A Unified Framework of FPT Approximation Algorithms for Clustering Problems

Qilong Feng
School of Computer Science and Engineering, Central South University, Changsha, China
csufeng@csu.edu.cn

Zhen Zhang
School of Computer Science and Engineering, Central South University, Changsha, China
csuzz@foxmail.com

Ziyun Huang
Department of Computer Science and Software Engineering, Penn State Erie,
The Behrend College, PA, USA
zxh201@psu.edu

Jinhui Xu
Department of Computer Science and Engineering, State University of New York at Buffalo,
NY, USA
jinhui@cse.buffalo.edu

Jianxin Wang
School of Computer Science and Engineering, Central South University, Changsha, China
jxwang@csu.edu.cn

Abstract

In this paper, we present a framework for designing FPT approximation algorithms for many k-clustering problems. Our results are based on a new technique for reducing search spaces. A reduced search space is a small subset of the input data that has the guarantee of containing k clients close to the facilities opened in an optimal solution for any clustering problem we consider. We show, somewhat surprisingly, that greedily sampling $O(k)$ clients yields the desired reduced search space, based on which we obtain FPT(k)-time algorithms with improved approximation guarantees for problems such as capacitated clustering, lower-bounded clustering, clustering with service installation costs, fault tolerant clustering, and priority clustering.

2012 ACM Subject Classification Theory of computation → Facility location and clustering

Keywords and phrases clustering, approximation algorithms, fixed-parameter tractability

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.5

Funding This work was supported by National Natural Science Foundation of China (61872450, 61672536, 61828205, and 61802441), Hunan Provincial Key Lab on Bioinformatics, and Hunan Provincial Science and Technology Program (2018WK4001).

1 Introduction

Clustering is a frequently encountered problem in computer science and has applications in many fields related to unsupervised learning. Different objectives have been introduced to estimate the quality of clustering results. Among them, the k-median and k-means cost functions are perhaps the most popular versions. In the k-median problem, we are given a set of clients and a set of facilities located in a metric space. The goal is to open a set of no more than k facilities, such that the sum of distance from each client to its nearest opened facility is minimized. The k-means problem is the same as the k-median problem, except that the clustering cost is measured by the squared distance for each client to its corresponding facility.

Both the k-median and k-means problems are NP-hard [21]. This leads to considerable efforts on obtaining approximation algorithms for the problems. The first constant factor approximation for k-median was given by Charikar et al. [12], who showed that LP-rounding yields a $6\frac{2}{3}$-approximation. The approximation guarantee was later improved by a series of work [26, 25, 5, 31] to the current best ratio of $2.675 + \epsilon$ [11]. For the k-means problem, Gupta and Tangwongsan [23] gave that a simple local search algorithm yields a $25 + \epsilon$ approximation. Ahmadian et al. [3] later gave a $(9 + \epsilon)$-approximation algorithm using a primal-dual method.

A commonly used way for relaxing the k-median and k-means problems is to assume that k is a fixed parameter. It was known that both problems are W[2]-hard if parameterized by k [21], implying that it is impossible to exactly solve the problems in FPT(k) time. However, this negative result does not rule out the possibility of obtaining better approximation ratios in FPT(k) time. Cohen-Addad et al. [14] showed that FPT(k)-time algorithms based on coresets yield $(1 + 2/e + \epsilon)$-approximation and $(1 + 8/e + \epsilon)$-approximation for k-median and k-means, respectively. Similar improvements have been achieved for other clustering problems such as capacitated k-median [15, 2], capacitated k-means [15], facility location [14], lower-bounded k-median [7, 8], and k-median with outliers [20] using FPT algorithms.

A crucial property used in the algorithms for the k-median and k-means problems is that all clients of a cluster lie fully in the Voronoi cell of the corresponding facility. However, in many applications involving clustering, clients are correlated and their clustering needs some additional constraints to ensure the legitimacy of the clustering result. One such example is the capacitated k-median problem, where the size of each cluster should be less than the capacity of the corresponding facility. In such applications, the clusters are no longer obtained from the Voronoi cell of the opened facilities, which means that the partition of clients and locations of the opened facilities in an optimal solution might be quite arbitrary. We curtly remark on the commonly used techniques to show the obstacles in obtaining FPT approximation algorithms in such settings.

- Several sampling based FPT(k)-time algorithms yield $(1+\epsilon)$-approximation for clustering problems in Euclidean space [30, 27, 28, 17, 9, 19]. The idea behind these algorithms is to exploit the fact that the location of the corresponding facility of a cluster can be approximated by the centroid of a small subset sampled from the cluster. The difficulty in extending this approach to the problems in the considered class is due to the constraints associated with the facilities and clients. For instance, in the lower-bounded k-median problem, the facilities have non-uniform lower bounds on the number of assigned clients. In such settings, it is quite hard to identify the clients from a cluster and the corresponding facility based on such approximate location. Moreover, these algorithms rely heavily on the properties of Euclidean space and seem difficult to be applied to general metric spaces. Indeed, assuming the Gap-Exponential Time Hypothesis, Cohen-Addad et al. [14] showed that in general metric spaces, any $(1 + 2/e - \epsilon)$-approximation algorithm for k-median and $(1 + 8/e - \epsilon)$-approximation algorithm for k-means have running time no better than $n^{k^{f(\epsilon)}}$.
- Coreset construction is a commonly used technique for designing FPT algorithms for clustering problems [13, 15, 14]. A coreset is a set of weighted clients such that one can get an approximation solution by minimizing the objective function on the coreset. Space decomposition is a frequently used approach for constructing such coresets. In [13, 15, 14], the space partition is given by a set of rings centered at a bi-criteria constant factor approximation to the problems. The clients located in each ring are replaced by a single client whose weight equals the number of clients in the ring. This yields coresets of size

poly$(\log n, k)$ for the considered problems. FPT-time approximation algorithms can then be obtained by performing enumeration methods on such coresets. Unfortunately, it is still unknown that whether such approach works for many widely-studied clustering problems, such as lower-bounded clustering [36], and fault tolerant clustering [24]. The challenge lies in the fact that the loss in the approximation ratio induced by each ring is difficult to analysis due to the additional constraints. How to approximate the input data by a small coreset in these problems is not clear.

1.1 Our techniques and results

We propose a *reduced search space* technique to design FPT approximation algorithms for clustering problems. Let $\mathcal{D}$ and $\mathcal{F}$ denote the given sets of clients and facilities respectively, where $|\mathcal{D}\cup\mathcal{F}| = n$. For each $i, j \in \mathcal{D}\cup\mathcal{F}$, let $\Delta(i, j)$ denote the distance and squared distance from i to j for k-median and k-means, respectively. For each $i \in \mathcal{D}\cup\mathcal{F}$ and $\mathcal{A} \subset \mathcal{D}\cup\mathcal{F}$, define $\Phi(i, \mathcal{A}) = \min_{j\in\mathcal{A}} \Delta(i, j)$ and $\Delta(\mathcal{A}, i) = \sum_{j\in\mathcal{A}} \Delta(j, i)$. The reduced search space can be formally defined as follows.

▶ **Definition 1** ((k, ϵ)-reduced search space)**.** *Given a set $\mathcal{D}$ of clients and a set $\mathcal{F}$ of facilities in a metric space, an integer $k > 0$, and a real number $0 < \epsilon \leq 1$, a subset $\mathcal{H} \subset \mathcal{D}$ is called a (k, ϵ)-reduced search space if for any partition $\mathbb{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ of $\mathcal{D}$ and any subset $\mathcal{C} = \{c_1, \ldots, c_k\}$ of $\mathcal{F}$, we have $\sum_{t=1}^{k} |\mathcal{D}_t|\Phi(c_t, \mathcal{H}) \leq (1 + \epsilon)\sum_{t=1}^{k} \Delta(\mathcal{D}_t, c_t)$ with constant probability.*

A reduced search space is a client set that has the guarantee of containing a set of clients close to the facilities opened in the unknown optimal solution. We consider the uniform capacitated k-median problem [15] for an example to illustrate the power of such a reduced search space. Let $\mathbb{D}^* = \{\mathcal{D}^*_1, \ldots, \mathcal{D}^*_k\}$ denote the partition of $\mathcal{D}$ and $\mathcal{C}^* = \{c^*_1, \ldots, c^*_k\}$ be the set of opened facilities in an optimal solution, where the clients from $\mathcal{D}^*_t$ are assigned to c^*_t for each $t \in [k]$. Let $\mathcal{H}$ be a (k, ϵ)-reduced search space. Define $h_t = \arg\min_{j\in\mathcal{H}} \Delta(j, c^*_t)$ and $c_t = \arg\min_{i\in\mathcal{F}} \Delta(i, h_t)$. With constant probability, we have

$$\sum_{t=1}^{k} \Delta(\mathcal{D}^*_t, c_t) \leq \sum_{t=1}^{k} \Delta(\mathcal{D}^*_t, h_t) + \sum_{t=1}^{k} |\mathcal{D}^*_t|\Delta(c^*_t, h_t) \leq \sum_{t=1}^{k} \Delta(\mathcal{D}^*_t, c^*_t) + 2\sum_{t=1}^{k} |\mathcal{D}^*_t|\Delta(c^*_t, h_t)$$
$$\leq (3 + 2\epsilon)\sum_{t=1}^{k} \Delta(\mathcal{D}^*_t, c^*_t),$$

where the first two steps follow from triangle inequality, and the last step is due to the definition of reduced search space. This implies that a $(3 + O(\epsilon))$-approximation solution can be easily found by enumerating the nearest facility to each $j \in \mathcal{H}$ with constant probability, which takes $|\mathcal{H}|^k n^{O(1)}$ time. The success probability can be boosted to $1 - \lambda^{-1}$ for any $\lambda > 1$ by repeatedly running the algorithm for $O(\log \lambda)$ times.

