Improved Local Search Based Approximation Algorithm for Hard Uniform Capacitated k-Median Problem

Sapna Grover, Neelima Gupta and Aditya Pancholi
Department of Computer Science, University of Delhi, India
E-mail: sgrover@cs.du.ac.in, ngupta@cs.du.ac.in, apancholi@cs.du.ac.in

Keywords: NP completeness, approximation algorithm, k-median problem

Received: January 7, 2017

In this paper, we study the hard uniform capacitated k - median problem. We give $(5 + \epsilon)$ factor approximation for the problem using local search technique, violating cardinality by a factor of 3. Though better results are known for the problem using LP techniques, local search algorithms are well known to be simpler. There is a trade-off viz-a-viz approximation factor and cardinality violation between our result and the result of Korupolu et al. [10] which is the only result known for the problem using local search. They gave $(1 + \alpha)$ approximation factor with $(5 + 5/\alpha)$ factor loss in cardinality. In a sense, our result is an improvement as they violate the cardinality by more than a factor of 6 to achieve 5 factor in approximation. Though in their result, the approximation factor can be made arbitrarily small, cardinality loss is at least 5 and small approximation factor is obtained at a big loss in cardinality. Thus, we improve upon their result with respect to cardinality.

Povzetek: Obravnavan je NP problem optimiranja iskanja k median in predlagana izvirna rešitev, ki dosega boljše rezultate v določenih primerjavah.

1 Introduction

k - Median Problem is one of the well studied NP-hard optimization problem. The input instance consists of a set of clients, a set of facilities, a non-negative number k and a non-negative cost of connecting a facility to a client. The goal is to select a set of at most k facilities as centers and assign clients to them such that the total cost of serving the clients from centers is minimum.

Several versions of the problem exist in literature with different properties, the most common being Uncapacitated k Median Problem (UkM) and Capacitated k Median Problem (CkM). In the former case, each facility has infinite capacity (*i.e.* there is no limit on the amount of demand it can serve) in comparison to finite capacity in the latter case. In CkM, capacities may be soft or hard. In soft capacitated version, multiple copies of a facility can be opened at a location whereas in case of hard capacities, each facility is either opened at some location or not. Also, the capacities may be uniform or non-uniform. In the former case, all facilities have the same capacity in contrast to the latter one where-in different facilities have different capacities. Another variation of CkM is with respect to assignments of clients to facilities: in un-splittable assignments, the entire demand of a client has to be served by only one facility, in comparison to splittable assignments in which the demand of a client can be split among multiple facilities.

Several techniques have been used to obtain results for the problem. One of the most widely used technique to approximate the problem is LP Rounding ([4, 5, 7, 8, 9, 11, 12, 13, 14]). Charikar *et al.* [7] gave a $20/3$ factor approximation algorithm for UkM, which was further improved to 3.25 factor by Charikar and Li in [8]. Li and Svensson [14] further improved the ratio to $1 + \sqrt{3} + \epsilon$. Their algorithm has a running time of O($n^{(1/\epsilon^2)}$).

Obtaining a constant approximation factor for CkM problem without violating capacity constraint and cardinality constraint is challenging as natural LP of the problem is known to have an unbounded integrality gap. Approximation results violate either capacity constraint or cardinality constraint, or both.

Cardinality violation: Li [12] gave a novel linear program called *rectangle LP* and presented an improved approximation algorithm $(exp(O(1/\epsilon^2)))$ using at most $(1 + \epsilon)k$ facilities for hard uniform CkM problem. The running time of the algorithm is $n^{O(1)}$, where the constant in the exponent does not depend on ϵ. He then extended this result to non-uniform soft capacitated variant of the problem in [13] and gave an $(O(1/\epsilon^2\ log(1/\epsilon)))$ approximation factor bounding softness by a factor of 2. The algorithm has a running time of $n^{O(1/\epsilon)}$.

Capacity violation: Charikar *et al.* [7] gave a 16 factor approximation algorithm for hard uniform CkM violating capacities by a factor of 3 in case of splittable demands and 4 in case of un-splittable demands. In 2015, Byrka *et al.* [4] gave an $O(1/\epsilon)$ approximation algorithm violating capacities by a factor of $(3 + \epsilon)$ for hard non-uniform CkM. Demirci *et al.* [9] improved the approximation ratio to $O(1/\epsilon^5)$ with capacity violation of $(1 + \epsilon)$ for the same version of the problem. The running time of their algorithm is $n^{O(1/\epsilon)}$. Recently, Byrka *et al.* [5] gave an $O(1/\epsilon^2)$ approximation violating capacities by a factor of $(1+\epsilon)$ for hard uniform CkM. The algorithm uses randomized rounding to round a fractional solution to the configuration LP.

