A $(2+\varepsilon)$-Approximation Algorithm for Metric k-Median

Vincent Cohen-Addad
Google Research
Grenoble, France
cohenaddad@google.com

Fabrizio Grandoni
IDSIA at USI-SUPSI
Lugano, Switzerland
fabrizio.grandoni@gmail.com

Euiwoong Lee
University of Michigan
Ann Arbor, USA
euiwoong@umich.edu

Chris Schwiegelshohn
Aarhus University
Aarhus, Denmark
cschwiegelshohn@gmail.com

Ola Svensson
EPFL
Lausanne, Switzerland
ola.svensson@epfl.ch

ABSTRACT

In the classical NP-hard (metric) k-median problem, we are given a set of n clients and centers with metric distances between them, along with an integer parameter $k \geq 1$. The objective is to select a subset of k *open* centers that minimizes the total distance from each client to its closest open center.

In their seminal work, Jain, Mahdian, Markakis, Saberi, and Vazirani presented the Greedy algorithm for facility location, which implies a 2-approximation algorithm for k-median that opens k centers in *expectation*. Since then, substantial research has aimed at narrowing the gap between their algorithm and the best achievable approximation by an algorithm guaranteed to open *exactly* k centers, as required in the k-median problem. During the last decade, all improvements have been achieved by leveraging their algorithm (or a small improvement thereof), followed by a second step called bi-point rounding, which inherently adds an additional factor to the approximation guarantee.

Our main result closes this gap: for any $\varepsilon > 0$, we present a $(2+\varepsilon)$-approximation algorithm for the k-median problem, improving the previous best-known approximation factor of 2.613. Our approach builds on a combination of two key algorithms. First, we present a non-trivial modification of the Greedy algorithm that operates with only $O(\log n/\varepsilon^2)$ adaptive phases. Through a novel walk-between-solutions approach, this enables us to construct a $(2+\varepsilon)$-approximation algorithm for k-median that consistently opens at most $k+O(\log n/\varepsilon^2)$ centers: via known results, this already implies a $(2+\varepsilon)$-approximation algorithm that runs in quasi-polynomial time. Second, we develop a novel $(2+\varepsilon)$-approximation algorithm tailored for stable instances, where removing any center from an optimal solution increases the cost by at least an $\Omega(\varepsilon^3/\log n)$ fraction. Achieving this involves several ideas, including a sampling approach inspired by the k-means++ algorithm and a reduction to submodular optimization subject to a partition matroid. This allows us to convert the previous result into a polynomial time algorithm that opens exactly k centers while maintaining the $(2+\varepsilon)$-approximation guarantee.

CCS CONCEPTS

• **Theory of computation** → **Approximation algorithms analysis**; **Facility location and clustering**.

KEYWORDS

Approximation Algorithm, Clustering, k-Median

ACM Reference Format:
Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn, and Ola Svensson. 2025. A $(2+\varepsilon)$-Approximation Algorithm for Metric k-Median. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC '25), June 23–27, 2025, Prague, Czechia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3717823.3718299

1 INTRODUCTION

Given a collection D of n *points*, a clustering is a partition of the points into (typically) disjoint subsets (the *clusters*) such that points in the same cluster are *similar* and points in different clusters are *dissimilar*. One of the classical and best-studied clustering problems is k-*median* (see also Table 1). Here, in addition to the set D (points are also called *clients* in this framework), we are given an integer parameter $k > 0$ and a set F[1] (the *facilities* or *centers*), with metric distances $d(a,b)$ for $a, b \in D \cup F$. Our goal is to select k centers $S \subseteq F$ (*open* centers) so as to minimize the sum of the distances from each client to the closest open center. In other words, the goal is to minimize the following objective function

$$\mathrm{cost}(S) := \sum_{j \in D} \min_{i \in S} d(i,j).$$

The k-clustering induced by S is obtained by considering, for each $c \in S$, the subset of points that have c as a closest center in S (breaking ties arbitrarily). We use opt_k to denote the optimal cost, and simply use opt when k is clear from the context.

For most clustering variants, it is NP-hard to compute an optimal clustering and metric k-median is no exception. For this reason a lot of effort was devoted to the design of (polynomial-time) approximation algorithms for this problem, i.e., algorithms that compute a feasible solution S whose cost is provably within some factor $\alpha > 1$ of opt (the *approximation factor* or *approximation ratio/guarantee*).

[1]Sometimes in the literature one assumes $F = D$ or $D \subseteq F$. We consider the more general case and make no such assumption.