In this paper, we show the advantages of reduced search spaces in the task of designing FPT algorithms, which include

- We show that the reduced search space is quite universal, based on which a unified framework for designing FPT approximation algorithms for a set of clustering problems is given. These problems include lower-bounded clustering [36], capacitated clustering [10], clustering with service installation costs [34], fault tolerant clustering [24], and priority clustering [29]. Our framework combines the reduced search space with a problem-specific selection algorithm to obtain the desired approximation solution. Note that the selection algorithms for the problems may not be as trivial as shown in the example (e.g., k-median with non-uniform capacities). The selection algorithms are given in Section 3.

- The running time of our algorithms depend heavily on the size of the reduced search space. It seems intuitively that a reduced search space should be quite large, since there are $n^{O(k)}$ choices of the partition $\mathbb{D}$ of $\mathcal{D}$ and the set $\mathcal{C}$ of opened facilities. The main challenge of applying reduced search space to design FPT algorithms is how to construct a small reduced search space. In this paper, we show, somewhat surprisingly, that a set of $O(k\epsilon^{-3})$ clients greedily sampled from $\mathcal{D}$ is a (k, ϵ)-reduced search space.

We summarize our results in Table 1 and the following paragraphs. We expect that our framework will be useful in other clustering problems and of broader interest.

1. **Capacitated clustering.** In this problem, each facility has a capacity, and the number of clients assigned to each open facility should be no more than the capacity of the facility. The current best approximation ratios for both the problems of capacitated k-median and k-means are $O(\log k)$. Our framework yields a $(3 + \epsilon)$-approximation algorithm for capacitated k-median and a $(9 + \epsilon)$-approximation algorithm for capacitated k-means that run in FPT(k) time. The approximation ratios are the same as that of the coresets-based FPT algorithms given by Cohen-Addad and Li [15]. However, our algorithms are simpler and only use random sampling.
2. **Lower-bounded clustering.** This problem generalizes the standard clustering problem in that each facility is associated with a lower bound, and the number of clients assigned to each facility should be more than the corresponding lower bound. Bera et al. [7, 8] gave a $(3.736 + \epsilon)$-approximation algorithm for the k-median with uniform lower bounds problem, which runs in FPT(k) time. The problem remains elusive for non-uniform lower bounds. For this more general case, our framework gives a $(3+\epsilon)$-approximation algorithm for lower-bounded k-median and a $(9 + \epsilon)$-approximation algorithm for lower-bounded k-means.
3. **Clustering with service installation costs.** This problem is frequently encountered in scenarios where clients require different kinds of services, and has applications in many fields such as network design [35] and data management [6]. For this problem, we are given a set of services. Each client is associated with a specific service, and we have a service installation cost for each facility-service pair that indicates the cost for installing the service at the facility (the service installation costs must satisfy a given ordering, which is detailed in Section 3.3). The goal is to open no more than k facilities, install services at the opened facilities, and assign each client to an opened facility where the associated service is installed, such that the sum of the assignment cost and the service installation cost is minimized. Shmoys et al. [34] showed that a primal-dual algorithm yields an 18-approximation for the k-median objective for the case where the cost for installing a service at each facility is uniform. For non-uniform service installation costs, we show that our framework gives a $(4.39+\epsilon)$-approximation algorithm and a $(19.53+\epsilon)$-approximation algorithm that run in FPT(k) time for the k-median and k-means objectives, respectively.
4. **Fault tolerant clustering.** For this problem, each client j is associated with a parameter $l_j \geq 1$, and counts the sum of its distances to the l_j nearest opened facilities as its assignment cost. Hajiaghayi et al. [24] gave a 93-approximation algorithm for the k-median objective. Our framework yields a FPT(k)-time $(3 + \epsilon)$-approximation algorithm and a FPT(k)-time $(9 + \epsilon)$-approximation algorithm for the k-median and k-means objectives, respectively.
5. **Priority clustering.** For this problem, each client has a priority and can only be assigned to a facility with the same or higher priority. Kumar and Sabharwal [29] introduced an $O(1)$-approximation algorithm for priority k-median for the case where the clients have no

Table 1 The results for the studied clustering problems. The results of this paper are marked with ⋆⋆.

Problems	Approx.	Time	Constraints	Ref.
Capacitated k-median	$O(\log k)$	poly(n)	No constraint	[2]
	$3+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	[15]
	$3+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆
Capacitated k-means	$O(\log k)$	poly(n)	No constraint	[18]
	$9+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	[15]
	$9+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆
Lower-bounded k-median	516	poly(n)	Uniform lower bounds	[22]
	$3.736+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	Uniform lower bounds	[7, 8]
	$3+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆
Lower-bounded k-means	$9+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆
k-median with service installation costs	18	poly(n)	Uniform installation costs	[34]
	$4.39+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆
k-means with service installation costs	$19.53+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆
Fault tolerant k-median	93	poly(n)	No constraint	[24]
	$3+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆
Fault tolerant k-means	$9+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆
Priority k-median	$O(1)$	poly(n)	No more than two priorities	[29]
	$3+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆
Priority k-means	$9+\epsilon$	$(k\epsilon^{-1})^{O(k)}n^{O(1)}$	No constraint	⋆⋆

more than two different priorities. The approximation ratio is implicit but seems to be a very large number. We give a $(3+\epsilon)$-approximation algorithm for priority k-median and a $(9+\epsilon)$-approximation algorithm for priority k-means that run in FPT(k) time using our framework.

2 A reduced search space for k-median

In this section, we prove our main technical result for the k-median objective: the construction of a reduced search space of size $O(k\epsilon^{-3})$. The analysis can be easily adapted to the k-means objective to get the desired FPT approximation algorithms. Let $\mathcal{I} = (\mathcal{D}, \mathcal{F}, k)$ denote an instance of k-median, where $\mathcal{D}$ is a set of clients and and $\mathcal{F}$ is a set of facilities in a metric space. Let $n = |\mathcal{D} \cup \mathcal{F}|$. The reduced search space is constructed using D-sampling [4]. It samples a client with probability proportional to its distance to the nearest previously sampled client, which can be defined as follows.

▶ **Definition 2** (D-sampling [4]). *Given a set $\mathcal{D}$ of clients and another set $\mathcal{H} \subset \mathcal{D}$ of clients, D-sampling is a sampling method which samples a client $j \in \mathcal{D}$ with respect to $\mathcal{H}$ with probability proportional to $\Phi(j, \mathcal{H})$. For the case where $\mathcal{H} = \emptyset$, D-sampling samples a client from $\mathcal{D}$ uniformly at random.*

Given an instance $\mathcal{I} = (\mathcal{D}, \mathcal{F}, k)$ of k-median and a real number $0 < \epsilon \le 1$, we sample a set $\mathcal{H}$ of $O(k\epsilon^{-3})$ clients from $\mathcal{D}$ using D-sampling, as detailed in Algorithm 1. The algorithm runs in $O(nk\epsilon^{-3})$ time. Recall that $\Phi(i, \mathcal{A}) = \min_{j\in\mathcal{A}} \Delta(i, j)$ and $\Delta(\mathcal{A}, i) = \sum_{j\in\mathcal{A}} \Delta(j, i)$ for each $i \in \mathcal{D} \cup \mathcal{F}$ and $\mathcal{A} \subset \mathcal{D} \cup \mathcal{F}$. Given two sets $\mathcal{A} \subset \mathcal{D} \cup \mathcal{F}$ and $\mathcal{B} \subset \mathcal{D} \cup \mathcal{F}$, define $\Delta(\mathcal{A}, \mathcal{B}) = \min_{i\in\mathcal{B}} \Delta(\mathcal{A}, i)$, and let $\Phi(\mathcal{A}, \mathcal{B}) = \sum_{j\in\mathcal{A}} \Phi(j, \mathcal{B})$.

Algorithm 1 Construct a reduced search space.

Input: An instance $\mathcal{I} = (\mathcal{D}, \mathcal{F}, k)$ of k-median and a real number $0 < \epsilon \leq 1$;
Output: A set $\mathcal{H} \subset \mathcal{D}$ of $O(k\epsilon^{-3})$ clients;

```
sample a client j ∈ D uniformly at random, and let H_1 = {j};
for t = 2 to 360kε^{-3} do
    sample a client j ∈ D using D-sampling with respect to H_{t-1};
    H_t ⇐ H_{t-1} ∪ {j};
    t ⇐ t + 1;
return H ⇐ H_t.
```

The following result is known as Chernoff Bound [32].

