Constant Approximation for Capacitated k-Median with $(1+\epsilon)$-Capacity Violation*

Gökalp Demirci[1] and Shi Li[2]

1 **Department of Computer Science, University of Chicago, Chicago, IL, USA,** demirci@cs.uchicago.edu.

2 **Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, USA,** shil@buffalo.edu. **Supported in part by NSF grant CCF-1566356.**

Abstract

We study the Capacitated k-Median problem for which existing constant-factor approximation algorithms are all pseudo-approximations that violate either the capacities or the upper bound k on the number of open facilities. Using the natural LP relaxation for the problem, one can only hope to get the violation factor down to 2. Li [SODA'16] introduced a novel LP to go beyond the limit of 2 and gave a constant-factor approximation algorithm that opens $(1+\epsilon)k$ facilities.

We use the configuration LP of Li [SODA'16] to give a constant-factor approximation for the Capacitated k-Median problem in a seemingly harder configuration: we violate only the capacities by $1+\epsilon$. This result settles the problem as far as pseudo-approximation algorithms are concerned.

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

Keywords and phrases Approximation Algorithms, Capacitated k-Median, Pseudo Approximation, Capacity Violation

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.XXX

1 Introduction

In the capacitated k-median problem (CKM), we are given a set F of facilities together with their capacities $u_i \in \mathbb{Z}_{>0}$ for $i \in F$, a set C of clients, a metric d on $F \cup C$, and a number k. We are asked to open some of these facilities $F' \subseteq F$ and give an assignment $\sigma : C \to F'$ connecting each client to one of the open facilities so that the number of open facilities is not bigger than k, i.e. $|F'| \leq k$ (*cardinality constraint*), and each facility $i \in F'$ is connected to at most u_i clients, i.e. $\left|\sigma^{-1}(i)\right| \leq u_i$ (*capacity constraint*). The goal is to minimize the sum of the connection costs, i.e. $\sum_{j\in C} d(\sigma(j), j)$.

Without the capacity constraint, i.e. $u_i = \infty$ for all $i \in F$, this is the famous k-median problem (KM). The first constant-factor approximation algorithm for KM is given by Charikar et al. [9], guaranteeing a solution within $6\frac{2}{3}$ times the cost of the optimal solution. Then the approximation ratio has been improved by a series of papers [13, 8, 3, 12, 17, 5]. The current best ratio for KM is $2.675+\epsilon$ due to Byrka et al. [5], which was obtained by improving a part of the algorithm given by Li and Svensson [17].

On the other hand, we don't have a true constant approximation for CKM. All known constant-factor results are pseudo-approximations which violate either the cardinality or the

[*] A full version of the paper can be found at http://arxiv.org/abs/1603.02324.

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi; Article No. XXX; pp. XXX:1–XXX:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

capacity constraint. Aardal et al. [1] gave an algorithm which finds a $(7+\epsilon)$-approximate solution to CKM by opening at most $2k$ facilities, i.e. violating the cardinality constraint by a factor of 2. Guha [11] gave an algorithm with approximation ratio 16 for the more relaxed *uniform* CKM, where all capacities are the same, by connecting at most $4u$ clients to each facility, thus violating the capacity constraint by 4. Li [14] gave a constant-factor algorithm for uniform CKM with capacity violation of only $2+\epsilon$ by improving the algorithm in [9]. For non-uniform capacities, Chuzhoy and Rabani [10] gave a 40-approximation for CKM by violating the capacities by a factor of 50 using a mixture of primal-dual schema and lagrangian relaxations. Their algorithm is for a slightly relaxed version of the problem called *soft* CKM where one is allowed to open multiple collocated copies of a facility in F. The CKM definition we gave above is sometimes referred to as *hard* CKM as opposed to this version. Recently, Byrka et al. [4] gave a constant-factor algorithm for hard CKM by keeping capacity violation factor under $3+\epsilon$.

All these algorithms for CKM use the basic LP relaxation for the problem which is known to have an unbounded integrality gap even when we are allowed to violate either the capacity or the cardinality constraint by $2-\epsilon$. In this sense, results of [1] and [14] can be considered as reaching the limits of the basic LP relaxation in terms of restricting the violation factor. In order to go beyond these limits, Li [15] introduced a novel LP called the *rectangle* LP and presented a constant-factor approximation algorithm for soft uniform CKM by opening $(1+\epsilon)k$ facilities. This was later generalized by the same author to non-uniform CKM [16], where he introduced an even stronger LP relaxation called the *configuration* LP. Very recently, independently of the work in this paper, Byrka et al. [6] used this configuration LP to give a similar algorithm for uniform CKM violating the capacities by $1+\epsilon$.

1.1 Our Result

In this paper, we use the configuration LP of [16] to give an $O(1/\epsilon^5)$-approximation algorithm for non-uniform hard CKM which respects the cardinality constraint and connects at most $(1+\epsilon)u_i$ clients to any open facility $i \in F$. The running time of our algorithm is $n^{O(1/\epsilon)}$. Thus, with this result, we now have settled the CKM problem from the view of pseudo-approximation algorithms: either $(1+\epsilon)$-cardinality violation or $(1+\epsilon)$-capacity violation is sufficient for a constant approximation for CKM.

The known results for the CKM problem have suggested that designing algorithms with capacity violation (satisfying the cardinality constraint) is harder than designing algorithms with cardinality violation. Note, for example, that the best known cardinality violation factor for non-uniform CKM among algorithms using only the basic LP relaxation (a factor of 2 in [1]) matches the smallest possible cardinality violation factor dictated by the gap instance. In contrast, the best capacity-violation factor is $3+\epsilon$ due to [4], but the gap instance for the basic LP with the largest known gap eliminates only the algorithms with capacity violation smaller than 2. Furthermore, we can argue that, for algorithms based on the basic LP and the configuration LP, a β-capacity violation can be converted to a β-cardinality violation, suggesting that allowing capacity violation is more restrictive than allowing cardinality violation. We leave the detail to the full version of the paper.

Our Techniques. Our algorithm uses the configuration LP introduced in [16] and the framework of [16] that creates a two-level clustering of facilities. [16] considered the $(1+\epsilon)$-cardinality violation setting, which is more flexible in the sense that one has the much freedom to distribute the ϵk extra facilities. In our $(1+\epsilon)$-capacity violation setting, each facility i can provide an extra ϵu_i capacity; however, these extra capacities are restricted by the locations of the facilities. In particular, we need one more level of clustering to form so-called "groups" so that each group contains $\Omega(1/\epsilon)$ fractional open facility. Only with groups of $\Omega(1/\epsilon)$ facilities, we can benefit from the extra capacities given by the $(1+\epsilon)$-capacity scaling. Our algorithm then constructs distributions of local solutions. Using a dependent rounding procedure we can select a local solution from each distribution such that the solution formed by the concatenation of local solutions has a small cost. This initial solution may contain more than k facilities. We then remove some already-open facilities, and bound the cost incurred due to the removal of open facilities. When we remove a facility, we are guaranteed that there is a close group containing $\Omega(1/\epsilon)$ open facilities and the extra capacities provided by these facilities can compensate for the capacity of the removed facility.