Most of the best-known approximation algorithms for k-median, including the current-best one [21, 29], are based on the following high-level approach (see Section 1.2 for alternative approaches in the literature). One considers the associated facility location problem, where instead of k we are given a (uniform) facility opening cost f. In this problem a feasible solution S can contain an arbitrary number of centers/facilities (rather than k), however in the objective function, besides the *connection cost* $cost(S)$, one has to pay also the *opening cost* $f|S|$. Then one uses a Lagrangian Multiplier Preserving (LMP) α_{LMP}-approximation for the problem (formal definition in Section 2) to construct a *bi-point* solution for k-median, i.e., a convex combination of a solution S_1 (opening fewer than k centers) and a solution S_2 (opening more than k centers), that opens exactly k centers in a fractional sense. In particular, sampling one of the two solutions according to the respective coefficients would give a solution of expected cost $\alpha_{LMP} \cdot \mathrm{opt}$ that opens k centers *in expectation*. Finally, a bi-point rounding algorithm is used to derive a feasible k-median solution, while losing another factor α_{BPR} in the approximation. The product $\alpha_{LMP} \cdot \alpha_{BPR}$ gives the overall approximation factor. We remark that the state of the art leaves a significant gap between the approximation factor for a solution with k centers *in expectation* and a true k-median solution.

In more detail, for the LMP stage, [37] gave a 3-approximation algorithm which was subsequently improved to 2 by [35, 36], and then to $2-\eta$ for some $\eta > 2.25 \cdot 10^{-7}$ by [21]. [37] also presented a factor 2 algorithm for the bi-point rounding stage. [44] subsequently improved this to a factor $\frac{1+\sqrt{3}}{2} \approx 1.366$ when allowing to use $k + O(1)$ many centers[2], and also gave a polynomial-time algorithm for obtaining a proper k-median clustering from a $k + O(1)$ clustering essentially without any loss in the approximation ratio. This approach was further refined by [11] who gave a 1.337-approximate bi-point rounding algorithm and also showed that no LP-based rounding algorithm can do better than $\frac{1+\sqrt{2}}{2}$. The state of the art for rounding methods is due to [29] who gave a 1.306-approximation algorithm, while showing that no algorithm can do better than $\sqrt{\phi} \approx 1.272$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio.

Unfortunately, there is an inherent limitation to this approach. No approximation algorithm for the LMP stage can do better than $1 + 2/e$ [35, 36], and using the lower bound for bi-point rounding algorithms by [29], the best possible analysis cannot yield an approximation factor better than $(1 + 2/e)\sqrt{\phi} \approx 2.207$. This leaves a significant gap to the inapproximability lower bound of $(1 + 2/e) \approx 1.735$ [35, 36], and even to the best-known approximation algorithm that opens k-centers in expectation [21], which achieves an approximation guarantee of $2 - \eta$ for some small $\eta > 0$.

1.1 Our Results

Our main result is stated as follows. (As commonly done, we use "with high probability" to mean that an event occurs with probability at least $1 - 1/\mathrm{poly}(n)$.)

Theorem 1.1. *For every $\varepsilon > 0$, there is a randomized polynomial-time algorithm for k-median that returns a solution with cost at most $(2 + \varepsilon)\mathrm{opt}$ with high probability.*

[2]Solutions opening slightly more than k centers are sometimes called *pseudo-solutions* in the literature.

The algorithm in our main theorem combines two novel algorithms. First, we address the limitations of the bi-point rounding method by introducing a non-trivial variant of the classic Greedy algorithm [35, 36]. The Greedy algorithm can be highly *adaptive* in that tiny changes to the opening cost f may lead to vastly different solutions S and S', where $|S| < k$ and $|S'| > k$, forcing previous bi-point rounding approaches to incur a large additional factor in the approximation guarantee. We resolve this high-adaptivity by presenting a low-adaptivity version, which makes adaptive decisions in only $O(\log(n)/\varepsilon^2)$ phases. This modification intuitively renders the algorithm less adaptive than the original, where each choice depends heavily on previous ones. We then introduce a novel walk-between-solutions algorithm that enables us to return a solution with at most $k + O(\log(n)/\varepsilon^2)$ centers, with no additional loss in the approximation factor (up to $1 + \epsilon$). Specifically, we achieve the following result.

Theorem 1.2. *For every $\varepsilon \in (0, 1/6)$, there is a polynomial-time algorithm for k-median that returns a solution containing at most $k + O(\log n/\varepsilon^2)$ many centers and of cost at most $(2 + \varepsilon)\mathrm{opt}$.*

Combining this result with the [44] reduction from k-median pseudo-solutions to true k-median solutions already yields a $(2 + \varepsilon)$-approximation in quasi-polynomial time. In order to improve this to a polynomial time algorithm, we use a completely different algorithm. We say that an instance is β-stable if $\mathrm{opt}_{k-1} \geq (1 + \beta) \cdot \mathrm{opt}_k$. Our result is as follows.

Theorem 1.3. *For any $\varepsilon, \zeta > 0$, there exists a randomized polynomial-time algorithm that, given a $(\zeta/\log n)$-stable k-median instance, returns a solution of cost at most $(2 + \varepsilon)\mathrm{opt}$ with high probability.*

The algorithm for stable instances is inspired by a recent fixed-parameter-tractable approach that essentially identifies, by enumeration, small-radius balls guaranteed to contain the optimal centers and then solves the problem by maximizing a submodular function subject to a partition matroid constraint. However, there are two major challenges: first, the running time of [24] is not polynomial even when $k = O(\log n)$; and second, because k may be large (potentially polynomial in n), we need to identify almost all clusters of the optimal solution before proceeding with any guessing/enumeration steps. We address these challenges through several novel ideas, including a sampling approach based on the k-means++ algorithm. An overview of the techniques is provided in Section 3.2.