▶ **Lemma 3.** *Let $a_1, \ldots, a_q$ be q independent random variables with values of 1 or 0, where a_i takes 1 with probability at least p for $i = 1, \ldots, q$. Let $a = \sum_{i=1}^{q} a_i$. For any real number $0 < \lambda < 1$, we have $\mathbf{Pr}[a < (1-\lambda)pq] < e^{-\frac{\lambda^2 pq}{2}}$.*

We will also use the following well known algebraic fact, which is called Abel's lemma [1].

▶ **Lemma 4.** *For two arbitrary sequences $\{a_t\}$ and $\{b_t\}$, we have $\sum_{t=1}^{N} a_t b_t = S_N b_N - \sum_{t=1}^{N-1} S_t(b_{t+1} - b_t)$, where $S_t = \sum_{t'=1}^{t} a_{t'}$.*

2.1 A general idea

Let $\mathbb{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ be an arbitrary partition of $\mathcal{D}$ and $\mathcal{C} = \{c_1, \ldots, c_k\}$ be an arbitrary subset of $\mathcal{F}$. Given a cluster $\mathcal{D}_t \in \mathbb{D}$, define $\mathbf{b}_\alpha(\mathcal{D}_t) = \{j \in \mathcal{D}_t : \Delta(j, c_t) \leq \alpha r_t\}$ for $\alpha > 0$, where $r_t = \Delta(\mathcal{D}_t, c_t)/|\mathcal{D}_t|$. This is the set of clients from $\mathcal{D}_t$ that lie on a closed ball centred at c_t with radius αr_t. If the value of α is small enough, then any client from $\mathbf{b}_\alpha(\mathcal{D}_t)$ is near to c_t. Our task is to find a client from $\mathbf{b}_\alpha(\mathcal{D}_t)$ for each cluster $\mathcal{D}_t \in \mathbb{D}$. The challenge is that we need to ensure that the clients from $\mathbf{b}_\alpha(\mathcal{D}_t)$ can be selected from the entire input data. By the definition of D-sampling, we know that if the clients from $\mathcal{D}_t$ are far from the set of previously sampled clients, they will be sampled with a high probability, even if $|\mathcal{D}_t|$ is very small compared to $|\mathcal{D}|$. We are able to show that $\mathbf{b}_\alpha(\mathcal{D}_t)$ contains a substantial portion of $\mathcal{D}_t$, such that the points from $\mathbf{b}_\alpha(\mathcal{D}_t)$ have a good chance to be sampled. However, using D-sampling causes yet another problem. If the clients in $\mathcal{D}_t$ are close to the previously sampled clients, then the probability of picking a client from $\mathbf{b}_\alpha(\mathcal{D}_t)$ with D-sampling will be close to 0. This problem is quite obvious for the clustering problems such as lower-bounded k-median and priority k-median, where the optimal clusters can be an arbitrary partition of the input data, and thus the clients in $\mathcal{D}_t$ are not guaranteed to be far from other clusters. Our method to deal with this problem is to use the replication of a previously sampled client to approximate facility c_t. We show that such a substitute works well if the probability of sampling points from $\mathcal{D}_t$ is very small. Let $\mathcal{H}$ denote the set of $O(k\epsilon^{-3})$ clients generated by Algorithm 1. The ideas above lead to the proof of the following result.

▶ **Theorem 5.** *$\mathcal{H}$ is a (k, ϵ)-reduced search space.*

2.2 Proof of Theorem 5

Let $\mathbb{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ be an arbitrary partition of $\mathcal{D}$. Let $\mathcal{H}_s$ denote the set of clients sampled with Algorithm 1 after the s-th iteration. Let $\mathcal{H}_0 = \mathbb{O}_0 = \emptyset$. Define $\mathbb{O}_s = \{\mathcal{D}_t \in \mathbb{D} : \Phi(c_t, \mathcal{H}_s) \leq (1 + \frac{\epsilon}{2})r_t\}$, where $r_t = \Delta(\mathcal{D}_t, c_t)/|\mathcal{D}_t|$. This is the set of the clusters whose corresponding facilities are close to $\mathcal{H}_s$. Rather than immediately proving Theorem 5, we first consider the following invariant $V(s)$, which says that the size of $\mathbb{O}_{s+1}$ is larger than that of $\mathbb{O}_s$ with a high probability, unless $\mathcal{H}_s$ is a (k,ϵ)-reduced search space. We will show that the invariant is maintained during each step of the algorithm. In Lemma 6, we further argue that $V(s)$ ensures the correctness of Theorem 5.

$V(s)$: Either $\mathcal{H}_s$ is a (k,ϵ)-reduced search space, or $\mathbf{Pr}[|\mathbb{O}_{s+1}| > |\mathbb{O}_s|] > \frac{\epsilon^3}{180}$.

Before proving this invariant property, we first show its implication.

▶ **Lemma 6.** *If invariant $V(s)$ is maintained during each step of Algorithm 1, then there exists a value $q = O(k\epsilon^{-3})$, such that $\mathcal{H}_q$ is a (k,ϵ)-reduced search space.*

Proof. At each iteration of Algorithm 1, define a variable a_s as follows: if $|\mathbb{O}_{s+1}| > |\mathbb{O}_s|$, then $a_s = 1$; otherwise, $a_s = 0$. Invariant $V(s)$ implies that $p = \mathbf{Pr}[|\mathbb{O}_{s+1}| > |\mathbb{O}_s|] > \frac{\epsilon^3}{180}$, unless $\mathcal{H}_s$ is the desired reduced search space. Let $q = 360k\epsilon^{-3}$ and $a = \sum_{s=1}^{q} a_s$. We have $pq > 2k$. Using Lemma 3, we get

$$\mathbf{Pr}[a < k] \le \mathbf{Pr}[a < \frac{1}{2}pq] < e^{-pq/8} < e^{-k/4} \le e^{-1/4},$$

which implies that

$$\mathbf{Pr}[\mathbb{O}_q = \mathbb{D}] = \mathbf{Pr}[a \ge k] = 1 - \mathbf{Pr}[a < k] > 1 - e^{-1/4}.$$

If $\mathbb{O}_q = \mathbb{D}$ holds, then by the definition of $\mathbb{O}_q$, we have $\Phi(c_t, \mathcal{H}_q) < (1+\epsilon)\Delta(\mathcal{D}_t, c_t)/|\mathcal{D}_t|$ for each $1 \le t \le k$, and thus $\sum_{t=1}^{k} |\mathcal{D}_t|\Phi(c_t, \mathcal{H}_q) < (1+\epsilon)\sum_{t=1}^{k} \Delta(\mathcal{D}_t, c_t)$. By the definition of reduced search space, $\mathcal{H}_q$ is a (k,ϵ)-reduced search space. This completes the proof of Lemma 6. ◀

Lemma 6 says that $V(s)$ is sufficient to ensure the validity of Theorem 5. It remains to prove the correctness of $V(s)$. Consider a cluster $\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s$. If a client from $\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t)$ is picked after the s-th iteration, then by the definitions of $\mathbf{b}_\alpha(\mathcal{D}_t)$ and $\mathbb{O}_s$, $\mathcal{D}_t$ should be added to $\mathbb{O}_{s+1}$ and thus $|\mathbb{O}_{s+1}| - |\mathbb{O}_s| \ge 1$. By the arguments above, we know that $\mathbf{Pr}[|\mathbb{O}_{s+1}| > |\mathbb{O}_s|]$ is no less than the probability of sampling a point from $\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t)$ for any cluster $\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s$. We now show the correctness of $V(s)$ for the case $s = 0$. Since the first client added to $\mathcal{H}$ is uniformly sampled from $\mathcal{D}$, we have

$$\mathbf{Pr}[|\mathbb{O}_1| > |\mathbb{O}_0|] \ge \frac{\sum_{\mathcal{D}_t \in \mathbb{D}} |\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t)|}{|\mathcal{D}|}. \tag{1}$$

We use the following lemma to show that for any cluster $\mathcal{D}_t \in \mathbb{D}$, $\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t)$ contains a substantial part of $\mathcal{D}_t$.

