathcal{V} \setminus \{r\}$ and $\mathcal{V}' \setminus \{r'\}$ are disjoint.*

PROOF. The first step of the construction is a simple iterative process. We maintain a spanning forest of rooted trees for $\mathcal{C}^*$. Initially, we have $|\mathcal{C}^*|$ singletons in the forest. At each iteration, we arbitrarily choose a tree $T = (\mathcal{V}, E, r)$ of size less than ℓ. Let $v^* = \arg\min_{v \in \mathcal{C}^* \setminus \mathcal{V}} d(r, v)$ be the nearest neighbor of r in $\mathcal{C}^* \setminus \mathcal{V}$. Assume v^* is in some rooted tree $T' = (\mathcal{V}', E', r')$. Then, we merge T and T' by adding an edge (r, v^*), and let v^* be the parent of r. i.e, the new tree will be $(\mathcal{V} \cup \mathcal{V}', E \cup E' \cup \{(r, v^*)\}, r')$. The process ends when all rooted trees have size at least ℓ. Since all singletons are neighborhood trees, by applying Lemma 4.6 repeatedly, we have that all rooted trees we constructed are neighborhood trees.

The neighborhood trees we constructed have size at least ℓ. However, they might have size much larger than ℓ^2. Thus, we need to break a large neighborhood tree. Let $T = (\mathcal{V}, E, r)$ be a neighborhood tree of size more than ℓ^2 we have constructed. We consider the growth of T by focusing on the tree containing r during the course of the iterative process. Initially, T contains a single vertex r. In each iteration in which T has grown, we merge T with some neighborhood tree $\tau = (\mathcal{V}_\tau, E_\tau, r_\tau)$ of size less than ℓ, by adding an edge (r_τ, v^*), where v^* is the nearest neighbor of r_τ in $\mathcal{C}^* \setminus \mathcal{V}_\tau$.

Let $\mathcal{L}$ be the set of treelets we constructed. For convenience, we also call the root r a treelet. Thus, we can view T as a tree over the set of treelets. To be more specific, we can construct a tree $\tilde{T} = (\mathcal{L}, E_{\tilde{T}})$ over the set $\mathcal{L}$ of treelets, where there is an edge $(\tau, \tau') \in \mathbb{E}_{\tilde{T}}$ if there is an edge between τ and τ' in T. We root $\tilde{T}$ at r; for every $\tau \in \mathcal{L}$, we use $\tilde{T}_\tau$ to denote the subtree of $\tilde{T}$ rooted at τ. The weight of a treelet τ is defined as the size of τ.

Consider the deepest treelet $\tau = (\mathcal{V}_\tau, E_\tau, r_\tau)$ in $\tilde{T}$ such that the total weight of the sub-tree of $\tilde{T}$ rooted at τ is at least $\ell(\ell-1)$. For each child τ' of τ in $\tilde{T}$, there is an edge e between τ' and τ in T. Then, we partition the children τ' according to the vertex in τ that e is incident to. Since $|\mathcal{V}_\tau| \le \ell - 1$, there must be a vertex $v \in \mathcal{V}_\tau$ such that the following holds. The total weight of all subtrees $\tilde{T}_{\tau'}$, over all children τ' of τ such that there is an edge between τ' and v in T, is at least $\frac{\ell(\ell-1)-|\mathcal{V}_\tau|}{|\mathcal{V}_\tau|}$. As $|\mathcal{V}_\tau| \le \ell - 1$, this is at least $\frac{\ell(\ell-1)}{\ell-1} - 1 = \ell - 1$. Let $\mathcal{L}'$ be the set of all such children of τ.

Focus on each $\tau' \in \mathcal{L}'$. If we un-contract each treelet in $\tilde{T}_{\tau'}$, we obtain a sub tree of T rooted at some child of v. Then, we construct tree T' as follows: take v, and these subtrees over all $\tau' \in \mathcal{L}'$ as well as the edges connecting v to the roots of these subtrees. Let v be the root of T'. Notice that by applying Lemma 4.6 repeatedly, we can prove that T' is a neighborhood tree. T' can be formed as follows. Initially, T' contains only the root v. Then we repeatedly merge some treelet $\tau'' = (\mathcal{V}_{\tau''}, E_{\tau''}, r_{\tau''}) \in \mathcal{L}$ with T' by adding an edge connecting $r_{\tau''}$ to some vertex $v^* \in T'$ such that v^* is the nearest neighbor of $r_{\tau''}$ in $\mathcal{V}_{\tau''}$.