Let us see why Theorems 1.2 and 1.3 combine to yield Theorem 1.1. We compute a few feasible solutions. Let Δ be the surplus number of centers used in Theorem 1.2. One solution is obtained by running the algorithm from Theorem 1.2 with the number of centers being $k' = k - \Delta$. If $\mathrm{opt}_{k'} \leq \mathrm{opt}_k \cdot (1 + \varepsilon)$, this is a good enough approximation. Then we run the algorithm from Theorem 1.3 for every $k'' \in (k', k]$. If the solution with k' centers computed before is not good enough, then there exists some $k'' \in (k', k]$ such that $\mathrm{opt}_{k''-1} \geq (1 + \beta) \cdot \mathrm{opt}_{k''}$ with $\beta \in \Omega(\varepsilon/\Delta) = \Omega(\varepsilon^3/\log n)$ and $\mathrm{opt}_{k''} \leq (1 + \varepsilon) \cdot \mathrm{opt}_k$. In this case, Theorem 1.3 yields the desired result for a proper choice of the parameters. Outputting the best of all computed solutions yields Theorem 1.1.

For Euclidean spaces, the set of candidate centers is infinite. By a standard reduction, we can find a polynomial-sized set $F \subset \mathbb{R}^d$ in polynomial time such that the cost of any constant factor

Reference	Approximation factor	Technique
[8]	$O(\log n \log\log n)$	tree embeddings
[12]	$O(\log k \log\log k)$	tree embeddings
[14]	6.667	dependent LP rounding
[37]	6	LMP + bi-point rounding
[35, 36]	4	LMP + bi-point rounding
[5]	$3+\varepsilon$	local search
[44]	2.733	LMP + bi-point rounding
[11]	2.675	LMP + bi-point rounding
[21, 29]	2.613	LMP + bi-point rounding
Theorem 1.1	$2+\varepsilon$	LMP interpolation + stable algorithm

Table 1: Progression of approximation factors for k-median. Dependencies on ε are omitted due to numerical overestimation, except for [5] and the final result.

approximate solution is approximated by a set of centers obtained from F, see for example Lemma 5.3 of [40]. Thus we also obtain a $(2+\varepsilon)$-approximation in Euclidean spaces, improving on 2.406 [19].

Finally, let us remark that Theorem 1.1 essentially closes the gap between the approximation guarantee of the best-known algorithm that opens k centers in expectation [21] and algorithms that consistently open at most k centers. However, a major open problem remains: obtaining a tight approximation result for k-median, as it is only known that it is hard to approximate the problem within a factor less than $(1+2/e)$ [36]. Building on this work, a natural next step would be to resolve this question or to significantly improve the approximation guarantee, even in the setting where k centers are opened in expectation.

1.2 Related Work

We already reviewed the LMP-based approximation algorithms earlier. But there are several other important results for k-median and closely related problems which we cover here.

There also exist other approaches for designing approximation algorithms for metric k-median, most notably local search [5, 23, 32, 38], combinatorial algorithms [46], metric embeddings [8, 9, 12], or other LP-based rounding approaches [2, 15, 16]. There has also been a lot of work for the related problems k-means and k-center, which minimize the sum of squared distances and the maximum distance, respectively. Specifically for k-center, [28, 34] gave a 2-approximation which is optimal unless $P \neq NP$. k-means admits a $(9+\varepsilon)$-approximation in general metrics [1]. In addition, both k-median and k-means have been studied in d-dimensional Euclidean spaces, where a PTAS is possible when either d or k are fixed [3, 20, 27, 40]. If neither d nor k are fixed, the approximation factors for Euclidean k-median and k-means are also better than their general metric counterparts [1, 30], with the best known approximation ratio being 5.913 for k-means and, prior to our work, being 2.406 for k-median by [19]. Both in high dimensional Euclidean spaces as well as general metrics, all of these aforementioned problems are APX hard [7, 18, 25, 33, 36, 41].

LMP-based approaches are also very relevant for the related facility location problem. Mahdian et al. [45] showed that an LMP $(1+2/e)$-approximation yields a $(\gamma_{GK} \approx 1.463)$-approximation for UFL, which matches the hardness result of Guha and Khuller [31] where γ_{GK} is the solution of $\gamma = 1+2e^{-\gamma}$. Nevertheless, most improvements do not rely on an LMP approach. Following a long line of research [10, 13, 17, 35, 36, 39, 45, 48], Li presented an algorithm with the currently best known approximation ratio of roughly 1.488 [42].