Organization. The remaining part of the paper is organized as follows. In Sections 2 and 3, we describe the configuration LP introduced in [16] and our three-level clustering procedure respectively. In Section 4, we show how to construct the distributions of local solutions. Then finally in Section 5, we show how to obtain our final solution by combining the distributions we constructed. Due to the page limit, some proofs are omitted and they can be found in the full version of the paper.

2 The Basic LP and the Configuration LP

In this section, we give the configuration LP of [16] for CKM. We start with the following basic LP relaxation:

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

$$\begin{array}{rlll|rll}
\sum_{i\in F} y_i \le k; & & (1) & & \sum_{j\in C} x_{i,j} \le u_i y_i, & \forall i\in F; & (4)\\
\sum_{i\in F} x_{i,j} = 1, & \forall j \in C; & (2) & & 0\le x_{i,j}, y_i \le 1, & \forall i\in F, j\in C. & (5)\\
x_{i,j} \le y_i, & \forall i\in F, j\in C; & (3) & & & &
\end{array}$$

In the LP, y_i indicates whether a facility $i\in F$ is open, and $x_{i,j}$ indicates whether client $j\in C$ is connected to facility $i\in F$. Constraint (1) is the cardinality constraint assuring that the number of open facilities is no more than k. Constraint (2) says that every client must be fully connected to facilities. Constraint (3) requires a facility to be open in order to connect clients. Constraint (4) is the capacity constraint.

It is well known that the basic LP has unbounded integrality gap, even if we are allowed to violate the cardinality constraint or the capacity constraint by a factor of $2-\epsilon$. The description of the instance can be found in the full version of the paper. In order to overcome the gap in the cardinality-violation setting, Li [16] introduced a novel LP for CKM called the configuration LP, which we formally state below. Let us fix a set $B\subseteq F$ of facilities. Let $\ell = \Theta(1/\epsilon)$ and $\ell_1 = \Theta(\ell)$ be sufficiently large integers. Let $\mathcal{S} = \{S\subseteq B : |S|\le \ell_1\}$ and $\widetilde{\mathcal{S}} = \mathcal{S}\cup\{\bot\}$, where $\bot$ stands for "any subset of B with size more than ℓ_1"; for convenience, we also treat $\bot$ as a set such that $i\in\bot$ holds for every $i\in B$. For $S\in\mathcal{S}$, let z^B_S indicate the event that the set of open facilities in B is exactly S and $z^B_\bot$ indicate the event that the number of open facilities in B is more than ℓ_1.

For every $S\in\widetilde{\mathcal{S}}$ and $i\in S$, $z^B_{S,i}$ indicates the event that $z^B_S = 1$ and i is open. (If $i\in B$ but $i\notin S$, then the event will not happen.) Notice that when $i\in S\neq\bot$, we always have $z^B_{S,i} = z^B_S$; we keep both variables for notational purposes. For every $S\in\widetilde{\mathcal{S}}, i\in S$ and client $j\in C$, $z^B_{S,i,j}$ indicates the event that $z^B_{S,i} = 1$ and j is connected to i. In an integral solution, all the above variables are $\{0,1\}$ variables. The following constraints are valid. To help understand the constraints, it is good to think of $z^B_{S,i}$ as $z^B_S\cdot y_i$ and $z^B_{S,i,j}$ as $z^B_S\cdot x_{i,j}$.

$$\sum_{S\in\widetilde{\mathcal{S}}} z^B_S = 1; \tag{6}$$

$$\sum_{S\in\widetilde{\mathcal{S}}:i\in S} z^B_{S,i} = y_i, \quad \forall i \in B; \tag{7}$$

$$\sum_{S\in\widetilde{\mathcal{S}}:i\in S} z^B_{S,i,j} = x_{i,j}, \ \ \forall i \in B, j \in C; \tag{8}$$

$$0 \leq z^B_{S,i,j} \leq z^B_{S,i} \leq z^B_S, \quad \forall S \in \widetilde{\mathcal{S}}, i \in S, j \in C; \tag{9}$$

$$z^B_{S,i} = z^B_S, \qquad \forall S \in \mathcal{S}, i \in S; \tag{10}$$

$$\sum_{i\in S} z^B_{S,i,j} \leq z^B_S, \qquad \forall S \in \widetilde{\mathcal{S}}, j \in C; \tag{11}$$

$$\sum_{j\in C} z^B_{S,i,j} \leq u_i z^B_{S,i}, \quad \forall S \in \widetilde{\mathcal{S}}, i \in S; \tag{12}$$

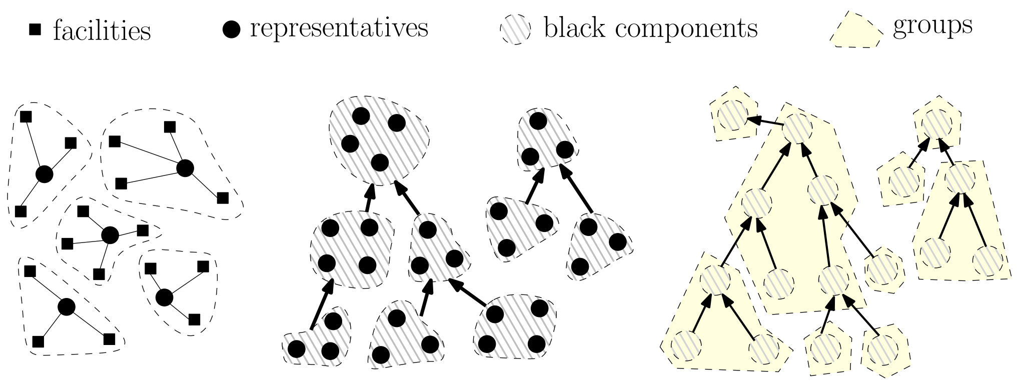
$$\sum_{i\in B} z^B_{\perp,i} \geq \ell_1 z^B_{\perp}. \tag{13}$$

Constraint (6) says that $z^B_S = 1$ for exactly one $S \in \widetilde{\mathcal{S}}$. Constraint (7) says that if i is open then there is exactly one $S \in \widetilde{\mathcal{S}}$ with $z^B_{S,i} = 1$. Constraint (8) says that if j is connected to i then there is exactly one $S \in \widetilde{\mathcal{S}}$ such that $z^B_{S,i,j} = 1$. Constraint (9) is by the definition of variables. Constraint (10) holds as we mentioned earlier. Constraint (11) says that if $z^B_S = 1$ then j can be connected to at most 1 facility in S. Constraint (12) is the capacity constraint. Constraint (13) says that if $z^B_\perp = 1$, there are at least ℓ_1 open facilities in B.