▶ **Lemma 7.** *For any $\mathcal{D}_t \in \mathbb{D}$ and $\alpha \ge 1$, we have $|\mathbf{b}_\alpha(\mathcal{D}_t)| \ge (1 - \frac{1}{\alpha})|\mathcal{D}_t|$.*

Proof. Suppose that the statement in the lemma does not hold. Then, we have $|\mathcal{D}_t \backslash \mathbf{b}_\alpha(\mathcal{D}_t)| > \frac{1}{\alpha}|\mathcal{D}_t|$. This implies that

$$\Delta(\mathcal{D}_t, c_t) \ge \Delta(\mathcal{D}_t \backslash \mathbf{b}_\alpha(\mathcal{D}_t), c_t) > |\mathcal{D}_t \backslash \mathbf{b}_\alpha(\mathcal{D}_t)|\alpha r_t > |\mathcal{D}_t| r_t = \Delta(\mathcal{D}_t, c_t),$$

where the second step is due to the definition of $\mathbf{b}_\alpha(\mathcal{D}_t)$, and the last step follows from the fact that $\Delta(\mathcal{D}_t, c_t) = |\mathcal{D}_t| r_t$. This result is not valid, and thus Lemma 7 is true. ◀

Using inequality (1) and Lemma 7, we have $\mathbf{Pr}[|\mathbb{O}_1| > |\mathbb{O}_0|] \geq \sum_{\mathcal{D}_t \in \mathbb{D}} |\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t)|/|\mathcal{D}| \geq \frac{\epsilon}{2+\epsilon}$, which implies that $V(s)$ holds for $s = 0$. We now consider the case of $s > 0$. By the definition of D-sampling, in the $s+1$-th iteration, Algorithm 1 samples a client from a cluster outside of $\mathbb{O}_s$ with probability $\sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s)/\Phi(\mathcal{D}, \mathcal{H}_s)$. We consider the following two cases: (1) $\sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) \leq \frac{\epsilon}{6}\Phi(\mathcal{D}, \mathcal{H}_s)$, and (2) $\sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) > \frac{\epsilon}{6}\Phi(\mathcal{D}, \mathcal{H}_s)$. In the following, we will show how invariant $V(s)$ is maintained in each case.

Case (1): $\sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) \leq \frac{\epsilon}{6}\Phi(\mathcal{D}, \mathcal{H}_s)$

For this case, D-sampling does not work since the probability of sampling clients from a cluster not covered by $\mathbb{O}_s$ is very small. For each cluster $\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s$, we will show that a previously sampled client is close to facility c_t. Let h_t denote the nearest client to c_t in $\mathcal{H}_s$ for each $\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s$. The following lemma shows that the distance from h_t to c_t can be bounded by a combination of r_t and $\frac{1}{|\mathcal{D}_t|}\Phi(\mathcal{D}_t, \mathcal{H}_s)$.

▶ **Lemma 8.** *For each $\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s$, we have $\Delta(h_t, c_t) \leq \frac{1}{|\mathcal{D}_t|}\Phi(\mathcal{D}_t, \mathcal{H}_s) + r_t$.*

Proof. For each $j \in \mathcal{D}_t$, let $h(j)$ denote the nearest client to j in $\mathcal{H}_s$. Consider a multi-set $\tilde{\mathcal{D}}_t = \{h(j) : j \in \mathcal{D}_t\}$. By triangle inequality, we have

$$\Delta(\tilde{\mathcal{D}}_t, c_t) \leq \sum_{j \in \mathcal{D}_t} \Delta(h(j), j) + \sum_{j \in \mathcal{D}_t} \Delta(j, c_t) = \Phi(\mathcal{D}_t, \mathcal{H}_s) + \Delta(\mathcal{D}_t, c_t). \tag{2}$$

Consequently, we get

$$\Delta(h_t, c_t) \leq \min_{i \in \tilde{\mathcal{D}}_t} \Delta(i, c_t) \leq \frac{1}{|\mathcal{D}_t|}\Delta(\tilde{\mathcal{D}}_t, c_t) \leq \frac{1}{|\mathcal{D}_t|}\Phi(\mathcal{D}_t, \mathcal{H}_s) + r_t,$$

where the first step follows from the fact that h_t is the nearest client to c_t in $\mathcal{H}_s$, the second step estimates the minimum by the average, and the last step is due to inequality (2). This completes the proof of Lemma 8. ◀

By the definition of $\mathbb{O}_s$ and Lemma 8, we have

$$\begin{aligned}
\sum_{t=1}^{k} |\mathcal{D}_t|\Phi(c_t, \mathcal{H}_s) &= \sum_{\mathcal{D}_t \in \mathbb{O}_s} |\mathcal{D}_t|\Phi(c_t, \mathcal{H}_s) + \sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} |\mathcal{D}_t|\Phi(c_t, \mathcal{H}_s) \\
&\leq (1+\epsilon/2) \sum_{\mathcal{D}_t \in \mathbb{O}_s} \Delta(\mathcal{D}_t, c_t) + \sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Delta(\mathcal{D}_t, c_t) + \sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) \\
&\leq (1+\epsilon/2) \sum_{t=1}^{k} \Delta(\mathcal{D}_t, c_t) + \sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s).
\end{aligned} \tag{3}$$

We now show that $\sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s)$ is much smaller than $\sum_{t=1}^{k} \Delta(\mathcal{D}_t, c_t)$.

▶ **Lemma 9.** *If $\sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) \leq \frac{\epsilon}{6}\Phi(\mathcal{D}, \mathcal{H}_s)$, then we have $\sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) \leq \frac{\epsilon}{2}\sum_{t=1}^{k} \Delta(\mathcal{D}_t, c_t)$.*

Proof. By the assumption that $\sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) \leq \frac{\epsilon}{6}\Phi(\mathcal{D}, \mathcal{H}_s)$, we have

$$\Phi(\mathcal{D}, \mathcal{H}_s) = \sum_{\mathcal{D}_t \in \mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) + \sum_{\mathcal{D}_t \in \mathbb{D}\backslash\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) \leq \sum_{\mathcal{D}_t \in \mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) + \frac{\epsilon}{6}\Phi(\mathcal{D}, \mathcal{H}_s),$$

which implies that

$$\Phi(\mathcal{D},\mathcal{H}_s) \le \frac{1}{1-\epsilon/6}\sum_{\mathcal{D}_t\in\mathbb{O}_s}\Phi(\mathcal{D}_t,\mathcal{H}_s) \le \frac{1}{1-\epsilon/6}\sum_{\mathcal{D}_t\in\mathbb{O}_s}\Delta(\mathcal{D}_t,\mathcal{H}_s)$$
$$\le \frac{1}{1-\epsilon/6}\sum_{\mathcal{D}_t\in\mathbb{O}_s}[\Delta(\mathcal{D}_t,c_t)+|\mathcal{D}_t|\Phi(c_t,\mathcal{H}_s)] \le \frac{2+\epsilon/2}{1-\epsilon/6}\sum_{\mathcal{D}_t\in\mathbb{O}_s}\Delta(\mathcal{D}_t,c_t),$$

where the second step follows from the definitions of $\Delta(\mathcal{D}_t,\mathcal{H}_s)$ and $\Phi(\mathcal{D}_t,\mathcal{H}_s)$, the third step follows from triangle inequality, and the last step is due to the definition of $\mathbb{O}_s$. Consequently, we get

$$\sum_{\mathcal{D}_t\in\mathbb{D}\backslash\mathbb{O}_s}\Phi(\mathcal{D}_t,\mathcal{H}_s) \le \frac{\epsilon}{6}\Phi(\mathcal{D},\mathcal{H}_s) \le \frac{\epsilon}{6}\cdot\frac{2+\epsilon/2}{1-\epsilon/6}\sum_{\mathcal{D}_t\in\mathbb{O}_s}\Delta(\mathcal{D}_t,c_t) \le \frac{\epsilon}{2}\sum_{\mathcal{D}_t\in\mathbb{O}_s}\Delta(\mathcal{D}_t,c_t)$$
$$\le \frac{\epsilon}{2}\sum_{t=1}^{k}\Delta(\mathcal{D}_t,c_t),$$

where the third step is derived from the fact that $0<\epsilon\le 1$. ◀

Using inequality (3) and Lemma 9, we get $\sum_{t=1}^{k}|\mathcal{D}_t|\Phi(c_t,\mathcal{H}_s) \le (1+\epsilon)\sum_{t=1}^{k}\Delta(\mathcal{D}_t,c_t)$. By the definition of reduced search spaces, $\mathcal{H}_s$ is a (k,ϵ)-reduced search space. Thus, invariant $V(s)$ holds for case (1).

Case (2): $\sum_{\mathcal{D}_t\in\mathbb{D}\backslash\mathbb{O}_s}\Phi(\mathcal{D}_t,\mathcal{H}_s) > \frac{\epsilon}{6}\Phi(\mathcal{D},\mathcal{H}_s)$

For this case, the clients from the clusters not covered by $\mathbb{O}_s$ have a good chance to be sampled. Given a cluster $\mathcal{D}_t\in\mathbb{D}\backslash\mathbb{O}_s$, let h_t denote the nearest client to c_t in $\mathcal{H}_s$. Define $d_t=\Delta(h_t,c_t)$ and $\beta_t=d_t/r_t$. The fact that $\mathcal{D}_t\in\mathbb{D}\backslash\mathbb{O}_s$ implies that $\beta_t>1+\frac{\epsilon}{2}$. As discussed above, we will show that if α has a proper value, then any client from $\mathbf{b}_\alpha(\mathcal{D}_t)$ is close to c_t, and can be sampled with a good chance. By the definitions of $\mathbf{b}_\alpha(\mathcal{D}_t)$ and $\mathbb{O}_s$, we know that if a client $j\in\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t)$ is sampled in the $s+1$-th iteration of Algorithm 1, then cluster $\mathcal{D}_t$ should be added to $\mathbb{O}_{s+1}$. We will argue that this happens with a high probability, and thus invariant $V(s)$ can be proven.