Then, we remove all vertices in $T' \setminus \{v\}$ from T. Meanwhile, we remove all subtrees in $\left\{\tilde{T}_{\tau'}\right\}_{\tau' \in \mathcal{L}'}$ from $\tilde{T}$. The remaining tree T has size at least $\ell^2 - \ell(\ell-1) = \ell$. The process ends when T has size at most ℓ^2. So, all neighborhood trees we constructed have size between ℓ and ℓ^2 and the union of all neighborhood trees cover all vertices of $\mathcal{C}^*$. Every vertex of $\mathcal{C}^*$ can appear at most once as a non-root of some neighborhood tree, since every time we constructed a tree T', we remove all non-root vertices of T' from T. $\square$

4.4. Moving demands within neighbourhood trees

Recall that all the demands are at the client representatives. Every representative $v \in \mathcal{C}^*$ has $uy'(\mathcal{U}_v)$ units of demand. *In this section, it is convenient for us to* **scale down** *the demands by u. Thus a representative $v \in \mathcal{C}^*$ has $y'(\mathcal{U}_v)$ units of demand. Due to the scaling, moving α units of demand from v to v' costs $u\alpha d(v, v')$. If finally some v has α_v units of demand, we need to open $\lceil \alpha_v \rceil$ facilities at v.* For analytical purposes, we also say that $v \in \mathcal{C}^*$ has $y(\mathcal{U}_v) \ge y'(\mathcal{U}_v)$ units of supply. The total supply is $\sum_{v \in \mathcal{C}^*} y(\mathcal{U}_v) = y_{\mathcal{F}} \le k$.

Assume $|\mathcal{C}^*| \ge \ell$ for now. We apply Lemma 4.7 to construct a set $\mathbb{T}$ of neighborhood trees satisfying Properties (4.7a) to (4.7c). We assign the supplies and demands to vertices in the set $\mathbb{T}$. Notice that every representative in $\mathcal{C}^*$ appears in $\mathbb{T}$, and it appears in $\mathbb{T}$ as a non-root at most once. If $v \in \mathcal{C}^*$ appears as a non-root, we assign the $y'(\mathcal{U}_v)$ units of demand and the $y(\mathcal{U}_v)$ units of supply to the non-root. Otherwise, we assign the $y'(\mathcal{U}_v)$ units of demand and the $y(\mathcal{U}_v)$ units of supply to an arbitrary root v in $\mathbb{T}$.

Fix a neighborhood tree $T = (\mathcal{V}, E, r) \in \mathbb{T}$ from now on. Each $v \in \mathcal{V}$ has α_v units of demand and β_v units of supply. For $v \in \mathcal{V} \setminus \{r\}$, we have $\alpha_v = y'(\mathcal{U}_v)$ and $\beta_v = y(\mathcal{U}_v)$. We have either $\alpha_r = y'(\mathcal{U}_r), \beta_r = y(\mathcal{U}_r)$ or $\alpha_r = \beta_r = 0$. Define $\alpha_{\mathcal{V}'} := \sum_{v \in \mathcal{V}'} \alpha_v$ and $\beta_{\mathcal{V}'} := \sum_{v \in \mathcal{V}'} \beta_v$ for every $\mathcal{V}' \subseteq \mathcal{V}$. We shall move demands and supplies within T. Moving supplies is only for analytic purposes and costs nothing. When moving demands and supplies, we update $\{\alpha_v : v \in \mathcal{V}\}$ and $\{\beta_v : v \in \mathcal{V}\}$ accordingly. Keep in mind that we always maintain the property that $\alpha_v \le \beta_v$ for every $v \in \mathcal{V}$; we do not change $\alpha_{\mathcal{V}}$ and $\beta_{\mathcal{V}}$ (we do not change the total demands or supplies in $\mathcal{V}$). After the moving process for T, we add $\lceil \alpha_v \rceil$ open facilities at v for every $v \in \mathcal{V}$. We shall compare $\sum_{v \in \mathcal{V}} \lceil \alpha_v \rceil$ to $\alpha_{\mathcal{V}}$.

To define the moving process for $T = (\mathcal{V}, E, r)$, we give each edge in E a rank as follows. An edge $e = (v, v') \in E$ has length $L_e := d(v, v')$. Sort edges in E according

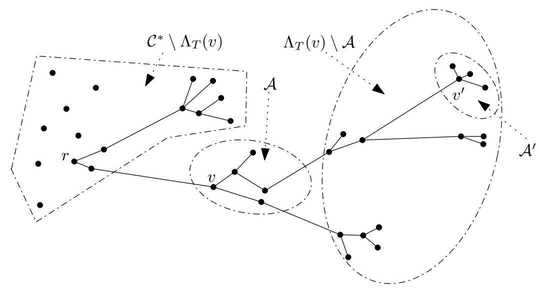

Fig. 3. Notations used in the proof of Claim 4.9. Solid circles are client representatives in $\mathcal{C}^*$ and solid lines give a neighborhood tree T.

to their lengths; assume $e_1, e_2, \cdots, e_{|\mathcal{V}|-1}$ is the ordering. Let the rank of e_1 be 1. For each $t = 2, 3, \cdots, e_{|\mathcal{V}|-1}$, if $L_{e_t} \leq 2\sum_{s=1}^{t-1} L_{e_s}$, then let the rank of e_t be the rank of e_{t-1}; otherwise let the rank of e_t be the rank of e_{t-1} plus 1. Let h be the rank of $e_{|\mathcal{V}|-1}$. For each $i \in [h]$, let E_i be the set of rank-i edges in E; for $i = 0, 1, \cdots, h$, let $E_{\leq i} = \bigcup_{i' \leq i} E_{i'}$ be the set of edges of rank at most i.