The configuration LP is obtained from the basic LP by adding the z variables and Constraints (6) to (13) for every $B \subseteq F$. Since there are exponentially many subsets $B \subseteq F$, we don't know how to solve this LP efficiently. However, note that there are only polynomially many $(n^{O(\ell_1)})$ z^B variables for a fixed $B \subseteq F$. Given a fractional solution (x, y) to the basic LP relaxation, we can construct the values of z^B variables and check their feasibility for Constraints (6) to (13) in polynomial time as in [16]. Our rounding algorithm either constructs an integral solution with the desired properties, or outputs a set $B \subseteq F$ such that Constraints (6) to (13) are infeasible. In the latter case, we can find a constraint in the configuration LP that (x, y) does not satisfy. Then we can run the ellipsoid method and the rounding algorithm in an iterative way (see, e.g., [7, 2]).

Notations From now on, we fix a solution $(\{x_{i,j} : i \in F, j \in C\}, \{y_i : i \in F\})$ to the basic LP. We define $d_{\mathsf{av}}(j) := \sum_{i\in F} x_{i,j} d(i,j)$ to be the connection cost of j, for every $j \in C$. Let $D_i := \sum_{j\in C} x_{i,j}\left(d(i,j) + d_{\mathsf{av}}(j)\right)$ for every $i \in F$, and $D_S := \sum_{i\in S} D_i$ for every $S \subseteq F$. We denote the value of the solution (x, y) by $\mathsf{LP} := \sum_{i\in F, j\in C} x_{i,j} d(i,j) = \sum_{j\in C} d_{\mathsf{av}}(j)$. Note that $D_F = \sum_{i\in F,j\in C} x_{i,j}\left(d(i,j) + d_{\mathsf{av}}(j)\right) = \sum_{i\in F,j\in C} x_{i,j} d(i,j) + \sum_{j\in C} d_{\mathsf{av}}(j) \sum_{i\in F} x_{i,j} = 2\mathsf{LP}$. For any set $F' \subseteq F$ of facilities and $C' \subseteq C$ of clients, we shall let $x_{F',C'} := \sum_{i\in F', j\in C'} x_{i,j}$; we simply use $x_{i,C'}$ for $x_{\{i\},C'}$ and $x_{F',j}$ for $x_{F',\{j\}}$. For any $F' \subseteq F$, let $y_{F'} := \sum_{i\in F'} y_i$. Let $d(A, B) := \min_{i\in A, j\in B} d(i, j)$ denote the minimum distance between A and B, for any $A, B \subseteq F \cup C$; we simply use $d(i, B)$ for $d(\{i\}, B)$.

Moving of Demands After the set of open facilities is decided, the optimum connection assignment from clients to facilities can be computed by solving the minimum cost b-matching problem. Due to the integrality of the matching polytope, we may allow the connections to be fractional. That is, if there is a good fractional assignment, then there is a good integral assignment. So we can use the following framework to design and analyze the rounding algorithm. Initially there is one unit of demand at each client $j \in C$. During the course of our algorithm, we move demands fractionally within $F \cup C$; moving α units of demand from i to j incurs a cost of $\alpha d(i, j)$. At the end, all the demands are moved to F and each facility $i \in F$ has at most $(1 + O(\frac{1}{\ell}))u_i$ units of demand. We open a facility if it has positive amount of demand. Our goal is to bound the total moving cost by $O(\ell^5)\mathsf{LP}$ and the number of open facilities by k.



(a). bundles $\{U_v\}_{v\in R}$ (b). black components $\mathcal{J}$ and forest $\Upsilon^*_{\mathcal{J}}$ (c). groups $\mathcal{G}$ and forest $\Upsilon_{\mathcal{G}}$

Figure 1 The three-phase clustering procedure. In the first phase (Figure (a)), we partition F into bundles, centered at the set R of representatives. In the second phase (Figure (b)), we partition R into a family $\mathcal{J}$ of black components and construct a degree-2 rooted forest over $\mathcal{J}$. In the third phase (Figure(c)), we partition $\mathcal{J}$ into a family $\mathcal{G}$ of groups; $\Upsilon_{\mathcal{G}}$ is formed from $\Upsilon^*_{\mathcal{J}}$ by contracting each group into a single node.

3 Representatives, Black Components, and Groups

Our algorithm starts with bundling facilities together with a three-phase process each of which creates bigger and bigger clusters. At the end, we have a nicely formed network of sufficiently big clusters of facilities. See Figure 1 for illustration of the three-phase clustering.

3.1 Representatives, Bundles and Initial Moving of Demands

In the first phase, we use a standard approach to facility location problems ([18, 19, 9, 16]) to partition the facilities into *bundles* $\{U_v\}_{v\in R}$, where each bundle U_v is associated with a center $v \in C$ that is called a *representative* and $R \subseteq C$ is the set of representatives. Each bundle U_v has a total opening at least $1/2$.

Let $R = \emptyset$ initially. Repeat the following process until C becomes empty: we select the client $v \in C$ with the smallest $d_{\mathsf{av}}(v)$ and add it to R; then we remove all clients j such that $d(j, v) \le 4d_{\mathsf{av}}(j)$ from C (thus, v itself is removed). We use v and its variants to index representatives, and j and its variants to index general clients. The family $\{U_v : v \in R\}$ is the Voronoi diagram of F with R being the centers: let $U_v = \emptyset$ for every $v \in R$ initially; for each location $i \in F$, we add i to U_v for $v \in R$ that is closest to i. For any subset $V \subseteq R$, we use $U(V) := \bigcup_{v\in V} U_v$ to denote the union of Voronoi regions with centers V.

▶ **Lemma 1.** *The following statements hold:*

(1a) *for all $v, v' \in R, v \ne v'$, we have $d(v, v') > 4\max\{d_{\mathsf{av}}(v), d_{\mathsf{av}}(v')\}$*
(1b) *for all $j \in C$, there exists $v \in R$, such that $d_{\mathsf{av}}(v) \le d_{\mathsf{av}}(j)$ and $d(v, j) \le 4d_{\mathsf{av}}(j)$;*
(1c) *$y_{U_v} \ge 1/2$ for every $v \in R$;*
(1d) *for any $v \in R$, $i \in U_v$, and $j \in C$, we have $d(i, v) \le d(i, j) + 4d_{\mathsf{av}}(j)$.*

The next lemma shows that moving demands from facilities to their corresponding representative doesn't cost much.

▶ **Lemma 2.** *For every $v \in R$, we have $\sum_{i\in U_v} x_{i,C} d(i, v) \le O(1)D_{U_v}$.*

Since $\{U_v : v \in R\}$ forms a partition of F, we get the following corollary.

▶ **Corollary 3.** $\sum_{v\in R, i\in U_v} x_{i,C} d(i,v) \leq O(1)\mathsf{LP}$.

Initial Moving of Demands With this corollary, we now move all the demands from C to V. First for every $j \in C$ and $i \in F$, we move $x_{i,j}$ units of demand from j to i. The moving cost of this step is exactly LP. After the step, all demands are at F and every $i \in F$ has $x_{i,C}$ units of demand. Then, for every $v \in R$ and $i \in U_v$, we move the $x_{i,C}$ units of demand at i to v. The moving cost for this step is $O(1)\mathsf{LP}$. Thus, after the initial moving, all demands are at the set R of representatives: a representative v has $x_{U_v,C}$ units of demand.