We now give a lower bound on $\Phi(\mathbf{b}_\alpha(\mathcal{D}_t),\mathcal{H}_s)$ for each $\mathcal{D}_t\in\mathbb{D}\backslash\mathbb{O}_s$, which is useful for analyzing the probability of sampling clients from $\mathbf{b}_\alpha(\mathcal{D}_t)$ in the $s+1$-th iteration of Algorithm 1.

▶ **Lemma 10.** *For any $\mathcal{D}_t\in\mathbb{D}\backslash\mathbb{O}_s$ and $1\le\alpha\le 1+\frac{\epsilon}{2}$, we have $\Phi(\mathbf{b}_\alpha(\mathcal{D}_t),\mathcal{H}_s)\ge\Delta(\mathcal{D}_t,c_t)(\beta_t-\frac{\beta_t}{\alpha}-\ln\alpha)$.*

Proof. Given a value $0\le\mu\le\alpha$, define $\mathcal{G}_\mu=\{j\in\mathcal{D}_t:\Delta(j,c_t)=\mu r_t\}$. This is the set of clients from $\mathcal{D}_t$ that lie on an annular region centred at c_t with radius μr_t. Given a client $j\in\mathcal{D}_t$, by $h(j)$ we denote the nearest client to j in $\mathcal{H}_s$. We have

$$\Phi(\mathcal{G}_\mu,\mathcal{H}_s)=\sum_{j\in\mathcal{G}_\mu}\Delta(j,h(j))\ge\sum_{j\in\mathcal{G}_\mu}[\Delta(h(j),c_t)-\Delta(c_t,j)]\ge\sum_{j\in\mathcal{G}_\mu}[d_t-\Delta(c_t,j)]$$
$$=\sum_{j\in\mathcal{G}_\mu}[\beta_t r_t-\Delta(c_t,j)]=\sum_{j\in\mathcal{G}_\mu}(\beta_t r_t-\mu r_t)=|\mathcal{G}_\mu|(\beta_t r_t-\mu r_t), \tag{4}$$

where the second step is due to triangle inequality, the third step is due to the fact that $d_t=\Delta(h_t,c_t)\le\Delta(h(j),c_t)$, and the fifth step follows from the definition of $\mathcal{G}_\mu$.

Define $f(\mu) = \beta_t r_t - \mu r_t$ and $\phi(\mu) = |\mathcal{G}_\mu|$. Let $\mathcal{L} = \{\mu \in (0,\alpha) : \phi(\mu) \neq 0\} \cup \{0, \alpha\}$. It can be seen that $|\mathcal{L}| \leq n+2$. We have

$$\Phi(\mathbf{b}_\alpha(\mathcal{D}_t), \mathcal{H}_s) = \sum_{\mu \in \mathcal{L}} \Phi(\mathcal{G}_\mu, \mathcal{H}_s) \geq \sum_{\mu \in \mathcal{L}} f(\mu)\phi(\mu), \tag{5}$$

where the last step is derived from inequality (4).

We will show that

$$\sum_{\mu \in \mathcal{L}} f(\mu)\phi(\mu) \geq \Delta(\mathcal{D}_t, c_t)(\beta_t - \frac{\beta_t}{\alpha} - \ln \alpha). \tag{6}$$

If inequality (6) is true, then combined with inequality (5), we have $\Phi(\mathbf{b}_\alpha(\mathcal{D}_t), \mathcal{H}_s) \geq \Delta(\mathcal{D}_t, c_t)(\beta_t - \frac{\beta_t}{\alpha} - \ln \alpha)$, which completes the proof of Lemma 10. It remains to show inequality (6). We sort each $\mu \in \mathcal{L}$ by increasing value, and let μ_τ denote the τ-th number in this order for each $1 \leq \tau \leq |\mathcal{L}|$. Using Lemma 4, we get

$$\begin{aligned}\sum_{\tau=1}^{|\mathcal{L}|} f(\mu_\tau)\phi(\mu_\tau) &= f(\mu_{|\mathcal{L}|}) \sum_{\tau=1}^{|\mathcal{L}|} \phi(\mu_\tau) - \sum_{\tau=1}^{|\mathcal{L}|-1} \Big[[f(\mu_{\tau+1}) - f(\mu_\tau)] \sum_{\tau'=1}^{\tau} \phi(\mu_{\tau'})\Big] \\ &= f(\alpha)\big|\mathbf{b}_\alpha(\mathcal{D}_t)\big| - \sum_{\tau=1}^{|\mathcal{L}|-1} \Big[[f(\mu_{\tau+1}) - f(\mu_\tau)]\big|\mathbf{b}_{\mu_\tau}(\mathcal{D}_t)\big|\Big],\end{aligned} \tag{7}$$

where the second step follows from the definition of $\phi(\mu)$. Let $f'(t)$ denote the first derivative of $f(t)$. By the definition of $\mathcal{L}$, for any $1 \leq \tau \leq |\mathcal{L}| - 1$ and $\mu' \in [\mu_\tau, \mu_{\tau+1})$, we have $\mathbf{b}_{\mu'}(\mathcal{D}_t) = \mathbf{b}_{\mu_\tau}(\mathcal{D}_t)$. Thus,

$$[f(\mu_{\tau+1}) - f(\mu_\tau)]\big|\mathbf{b}_{\mu_\tau}(\mathcal{D}_t)\big| = \big|\mathbf{b}_{\mu_\tau}(\mathcal{D}_t)\big| \int_{\mu_\tau}^{\mu_{\tau+1}} f'(\mu)\mathrm{d}\mu = \int_{\mu_\tau}^{\mu_{\tau+1}} f'(\mu)\big|\mathbf{b}_\mu(\mathcal{D}_t)\big|\mathrm{d}\mu,$$

which implies that

$$\sum_{\tau=1}^{|\mathcal{L}|-1} \Big[[f(\mu_{\tau+1}) - f(\mu_\tau)]\big|\mathbf{b}_{\mu_\tau}(\mathcal{D}_t)\big|\Big] = \int_0^\alpha f'(\mu)\big|\mathbf{b}_\mu(\mathcal{D}_t)\big|\mathrm{d}\mu. \tag{8}$$

Define the following function that depends on μ $(0 \leq \mu \leq \alpha)$:

$$\eta(\mu) = \begin{cases} \frac{1}{\mu^2}|\mathcal{D}_t| & \mu > 1 \\ 0 & 0 \leq \mu \leq 1 \end{cases}$$

A primitive function of $\eta(\mu)$ is

$$G(\mu) = \begin{cases} (1 - \frac{1}{\mu})|\mathcal{D}_t| & \mu > 1 \\ 0 & 0 \leq \mu \leq 1 \end{cases}$$

Using integration by parts, we have

$$\int_0^\alpha \eta(\mu)f(\mu)\mathrm{d}\mu = G(\mu)f(\mu)\big|_0^\alpha - \int_0^\alpha f'(\mu)G(\mu)\mathrm{d}\mu = G(\alpha)f(\alpha) - \int_0^\alpha f'(\mu)G(\mu)\mathrm{d}\mu. \tag{9}$$

Consequently, we get

$$\begin{aligned}
&\sum_{\tau=1}^{|\mathcal{L}|} f(\mu_\tau)\phi(\mu_\tau) - \int_0^\alpha \eta(\mu) f(\mu) \mathrm{d}\mu \\
&\qquad = [|\mathbf{b}_\alpha(\mathcal{D}_t)| - G(\alpha)] f(\alpha) + \int_0^\alpha f'(\mu)[G(\mu) - |\mathbf{b}_\mu(\mathcal{D}_t)|] \mathrm{d}\mu \\
&\qquad \geq \int_0^\alpha f'(\mu)[G(\mu) - |\mathbf{b}_\mu(\mathcal{D}_t)|] \mathrm{d}\mu, \qquad (10)
\end{aligned}$$

where the first step is derived from equalities (7), (8), and (9), and the second step follows from the fact that $|\mathbf{b}_\alpha(\mathcal{D}_t)| \geq G(\alpha)$ and $f(\alpha) > 0$, which is due to Lemma 7 and the fact that $\alpha \leq 1 + \frac{\epsilon}{2} < \beta_t$.