Aardal *et al.* [1] exploited the structure of an extreme point solution to give a $(7+\epsilon)$ factor algorithm for hard non-uniform Capacitated k- Facility Location Problem (CkFLP) violating cardinality constraint by a factor of 2. As a special case of CkFLP, their result applies on hard non-uniform CkM with all facility costs being zero. In the same manner, the CkFLP result $(1/\epsilon^2)$ of Byrka *et al.* [4] is applicable on hard uniform CkM. The result violates capacities by a factor of $2+\epsilon$.

The other commonly used technique for the problem is local search [2, 6, 10]. Charikar and Guha [6] gave 4 factor algorithm without violating cardinality constraint for the un-capacitated variant of the problem. Korupolu *et al.* [10] gave O$(1+\epsilon)$ factor approximation algorithm for UkM using at most $3+5/\epsilon$ facilities. Arya *et al.* [2] gave an improvised result of $3+2/p$ factor algorithm for UkM by using p-swaps.

We present a $(5+\epsilon)$ factor algorithm for hard uniform CkM violating the cardinality by a factor of 3 using Local Search. Algorithms based on local search are well known to be simpler as compared to the LP-based algorithms. The only result known for the problem using local search is due to Korupolu *et al.* [10]. They give an algorithm with a trade-off between approximation factor and cardinality loss. They give $(1+\alpha)$ approximation factor with $(5+5/\alpha)$ factor loss in cardinality. To achieve 5 factor in approximation, cardinality violation is more than 6. Though the approximation factor can be made arbitrarily small, cardinality loss is at least 5. Note that small approximation factor is obtained at a big loss in cardinality. For example, for α anything less than 1, cardinality violation is more than 10. Though we somewhat loose on the approximation factor, we surely improve upon the cardinality violation. Thus, there is a trade-off between cardinality violation and approximation factor amongst their result and ours. In particular, we present the following result:

Theorem 1. *There is a polynomial time algorithm that approximates hard uniform capacitated k median problem within* 5 *factor violating the cardinality by a factor of* 3.

High Level Idea: We extend the idea of 'mapping' of Arya *et al.* [2] to the capacitated version of the problem. However, for the capacitated case, mapping needs to be done a little intelligently. Mapping to an almost fully utilized facility may not be able to accommodate all the clients mapped to it and vice-versa. That is, a partially utilized facility may not be able to accommodate the load of an almost fully utilized facility. Thus, mapping is done only between the partially utilized facilities. To ensure that there are sufficient number of partially utilized facilities, we need to assume that we have sufficient number $(3k)$ of opened centers.

2 Notation and preliminaries

2.1 Capacitated k-median problem

In Capacitated k-Median Problem, we are given a set of $\mathcal{F}$ of facilities, a set $\mathcal{C}$ of clients and a real valued distance function c on $\mathcal{F} \cup \mathcal{C}$ in metric space. Each client $j \in \mathcal{C}$ has a non-negative demand d_j and each facility $i \in \mathcal{F}$ has a capacity u_i indicating the amount of demand it can serve. The cost of serving one unit of demand of a client $j \in \mathcal{C}$ from facility $i \in \mathcal{F}$ is denoted as $c(i,\ j)$. The goal is to select a subset $\mathcal{S} \subseteq \mathcal{F}$ of at most k facilities and assign clients to them without violating the capacities such that the total cost of serving all the clients by the opened facilities is minimum.

We consider the hard uniform capacitated k-median version of the problem *i.e.* $u_i = U\ \forall i \in \mathcal{F}$ and at most one instance of a facility can be opened at its location. We assume unit demand at each client *i.e.* $d_j = 1\ \forall j \in \mathcal{C}$.