3.2 Black Components

In the second phase, we employ the minimum-spanning-tree construction of [16] to partition the set R of representatives into a family $\mathcal{J}$ of so-called *black components*. There is a degree-2 rooted forest $\Upsilon^*_{\mathcal{J}}$ over $\mathcal{J}$ with many good properties. For example, each non-root black component is not far away from its parent, and each root black component of $\Upsilon^*_{\mathcal{J}}$ contains a total opening of $\Omega(\ell)$. (For simplicity, we say the total opening at a representative $v \in R$ is y_{U_v}, which is the total opening at the bundle U_v.) The forest in [16] can have a large degree, while our algorithm requires the forest to have degree 2. This property is guaranteed by using the left-child-right-sibling representation.

Due to the page limit, we leave the description of the framework of [16] to the full version of the paper, and give its summary in the following lemma:

▶ **Lemma 4.** *There is an efficient algorithm to partition R into a set $\mathcal{J}$ of* black components *(or components, for simplicity) and construct a rooted forest $\Upsilon^*_{\mathcal{J}}$ over $\mathcal{J}$, such that if we let $L(J) = d(J, R \setminus J)$ for every black component $J \in \mathcal{J}$, then the following properties hold:*

(4a) *for every $J \in \mathcal{J}$, there is a spanning tree over the representatives in J such that for every edge (v, v') in the spanning tree we have $d(v, v') \leq L(J)$;*

(4b) *every root component $J \in \mathcal{J}$ of $\Upsilon^*_{\mathcal{J}}$ has $y_{U(J)} \geq \ell$ and every non-root component $J \in \mathcal{J}$ has $y_{U(J)} < \ell$;*

(4c) *every root component $J \in \mathcal{J}$ of $\Upsilon^*_{\mathcal{J}}$ has either $y_{U(J)} < 2\ell$ or $|J| = 1$;*

(4d) *for any non-root component J and its parent J', we have $L(J) \geq L(J')$;*

(4e) *for any non-root component J and its parent J', we have $d(J, J') \leq O(\ell)L(J)$;*

(4f) *every component J has at most two children.*

3.3 Groups

In the third phase, we apply a simple greedy algorithm to the forest $\Upsilon^*_{\mathcal{J}}$ to partition the set $\mathcal{J}$ of black components into a family $\mathcal{G}$ of *groups*, where each group $G \in \mathcal{G}$ contains many black components that are connected in $\Upsilon^*_{\mathcal{J}}$. By contracting each group $G \in \mathcal{G}$, the forest $\Upsilon^*_{\mathcal{J}}$ over the set $\mathcal{J}$ of black components becomes a forest $\Upsilon_{\mathcal{G}}$ over the set $\mathcal{G}$ of groups. Each group has a total opening of $\Omega(\ell)$, unless it is a leaf-group in $\Upsilon_{\mathcal{G}}$.

We partition the set $\mathcal{J}$ into groups using a technique similar to [4, 6]. For each rooted tree $T = (\mathcal{J}_T, E_T)$ in $\Upsilon^*_{\mathcal{J}}$, we construct a group G of black components as follows. Initially, let G contain the root component of T. While $\sum_{J\in G} y_{U(J)} < \ell$ and $G \neq \mathcal{J}_T$, repeat the following procedure. Choose the component $J \in \mathcal{J}_T \setminus G$ that is adjacent to G in T, with the smallest L-value, and add J to G.

Thus, by the construction G is connected in T. After we have constructed the group G, we add G to $\mathcal{G}$. We remove all black components in G from T. Then, each T is broken into many rooted trees; we apply the above procedure recursively for each rooted tree.

So, we have constructed a partition $\mathcal{G}$ for the set $\mathcal{J}$ of components. If for every $G \in \mathcal{G}$, we contract all components in G into a single node, then the rooted forest $\Upsilon^*_{\mathcal{J}}$ over $\mathcal{J}$ becomes a rooted forest $\Upsilon_{\mathcal{G}}$ over the set $\mathcal{G}$ of groups. $\Upsilon_{\mathcal{G}}$ naturally defines a parent-child relationship over $\mathcal{G}$. The following lemma uses Properties (4a) to (4f) of $\mathcal{J}$ and the way we construct $\mathcal{G}$.

▶ **Lemma 5.** *The following statements hold for the set $\mathcal{G}$ of groups and the rooted forest $\Upsilon_{\mathcal{G}}$ over $\mathcal{G}$:*

(5a) *any root group $G \in \mathcal{G}$ contains a single root component $J \in \mathcal{J}$;*
(5b) *if $G \in \mathcal{G}$ is not a root group, then $\sum_{J \in G} y_{U(J)} < 2\ell$;*
(5c) *if $G \in \mathcal{G}$ is a non-leaf group, then $\sum_{J \in G} y_{U(J)} \geq \ell$;*
(5d) *let $G \in \mathcal{G}, G' \in \mathcal{G}$ be the parent of G, $J \in G$ and $v \in J$, then the distance between v and any representative in $\bigcup_{J' \in G'} J'$ is at most $O(\ell^2)L(J)$;*
(5e) *any group G has at most $O(\ell)$ children.*

4 Constructing Local Solutions

In this section, we shall construct a local solution, or a distribution of local solutions, for a given set $V \subseteq R$ which is the union of some black components. A local solution for V contains a pair $(S \subseteq U(V), \beta \in \mathbb{R}^{U(V)}_{\geq 0})$, where S is the facilities we open in $U(V)$ and β_i for each $i \in U(V)$ is the amount of supply at i: the demand that can be satisfied by i. Thus $\beta_i = 0$ if $i \in U(V) \setminus S$. We shall use the supplies at $U(V)$ to satisfy the $x_{U(V),C}$ demands at V after the initial moving of demands; thus, we require $\sum_{i \in U(V)} \beta_i = x_{U(V),C}$. There are two other main properties we need the distribution to satisfy: (a) the expected size of S from the distribution is not too big, and (b) the cost of matching the demands at V and the supplies at $U(V)$ is small.

We distinguish between *concentrated* black components and *non-concentrated* black components. Roughly speaking, a component $J \in \mathcal{J}$ is concentrated if in the fractional solution (x, y), for most clients $j \in C$, j is either almost fully served by facilities in $U(J)$, or almost fully served by facilities in $F \setminus U(J)$. We shall construct a distribution of local solutions for each concentrated component J. We require Constraints (6) to (13) to be satisfied for $B = U(J)$ (if not, we return the set $U(J)$ to the separation oracle) and let z^B be the vector satisfying the constraints. Roughly speaking, the z^B-vector defines a distribution of local solutions for V. A local solution (S, β) is good if S is not too big and the total demand $\sum_{i \in S} \beta_i$ satisfied by S is not too small. Then, our algorithm randomly selects (S, β) from the distribution defined by z^B, under the condition that (S, β) is good. The fact that J is concentrated guarantees that the total mass of good local solutions in the distribution is large; therefore the factors we lose due to the conditioning are small.