2 PRELIMINARIES

We are given a set of n clients D and set of m potential centers or facilities F. Given a set $S \subseteq F$, we use $d(j,S) = \min_{i\in S} d(j,i)$, where d is the underlying distance function. $\tilde{O}(x)$ suppresses polylog(x) factors, and $O_\varepsilon(x)$ suppresses polynomial factors in ε. We use $[a]^+ = \max(a,0)$. We already defined the k-median problem. The uncapacitated facility location problem (UFL) with uniform opening costs is given a non-negative number f and asks to find a set $S \subseteq F$ minimizing

$$\mathrm{cost}_{UFL}(S) = \sum_{j\in D} d(j,S) + f\cdot|S|.$$

We refer to $\sum_{j\in D} d(j,S)$ as the *connection cost* and $f\cdot|S|$ as the *opening cost*. Let $x_{i,j}$ denote in the indicator variable for serving client j with facility i and let y_i be the indicator variable for opening facility i. Relaxing these to continuous variables in $[0,1]$, the LP relaxation to k-median and UFL are given by the programs

$$\begin{aligned}
\min \sum_{i\in F, j\in D} d(i,j)x_{i,j} & & (LP_{km})\\
s.t. \sum_{i\in F} x_{i,j} \geq 1 & & \forall j\in D\\
y_i - x_{i,j} \geq 0 & & \forall j\in D, i\in F\\
\sum_{i\in F} y_i \leq k & & \\
x,y \geq 0 & &
\end{aligned}$$

and

$$\begin{aligned}
\min \sum_{i\in F, j\in D} d(i,j)x_{i,j} + f\cdot \sum_{i\in F} y_i \qquad & (LP_{UFL}(f)) \\
s.t. \sum_{i\in F} x_{i,j} \geq 1 \qquad & \forall j \in D \\
y_i - x_{i,j} \geq 0 \qquad & \forall j \in D, i \in F \\
x, y \geq 0 &
\end{aligned}$$

The constraint $\sum_{i\in F} x_{i,j} \geq 1$ ensures that every client is served by some facility and the constraint $y_i - x_{i,j} \geq 0$ ensures that a client j can only be served by facility i if i is open. Moreover, the dual for the facility location LP relaxation is given by the equations:

$$\begin{aligned}
\max \sum_{j\in D} \alpha_j \qquad & (DP_{UFL}(f)) \\
s.t. \sum_{j\in D} [\alpha_j - d(i,j)]^+ \leq f \qquad & \forall i \in F \\
\alpha \geq 0 &
\end{aligned}$$

The usual interpretation of the α_j is that they are the budgets for each client, accounting for both connecting the client to a facility, as well as opening a facility. The opening contribution dual variables $\beta_{i,j}$ obtained from the $y_i - x_{i,j} \geq 0$ constraints, as well as the dual constraints $\alpha_j - \beta_{i,j} \leq d(i,j)$ are removed here, as we can always set $\beta_{i,j} = [\alpha_j - d(i,j)]^+$. Recall that opt $=$ opt$_k$ is the cost of an optimal k-median clustering. We use opt$_{LP}$ and opt$_{LP}(f)$ to denote the cost of an optimal solution to LP_{km} and $LP_{UFL}(f)$, resp. Trivially opt $\geq$ opt$_{LP}$. Since any feasible solution for LP_{km} is also feasible for $LP_{UFL}(f)$, one has opt$_{LP} \geq$ opt$_{LP}(f) - kf$. We will use the fact that, by weak duality, any feasible solution to $DP_{UFL}(f)$ induces a lower bound on opt$_{LP}(f)$. In particular, for any feasible solution α to $DP_{UFL}(f)$, we have opt $\geq$ opt$_{LP}(f) - kf \geq \sum_{j\in D} \alpha_j - kf$.

We say that an algorithm is a Lagrangian Multiplier Preserving (LMP) α-approximation algorithm for UFL if it returns a solution S with

$$\text{cost}(S) \leq \alpha \cdot (\text{opt}_{LP}(f) - f|S|).$$

Finally, we will make the assumption that the distances are integers in $[1, n^3/\varepsilon]$. The justification is given by the following standard reduction.

Lemma 2.1. *For any constants $\varepsilon > 0$ and $\alpha > 1$, given a polynomial-time α-approximation algorithm for k-median on instances with distances in $\{1, \ldots, n^3/\varepsilon\}$, there exists a polynomial-time $\alpha(1 + O(\varepsilon))$-approximation algorithm for k-median on general instances.*

Proof. Consider any input instance $(D \cup F, d)$ of k-median. Assume that $n := |D|$ is large enough w.r.t. α, otherwise the problem can be solved by brute force in polynomial time. We guess $M := \max_{j\in D} d(j, \text{OPT})$, where OPT is some optimal k-median solution, by trying all the $n \cdot m$ possibilities. Next, for each $(i,j) \in F \times D$, define $d'(j,i) = d'(i,j) = \max\{1, \lceil \frac{d(i,j)}{M} \frac{n}{\varepsilon} \rceil\}$ if $d(i,j) \leq M$. Set all the remaining $d'(a,b)$, $a \neq b$, to $\frac{n^3}{\varepsilon}$. Finally, replace the $d'(i,j)$'s with the corresponding metric closure. We run the given α-approximation algorithm on the k-median instance $(D \cup F, d')$, hence obtaining a solution S. It is easy to check that S is a good enough approximation for the input instance. □

3 TECHNICAL OVERVIEW

In this section, we give an overview of our result. Section 3.1 introduces ideas behind Theorem 1.2 that achieves a (2+ε)-approximation while opening $k + O(\log n/\varepsilon^2)$ centers, and Section 3.2 discusses Theorem 1.3 for stable instances. Due to space constraints, we are not able to include full proofs in this extended abstract. For this, we refer the reader to the arxiv version of our paper [22].