2.2 Local search paradigm

Given a Problem $\mathtt{P}$, let $\mathtt{S}$ be any arbitrary feasible solution to it. A new solution $\mathtt{S}'$ is called a neighborhood solution of $\mathtt{S}$ if it can be obtained by performing local search operations such as adding one or more facilities $\mathtt{s} \notin \mathtt{S}$ to $\mathtt{S}$ or deleting one or more facilities $\mathtt{s}$ from $\mathtt{S}$ or swapping one or more facilities of $\mathtt{S}$ with facilities not in $\mathtt{S}$. We now formally describe the steps of the algorithm.

The paradigm:

1. Compute an arbitrary feasible solution $\mathtt{S}$ to $\mathtt{P}$.
2. While $\mathtt{S}'$ is a neighborhood solution of $\mathtt{S}$ such that $cost(\mathtt{S}') < cost(\mathtt{S})$ do, set $\mathtt{S} = \mathtt{S}'$.

The solution $\mathtt{S}$ so obtained is called a locally optimal solution. Note that $cost(\mathtt{S}') \geq cost(\mathtt{S})$ for every neighborhood solution $\mathtt{S}'$, for otherwise $\mathtt{S}$ would not have been locally optimal. More formally, a solution $\mathtt{S}$ is said to be locally optimal if no further operation results in improvement in cost.

3 $(5+\epsilon, 3)$ algorithm

For the k-median problems, we define an (a, b) - approximation algorithm as a polynomial-time algorithm that computes a solution using at most bk number of facilities with cost at most a times the cost of an optimal solution using at most k facilities.

We select an arbitrary set of facilities $\mathcal{S} \subseteq \mathcal{F}$ such that $|\mathcal{S}| = 3k$. This set acts as our initial feasible solution. Note that, defining a subset of opened facilities completely specifies a solution. We can obtain the assignments by solving an appropriately defined instance of transportation problem.

The only operation permitted by our algorithm is $swap(s,o)$, defined as follows: $\mathcal{S} = \mathcal{S} - \{s\} + \{o\}$, $o \in \mathcal{F} \setminus \mathcal{S}$, $s \in \mathcal{S}$. Reassign all the clients served by o in optimal to o in our new solution.

We run the local search algorithm on $\mathcal{S}$. Since $\mathcal{S}$ is now locally optimal, for all neighborhood solutions $\mathcal{S}'$ of $\mathcal{S}$, we have, $cost(\mathcal{S}') \geq cost(\mathcal{S})$.

3.1 Analysis

Let $\mathcal{O}$ denote the optimal solution to the problem. We now show that the local optimal solution $\mathcal{S}$ is within 5 factor of the optimal solution *i.e.* $cost(\mathcal{S}) \leq 5cost(\mathcal{O})$.

For a client j, let $\pi_{\mathcal{S}}(j)$ and $\pi_{\mathcal{O}}(j)$ denote the facilities serving j in $\mathcal{S}$ and $\mathcal{O}$ respectively. Also, let S_j and O_j denote the service costs paid by j in $\mathcal{S}$ and $\mathcal{O}$ respectively.

Let $s \in \mathcal{S}$ and $o \in \mathcal{O}$. Consider Figure 1. Let $\mathcal{B}_{\mathcal{S}}(s)$ denote the ball of s, that is, the set of clients served by s in $\mathcal{S}$. Similarly, let $\mathcal{B}_{\mathcal{O}}(o)$ denote the ball of $o \in \mathcal{O}$. Also, let $\mathcal{B}(s,o)$ be the set of clients served by $s \in \mathcal{S}$ and $o \in \mathcal{O}$.

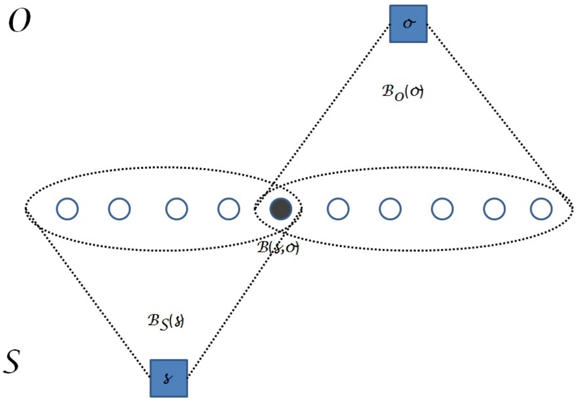

Figure 1: Balls of facilities

To deal with capacities, we classify the facilities in $\mathcal{S}$ based on the number of clients served by them. A facility $s \in \mathcal{S}$ is said to be *heavy* if it serves more than $U/2$ clients in $\mathcal{S}$, else it is said to be *light*. Note that the number of heavy facilities can be at most $2k$. Let $\mathcal{S}_{\mathcal{L}}$ denote the set of light facilities in $\mathcal{S}$. Since $|S| = 3k$, $|\mathcal{S}_{\mathcal{L}}| \geq k$.