For non-concentrated components, we construct a single local solution (S, β), instead of a distribution of local solutions. Moreover, the construction is for the union V of some non-concentrated components, instead of an individual component. The components that comprise V are close to each other; by the fact that they are non-concentrated, we can move demands arbitrarily within V, without incurring too much cost. Thus we can essentially treat the distances between representatives in V as 0. Then we are only concerned with two parameters for each facility $i \in U(V)$: the distance from i to V and the capacity u_i. Using a simple argument, the optimum fractional local solution (that minimizes the cost of matching the demands and supplies) is almost integral: it contains at most 2 fractionally open facilities. By fully opening the two fractional facilities, we find an integral local solution with small number of open facilities.

The remaining part of this section is organized as follows. We first formally define concentrated black components, and explain the importance of the definition. We then define the earth-mover-distance, which will be used to measure the cost of satisfying demands using supplies. The construction of local solutions for concentrated components and non-concentrated components will be stated in Theorem 9 and Lemma 10 respectively. Due to the page limit, their proofs will only appear in the full version of the paper.

Concentrated Black Components The definition of concentrated black component is the same as that of [16], except that we choose the parameter ℓ_2 differently.

▶ **Definition 6.** Define $\pi_J = \sum_{j\in C} x_{U(J),j}(1-x_{U(J),j})$, for every black component $J \in \mathcal{J}$. A black component $J \in \mathcal{J}$ is said to be *concentrated* if $\pi_J \leq x_{U(J),C}/\ell_2$, and *non-concentrated* otherwise, where $\ell_2 = \Theta(\ell^3)$ is large enough.

We use $\mathcal{J}^{\mathsf{C}}$ to denote the set of concentrated components and $\mathcal{J}^{\mathsf{N}}$ to denote the set of non-concentrated components. The next lemma from [16] shows the importance of π_J.

▶ **Lemma 7.** *For any $J \in \mathcal{J}$, we have $L(J)\pi_J \leq O(1)D_{U(J)}$.*

Recall that $L(J) = d(J, R \setminus J)$ and $x_{U(J),C}$ is the total demand in J after the initial moving. Thus, according to Lemma 7, if J is not concentrated, we can use $D_{U(J)}$ to charge the cost for moving all the $x_{U(J),C}$ units of demand out of J, provided that the moving distance is not too big compared to $L(J)$. This gives us freedom for handling non-concentrated components. If J is concentrated, the amount of demand that is moved out of J must be comparable to π_J; this will be guaranteed by the configuration LP.

Earth Mover Distance In order to measure the moving cost of satisfying demands using supplies, we define the earth mover distance:

▶ **Definition 8** (Earth Mover Distance)**.** Given a set $V \subseteq R$ with $B = U(V)$, a demand vector $\alpha \in \mathbb{R}^V_{\geq 0}$ and a supply vector $\beta \in \mathbb{R}^B_{\geq 0}$ such that $\sum_{v\in V}\alpha_v \leq \sum_{i\in B}\beta_i$, the earth mover distance from α to β is defined as $\mathrm{EMD}_V(\alpha,\beta) := \inf_f \sum_{v\in V, i\in B} f(v,i)d(v,i)$, where f is over all functions from $V \times B$ to $\mathbb{R}_{\geq 0}$ such that

- $\sum_{i\in B} f(v,i) = \alpha_v$ for every $v \in V$;
- $\sum_{v\in V} f(v,i) \leq \beta_i$ for every $i \in B$.

For some technical reason, we allow some fraction of a supply to be unmatched. From now on, we shall use $\alpha_v = x_{U_v,C}$ to denote the amount of demand at v after the initial moving. For any set $V \subseteq R$ of representatives, we use $\alpha|_V$ to denote the vector α restricted to the coordinates in V.

We now summarize our constructions of local solutions for concentrated and non-concentrated black components, respectively.

▶ **Theorem 9.** *Let $J \in \mathcal{J}^{\mathsf{C}}$ and let $B = U(J)$. Assume Constraints* (6) *to* (13) *are satisfied for B. Then, we can find a distribution $(\phi_{S,\beta})_{S\subseteq B, \beta\in\mathbb{R}^B_{\geq 0}}$ of pairs (S,β), such that*

(9a) $s_\phi := \mathbb{E}_{(S,\beta)\sim\phi}|S| \in [y_B, y_B(1+2\ell\pi_J/x_{B,C})]$*, and* $s_\phi = y_B$ *if* $y_B > 2\ell$*,*

and for every (S,β) in the support of ϕ, we have

(9b) $|S| \in \{\lfloor s_\phi \rfloor, \lceil s_\phi \rceil\}$*;*
(9c) $\beta_i \leq (1+O(1/\ell))u_i$ *if* $i \in S$ *and* $\beta_i = 0$ *if* $i \in B \setminus S$*;*
(9d) $\sum_{i\in S}\beta_i = x_{B,C} = \sum_{v\in J}\alpha_v$*.*

Moreover, the distribution ϕ satisfies

(9e) *the support of ϕ has size at most $n^{O(\ell)}$;*

(9f) $\mathbb{E}_{(S,\beta)\sim\phi}\,\mathrm{EMD}_J(\alpha|_J,\beta) \le O(\ell^4)D_B$.

▶ **Lemma 10.** *Let $\mathcal{J}' \subseteq \mathcal{J}^{\mathsf{N}}$ be a set of non-concentrated black components, $V = \bigcup_{J\in\mathcal{J}'} J$ and $B = U(V)$. Assume there exists $v^* \in R$ such that $d(v, v^*) \le O(\ell^2)L(J)$ for every $J \in \mathcal{J}'$ and $v \in J$. Then, we can find a pair $(S \subseteq B, \beta \subseteq \mathbb{R}^B_{\ge 0})$ such that*

(10a) $|S| \in \{\lceil y_B\rceil, \lceil y_B\rceil + 1\}$;
(10b) $\beta_i \le u_i$ *if* $i \in S$ *and* $\beta_i = 0$ *if* $i \in B \setminus S$;
(10c) $\sum_{i\in S}\beta_i = x_{B,C} = \sum_{v\in V}\alpha_v$;
(10d) $\mathrm{EMD}_V(\alpha|_V,\beta) \le O(\ell^2\ell_2)D_B$.

5 Rounding Algorithm

In this section we describe our rounding algorithm. We start by giving the intuition behind the algorithm. For each concentrated component $J \in \mathcal{J}$, we construct a distribution of local solutions using Theorem 9. We shall construct a partition $\mathcal{V}^{\mathsf{N}}$ of the representatives in $\bigcup_{J\in\mathcal{J}^{\mathsf{N}}} J$ so that each $V \in \mathcal{V}^{\mathsf{N}}$ is the union of some nearby components in $\mathcal{J}^{\mathsf{N}}$. For each set $V \in \mathcal{V}^{\mathsf{N}}$, we apply Lemma 10 to construct a local solution. If we independently and randomly choose a local solution from every distribution we constructed, then we can move all the demands to the open facilities at a small cost, by Property (9f) and Property (10d).