3.1 $(2+\varepsilon)$-Approximation with $O_\varepsilon(\log n)$ Extra Centers

Let us motivate our high-level plan by revisiting the previous framework. The previous best approximation algorithms [11, 21, 29, 43] for k-median start by the following initialization step: starting from two LMP approximate solutions for UFL $S, S' \subseteq F$ with the respective facility costs f, f' such that $|S| < k, |S'| > k$[3] and f possibly much larger than f', the algorithm uses binary search to obtain two new solutions $(S,f), (S',f')$ such that $|S| < k, |S'| > k$ and $|f - f'| \leq 2^{-n}$. This step can be already viewed as a *merging* step that takes two solutions whose number of facilities *sandwich k* and makes them (specifically the facility cost) closer. Then, the natural question is *can we merge S and S' into the (almost) same set of open facilities so that both open (almost) exactly k facilities?*

One naive implementation of this idea might be the following (from now on, let us assume $f = f'$). Just like binary search, suppose that we find an LMP approximate solution S'' which is *strictly between* S and S' in some formal sense. If $|S''| > k$, we will let $S' \leftarrow S''$ and otherwise $S \leftarrow S''$. Then the invariant $|S| < k$, $|S'| > k$ is maintained while the two solutions got *strictly closer*. So we can continue this process until they become the same!

Of course, the main question is whether one can find such S'' given S and S'. Just like S and S', S'' must be an outcome of the LMP approximation algorithm, or at least close to such an outcome in some formal sense. Since *being a valid outcome of an algorithm* is a highly complex property, especially when the algorithm has a large number of *adaptive* steps whose action depends on the previous actions, this task remains challenging for a general LMP approximation algorithm.

However, there is an example of success for Jain and Vazirani's primal-dual [37] algorithm, which gives a LMP 3-approximation. At a high level, this algorithm consists of two phases, where the first phase yields a set T of tentative open facilities, and the second phase is allowed to choose an arbitrary $S \subseteq T$ that forms a maximal independent set in an auxiliary graph, so there is a freedom in choosing the independent set. Ahmadian et al. [1] used this freedom to give a walking procedure that finds a maximal independent set in the auxiliary graph of cardinality exactly k, which led to an improved approximation ratio for Euclidean k-means/k-median and metric k-means.

However, the Greedy algorithm [35, 36] with dual-fitting analysis, which guarantees a better 2-LMP approximation, is more adaptive; in fact, it can be a completely deterministic algorithm without leaving any freedom. Therefore, in order to implement our idea, it would be better to find a low-adaptive version of the Greedy algorithm, where the algorithm runs in a small ($O_\varepsilon(\log n)$ in our

[3]Throughout this overview, we assume no solution opens exactly k centers, because we are done if this happens.

case) number of *phases*, and each phase presents some freedom of choices for the algorithm.

The Greedy Algorithm. Let us recall the Greedy algorithm [35, 36]. The algorithm maintains the set A of *active* clients and the set S of open facilities. Initially, all clients are active, i.e., $A = D$, and no facilities are opened, i.e., $S = \emptyset$. Additionally, the algorithm maintains dual variables $(\alpha_j)_{j\in D}$, initialized to 0.

While there is at least one active client, the algorithm uniformly increases the α-values of the active clients until there is a facility i whose "bids" are sufficient to open it (or an active client becomes "close" to an already opened facility). The bids from a client j to a facility i depend on whether j is active or not. If $j \in A$, then a portion of its budget α_j, specifically $d(i,j)$, is allocated to cover the connection cost, and the remainder, $\alpha_j - d(i,j)$, is used to bid towards the facility's opening cost $\hat{f} := 2f$. For a client $j \notin A$, which is already connected to an open facility at a distance $d(j,S)$, the bid towards i is the difference in distance, i.e., $d(j,S) - d(i,j)$ (or 0 if i is farther). Finally, the algorithm marks any client j with $\alpha_j \geq d(j,S)$ as inactive, ensuring that at any point in time, active clients j satisfy $\alpha_j \leq d(j,S)$.

In order to prove that the algorithm yields an LMP 2-approximation, let $(\alpha_j^*)_{j\in D}$ be the final α values and S^* be the set of opened facilities. The design of the algorithm guarantees that the α^* values pay for both the opening costs of the facilities in S^* and the connection costs of the clients, leading to $\sum_{j\in D}\alpha_j^* = \sum_{j\in D} d(j,S^*) + \hat{f}\cdot|S^*|$.

The more interesting part of the analysis is to show that $\alpha^*/2$ is a dual feasible solution for the dual facility location LP with facility cost f, namely $\sum_{j\in D}[\alpha_j^* - 2d(i,j)]^+ \leq \hat{f}$ for every facility i. The dual feasibility follows from two key properties that, at any point in time, (1) $\alpha_j \leq d(j,S)$ for every $j \in A$ and (2) for every $i \in F$, no facility is overbid:

$$\sum_{j\in A}[\alpha_j - d(i,j)]^+ + \sum_{j\in D\setminus A}[d(j,S) - d(j,i)]^+ \leq \hat{f}\,. \qquad (1)$$

3.1.1 Logarithmic Adaptivity. As stated, the Greedy Algorithm is a highly adaptive algorithm; for instance, consider the case when two facilities i and i' both become paid for at the same time θ. Whether we open i or i' can dramatically change the bids to future facilities and completely change the trace of the algorithm.