CLAIM 4.8. *For any $i \in [h]$ and $e, e' \in E_i$, we have $L_e/L_{e'} \leq 3^{|\mathcal{V}|-1}$.*

PROOF. It suffices to prove the lemma for the case where e' is the shortest rank-i edge, and e is the longest rank-i edge. Suppose $e' = e_{t'}$ and $e = e_t$ for $t' < t$. Let $L = \sum_{e'' \in E_{\leq i-1}} L_{e''}$. Then, $L_{e'} > 2L$. For every $s \in \{t', t'+1, \cdots, t-1\}$, we have $L_{e_{s+1}} \leq 2(L + L_{e_{t'}} + L_{e_{t'+1}} + \cdots + L_{e_s})$. Thus, $L + L_{e_{t'}} + L_{e_{t'+1}} + \cdots + L_{e_s} + L_{e_{s+1}} \leq 3(L + L_{e_{t'}} + L_{e_{t'+1}} + \cdots + L_{e_s})$. Thus, $L_e \leq 3^{t-t'}(L + L_{e'}) < \frac{3}{2} \cdot 3^{t-t'} L_{e'} \leq 3^{|\mathcal{V}|-1} L_{e'}$ as $t - t' \leq |\mathcal{V}| - 2$. $\square$

For every $i \in \{0, 1, \cdots, h\}$, we call the set of vertices in a connected component of $(\mathcal{V}, E_{\leq i})$ a level-i set. The family of level-i sets forms a partition of $\mathcal{V}$; and the union of families over all $i \in \{0, 1, \cdots, h\}$ is a laminar family. For every $i \in [h]$ and every level-i set $\mathcal{A}$, we check if Constraint (7) is satisfied for $\mathcal{B} = \mathcal{U}_{\mathcal{A}}$ and every $\mathcal{J} \subseteq \mathcal{C}$ (recall that this can be checked efficiently). If not, we find a violation of Constraint (7); from now on, we assume Constraint (7) holds for all these rectangles $(\mathcal{B}, \mathcal{J})$.

CLAIM 4.9. *If a level-i set $\mathcal{A}$ does not contain the root r, then $d(\mathcal{A}, \mathcal{C}^* \setminus \mathcal{A}) \geq \frac{L'}{2}$, where L' is the length of the shortest edge in E_{i+1}.*

PROOF. See Figure 3 for the notations used in the proof. Let v be the highest vertex in $\mathcal{A}$ according to T, and $L = \sum_{e \in E_{\leq i}} L_e$ be the total length of edges of rank at most i.

Notice that $\mathcal{C}^* \setminus \mathcal{A} = (\mathcal{C}^* \setminus \Lambda_T(v)) \cup (\Lambda_T(v) \setminus \mathcal{A})$. $d(v, \mathcal{C}^* \setminus \Lambda_T(v)) = d(v, \rho_T(v)) \geq L'$ since T is a neighborhood tree and the rank of $(v, \rho_T(v))$ is at least $i+1$. Thus $d(\mathcal{A}, \mathcal{C}^* \setminus \Lambda_T(v)) \geq L' - L \geq \frac{L'}{2}$ as the distance from v to any vertex in $\mathcal{A}$ is at most L. We now bound $d(\mathcal{A}, \Lambda_T(v) \setminus \mathcal{A})$. Consider each connected component in $(\Lambda_T(v) \setminus \mathcal{A}, E_{\leq i})$. Let $\mathcal{A}'$ be the set of vertices in the component and v' be its root. Since $d(v', \rho_T(v')) \geq L'$, we have $d(v', v) \geq L'$ as $\rho_T(v')$ is the nearest representative to v' in $\mathcal{C}^* \setminus \Lambda_T(v') \ni v$. Since both $\mathcal{A}$ and $\mathcal{A}'$ are connected by edges in $E_{\leq i}$, $v \in \mathcal{A}, v' \in \mathcal{A}'$ and the total length of edges in $E_{\leq i}$ is L, we have that $d(\mathcal{A}, \mathcal{A}') \geq L' - L \geq \frac{L'}{2}$. As this is true for any such $\mathcal{A}'$, we have $d(\mathcal{A}, \Lambda_T(v) \setminus \mathcal{A}) \geq \frac{L'}{2}$, which, combined with $d(\mathcal{A}, \mathcal{C}^* \setminus \Lambda_T(v)) \geq \frac{L'}{2}$, implies the claim. $\square$

Recall that the family of all level-i sets, over all $i = 0, 1, 2, \cdots, h$ forms a laminar family. Level-0 sets are singletons and the level-h set is the whole set $\mathcal{V}$. Our moving operation is level-by-level: for every $i = 1, 2, \cdots, h$ in this order, for every level-i set

$\mathcal{A} \subseteq \mathcal{V}$, we define a moving process for $\mathcal{A}$, in which we move demands and supplies within vertices in $\mathcal{A}$. After the moving operation for $\mathcal{A}$, we guarantee the following properties.