Observe that $f(\mu)$ decreases monotonously for $0 \leq \mu \leq \alpha$. This implies that $f'(\mu) \leq 0$ for any $0 \leq \mu \leq \alpha$. Moreover, we have $G(\mu) - |\mathbf{b}_\mu(\mathcal{D}_t)| \leq 0$ by Lemma 7. Thus, we get $f'(\mu)[G(\mu) - |\mathbf{b}_\mu(\mathcal{D}_t)|] \geq 0$ for any $0 \leq \mu \leq \alpha$ and $\int_0^\alpha f'(\mu)[G(\mu) - |\mathbf{b}_\mu(\mathcal{D}_t)|] \mathrm{d}\mu \geq 0$. Consequently, inequality (10) implies that

$$\begin{aligned}
&\sum_{\tau=1}^{|\mathcal{L}|} f(\mu_\tau)\phi(\mu_\tau) \geq \int_0^\alpha \eta(\mu) f(\mu) \mathrm{d}\mu = \int_1^\alpha \eta(\mu) f(\mu) \mathrm{d}\mu = \int_1^\alpha \frac{|\mathcal{D}_t|}{\mu^2}(\beta_t r_t - \mu r_t) \mathrm{d}\mu \\
&\qquad = \Delta(\mathcal{D}_t, c_t) \int_1^\alpha \frac{1}{\mu^2}(\beta_t - \mu) \mathrm{d}\mu = \Delta(\mathcal{D}_t, c_t)(\beta_t - \frac{\beta_t}{\alpha} - \ln \alpha),
\end{aligned}$$

where the second step follows from $g(\mu) = 0$ for any $0 \leq \mu \leq 1$, and the fourth step follows from the fact that $\Delta(\mathcal{D}_t, c_t) = |\mathcal{D}_t| r_t$. This implies that inequality (6) holds, which in turn implies that Lemma 10 is true. ◀

The following result implies that the ratio of $\Phi(\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t), \mathcal{H}_s)$ and $\Phi(\mathcal{D}_t, \mathcal{H}_s)$ can be bounded by a constant for each $\mathcal{D}_t \in \mathbb{D} \backslash \mathbb{O}_s$.

▶ **Lemma 11.** *For each $\mathcal{D}_t \in \mathbb{D} \backslash \mathbb{O}_s$, we have $\Phi(\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t), \mathcal{H}_s) > \frac{\epsilon^2}{30} \Phi(\mathcal{D}_t, \mathcal{H}_s)$.*

Proof. Observe that

$$\begin{aligned}
\Phi(\mathcal{D}_t, \mathcal{H}_s) &\leq \Delta(\mathcal{D}_t, h_t) \leq \Delta(\mathcal{D}_t, c_t) + |\mathcal{D}_t| d_t = \Delta(\mathcal{D}_t, c_t) + \Delta(\mathcal{D}_t, c_t)\frac{d_t}{r_t} \\
&= (1 + \beta_t)\Delta(\mathcal{D}_t, c_t),
\end{aligned}$$

where the second step follows from triangle inequality, and the third step is due to the fact that $\Delta(\mathcal{D}_t, c_t) = |\mathcal{D}_t| r_t$. Thus, using Lemma 10, for any $1 \leq \alpha \leq 1 + \frac{\epsilon}{2}$, we have

$$\frac{\Phi(\mathbf{b}_\alpha(\mathcal{D}_t), \mathcal{H}_s)}{\Phi(\mathcal{D}_t, \mathcal{H}_s)} \geq \frac{\Phi(\mathbf{b}_\alpha(\mathcal{D}_t), \mathcal{H}_s)}{(1 + \beta_t)\Delta(\mathcal{D}_t, c_t)} \geq \frac{1}{1 + \beta_t}(\beta_t - \frac{\beta_t}{\alpha} - \ln \alpha). \qquad (11)$$

It can be seen that $\frac{1}{1+\beta_t}(\beta_t - \frac{\beta_t}{\alpha} - \ln \alpha)$ increases monotonously with increasing value of β_t for $\alpha \geq 1$. This implies that

$$\begin{aligned}
\frac{\Phi(\mathbf{b}_\alpha(\mathcal{D}_t), \mathcal{H}_s)}{\Phi(\mathcal{D}_t, \mathcal{H}_s)} &> \frac{1}{1 + \beta_t}(\beta_t - \frac{\beta_t}{1 + \epsilon/2} - \ln(1 + \frac{\epsilon}{2})) > \frac{1}{2 + \epsilon/2}(\frac{\epsilon}{2} - \ln(1 + \frac{\epsilon}{2})) \\
&= \frac{1}{2 + \epsilon/2}(\frac{\epsilon}{2} - \ln(1 + \frac{\epsilon}{2})) \geq \frac{1}{2 + \epsilon/2} \cdot \frac{\epsilon^2}{12} \geq \frac{\epsilon^2}{30},
\end{aligned}$$

where the first step is due to inequality (11), the second step follows from the fact that $\beta_t > 1 + \frac{\epsilon}{2}$, and the last two steps follow from the fact that $0 < \epsilon \leq 1$. This completes the proof of Lemma 11. ◀

By the assumption that $\sum_{\mathcal{D}_t \in \mathbb{D}\setminus\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s) > \frac{\epsilon}{6}\Phi(\mathcal{D}, \mathcal{H}_s)$ and Lemma 11, we get

$$\begin{aligned}\Pr[|\mathbb{O}_{s+1}| > |\mathbb{O}_s|] &\geq \frac{\sum_{\mathcal{D}_t \in \mathbb{D}\setminus\mathbb{O}_s} \Phi(\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t), \mathcal{H}_s)}{\Phi(\mathcal{D}, \mathcal{H}_s)} \\ &= \frac{\sum_{\mathcal{D}_t \in \mathbb{D}\setminus\mathbb{O}_s} \Phi(\mathbf{b}_{1+\epsilon/2}(\mathcal{D}_t), \mathcal{H}_s)}{\sum_{\mathcal{D}_t \in \mathbb{D}\setminus\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s)} \cdot \frac{\sum_{\mathcal{D}_t \in \mathbb{D}\setminus\mathbb{O}_s} \Phi(\mathcal{D}_t, \mathcal{H}_s)}{\Phi(\mathcal{D}, \mathcal{H}_s)} \\ &> \frac{\epsilon^2}{30} \cdot \frac{\epsilon}{6} = \frac{\epsilon^3}{180}.\end{aligned}$$

This implies that invariant $V(s)$ holds for case (2). Using Lemma 6, we complete the proof of Theorem 5.

3 The selection algorithms

Given a clustering problem, let $\mathbb{D}^* = \{\mathcal{D}_1^*, \ldots, \mathcal{D}_k^*\}$ denote the partition of $\mathcal{D}$ and $\mathcal{C}^* = \{c_1^*, \ldots, c_k^*\}$ be the set of opened facilities in an optimal solution, where the clients from $\mathcal{D}_t^*$ are assigned to c_t^* for each $1 \leq t \leq k$. Using the standard discretization method [3, 33], we can assume that the ratio of the maximum and minimum distances between any two points from $\mathcal{F} \cup \mathcal{D}$ is upper bounded by $n^{O(1)}$, which induces an arbitrarily small loss in the approximation guarantee.[1] As shown in Theorem 5, greedy sampling yields a (k, ϵ)-reduced search space with $O(nk\epsilon^{-3})$ time. In this section, we show how to obtain the desired approximation solution for each studied problem using such a reduced search space.

3.1 Lower-bounded k-median

Given a set $\mathcal{D}$ of clients and a set $\mathcal{F}$ of facilities in a metric space, where each facility $i \in \mathcal{F}$ is associated with a lower bound $\varphi(i)$, the lower-bounded k-median problem [36] is to open at most k facilities and assign each client to an opened facility, such that the number of clients assigned to each open facility $i \in \mathcal{F}$ is at least $\varphi(i)$ and the assignment cost is minimized.

Since the facilities are associated with lower bounds, it may be the case that in an optimal solution, the clients are partitioned into k' clusters for an integer $0 < k' < k$ and we need to guess the value of k', which multiplies the running time by a factor of k. For each integer $k' + 1 \leq t \leq k$, we can assume that $\mathcal{D}_t^* = \emptyset$, and let c_t^* be an arbitrary facility.

We first run Algorithm 1 to obtain a reduced search space $\mathcal{H}$. For each $t \in [k]$, let h_t denote the nearest client to c_t^* in $\mathcal{H}$, and define $d_t = \Delta(c_t^*, h_t)$. We round d_t down to the closest integer power of $1 + \epsilon$ and define $\mathcal{Q}_t = \{i \in \mathcal{F} : d_t \leq \Delta(i, h_t) \leq (1 + \epsilon)d_t\}$. We have $c_t^* \in \mathcal{Q}_t$. The idea of our selection algorithm is to choose facilities from the sets $\mathcal{Q}_1, \ldots, \mathcal{Q}_k$ to open. For each $t \in [k']$, let $c_t = \arg\min_{i \in \mathcal{Q}_t} \varphi(i)$ be the facility in $\mathcal{Q}_t$ associated with the smallest lower bound. We have $\Delta(\mathcal{D}_t^*, c_t) \leq \Delta(\mathcal{D}_t^*, h_t) + |\mathcal{D}_t^*|\Delta(h_t, c_t) \leq \Delta(\mathcal{D}_t^*, h_t) + (1 + \epsilon)|\mathcal{D}_t^*|\Delta(h_t, c_t^*) \leq \Delta(\mathcal{D}_t^*, c_t^*) + (2 + \epsilon)|\mathcal{D}_t^*|\Delta(h_t, c_t^*)$, where the first and last steps follow from triangle inequality, and the second step is due to the fact that $c_t \in \mathcal{Q}_t$.