To reduce such high adaptivity, we limit the number of time steps where such decisions are made. Specifically, our high-level goal is to only consider the time steps of the form $(1+\varepsilon^2)^0, (1+\varepsilon^2)^1, (1+\varepsilon^2)^2, \ldots$. Then, as the minimum nonzero distance is 1 and the maximum distance is $\mathrm{poly}(n)$, the total number of time steps that we need to consider is $O(\log n/\varepsilon^2)$, and this is what we refer to as $O(\log n)$ adaptivity.

The main difficulty in implementing this idea is to ensure (1). Consider the situation where all active clients have α value $\theta = (1+\varepsilon^2)^t$. If their α values jump from θ to $(1+\varepsilon^2)\theta$, it is possible that a previously underbid facility becomes strictly overbid.

We fix this issue in two steps. The first easy fix is to *scale down the metric* by a factor of $(1-\delta)$ where $\delta = O(\varepsilon)$; that is, we consider a facility i paid for if

$$\sum_{j\in A}[\alpha_j - (1-\delta)d(i,j)]^+ + (1-\delta)\sum_{j\in D\setminus A}[d(j,S) - d(j,i)]^+ \geq \hat{f}\,. \quad (2)$$

At the cost of losing a factor $1/(1-\delta)$ in the approximation guarantee, this has the following benefit. If i is not paid according to (2) when all active clients have $\alpha_j = \theta$, then even when α_j's are increased to $(1+\varepsilon^2)\theta$, for a client j with $d(j,i) \geq \varepsilon\theta$, its contribution $[(1+\varepsilon^2)^{t+1} - d(i,j)]^+$ to (1) is at most its previous contribution $[(1+\varepsilon^2)^t - (1-\delta)d(i,j)]^+$ to (2), which makes (1) unlikely to be violated.

Then our second fix deals with active clients close to co-located with i. We do so by allowing the clients j in the ball $B(i,\varepsilon\theta)$ to increase their bids for i up to $\min((1+\varepsilon^2)\theta, (1-\delta)d(j,S))$, so that if i could not be opened even with increased bids from nearby clients, then increasing active clients' α values from θ to $(1+\varepsilon^2)\theta$ should not make i overbid. These increased bids bring other considerations (e.g., ensuring dual feasibility of other facilities), leading to the following definition of *openable* facilities, which can be efficiently checked by a linear program.

Definition 1. *A facility i is* openable *if there are bids $(\tau_j)_{j\in A}$ of active clients that satisfy the following conditions.*

- *For every $j \in A \setminus B(i,\varepsilon\theta)$, $\tau_j = \alpha_j$.*
- *For every $j \in A \cap B(i,\varepsilon\theta)$, $\tau_j \in [\alpha_j, \min\{(1-\delta)d(j,S), (1+\varepsilon^2)\theta\}]$*
- $\sum_{j\in A}[\tau_j - (1-\delta)d(i,j)]^+ + (1-\delta)\sum_{j\in I}[d(j,S) - d(i,j)]^+ \geq \hat{f}$.
- *For every facility i_0 and $k \in A$,* $\sum_{j\in A}[\tau_j - d(i_0,j)]^+ + \sum_{j\in I}[\tau_k - 2d(j,i_0) - d(k,i_0)]^+ \leq \hat{f}$.

Given this definition of openability, we present our modified Greedy algorithm with $O(\log n)$ phases, LogAdaptiveAlgorithm, which is restated here as Algorithm 1. Here a *phase* is defined to be the run of the algorithm for a fixed value of $\theta = 1, (1+\varepsilon^2), (1+\varepsilon^2)^2, \ldots$, which is broken down to two *stages* according to the description of Algorithm 1. Note that at least in the beginning, stage 1 allows opening of any openable facility, providing the desired freedom to walk between two solutions within the phase. While stage 1 still presents some adaptivity as an opening of a facility might impact the openability of another, it is sufficient for the merging procedure described in Section 3.1.2.

For the analysis, we prove that LogAdaptiveAlgorithm yields an LMP $(2+O(\varepsilon))$-approximate solution for UFL. It is a nontrivial extension of the dual-fitting analysis of the Greedy algorithm for our definition of openability and design of the algorithm.

3.1.2 Merging Two Solutions. We describe our final algorithm MergeSolutions that maintains two LMP approximate solutions whose number of facilities sandwich k and gradually merges them until one of them opens almost exactly k facilities.

Setup. Based on the discussion about LogAdaptiveAlgorithm, we will describe a solution $\mathcal{H}$ as $(H_1, \ldots, H_L)$ with $L = O(\log n/\varepsilon^2)$, where H_i is a *valid execution* of stage 1 of phase i (i.e., when $\theta = (1+\varepsilon^2)^{i-1}$), which is formally represented as a sequence of opened facilities. H_i's validity also requires *maximality*, meaning that no openable facility remains in the phase. In order to facilitate the merging, given a phase, we extend the notion of a valid execution to include the following actions.