If $r \notin \mathcal{A}$, then either

(N1) all but one vertices $v \in \mathcal{A}$ have $\alpha_v = \beta_v \in \mathbb{Z}_{\geq 0}$; or
(N2) every vertex $v \in \mathcal{A}$ has $\beta_v \geq \lceil \alpha_v \rceil - 1/\ell$.

If $r \in \mathcal{A}$, then

(I1) every vertex $v \in \mathcal{A} \setminus \{r\}$ has $\beta_v \geq \lceil \alpha_v \rceil - 1/\ell$.

The above properties hold for all level-0 sets: they are all singletons; Property (N1) holds if $r \notin \mathcal{A}$ and Property (I1) holds if $r \in \mathcal{A}$. Now, suppose the properties hold for all level-$(i-1)$ sets. We define a moving operation for a level-i set $\mathcal{A}$ after which $\mathcal{A}$ satisfies the properties.

The first step is a collection step, in which we collect demands and supplies from $\mathcal{A}$. For every $v \in \mathcal{A} \setminus \{r\}$ such that $\beta_v < \lceil \alpha_v \rceil - 1/\ell$, we collect $\lfloor \alpha_v \rangle$ units of demand and $\beta_v - \lfloor \alpha_v \rfloor$ units of supply from v and keep them in a temporary holder. For all vertices $v \in \mathcal{A}$ with $\beta_v > \lceil \alpha_v \rceil$, we collect $\beta_v - \lceil \alpha_v \rceil$ units of supply from v. Now, we have $\lceil \alpha_v \rceil - 1/\ell \leq \beta_v \leq \lceil \alpha_v \rceil$ for every $v \in \mathcal{A} \setminus \{r\}$.

The second step is a redistribution step, in which we move the demand and supply in the temporary holder back to $\mathcal{A}$. If $r \in \mathcal{A}$, we simply move the demand and the supply in the holder to r and terminate the process. $\mathcal{A}$ will satisfy Property (I1). From now on we assume $r \notin \mathcal{A}$. We try to move the demand and the supply in the holder to each $v \in \mathcal{A}$ continuously until we have $\alpha_v = \beta_v \in \mathbb{Z}_{\geq 0}$: we first move demand from the holder to v until $\alpha_v = \beta_v$, then move demand and supply at the same rate until $\alpha_v = \beta_v \in \mathbb{Z}_{\geq 0}$. If we succeeded in making all vertices $v \in \mathcal{A}$ satisfy $\alpha_v = \beta_v \in \mathbb{Z}_{\geq 0}$, then we can move the remaining supplies and demands in the holder to an arbitrary vertex in $\mathcal{A}$. In this case $\mathcal{A}$ satisfies Property (N1). Suppose we failed to make $\alpha_v = \beta_v \in \mathbb{Z}_{\geq 0}$ for some $v \in \mathcal{A}$. The failure is due to the insufficient demand in the holder: we have collected at least the same amount of supply as demand; in the redistribution step, we either move the demand from the holder or move the demand and the supply at the same rate. We then move all the remaining supply in the holder to an arbitrary vertex $v \in \mathcal{A}$. Notice that during the continuous redistribution process for v, we always maintain the property that $\lceil \alpha_v \rceil - 1/\ell \leq \beta_v \leq \lceil \alpha_v \rceil$. Moving the remaining supply to an arbitrary vertex v also maintain the property that $\lceil \alpha_v \rceil - 1/\ell \leq \beta_v$. Thus $\mathcal{A}$ will satisfy Property (N2) in the end.

After we finished the moving operation for the level-h set $\mathcal{V}$, our set $\mathcal{V}$ satisfies Property (I1) as $r \in \mathcal{V}$. Thus $\sum_{v \in \mathcal{V}} \lceil \alpha_v \rceil \leq \sum_{v \in \mathcal{V} \setminus \{r\}} (\beta_v + 1/\ell) + \alpha_r + 1 \leq \beta_{\mathcal{V}} + 1 + (|\mathcal{V}| - 1)/\ell \leq \beta_{\mathcal{V}} + \frac{2\ell-1}{(\ell-1)^2}\beta_{\mathcal{V}}$ as $\beta_{\mathcal{V}} \geq (|\mathcal{V}| - 1)(1 - 1/\ell)$ and $|\mathcal{V}| \geq \ell$. Taking this sum over all trees in $\mathbb{T}$, we have that the number of open facilities is at most $k + \frac{2\ell-1}{(\ell-1)^2}k$. By setting $\ell = \lceil 3/\epsilon \rceil$, the number of open facilities is at most $(1+\epsilon)k$.