However, we may open more than k facilities, even in expectation. Noticing that the fractional solution opens y_B facilities in a set B, the extra number of facilities come from two places. In Property (9a) of Theorem 9, we may open in expectation $y_B \cdot 2\ell\pi_J/x_{B,C}$ more facilities in B than y_B. Then in Property (10a) of Lemma 10, we may open $\lceil y_B\rceil$ or $\lceil y_B\rceil + 1$ facilities in B. To reduce the number of open facilities to k, we shall shut down (or remove) some already-open facilities and move the demands satisfied by these facilities to the survived open facilities: a concentrated component $J \in \mathcal{J}^{\mathsf{C}}$ is responsible for removing $y_B \cdot 2\ell\pi_J/x_{B,C} < 1$ facilities in expectation; a set $V \in \mathcal{V}^{\mathsf{N}}$ is responsible for removing up to 2 facilities. Lemma 7 allows us to bound the cost of moving demands caused by the removal, provided that the moving distance is not too big. To respect the capacity constraint up to a factor of $1 + \epsilon$, we are only allowed to scale the supplies of the survived open facilities by a factor of $1 + O(1/\ell)$. Both requirements will be satisfied by the forest structure over groups and the fact that each non-leaf group contains $\Omega(\ell)$ fractional opening (Property (5c)). Due to the forest structure and Property (5c), we always have enough open facilities locally that can support the removing of facilities.

In order to guarantee that we always open k facilities, we need to use a dependent rounding procedure for opening and removing facilities. As in many of previous algorithms, we incorporate the randomized rounding procedure into random selections of vertex points of polytopes respecting marginal probabilities. In many cases, a randomized selection procedure can be derandomized since there is an explicit linear objective we shall optimize.

We now formally describe our rounding algorithm. For every group $G \in \mathcal{G}$, we use Λ_G to denote the set of child-groups of G. We construct a partition $\mathbb{J}^{\mathsf{C}}$ of $\mathcal{J}^{\mathsf{C}}$ as follows. For each root group $G \in \mathcal{G}$, we add $G \cap \mathcal{J}^{\mathsf{C}}$ to $\mathbb{J}^{\mathsf{C}}$ if it is not empty. For each non-leaf group $G \in \mathcal{G}$, we add $\bigcup_{G'\in\Lambda_G}(G' \cap \mathcal{J}^{\mathsf{C}})$ to $\mathbb{J}^{\mathsf{C}}$, if it is not empty. We construct the partition $\mathbb{J}^{\mathsf{N}}$ for $\mathcal{J}^{\mathsf{N}}$ in the same way, except that we consider components in $\mathcal{J}^{\mathsf{N}}$. We also define a set $\mathcal{V}^{\mathsf{N}}$ as follows: for every $\mathcal{J}' \in \mathbb{J}^{\mathsf{N}}$, we add $\bigcup_{J\in\mathcal{J}'} J$ to $\mathcal{V}^{\mathsf{N}}$; thus, $\mathcal{V}^{\mathsf{N}}$ forms a partition for $\bigcup_{J\in\mathcal{J}^{\mathsf{N}}} J$. See Figure 2 for the definition of $\mathcal{V}^{\mathsf{N}}$.

In Section 5.1, we describe the procedure for opening a set S^* of facilities, whose cardinality may be larger than k. Then in Section 5.2, we define the procedure remove, which removes one open facility. We wrap up the algorithm in Section 5.3.

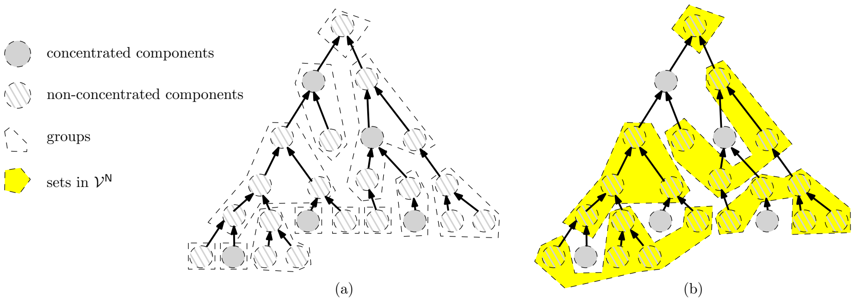

concentrated components
non-concentrated components
groups
sets in $\mathcal{V}^{\mathsf{N}}$
(a)
(b)

Figure 2 Figure (a) gives the forest $\Upsilon^*_{\mathcal{J}}$ over $\mathcal{J}$ and the set $\mathcal{G}$ of groups (denoted by empty polygons). Figure (b) gives $\mathcal{V}^{\mathsf{N}}$: each set $V \in \mathcal{V}^{\mathsf{N}}$ is the union of components in a solid polygon.

5.1 Constructing Initial Set S^* of Open Facilities

In this section, we open a set S^* of facilities, whose cardinality may be larger than k, and construct a supply vector $\beta^* \in \mathbb{R}^F_{\geq 0}$ such that $\beta^*_i = 0$ if $i \notin S^*$. (S^*, β^*) will be the concatenation of all local solutions we constructed.

It is easy to construct local solutions for non-concentrated components. For each set $\mathcal{J}' \in \mathbb{J}^{\mathsf{N}}$ of components and its correspondent $V = \bigcup_{J \in \mathcal{J}'} J \in \mathbb{J}^{\mathsf{N}}$, we apply Lemma 10 to obtain a local solution $\left(S \subseteq U(V), \beta \in \mathbb{R}^{U(V)}_{\geq 0}\right)$. Then, we add S to S^* and let $\beta^*_i = \beta_i$ for every $i \in U(V)$. Notice that $\mathcal{J}'$ either contains a single root black component J, or contains all the non-concentrated black components in the child-groups of some group G. In the former case, the diameter of J is at most $O(\ell)L(J)$ by Property (4a); in the latter case, we let v^* be an arbitrary representative in $\bigcup_{J' \in G} J'$ and then any representative $v \in J, J \in \mathcal{J}'$ has $d(v, v^*) \leq O(\ell^2)L(J)$ by Property (5d). Thus, all the properties in Lemma 10 are satisfied.

For concentrated components, we only obtain distributions of local solutions by applying Theorem 9. For every $J \in \mathcal{J}^{\mathsf{C}}$, we check if Constraints (6) to (13) are satisfied for $B = U(J)$. If not, we return a separation plane for the fractional solution; otherwise we apply Theorem 9 to each component J to obtain a distribution $\left(\phi^J_{S,\beta}\right)_{S \subseteq U(J), \beta \in \mathbb{R}^{U(J)}_{\geq 0}}$. To produce local solutions for concentrated components, we shall use a dependent rounding procedure that respects the marginal probabilities. As mentioned earlier, we shall define a polytope and the procedure randomly selects a vertex point of the polytope.