- We allow opening a *free facility*, at most three times per phase. A free facility $\tilde{i}$ is based on a regular facility $i \in F$, and

Algorithm 1 (LogAdaptiveAlgorithm).

Initialization: Set $\hat{f} = 2f, S = \emptyset, A = D, \theta = 1$ and $\alpha_j = \theta$ for every $j \in A = D$.

Repeat the following while $A \neq \emptyset$:

(1) While there is an unopened facility i that is openable (chosen arbitrarily if there are many),
    Calculate $(\tau_j)_{j\in A}$ so that i is openable with bids $(\tau_j)_{j\in A}$. Open i (i.e., add it to S), let $\alpha_j \leftarrow \tau_j$ for every $j \in A$ (note that only the values of the clients in $A \cap B(i, \varepsilon\theta)$ change), mark all clients $j \in A$ such that $\alpha_j \geq (1-\delta)d(j,S)$ as served (i.e., remove them from A).

(2) Perform the following step and proceed to the next phase.
    For every remaining $j \in A$, let $\alpha_j \leftarrow \min((1+\varepsilon^2)\theta, (1-\delta)d(j,S))$, remove it from A when $\alpha_j = (1-\delta)d(j,S)$. Update $\theta = (1+\varepsilon^2)\theta$.

when it is created, we can choose the distance parameter $u(\tilde{i}) = d(\tilde{i}, i)$. All other distances $d(\tilde{i}, x)$ are defined to be $u(\tilde{i}) + d(i,x)$.

- We slightly relax the notion of openability so that one can open a facility i by paying $\hat{f} - n\eta$ instead of $\hat{f}$ for parameter $\eta := 2^{-n}$ (in the third bullet of Definition 1).

Call $\mathcal{H} = (H_1, \ldots, H_L)$ a *solution of η-valid sequences* if H_i is a valid execution (i.e., an η-valid sequence) of the ith phase in the above relaxed notion given $H_1, \ldots, H_{i-1}$. We extend the analysis of LogAdaptiveAlgorithm to show that such $\mathcal{H}$ in the above sense still yields an LMP $(2+O(\varepsilon+n\eta))$-approximation, if we do not count the opening cost of the free facilities (while using the connections they provide). Therefore, a solution of η-valid sequences that opens exactly k regular facilities, and at most 3 free facilities per phase, is a $(2+O(\varepsilon+n\eta))$-approximation to k-median that opens at most $O(L) = O(\log n/\varepsilon^2)$ extra facilities.

Initially, by the standard binary search on the facility cost and running LogAdaptiveAlgorithm, we start with two solutions $\mathcal{H} = (H_1, \ldots, H_L)$ and $\mathcal{H}' = (H'_1, \ldots, H'_L)$ with facility costs f and f' such that $|f - f'| \leq \eta$ and their number of opened facilities sandwich k. (We will associate parameters (f,u) for $\mathcal{H}$ and (f',u') for $\mathcal{H}'$.) Throughout the algorithm, we will maintain the invariants that (1) $\mathcal{H}$ and $\mathcal{H}'$ open the same set of free facilities (so that u and u' are defined on the same set), (2) their number of opened regular facilities sandwich k, and (3) $(f,u) = (f',u')$ except for one parameter f or $u(\tilde{i})$ for one free $\tilde{i}$, whose values in the two solutions differ by at most η. Call such a parameter the *difference parameter*.

For each $i = 1, \ldots, L$, we will outline a procedure that, given that the prefixes $(H_1, \ldots, H_{p-1})$ and $(H'_1, \ldots, H'_{p-1})$ are identical, modifies the suffixes $(H_p, \ldots, H_L)$ and $(H'_p, \ldots, H'_L)$ such that H_p and H'_p become identical while satisfying the above invariants, unless we already find a solution that opens exactly k regular facilities. This will allow us to gradually merge the solution until one of them opens exactly k regular facilities (and at most $O(L)$ free facilities).

How to walk in each phase. Finally, we focus on phase p and introduce ideas behind ensuring $H_p = H'_p$. Our first step is to change the difference parameter value of $\mathcal{H}'$ to that of $\mathcal{H}$; formally, we let $\mathcal{H}'' = \text{CompleteSolution}_{(f,u)}(H'_1, \ldots, H'_p)$, which is the solution resulting from running LogAdaptiveAlgorithm from a given partial solution $(H'_1, \ldots, H'_p)$ with parameters (f,u) instead of (f',u'). Such a change of the difference parameter is the reason for introducing the $n\eta$ slack for η-valid sequences; for instance, if H_i was the exact execution with parameters (f',u'), we show that it is still η-valid with respect to (f,u).

If $\mathcal{H}''$ opens k regular facilities we are done. Also, if the number of open regular facilities of $\mathcal{H}''$ and $\mathcal{H}'$ sandwich k, then we already made progress as they now coincide on the pth phase while still having one difference parameter. Therefore, we only need to handle the case where $\mathcal{H}''$ and $\mathcal{H}'$ sandwich k, where the difference parameters already became identical but H_p and H'_p are different. Let us rename $\mathcal{H}''$ to $\mathcal{H}'$.