It suffices to bound the moving cost for T.

LEMMA 4.10. *The moving cost of the operation for $T = (\mathcal{V}, E, r) \in \mathbb{T}$ is at most*

$$\exp\left(O(\ell^2)\right)\left(D(\mathcal{U}_{\mathcal{V}\setminus\{r\}}) + D'(\mathcal{U}_{\mathcal{V}\setminus\{r\}})\right).$$

PROOF. Consider the moving process for a level-i set $\mathcal{A}$. Suppose we collected some demand from $v \in \mathcal{A}$. It must be the case that $v \neq r$ and $\beta_v < \lceil \alpha_v \rceil - 1/\ell$ before the collection, as otherwise we would not collect demand from v. If we let $\mathcal{A}' \subseteq \mathcal{A}$ be the level-$(i-1)$ set containing v, then $\mathcal{A}'$ must satisfy $r \notin \mathcal{A}'$ and Property (N2) by the induction assumption. This implies that we did not collect demands from any other vertices in $\mathcal{A}'$. Notice that $\beta_v < \lceil \alpha_v \rceil - 1/\ell$ implies $\alpha_v > \lfloor \beta_v \rfloor$ and $\lceil \beta_v \rangle > 1/\ell$. Then, $\alpha_{\mathcal{A}'} > \lfloor \beta_{\mathcal{A}'} \rfloor$ and $\lceil \beta_{\mathcal{A}'} \rangle > 1/\ell$ as all vertices $v' \in \mathcal{A}' \setminus \{v\}$ have $\alpha_{v'} = \beta_{v'} \in \mathbb{Z}_{\geq 0}$. Since we never moved demands or supplies in or out of $\mathcal{A}'$ before, we have $\alpha_{\mathcal{A}'} = y'_{\mathcal{S}}$ and $\beta_{\mathcal{A}'} = y_{\mathcal{S}}$, where $\mathcal{S} = \mathcal{U}_{\mathcal{A}'}$. Then $y'_{\mathcal{S}} > \lfloor y_{\mathcal{S}} \rfloor$ and $\lceil y_{\mathcal{S}} \rangle > 1/\ell$.

As we assumed that Constraint (7) is satisfied for $\mathcal{B} = \mathcal{S}$ and every $\mathcal{J} \subseteq \mathcal{C}$, we can apply Lemma 4.3 to show that $\lfloor y'_{\mathcal{S}} \rangle \lceil y_{\mathcal{S}} \rangle \, d(\mathcal{A}', \mathcal{C}^* \setminus \mathcal{A}') \leq \frac{4}{u} D_{\mathcal{S}} + \frac{4\ell+2}{u} D'_{\mathcal{S}}$. The demands collected from v will be moved to vertices in $\mathcal{A}$. The moving distance is at most $\sum_{e \in E_{\leq i}} L_e \leq |\mathcal{V}| \, 3^{|\mathcal{V}|} L'$ by Claim 4.8, where L' is the length of the shortest edge in E_i. Now, by Claim 4.9, $L' \leq 2d(\mathcal{A}', \mathcal{C}^* \setminus \mathcal{A}')$. Thus, the moving distance is at most

$$2 \, |\mathcal{V}| \, 3^{|\mathcal{V}|} d(\mathcal{A}', \mathcal{C}^* \setminus \mathcal{A}') \leq \frac{2}{2u} \frac{|\mathcal{V}| \, 3^{|\mathcal{V}|}}{\lfloor y'_{\mathcal{S}} \rangle \lceil y_{\mathcal{S}} \rangle} (2D_{\mathcal{S}} + (2\ell + 1) D'_{\mathcal{S}}).$$

Notice that $\lceil y_{\mathcal{S}} \rangle > 1/\ell$ and $|\mathcal{V}| \leq \ell^2$. The distance is at most $\frac{\exp\left(O(\ell^2)\right)}{u \lfloor y'_{\mathcal{S}} \rangle} (D_{\mathcal{S}} + D'_{\mathcal{S}})$. As we moved $\lfloor y'_{\mathcal{S}} \rangle$ units of demand from $\mathcal{A}'$, the moving cost is at most $\exp\left(O(\ell^2)\right)(D_{\mathcal{S}} + D'_{\mathcal{S}})$.