We let $s_J := s_{\phi_J} := \mathbb{E}_{(S,\beta) \sim \phi^J} |S|$ be the expectation of $|S|$ according to distribution ϕ^J. For notational convenience, we shall use $a \approx b$ to denote $a \in \left[\lfloor b \rfloor, \lceil b \rceil\right]$. Consider the following polytope $\mathcal{P}$ defined by variables $\{\psi^J_{S,\beta}\}_{J \in \mathcal{J}^{\mathsf{C}}, S, \beta}$ and $\{q_J\}_{J \in \mathcal{J}^{\mathsf{C}}}$.[1]

$$\psi^J_{S,\beta}, p_J \in [0,1] \qquad \forall J \in \mathcal{J}^{\mathsf{C}}, S, \beta; \qquad (14)$$

$$\sum_{J \in \mathcal{J}'} q_J \leq 1, \qquad \forall \mathcal{J}' \in \mathbb{J}^{\mathsf{C}}; \qquad (16)$$

$$\sum_{S,\beta} \psi^J_{S,\beta} = 1, \qquad \forall J \in \mathcal{J}^{\mathsf{C}}; \qquad (15)$$

$$\sum_{S,\beta} \psi^J_{S,\beta}|S| - q_J \approx y_{U(J)}, \quad \forall J \in \mathcal{J}^{\mathsf{C}}; \qquad (17)$$

$$\sum_{J \in \mathcal{J}'} \left(\sum_{S,\beta} \psi^J_{S,\beta}|S| - q_J\right) \approx \sum_{J \in \mathcal{J}'} y_{U(J)}, \quad \forall \mathcal{J}' \in \mathbb{J}^{\mathsf{C}}; \qquad (18)$$

$$\sum_{J \in \mathcal{J}^{\mathsf{C}}} \left(\sum_{S,\beta} \psi^J_{S,\beta}|S| - q_J\right) \approx \sum_{J \in \mathcal{J}^{\mathsf{C}}} y_{U(J)}. \qquad (19)$$

[1] For every $J \in \mathcal{J}^{\mathsf{C}}$, we only consider the pairs (S, β) in the support of ϕ^J; thus the total number of variables is $n^{O(\ell)}$.

In the above LP, ψ^J is the indicator vector for local solutions for J and q_J indicates whether J is responsible for removing one facility; if $q_J = 1$, we shall call $\mathsf{remove}(J)$ later. Up to changing of variables, any vertex point of $\mathcal{P}$ is defined by two laminar families of tight constraints and thus $\mathcal{P}$ is integral:

▶ **Lemma 11.** *$\mathcal{P}$ is integral.*

We set $\psi^{*J}_{S,\beta} = \phi^J_{S,\beta}$ and $q^*_J = s_J - y_{U(J)}$ for every $J \in \mathcal{J}^{\mathsf{C}}$ and (S,β). Then,

▶ **Lemma 12.** *(ψ^*, q^*) is a point in polytope $\mathcal{P}$.*

We randomly select a vertex point (ψ, q) of $\mathcal{P}$ such that $\mathbb{E}[\psi^J_{S,\beta}] = \psi^{*J}_{S,\beta} = \phi^J_{S,\beta}$ for every $J \in \mathcal{J}^{\mathsf{C}}, (S,\beta)$, and $\mathbb{E}[q_J] = q^*_J = s_J - y_{U(J)}$ for every $J \in \mathcal{J}^{\mathsf{C}}$. Since ψ is integral, for every $J \in \mathcal{J}$, there is a unique local solution $\left(S \subseteq U(J), \beta \in \mathbb{R}^{U(J)}_{\geq 0}\right)$ such that $\psi^J_{S,\beta} = 1$; we add S to S^* and let $\beta^*_i = \beta_i$ for every $i \in U(J)$.

This finishes the definition of the initial S^* and β^*. Let $\alpha^* = \alpha$ (recall that $\alpha_v = x_{U_v,C}$ is the demand at v after the initial moving, for every $v \in R$) be the initial demand vector. Later we shall remove facilities from S^* and update α^* and β^*. S^*, α^*, β^* satisfy the following properties, which will be maintained as the rounding algorithm proceeds.

(13a) $\sum_{v\in V} \alpha^*_v = \sum_{v \in V} \beta^*_v$ for every $V \in \mathcal{J}^{\mathsf{C}} \cup \mathcal{V}^{\mathsf{N}}$;
(13b) $\sum_{v \in R} \alpha^*_v = |C|$.

Property (13a) is due to Properties (9d) and (10c). Property (13b) holds since $\sum_{v\in R}\alpha^*_v = \sum_{v \in R} x_{U_v,C} = x_{F,C} = |C|$.

5.2 The remove procedure

In this section, we define the procedure remove that removes facilities from S^* and updates α^* and β^*. The procedure takes a set $V \in \mathcal{J}^{\mathsf{C}} \cup \mathcal{V}^{\mathsf{N}}$ as input. If V is a root black component, then we let $G = \{V\}$ be the root group containing V; if V is a non-root concentrated component, let G be the parent group of the group containing V; otherwise V is the union of non-concentrated components in all child-groups of some group, and we let G be this group. Let $V' = \bigcup_{J' \in G} J'$. Before calling $\mathsf{remove}(V)$, we require the following properties to hold:

(14a) $|S^* \cap U(V)| \geq 1$;
(14b) $\left|S^* \cap U(V')\right| \geq \ell - 6$.

While maintaining Properties (13a) and (13b), the procedure $\mathsf{remove}(V)$ will

(15a) remove from S^* exactly one open facility, which is in $U(V \cup V')$,
(15b) not change $\alpha^*|_{R\setminus(V\cup V')}$ and $\beta^*|_{F\setminus U(V \cup V')}$,
(15c) increase α^*_v by at most a factor of $1 + O(1/\ell)$ for every $v \in V \cup V'$ and increase β^*_i by at most a factor of $1 + O(1/\ell)$ for every $i \in U(V \cup V')$.

Moreover,

(15d) the moving cost for converting the old α^* to the new α^* is at most $O(\ell^2)\beta^*_{i^*}L(J)$ for some black component $J \subsetneq V$ and facility $i^* \in U(J)$;
(15e) for every $V'' \in \mathcal{J}^{\mathsf{C}} \cup \mathcal{V}^{\mathsf{N}}$, $\mathrm{EMD}_{V''}\left(\alpha^*|_{V''}, \beta^*|_{U(V'')}\right)$ will be increased by at most a factor of $1 + O(1/\ell)$.

Due to the page limit, we only highlight the key ideas used to implement $\mathsf{remove}(V)$ and leave the formal description to the full version of the paper. Assume V is not a root component. We choose an arbitrary facility $i \in S^* \cap U(V)$. Notice that there are $\Omega(\ell)$ facilities in $S^* \cap U(V')$. If the $\beta^*_i \leq \sum_{v' \in V'} \alpha^*_{v'}/\ell$, then we can shut down i and send the demands that should be sent to i to V'. We only need to increase the supplies in $U(V')$ by a factor of $1 + O(1/\ell)$. Otherwise, we shall shut down the facility $i' \in S^* \cap U(V')$ with the smallest $\beta^*_{i'}$ value. Since there are at least $\Omega(\ell)$ facilities in $U(V')$, we can satisfy the $\beta^*_{i'}$ units of unsatisfied demands using other facilities in $S^* \cap U(V')$. For this i', we have $\beta^*_{i'} \leq O(1)\beta^*_i$. Thus, the total amount of demands that will be moved is comparable to β^*_i. In either case, the cost of redistributing the demands is not too big. When V is a root component, we shall shut down the facility $i' \in S^* \cap U(V)$ with the smallest $\beta^*_{i'}$ value.