---

[1] For example, we can polynomially bound the ratio as follows. First, guess the cost *opt* of an optimal solution. We can enumerate the distance between each client-facility pair to find a value M that satisfies $opt/n < M < opt$. For each $i, j \in \mathcal{D} \cup \mathcal{F}$ with $\Delta(i, j) > Mn^2$, let $\Delta(i, j) = Mn^2$. No $O(1)$-approximation solution will use such edges since $Mn^2 > nopt$. Let $\Delta(i, j) = M/n^2$ for each $i, j \in \mathcal{D} \cup \mathcal{F}$ with $\Delta(i, j) < M/n^2$, which loses a factor $1 + 1/n$ in the approximation guarantee. Now the ratio is at most n^4.

Summing both sides of the inequality over $\mathcal{D}_t^* \in \mathbb{D}^*$, we know that

$$\sum_{t=1}^{k'} \Delta(\mathcal{D}_t^*, c_t) \leq \sum_{t=1}^{k'} \Delta(\mathcal{D}_t^*, c_t^*) + (2+\epsilon) \sum_{t=1}^{k'} |\mathcal{D}_t^*| \Delta(h_t, c_t^*)$$
$$= \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t^*) + (2+\epsilon) \sum_{t=1}^{k} |\mathcal{D}_t^*| \Delta(h_t, c_t^*) \leq (3+4\epsilon) \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t^*) \quad (12)$$

holds with constant probability, where the last step follows from the definition of reduced search space.

By the fact that $c_t^* \in \mathcal{Q}_t$ and $c_t = \arg\min_{i \in \mathcal{Q}_t} \varphi(i)$ for each $t \in [k']$, we have $\varphi(c_t) \leq \varphi(c_t^*)$. Thus, assigning the clients from cluster $\mathcal{D}_t^*$ to c_t for each $t \in [k']$ is feasible for lower-bounded k-median. Using the partition approach given by Ding and Xu [17], we can partition $\mathcal{D}$ into k' clusters $\{\mathcal{D}_1, \ldots, \mathcal{D}_{k'}\}$ based on facilities $c_1, \ldots, c_{k'}$ with $n^{O(1)}$ time, such that the lower bounds of the facilities are satisfied and $\sum_{t=1}^{k'} \Delta(\mathcal{D}_t, c_t) \leq \sum_{t=1}^{k'} \Delta(\mathcal{D}_t^*, c_t) \leq (3+4\epsilon) \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t^*)$ holds with constant probability, where the last step follows from inequality (12). This implies a $(3+O(\epsilon))$-approximation for lower-bounded k-median.

It remains to show how to find the facilities $c_1, \ldots, c_{k'}$. We use an enumeration method similar to that in [14]. Observe that there are at most $|\mathcal{H}|^k O(\epsilon^{-1} \log n)^k = O(k\epsilon^{-4} \log n)^k$ choices for the facilities $c_1, \ldots, c_{k'}$ (we have at most $|\mathcal{H}|^k$ choices for clients $h_1, \ldots, h_{k'}$ and no more than $O(\epsilon^{-1} \log n)^k$ choices for distances $d_1, \ldots, d_{k'}$). Thus, these facilities can be guessed by paying a factor of $O(k\epsilon^{-4} \log n)^k$ in the running time, which is upper bounded by $(k\epsilon^{-1})^{O(k)} n^{O(1)}$ using the trick given in [14, 15]: If $k < \log n / \log \log n$, then we have $(\log n)^k = n^{O(1)}$. Otherwise, $\log n = O(k \log k)$, which implies that $(\log n)^k = k^{O(k)}$.

▶ **Theorem 12.** *There is an algorithm yielding a $(3+\epsilon)$-approximation for lower-bounded k-median with constant probability, which runs in $(k\epsilon^{-1})^{O(k)} n^{O(1)}$ time.*

3.2 Capacitated k-median

The capacitated k-median problem [10] considers a set $\mathcal{D}$ of clients and a set $\mathcal{F}$ of facilities in a metric space, where each facility $i \in \mathcal{F}$ has a capacity $\varphi(i)$. The goal is to open at most k facilities and assign each client to an opened facility, such that the number of clients assigned to each facility $i \in \mathcal{F}$ is at most $\varphi(i)$ and the assignment cost is minimized.

We run Algorithm 1 to obtain a reduced search space $\mathcal{H}$. The selection algorithm is similar to that of lower-bounded k-median. We replace the lower bounds of the facilities with capacities. However, we can no longer immediately open the facility associated with the largest capacity from $\mathcal{Q}_t$ for each $t \in [k]$, since it may be the case that the sets $\mathcal{Q}_t$ are not disjoint and a facility can be chosen for more than once, which might violate the capacity of the facility.

We use a *color-coding* technique [15] to deal with this issue. We randomly associate each $i \in \mathcal{F}$ with a label from $\{1, \ldots, k\}$. Each facility $c_t^* \in \mathcal{C}^*$ is assigned label t with probability k^{-k}. The probability can be boosted to a constant by repeating the process for $O(k^k)$ times. We now open the facility with the largest capacity among the ones from $\mathcal{Q}_t$ that are assigned label t for each $t \in [k]$, which is denoted by c_t. Using the partition algorithm given by Adamczyk et al. [2], we can partition $\mathcal{D}$ into k clusters $\{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ by facilities $c_1, \ldots, c_k$, such that the capacities of the facilities are satisfied and $\sum_{t=1}^{k} \Delta(\mathcal{D}_t, c_t) \leq \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t) \leq (3+4\epsilon) \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t^*)$.

▶ **Theorem 13.** *There is an algorithm yielding a $(3+\epsilon)$-approximation for capacitated k-median with constant probability, which runs in $(k\epsilon^{-1})^{O(k)} n^{O(1)}$ time.*

3.3 k-median with service installation costs

The k-median with service installation costs problem [34] considers a set $\mathcal{D}$ of clients and a set $\mathcal{F}$ of facilities in a metric space, and a set $\mathcal{S}$ of services, where each client $j \in \mathcal{D}$ is associated with a service $g(j) \in \mathcal{S}$, each service $\sigma \in \mathcal{S}$ is associated with a cost $f_i(\sigma)$ for installing it at each $i \in \mathcal{F}$, and there exists an ordering on the facilities from $\mathcal{F}$ satisfying that if i comes before i' in the ordering, then $f_i(\sigma) \leq f_{i'}(\sigma)$ for each $i, i' \in \mathcal{F}$ and $\sigma \in \mathcal{S}$. The goal is to open no more than k facilities, install services at the opened facilities, and assign each $j \in \mathcal{D}$ to a facility at which $g(j)$ is installed, such that the sum of the assignment cost and the service installation costs is minimized.

We use a selection algorithm similar to that of lower-bounded k-median. The only difference is that we open the facility c_t from $\mathcal{Q}_t$ with the smallest service installation cost for each $t \in [k]$. By the arguments above, a $(3 + O(\epsilon))$-approximation for the problem can be obtained if we install services on facilities $c_1, \ldots, c_k$ and assign clients in an optimal way. For each $\sigma \in \mathcal{S}$, let $\mathcal{D}_\sigma = \{j \in \mathcal{D} : g(j) = \sigma\}$. Define $\mathcal{C} = \{c_1, \ldots, c_k\}$. The problem of installing services and assigning clients can be formalized as the following integer programming (IP) for each $\sigma \in \mathcal{S}$.

$$\min \quad \sum_{i\in\mathcal{C}, j\in\mathcal{D}_\sigma} \Delta(j,i)x_{ij} + \sum_{i\in\mathcal{C}} f_i(\sigma)y_i \qquad \text{IP}(\sigma)$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{C}} x_{ij} = 1 \qquad \forall j \in \mathcal{D}_\sigma \qquad (13)$$

$$x_{ij} \leq y_i \qquad \forall i \in \mathcal{C}, j \in \mathcal{D}_\sigma \qquad (14)$$

$$x_{ij}, y_i \in \{0,1\} \qquad \forall i \in \mathcal{C}, j \in \mathcal{D}_\sigma \qquad (15)$$

IP(σ) associates a variable x_{ij} with each $j \in \mathcal{D}_\sigma$ and $i \in \mathcal{F}$, which indicates whether j is assigned to i, and associates a variable y_i with each $i \in \mathcal{F}$ that indicates whether service σ is installed at i. Constraint (13) ensures that each client is assigned to a facility, and constraint (14) ensures that the service requirements of the clients from $\mathcal{D}_\sigma$ are satisfied.

Observe that IP(σ) is a formulation of uncapacitated facility location, where each facility $i \in \mathcal{F}$ has an opening cost $f_i(\sigma)$. Using the FPT approximation algorithm given by Cohen-Addad et al. [14] to solve IP(σ) for each $\sigma \in \mathcal{S}$ with $\mathcal{D}_\sigma \neq \emptyset$, we can install services and partition the clients in $(k\epsilon^{-1})^{O(k)}n^{O(1)}$ time, which loses a factor $1.463 + \epsilon$ in the approximation ratio. This implies a $(4.389 + \epsilon)$-approximation for k-median with service installation costs in time $(k\epsilon^{-1})^{O(k)}n^{O(1)}$.