Taking the sum of the upper bounds over all level-$(i-1)$ sets $\mathcal{A}' \subseteq \mathcal{A} \setminus \{r\}$, the cost is at most $\exp\left(O(\ell^2)\right)\left(D(\mathcal{U}_{\mathcal{A} \setminus \{r\}}) + D'(\mathcal{U}_{\mathcal{A} \setminus \{r\}})\right)$. Taking the sum over $i \in [h]$ and all level-i sets $\mathcal{A}$, the cost is at most $\exp\left(O(\ell^2)\right)\left(D(\mathcal{U}_{\mathcal{V} \setminus \{r\}}) + D'(\mathcal{U}_{\mathcal{V} \setminus \{r\}})\right)$, as the number h of levels is absorbed by $\exp\left(O(\ell^2)\right)$. This finishes the proof of Lemma 4.10. $\square$

Finally, taking the bound over all neighborhood trees $T = (\mathcal{V}, E, r)$, the moving cost is at most $\exp(O(\ell^2))(D_{\mathcal{F}} + D'_{\mathcal{F}})$ due to Property (4.7c) and the fact that $\{\mathcal{U}_v : v \in \mathcal{C}^*\}$ forms a partition of $\mathcal{F}$. Since $D_{\mathcal{F}} = D'_{\mathcal{F}} = \mathsf{LP}$, the moving cost is at most $\exp(O(\ell^2))\mathsf{LP}$.

This finishes the proof of Theorem 1.1 for the case $|\mathcal{C}^*| \geq \ell$. When $|\mathcal{C}^*| < \ell$, we only build one neighborhood tree $(\mathcal{C}^*, E, r)$. Any minimum spanning tree over $\mathcal{C}^*$ will be a neighborhood tree. We run the algorithm for this neighborhood tree. The argument for moving cost still works; it suffices to bound the number of open facilities. After the moving process, we have $\beta_v \geq \lceil \alpha_v \rceil - 1/\ell$ for every $v \in \mathcal{C}^* \setminus \{r\}$. Also $\beta_{\mathcal{C}^*} \leq k$. Thus, $\beta_r \leq k - \beta_{\mathcal{C}^* \setminus \{r\}} \leq k - \sum_{v \in \mathcal{C}^* \setminus \{r\}} (\lceil \alpha_v \rceil - 1/\ell) \leq k - \sum_{v \in \mathcal{C}^* \setminus \{r\}} \lceil \alpha_v \rceil + (\ell - 2)/\ell$ as $|\mathcal{C}^*| < \ell$. Thus, $\lceil \alpha_r \rceil \leq \lceil \beta_r \rceil \leq k - \sum_{v \in \mathcal{C}^* \setminus \{r\}} \lceil \alpha_v \rceil + 1$, implying $\sum_{v \in \mathcal{C}^*} \lceil \alpha_v \rceil \leq k + 1$. Thus, the number of open facilities is at most $k + 1 \leq \lceil (1 + \epsilon) k \rceil$.

5. INTEGRALITY GAP FOR THE RECTANGLE LP RELAXATION

In this section, we show that the integrality gap of the rectangle LP relaxation is $\Omega(\log n)$, if the cardinality constraint can not be violated. Let $G = (V, E)$ be an expander of degree 3 of size $u = |V|$, where u is also the capacity of each facility; let $\alpha = \min_{S \subseteq V : |S| \leq |V|/2} \frac{|E(S, V \setminus S)|}{|S|}$ be the expansion of G. Let $\mathcal{F} = V$ and $\mathcal{C}$ contains $n := u(u+1)$ clients: for each $v \in V$, there are $u + 1$ clients in $\mathcal{C}$ co-located with v. We are allowed to open $k = u + 1$ facilities; multiple copies of each facility can be opened. The metric d is the shortest-path metric defined by G.

We first show that the cost of the optimum solution to the instance is $\Omega(u \log u)$. Intuitively, in the optimum solution, we first open one facility at each location $v \in V$, and let it serve u clients at v. So, one client remains at each location. Then we open one additional facility to serve the remaining u clients; the cost of serving the u clients will be $\Omega(u \log u)$. However, it is a little involved to prove this intuition; thus we shall use a different way to prove the lower bound.

If there are more than $\log u$ locations in $\mathcal{F}$ without open facilities (we call them empty locations), then the cost of the facilities is at least $(u + 1) \log u = \Omega(u \log u)$. Thus we assume there are less than $\log u$ empty locations. At an non-empty location, we can assume there is one open facility which serves u clients at the location; so we remove this open facility and the u clients it serves. Now, we have at least one client left at each location; at most $\log u + 1$ open facilities remain. Connecting the remaining clients to the remaining facilities will cost $\Omega(u \log u)$, even if there are no capacity constraints: the number of locations with distance at most $(\log u)/2$ to one of the $\log u + 1$ open facilities is at most $(\log u + 1)(3 \times 2^{(\log u)/2}) \le u/2$ for large enough u; so the connection cost is at least $u/2 \times (\log u)/2 = \Omega(u \log u)$.