Let $\mathcal{B}^L_{\mathcal{O}}(o)$ be the set of clients served by o in optimal and by light facilities in $\mathcal{S}$ and $\mathcal{M}_o = |\mathcal{B}^L_{\mathcal{O}}(o)|$. We say that a facility $s \in \mathcal{S}_{\mathcal{L}}$ *dominates* o, if it serves more than half the clients served by light facilities in $\mathcal{S}$ and by $o \in \mathcal{O}$, *i.e.* $\mathcal{B}(s,o) > \mathcal{M}_o/2$. A facility belonging to $\mathcal{S}_{\mathcal{L}}$ is called *bad* if it dominates more than one facilities in $\mathcal{O}$, it is called *good* if it dominates exactly one facility in $\mathcal{O}$, else it is called *nice*

We now devise a $1-1$ and onto mapping $\tau : \mathcal{B}^L_{\mathcal{O}}(o) \rightarrow \mathcal{B}^L_{\mathcal{O}}(o)$. Order the clients in $\mathcal{B}^L_{\mathcal{O}}(o)$ as $j_0, j_1, ..., j_{\mathcal{M}_o - 1}$ such that for every $s \in S$ with a nonempty $\mathcal{B}(s,o)$, the clients in $\mathcal{B}(s,o)$ are consecutive; that is, there exists r, s, $0 \leq r \leq s \leq \mathcal{M}_o - 1$, such that $\mathcal{B}(s,o) = \{j_r, ..., j_s\}$. Define $\tau(j_p) = (j_q)$, where $q = (p + \lfloor \mathcal{M}_o/2 \rfloor)\ modulo\ \mathcal{M}_o$. Consider Figure 2a which shows the set $\mathcal{B}_{\mathcal{O}}(o)$. The corresponding mapping is shown in Figure 2b.

The following claim holds on mapping:

Claim 1. *If $s \in \mathcal{S}_{\mathcal{L}}$ does not dominate o, then $\tau(\mathcal{B}(s,o)) \cap \mathcal{B}(s,o) = \phi$.*

Proof. For contradiction, assume that both j_p, $\tau(j_p) = j_q \in \mathcal{B}(s,o)$ for some s, where $|\mathcal{B}(s,o)| \leq \mathcal{M}_o/2$. If $q = p + \lfloor \mathcal{M}_o/2 \rfloor$, then $|\mathcal{B}(s,o)| \geq q - p + 1 = \lfloor \mathcal{M}_o/2 \rfloor + 1 > \mathcal{M}_o/2$. If $q = p + \lfloor \mathcal{M}_o/2 \rfloor - \mathcal{M}_o$, then $|\mathcal{B}(s,o)| \geq p - q + 1 = \mathcal{M}_o - \lfloor \mathcal{M}_o/2 \rfloor + 1 > \mathcal{M}_o/2$. In either case, we have a contradiction, and hence mapping τ satisfies the claim. □

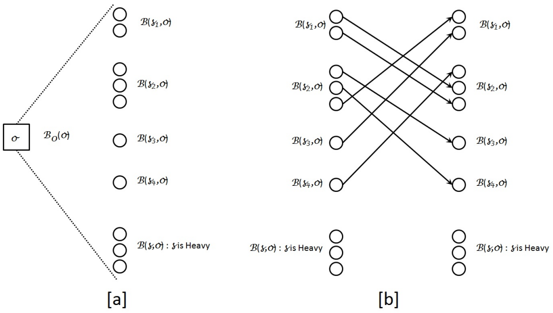

Figure 2: Mapping

The notion of dominate can be used to construct a bipartite graph $H = (\mathcal{S}, \mathcal{O}, E)$. For each facility in $\mathcal{S}_{\mathcal{L}}$, we have a vertex on the $\mathcal{S}$-side and for each facility in $\mathcal{O}$, we have a vertex on the $\mathcal{O}$-side. We add an edge between $s \in \mathcal{S}_{\mathcal{L}}$ and $o \in \mathcal{O}$ if s dominates o. Note that the degree of each vertex on $\mathcal{O}$-side is at most one while the vertices on the $\mathcal{S}$-side can have degree up to k.