▶ **Theorem 14.** *There is an algorithm yielding a $(4.389 + \epsilon)$-approximation for k-median with service installation costs, which runs in $(k\epsilon^{-1})^{O(k)}n^{O(1)}$ time.*

3.4 Fault tolerant k-median

In the fault tolerant k-median problem [24], each client j should be assigned to l_j opened facilities, and the assignment cost of j is the sum of its distances to the l_j facilities, where $l_j > 0$ is a given integer.

A closely related problem to fault tolerant k-median is chromatic k-median [16], which considers a set of colored clients and has the constraint that no pair of clients with the same color can be assigned to the same facility. As shown by Ding and Xu [17], fault tolerant k-median can be reduced to chromatic k-median: Given an instance of fault tolerant k-median, we construct an instance of chromatic k-median by making l_j mono color copies for each $j \in \mathcal{D}$. Thus, it suffices to give an FPT approximation algorithm for chromatic k-median.

We start with finding a reduced search space $\mathcal{H}$ for the instance of chromatic k-median using Algorithm 1 (each client $j \in \mathcal{D}$ has $l_j - 1$ copies in this instance). For each $t \in [k]$, let $h_t = \arg\min_{j \in \mathcal{H}} \Delta(j, c_t^*)$ and $c_t = \arg\min_{i \in \mathcal{F}} \Delta(i, h_t)$. Then, $\sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t) \leq \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, h_t) + \sum_{t=1}^{k} |\mathcal{D}_t^*| \Delta(c_t^*, h_t) \leq \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t^*) + 2 \sum_{t=1}^{k} |\mathcal{D}_t^*| \Delta(c_t^*, h_t) \leq (3 + 2\epsilon) \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t^*)$ holds with constant probability, where the first two steps follow from triangle inequality, and the last step is due to the definition of reduced search space. By multiplying the running time by a factor of $|\mathcal{H}|^k = O(k\epsilon^{-3})^k$, we can assume that we have guessed the facilities $c_1, \ldots, c_k$. Using the partition algorithm given by Ding and Xu [17] and the color-coding technique given in Section 3.2, we can partition the clients into k clusters $\{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ by the facilities $c_1, \ldots, c_k$, such that the color constrained is satisfied and $\sum_{t=1}^{k} \Delta(\mathcal{D}_t, c_t) \leq \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t) \leq (3 + 4\epsilon) \sum_{t=1}^{k} \Delta(\mathcal{D}_t^*, c_t^*)$ holds with constant probability.

▶ **Theorem 15.** *There is an algorithm yielding a $(3 + \epsilon)$-approximation for fault tolerant k-median, which runs in $(k\epsilon^{-1})^{O(k)} n^{O(1)}$ time.*

3.5 Priority k-median

In the priority k-median problem [29], we are given a set $\mathcal{D}$ of clients and a set $\mathcal{F}$ of facilities in a metric space, a set $\mathcal{P} = \{1, \ldots, |\mathcal{P}|\}$ of priorities, and $|\mathcal{P}|$ integers $k_1, \ldots, k_{|\mathcal{P}|}$, where $\sum_{p=1}^{|\mathcal{P}|} k_p = k$. Each $i \in \mathcal{D} \cup \mathcal{F}$ has a priority $p_i \in \mathcal{P}$. The goal is to open no more than k_p facilities with priority p for each $p \in \mathcal{P}$ and assign each client to an opened facility with the same or higher priority, such that the assignment cost is minimized.

With an $O(k^k)$ multiplicative overhead in the running time, we can assume that we have guessed the priority associated with each $c_t^* \in \mathcal{C}^*$. For each $c_t^* \in \mathcal{C}^*$, we open the facility with priority $p_{c_t^*}$ that is nearest to c_t^*, and assign each client to its nearest facility with the same or higher priority. By the arguments above, this induces a $(3 + O(\epsilon))$-approximation for priority k-median.

▶ **Theorem 16.** *There is an algorithm yielding a $(3 + \epsilon)$-approximation for priority k-median, which runs in $(k\epsilon^{-1})^{O(k)} n^{O(1)}$ time.*

References

1 Niels H. Abel. Untersuchungen uber die reihe $1 +_m^1 x +_{1\cdot 2}^{m(m-1)} x^2 + \cdots$. *Reine Angew. Math.*, 1:311–339, 1826.

2 Marek Adamczyk, Jaroslaw Byrka, Jan Marcinkowski, Syed Mohammad Meesum, and Michal Wlodarczyk. Constant-factor FPT approximation for capacitated k-median. In *Proc. 27th Annual European Symposium on Algorithms*, pages 1:1–1:14, 2019.

3 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and Euclidean k-median by primal-dual algorithms. In *Proc. 58th IEEE Annual Symposium on Foundations of Computer Science*, pages 61–72, 2017.

4 David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, 2007.

5 Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

6 Ivan D. Baev, Rajmohan Rajaraman, and Chaitanya Swamy. Approximation algorithms for data placement problems. *SIAM J. Comput.*, 38(4):1411–1429, 2008.

7 Suman K. Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. In *Proc. 32nd Annual Conference on Neural Information Processing Systems*, pages 4955–4966, 2019.

8 Suman K. Bera, Deeparnab Chakrabarty, and Maryam Negahbani. Fair algorithms for clustering. *CoRR*, abs/1901.02393, 2019. arXiv:1901.02393.

9 Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Faster algorithms for the constrained k-means problem. *Theory Comput. Syst.*, 62(1):93–115, 2018.

10 Jaroslaw Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor approximation algorithms for hard capacitated k-median problems. In *Proc. 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 722–736, 2015.

11 Jaroslaw Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2):23:1–23:31, 2017.

12 Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002.

13 Ke Chen. On coresets for k-median and k-means clustering in metric and Euclidean spaces and their applications. *SIAM J. Comput.*, 39(3):923–947, 2009.

14 Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT approximations for k-median and k-means. In *Proc. 46th International Colloquium on Automata, Languages, and Programming*, pages 42:1–42:14, 2019.

15 Vincent Cohen-Addad and Jason Li. On the fixed-parameter tractability of capacitated clustering. In *Proc. 46th International Colloquium on Automata, Languages, and Programming*, pages 41:1–41:14, 2019.

16 Hu Ding and Jinhui Xu. Solving the chromatic cone clustering problem via minimum spanning sphere. In *Proc. 38th International Colloquium on Automata, Languages, and Programming*, pages 773–784, 2011.

17 Hu Ding and Jinhui Xu. A unified framework for clustering constrained data without locality property. In *Proc. 26th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1471–1490, 2015.

18 Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004.

19 Qilong Feng, Jiaxin Hu, Neng Huang, and Jianxin Wang. Improved PTAS for the constrained k-means problem. *J. Comb. Optim.*, 37(4):1091–1110, 2019.

20 Qilong Feng, Zhen Zhang, Ziyun Huang, Jinhui Xu, and Jianxin Wang. Improved algorithms for clustering with outliers. In *Proc. 30th International Symposium on Algorithms and Computation*, pages 61:1–61:12, 2019.

21 Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *J. Algorithms*, 31(1):228–248, 1999.

22 Yutian Guo, Junyu Huang, and Zhen Zhang. A constant factor approximation for lower-bounded k-median. In *Proc. 16th Annual Conference on Theory and Applications of Models of Computation*, 2020.

23 Anupam Gupta and Kanat Tangwongsan. Simpler analyses of local search algorithms for facility location. *CoRR*, abs/0809.2554, 2008. arXiv:0809.2554.

24 Mohammad T. Hajiaghayi, Wei Hu, Jian Li, Shi Li, and Barna Saha. A constant factor approximation algorithm for fault-tolerant k-median. *ACM Trans. Algorithms*, 12(3):36:1–36:19, 2016.

25 Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM*, 50(6):795–824, 2003.

26 Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

27 Ragesh Jaiswal, Amit Kumar, and Sandeep Sen. A simple D^2-sampling based PTAS for k-means and other clustering problems. *Algorithmica*, 70(1):22–46, 2014.

28 Ragesh Jaiswal, Mehul Kumar, and Pulkit Yadav. Improved analysis of D^2-sampling based PTAS for k-means and other clustering problems. *Inf. Process. Lett.*, 115(2):100–103, 2015.

29 Amit Kumar and Yogish Sabharwal. The priority k-median problem. In *Proc. 27th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 71–83, 2007.

30 Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM*, 57(2):5:1–5:32, 2010.

31 Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. *SIAM J. Comput.*, 45(2):530–547, 2016.

32 Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.

33 David Saulpic, Vincent Cohen-Addad, and Andreas E. Feldmann. Near-linear time approximations schemes for clustering in doubling metrics. In *Proc. 60th IEEE Annual Symposium on Foundations of Computer Science*, pages 540–559, 2019.

34 David B. Shmoys, Chaitanya Swamy, and Retsef Levi. Facility location with service installation costs. In *Proc. 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1088–1097, 2004.

35 Mauro Sozio, Thomas Neumann, and Gerhard Weikum. Near-optimal dynamic replication in unstructured peer-to-peer networks. In *Proc. 27th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 281–290, 2008.

36 Zoya Svitkina. Lower-bounded facility location. *ACM Trans. Algorithms*, 6(4):69:1–69:16, 2010.