Now we give a fractional solution to the rectangle LP whose cost is $O(u)$. For each facility $i \in \mathcal{F}$, we have $y_i = 1 + 1/u$. For each $i \in \mathcal{F}$, the assignment of clients is as follows. For every client j that is collocated with i, we have $x_{i,j} = 1 - 3/(\alpha u)$; for every j that has distance 1 to i, we have $x_{i,j} = 1/(\alpha u)$; for all other clients j, we have $x_{i,j} = 0$. The constraints in the basic LP relaxation are satisfied: every client j has $\sum_{i \in \mathcal{F}} x_{i,j} = 1 - 3/(\alpha u) + 3 \cdot 1/(\alpha u) = 1$; every facility i has $\sum_{j \in \mathcal{C}} x_{i,j} = u + 1 = uy_i$; we have $x_{i,j} \le y_i$ for every $i \in \mathcal{F}, j \in \mathcal{C}$. The cost of the fractional solution is $\sum_{i,j} d(i,j)x_{i,j} = u(u+1) \cdot 3/(\alpha u) = 3(u+1)/\alpha = O(u)$.

We now show that Constraint (7) is satisfied for every set $\mathcal{B} \subsetneq \mathcal{F}$ of facilities. Let $t = |\mathcal{B}|$ and $q = y_{\mathcal{B}} = t(1 + 1/u)$. We identify $\mathcal{C}$ with $[n]$ and assume $x_{\mathcal{B},1} \ge x_{\mathcal{B},2} \ge \cdots \ge x_{\mathcal{B},n}$. It suffices to prove that for every $p \in [n]$, we have $x_{\mathcal{B},[p]} \le f(p,q)$.

Given a concave function g on $\{0, 1, 2, \cdots, n\}$, we say an integer $t \in [0, n]$ is a break point if either $t \in \{0, n\}$ or $2g(t) > g(t-1) + g(t+1)$. Notice that $x_{\mathcal{B},[p]}$ is a concave function of p. Since the $u + 1$ clients at the same location i have the same $x_{\mathcal{B},\cdot}$ value, a break point of $x_{\mathcal{B},[\cdot]}$ must be a multiply of $u+1$. $f(\cdot, q)$ has four break points: 0, ut, $ut+u$ and n.

Assume $x_{\mathcal{B},[p]} > f(p,q)$ for some p. Then $x_{\mathcal{B},[p']} > f(p', q)$ must hold for some p' which is either a break point of $x_{\mathcal{B},[\cdot]}$, or a break point of $f(\cdot, q)$. As (x, y) is a valid solution to the basic LP, we have $x_{\mathcal{B},[p]} \le \min\{p, qu\}$ for every p. From the definition of f, it must be the case that $p' \in (u\lfloor q \rfloor, u\lceil q \rceil) = (ut, ut+u)$. Thus we must have $p' = (u+1)t$. In this case, $[p']$ contain the $(u+1)t$ clients co-located with facilities in $\mathcal{B}$.

First assume $t \le u/2$. Notice that $E(\mathcal{B}, \mathcal{F} \setminus \mathcal{B}) \ge \alpha|\mathcal{B}| = \alpha t$ as α is the expansion of G. So, $x_{\mathcal{B},[p']} \le t(u+1) - \alpha t(u+1) \cdot 1/(\alpha u) \le t(u+1) - t = tu \le tu + t^2/u = f(p', q)$. Now assume $t > u/2$. Then $x_{\mathcal{B},[p']} \le t(u+1) - \alpha(u-t)(u+1) \cdot 1/(\alpha u) \le t(u+1) - (u-t) = tu + 2t - u \le tu + t^2/u = f(p', q)$. This leads to a contradiction. Thus, the fractional solution satisfies Constraint (7).

Thus, we have proved that the integrality gap for the rectangle LP is $\Omega(\log u) = \Omega(\log n)$. So, in order to obtain a real constant approximation algorithm, a stronger LP relaxation is needed.

REFERENCES

Ankit Aggarwal, L. Anand, Manisha Bansal, Naveen Garg, Neelima Gupta, Shubham Gupta, and Surabhi Jain. 2010. A 3-approximation for Facility Location with Uniform Capacities. In *Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization (IPCO'10)*. Springer-Verlag, Berlin, Heidelberg, 149–162. DOI:http://dx.doi.org/10.1007/978-3-642-13036-6_12

Hyung-Chan An, Mohit Singh, and Ola Svensson. LP-Based Algorithms for Capacitated Facility Location. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2014*.

V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. 2001. 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)*. ACM, New York, NY, USA, 21–29. DOI:http://dx.doi.org/10.1145/380752.380755

Manisha Bansal, Naveen Garg, and Neelima Gupta. 2012. A 5-approximation for Capacitated Facility Location. In *Proceedings of the 20th Annual European Conference on Algorithms (ESA'12)*. Springer-Verlag, Berlin, Heidelberg, 133–144. DOI:http://dx.doi.org/10.1007/978-3-642-33090-2_13

J. Byrka. 2007. An Optimal Bifactor Approximation Algorithm for the Metric Uncapacitated Facility Location Problem. In *APPROX '07/RANDOM '07: Proceedings of the 10th International*

Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques. Springer-Verlag, Berlin, Heidelberg, 29–43. DOI:http://dx.doi.org/10.1007/978-3-540-74208-1_3

Jaroslaw Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-Factor Approximation Algorithms for Hard Capacitated k-Median Problems. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015)*.