We now consider all k swaps, one for each facility in $\mathcal{O}$. If $s \in \mathcal{S}_{\mathcal{L}}$ is good, then we consider the $swap(s,o)$, where o is the facility in $\mathcal{O}$ dominated by s. Let λ be the number of facilities in $\mathcal{O}$ that did not participate in the above swaps. Then the total number of bad and nice facilities in $\mathcal{S}_{\mathcal{L}}$ is at least λ and at least $\lambda/2$ of them must be nice. The remaining λ facilities in $\mathcal{O}$ get swapped with the nice facilities in $\mathcal{S}_{\mathcal{L}}$ such that each nice facility is considered in at most two swaps. The bad facilities are not considered for swapping. The swaps considered above satisfy the following properties:

1. Each $o \in \mathcal{O}$ is considered in exactly one swap.
2. Facilities in $\mathcal{S} \setminus \mathcal{S}_{\mathcal{L}}$ are not considered in any swap operation.
3. Bad facilities in $\mathcal{S}_{\mathcal{L}}$ are not considered in any swap operation.
4. Each nice facility $s \in \mathcal{S}_{\mathcal{L}}$ is considered in at most two swap operations.
5. If $swap(s,o)$ is considered then s does not dominate any facility $o' \neq o : o' \in \mathcal{O}$.

Lemma 1. *Let $cost(\mathcal{S})$ denote the cost of the local optimal solution $\mathcal{S}$ and, $cost(\mathcal{O})$ denote the cost of the global optimal solution $\mathcal{O}$. Then, $cost(\mathcal{S}) \leq 5cost(\mathcal{O})$.*

Proof. Consider $swap(s, o)$. Let $j \in \mathcal{B}_{\mathcal{S}}(s)$. We first reassign the clients in $\mathcal{B}_{\mathcal{S}}(s)$.

1. If $j \in \mathcal{B}_{\mathcal{O}}(o)$, assign j to o.

2. If $j \notin \mathcal{B}_{\mathcal{O}}(o)$, assign j to $s' \in \mathcal{S}_{\mathcal{L}}$ such that $\tau(j) = j'$ and $j' \in \mathcal{B}_{\mathcal{S}}(s')$.

In case 1, the change in cost is given by $(O_j - S_j)$. In case 2, the change in cost is $(c(j,\ s') - S_j)$. Let $j \in \mathcal{B}_{\mathcal{O}}(o')$. From triangle inequality, we get $c(j,\ s') \leq c(j,\ o') + c(o',\ \tau(j)) + c(\tau(j),\ s') = O_j + O_{\tau(j)} + S_{\tau(j)}$.

As $\mathcal{S}$ is a locally optimal solution, we have

$$\sum_{j \in \mathcal{B}_{\mathcal{S}}(s) \cap \mathcal{B}_{\mathcal{O}}(o)} (O_j - S_j) + \sum_{j \in \mathcal{B}_{\mathcal{S}}(s) \setminus \mathcal{B}_{\mathcal{O}}(o)} (O_j + O_{\tau(j)} + S_{\tau(j)} - S_j) > 0 \quad (1)$$

Each facility $o \in \mathcal{O}$ is considered in exactly one swap operation. Thus the first term of inequality when added over all k swaps gives exactly $cost(\mathcal{O}) - cost(\mathcal{S})$. Each $s \in \mathcal{S}$ is considered in at most two swaps. The second term of inequality when added over all k swaps is no greater than $2(O_j + O_{\tau(j)} + S_{\tau(j)} - S_j)$. As τ is a $1-1$ and onto mapping, $\sum_{j \in \mathcal{C}} O_j = \sum_{j \in \mathcal{C}} O_{\tau(j)}$ and $\sum_{j \in \mathcal{C}} (S_{\tau(j)} - S_j) = 0$. Thus, $2(O_j + O_{\tau(j)} + S_{\tau(j)} - S_j) = 4cost(\mathcal{O})$. Combining the two terms, we get $cost(\mathcal{O}) - cost(\mathcal{S}) + 4cost(\mathcal{O}) \geq 0$. Thus, $cost(\mathcal{S}) \leq 5cost(\mathcal{O})$. □