5.3 Obtaining the Final Solution

To obtain our final set S^* of facilities, we call the remove procedures in some order. We consider each group G using the top-to-bottom order. That is, before we consider a group G, we have already considered its parent group. If G is a root group, then it contains a single root component J. If $J \in \mathcal{J}^{\mathsf{N}}$, repeat the the following procedure twice: if there is some facility in $S^* \cap U(J)$ then we call remove(J). If $J \in \mathcal{J}^{\mathsf{C}}$ and $q_J = 1$ then we call remove(J). Now if G is a non-leaf group, then do the following. Let $V = \bigcup_{G' \in \Lambda_G, J \in G' \cap \mathcal{J}^{\mathsf{N}}} J$. Repeat the following procedure twice: if there is some facility in $S^* \cap U(V)$ then we call remove(V). For every $G' \in \Lambda_G$ and $J \in G' \cap \mathcal{J}^{\mathsf{C}}$ such that $q_J = 1$ we call remove(J).

▶ **Lemma 16.** *After the above procedure, we have $|S^*| \leq y_F \leq k$.*

By Properties (15b) and (15c), and Constraint (16), our final β^*_i is at most $1 + O(1/\ell)$ times the initial β^*_i for every $i \in V$. Finally we have $\beta^*_i \leq (1 + O(1/\ell))u_i$ for every $i \in F$. Thus, the capacity constraint is violated by a factor of $1 + \epsilon$ if we set ℓ to be large enough.

It remains to bound the expected cost of the solution S^*; this is done by bounding the cost for transferring the original α^* to the final α^*, as well as the cost for matching our final α^* and β^*.

We first focus on the transferring cost. By Property (15e), when we call remove(V), the transferring cost is at most $O(\ell^2)\beta^*_{i^*}L(J)$ for some black component $J \subseteq V$ and i^*. Notice that $\beta^*_{i^*}$ is scaled by at most a factor of $(1 + O(1/\ell))$, we always have $\beta^*_{i^*} \leq (1 + O(1/\ell))\alpha_{U(J),C}$. So, the cost is at most $O(\ell^2)x_{U(J),C}L(J)$. If V is the union of some non-concentrated components, then this quantity is at most $O(\ell^2)\ell_2\pi_J L(J) \leq O(\ell^2\ell_2)D_{U(J)} \leq O(\ell^2\ell_2)D_{U(V)}$. We call remove$(V)$ at most twice, thus the contribution of V to the transferring cost is at most $O(\ell^2\ell_2)D_{U(V)}$. If V is a concentrated component J, then the quantity might be large. However, the probability we call remove(J) is $\mathbb{E}[q_J] = q^*_J = s_J - y_{U(J)} \leq 2\ell y_{U(J)}\pi_J/x_{U(J),C}$ if $y_{U(J)} \leq 2\ell$ and it is 0 otherwise (by Property (9a)). So, the expected contribution of this V to the transferring cost is at most $O(\ell^2)x_{U(J),C}L(J) \times 2\ell y_{U(J)}\pi_J/x_{U(J),C} \leq O(\ell^4)\pi_J L(J) \leq O(\ell^4)D_{U(J)}$ by Lemma 7. Thus, overall, the expected transferring cost is at most $O(\ell^5)D_F = O(\ell^5)\mathsf{LP}$.

Then we consider the matching cost. Since we maintained Property (13a), the matching cost is bounded by $\sum_{V \in \mathcal{J}^{\mathsf{C}} \cup \mathcal{V}^{\mathsf{N}}} \mathrm{EMD}_V(\alpha^*|_V, \beta^*|_{U(V)})$. Due to Property (15e), this quantity has only increased by a factor of $1 + O(1/\ell)$ during the course of removing facilities. For the initial α^* and β^*, the expectation of this quantity is at most $\sum_{J \in \mathcal{J}^{\mathsf{C}}} O(\ell^4)D_{U(J)} + \sum_{V \in \mathcal{V}^{\mathsf{N}}} O(\ell^2\ell_2)D_{U(V)}$ due to Properties (9f) and (10d). This is at most $O(\ell^5)D_F = O(\ell^5)\mathsf{LP}$.

We have found a set S^* of at most k facilities and a vector $\beta^* \in \mathbb{R}^F_{\geq 0}$ such that $\beta^*_i = 0$ for every $i \notin S^*$ and $\beta^*_i \leq (1 + O(1/\ell))u_i$. If we set $\ell = \Theta(1/\epsilon)$ to be large enough, then $\beta^*_i \leq (1 + \epsilon)u_i$. The cost for matching the α-demand vector and the β^* vector is at most $O(\ell^5)\mathsf{LP} = O(1/\epsilon^5)\mathsf{LP}$. Thus, we obtained a $O(1/\epsilon^5)$-approximation for CKM with $(1 + \epsilon)$-capacity violation.

References

1 Karen Aardal, Pieter L. van den Berg, Dion Gijswijt, and Shanfei Li. Approximation algorithms for hard capacitated k-facility location problems. *European Journal of Operational Research*, 242(2):358 – 368, 2015. doi:http://dx.doi.org/10.1016/j.ejor.2014.10.011.

2 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*.

3 V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. 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. URL: http://doi.acm.org/10.1145/380752.380755, doi:http://doi.acm.org/10.1145/380752.380755.

4 Jarosław 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)*.

5 Jarosław 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)*.

6 Jarosław Byrka, Bartosz Rybicki, and Sumedha Uniyal. An approximation algorithm for uniform capacitated k-median problem with $1+\epsilon$ capacity violation, 2015. arXiv:1511.07494. arXiv:arXiv:1511.07494.

7 Robert D. Carr, Lisa K. Fleischer, Vitus J. Leung, and Cynthia A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, pages 106–115, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=338219.338241.

8 M. Charikar and S. Guha. 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*, pages 378–388, 1999.

9 M. Charikar, S. Guha, É. Tardos, and D. 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. URL: http://doi.acm.org/10.1145/301250.301257, doi:http://doi.acm.org/10.1145/301250.301257.

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

11 Sudipto Guha. *Approximation Algorithms for Facility Location Problems*. PhD thesis, Stanford, CA, USA, 2000.

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

13 K Jain and V. 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, 2001. doi:http://doi.acm.org/10.1145/375827.375845.

14 Shanfei Li. 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, 2014.

15 Shi Li. On uniform capacitated k-median beyond the natural LP relaxation. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2015).*

16 Shi Li. Approximating capacitated k-median with $(1+\epsilon)k$ open facilities. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*, pages 786–796, 2016.

17 Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 901–910, New York, NY, USA, 2013. ACM. URL: http://doi.acm.org/10.1145/2488608.2488723, doi:10.1145/2488608.2488723.

18 J. Lin and J. S. Vitter. ϵ-approximations with minimum packing constraint violation (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC), Victoria, British Columbia, Canada*, pages 771–782, 1992.

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