Jaroslaw Byrka, Thomas Pensyl, Bartosz Rybicki, Joachim Spoerhase, Aravind Srinivasan, and Khoa Trinh. 2015. An Improved Approximation Algorithm for Knapsack Median Using Sparsification. In *Proc. 23rd Annual European Symposium on Algorithms (ESA'15)*. 275–287. http://dx.doi.org/10.1007/978-3-662-48350-3_24

Jaroslaw Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An Improved Approximation for k-median, and Positive Correlation in Budgeted Optimization. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015)*.

M. Charikar and S. Guha. 1999. Improved Combinatorial Algorithms for the Facility Location and k-Median Problems. In *In Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*. 378–388.

M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. 1999. 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)*. ACM, New York, NY, USA, 1–10. DOI:http://dx.doi.org/10.1145/301250.301257

F. A. Chudak and D. B. Shmoys. 2004. Improved Approximation Algorithms for the Uncapacitated Facility Location Problem. *SIAM J. Comput.* 33, 1 (2004), 1–25. DOI:http://dx.doi.org/10.1137/S0097539703405754

Fabian A. Chudak and David P. Williamson. 2005. Improved Approximation Algorithms for Capacitated Facility Location Problems. *Math. Program.* 102, 2 (March 2005), 207–222. DOI:http://dx.doi.org/10.1007/s10107-004-0524-9

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

Dion Gijswijt and Shanfei Li. 2013. Approximation algorithms for the capacitated k-facility location problems. *CoRR* abs/1311.4759 (2013).

S Guha and S Khuller. 1998. Greedy strikes back: Improved Facility Location Algorithms. In *Journal of Algorithms*. 649–657.

K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. 2003. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM* 50 (November 2003), 795–824. Issue 6. DOI:http://dx.doi.org/10.1145/950620.950621

K. Jain, M. Mahdian, and A. Saberi. 2002. A new greedy approach for facility location problems. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing (STOC '02)*. ACM, New York, NY, USA, 731–740. DOI:http://dx.doi.org/10.1145/509907.510012

K Jain and V. V. Vazirani. 2001. Approximation algorithms for metric facility location and k-Median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM* 48, 2 (2001), 274–296. DOI:http://dx.doi.org/10.1145/375827.375845

M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. 1998. Analysis of a local search heuristic for facility location problems. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms (SODA '98)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1–10. http://portal.acm.org/citation.cfm?id=314613.314616

S. Li. 2011. A 1.488 Approximation Algorithm for the Uncapacitated Facility Location Problem. In *Automata, Languages and Programming - 38th International Colloquium (ICALP)*. 77–88.

Shanfei Li. 2014. An Improved Approximation Algorithm for the Hard Uniform Capacitated k-median Problem. In *APPROX '14/RANDOM '14: Proceedings of the 17th International Workshop on Combinatorial Optimization Problems and the 18th International Workshop on Randomization and Computation (APPROX '14/RANDOM '14)*.

Shi Li. 2016. Approximating capacitated k-median with $(1+\epsilon)k$ open facilities. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*.

Shi Li and Ola Svensson. 2013. Approximating K-median via Pseudo-approximation. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing (STOC '13)*. ACM, New York, NY, USA, 901–910. DOI:http://dx.doi.org/10.1145/2488608.2488723

J. Lin and J. S. Vitter. 1992. Approximation algorithms for geometric median problems. *Inf. Process. Lett.* 44 (December 1992), 245–249. Issue 5. DOI:http://dx.doi.org/10.1016/0020-0190(92)90208-D

M. Mahdian, Y. Ye, and J. Zhang. 2006. Approximation Algorithms for Metric Facility Location Problems. *SIAM J. Comput.* 36, 2 (2006), 411–432. DOI:http://dx.doi.org/10.1137/S0097539703435716

D. B. Shmoys, E. Tardos, and K. Aardal. 1997. Approximation algorithms for facility location problems (extended abstract). In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 265–274. DOI:http://dx.doi.org/10.1145/258533.258600

D Williamson and Shmoys D. 2011. *The Design of Approximation Algorithms*. Cambridge University Press.

Jiawei Zhang, Bo Chen, and Yinyu Ye. 2005. A Multiexchange Local Search Algorithm for the Capacitated Facility Location Problem. *Math. Oper. Res.* 30, 2 (May 2005), 389–403. DOI:http://dx.doi.org/10.1287/moor.1040.0125