In the algorithm presented so far, we move to a new solution if it gives some improvement in the cost, however small that improvement may be. This may lead to an algorithm taking lot of time. To ensure that the algorithm terminates in polynomial time, a local search step is performed only when the cost of the current solution $\mathcal{S}$ is reduced by at least $\frac{cost(\mathcal{S})}{p(n,\epsilon)}$, where n is the size of the problem instance and $p(n, \epsilon)$ is an appropriate polynomial in n and $1/\epsilon$ for a fixed $\epsilon > 0$. This modification in the algorithm incurs a cost of additive ϵ in the approximation factor.

It is easy to see that if we have $3.5k$ facilities then the total number of bad and nice facilities in $\mathcal{S}_{\mathcal{L}}$ is at least $\lambda + k/2$ and at least $(\lambda + k)/2 \geq \lambda$ of them must be nice. The remaining λ facilities in $\mathcal{O}$ get swapped with the nice facilities in $\mathcal{S}_{\mathcal{L}}$ such that each nice facility is considered in at most one swap. This saves us factor 2 coming from the second term of equation (1). Thus, we get $(3 + \epsilon, 3.5)$ algorithm. Also, using p-swaps of Arya *et al.* [2], we can get $(3 + 2/p, 3)$ algorithm.

4 Conclusion and future work

We gave a $(5 + \epsilon)$ factor approximation algorithm for hard uniform capacitated k median problem using local search technique, violating cardinality by a factor of 3. It improves upon the existing results known for the problem using local search, with respect to cardinality violation. It would be interesting to obtain a constant factor algorithm reducing the cardinality violation to $(1 + \epsilon)$. Though such a result is known using LP-techniques, it would be interesting to obtain similar result using local search. Another direction to extend the work would be to consider the non-uniform capacitated version of the problem using local search.

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. https://doi.org/10.1016/j.ejor.2014.10.011

[2] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, Heraklion, Crete, Greece*, pages 21–29, 2001. https://doi.org/10.1145/380752.380755

[3] Manisha Bansal. Approximation algorithms for facility location problems- https://drive.google.com/file/d/0bxmghjb2ede3nkvka2rzcfnethm/view. In *PhD Thesis, October, 2013.*

[4] Jaroslaw Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor approximation algorithms for hard capacitated *k*-median problems. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 722–736, 2015. https://doi.org/10.1137/1.9781611973730.49

[5] Jaroslaw Byrka, Bartosz Rybicki, and Sumedha Uniyal. An approximation algorithm for uniform capacitated k-median problem with (1 + ϵ) capacity violation. In *Integer Programming and Combinatorial Optimization - 18th International Conference, IPCO 2016, Liège, Belgium, June 1-3, 2016, Proceedings*, pages 262–274, 2016. https://doi.org/10.1007/978-3-319-33461-5_22

[6] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS), New York, NY, USA*, pages 378–388, 1999. https://doi.org/10.1137/s0097539701398594

[7] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem (extended abstract). In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 1–10, 1999. https://doi.org/10.1145/301250.301257

[8] Moses Charikar and Shi Li. A dependent lp-rounding approach for the k-median problem. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, pages 194–205, 2012. https://doi.org/10.1007/978-3-642-31594-7_17

[9] H. Gökalp Demirci and Shi Li. Constant approximation for capacitated k-median with $(1 + \epsilon)$ - capacity violation. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 73:1–73:14, 2016.

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

[11] Shanfei Li. An Improved Approximation Algorithm for the Hard Uniform Capacitated k-median Problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014), Germany*, pages 325–338. https://doi.org/10.1016/j.ejor.2014.10.011

[12] Shi Li. On uniform capacitated k-median beyond the natural LP relaxation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 696–707, 2015. https://doi.org/10.1137/1.9781611973730.47

[13] Shi Li. Approximating capacitated k-median with $(1 + \epsilon)k$ open facilities. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 786–796, 2016. https://doi.org/10.1137/1.9781611974331.ch56

[14] Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. In *Proceedings of Symposium on Theory of Computing Conference (STOC), Palo Alto, CA, USA*, pages 901–910, 2013. https://doi.org/10.1145/2488608.2488723