Now our goal is to gradually change H'_p to H_p (i.e., *walk* from H'_p to H_p). Instead of describing the full procedure, we illustrate the procedure on a particular example. Suppose that $H_p = \langle i_1 \rangle$, $H'_p = \langle i_2, \ldots, i_6 \rangle$ where one can open at most one between i_1 and i_j for any $j \in \{2, \ldots, 6\}$; this constraint can be expressed as a star graph with i_1 as the center vertex and $i_2, \ldots, i_6$ are only connected to i_1, and a valid solution (i.e., η-valid sequence) here means a maximal independent set. (Suppose that this is the only constraint on openings, so H_p and H'_p can be thought of as sets instead of sequences.)

In this example, H_p and H'_p are the only two maximal independent sets, so it is impossible to *smoothly* walk from H'_p to H_p. Here is where the notion of a free facility becomes useful because we can assume that having a free facility helps *maximality* while not hurting *independence*. Then, from H'_p, by (Step 1) adding a free copy of i_1 called $\tilde{i}_1$ with $u(\tilde{i}_1) = 0$, (Step 2) removing $i_2, \ldots, i_6$ one-by-one, and (Step 3) converting $\tilde{i}_1$ to a regular i_1, we obtain a sequence of maximal independent sets where a consecutive pair of independent sets differ by at most one facility. Figure 1 illustrates this process.

For each intermediate H'_p, we consider its completion (i.e., $\text{CompleteSolution}_{(f,u)}(H'_1, \ldots, H'_p)$) and see how many regular facilities it opens. Since the completions from the initial H'_p and H_p sandwich k, there must be a consecutive pair in the above sequence of independent sets (say H^a_p and H^b_p) that sandwich k, and they differ by only one facility i_q (say H^a_p contains i_q while H^b_p does not). For future execution of the algorithm, H^a_p and H^b_p are the same as the solutions that have the free copy $\tilde{i}_q$ (instead of i_q) with the $u(\tilde{i}_q)$ value 0 (for H^a_p) and ∞ (or sufficiently large number, for H^b_p). Therefore, by performing binary search on the $u(\tilde{i}_q)$, just

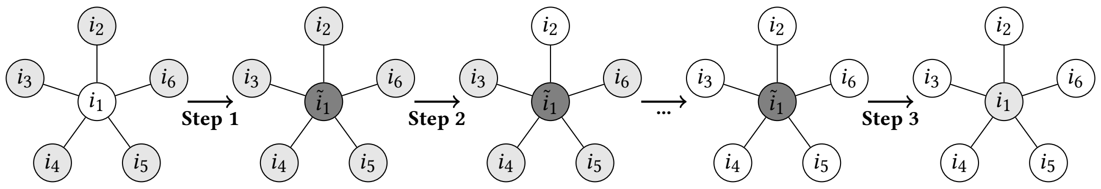

Figure 1: How we walk from H'_p to H_p. Dark grey color denotes a free facility and light grey color denotes regular open facilities.

like we do on the facility cost f, we yield two solutions that are now identical up to phase p except the η difference on the only one different parameter $u(\tilde{i}_q)$, which archives our goal of merging two solutions at phase p!

In general, the fact that H_p and H'_p are not just sets but sequences of opened facilities requires more technical consideration. However, the full procedure is still an extension of this idea, trying to increase the length of the common prefix between H_p and H'_p by walking from H'_p to H_p by one coordinate at a time, until two consecutive solutions are sandwiching k.

3.2 $(2+\varepsilon)$-Approximation for $O_\varepsilon(\frac{1}{\log n})$-Stable Instances

The other component of our algorithm is an improved approximation to k-median assuming that the instance has a weak version of a structural property known as stability, initially formulated by [47]. Recall that an instance is β-stable if $\text{opt}_{k-1} \geq (1+\beta)\text{opt}_k$. For any β-stable instance, [6, 26] showed how to obtain a PTAS for the problem in time $n^{\beta^{-1}\varepsilon^{-O(1)}}$ and [6] left as an open question as whether one could obtain a PTAS with running time $f(\beta,\varepsilon)n^{O_\varepsilon(1)}$ for any computable function f for general metrics (they show it is possible for Euclidean metrics). We would indeed like to have an algorithm running in time $2^{\text{poly}(\beta,\varepsilon)}n^{O_\varepsilon(1)}$ to solve our $(\varepsilon/\log n)$-stable instances.

Unfortunately, since any k-median instance is $\Omega(1/k)$-stable, obtaining a PTAS with the above running time would violate Gap-ETH [24]: The k-median problem cannot be approximated better than $1+2/e$ in time $f(k,\varepsilon)n^{O_\varepsilon(1)}$ assuming Gap-ETH.

The saving grace here is that we are not after a $(1+\varepsilon)$-approximation algorithm but we can use any 2-approximation with the above running time. Since a connection between β-stable and FPT algorithm exists (as explained in the above discussion), it is tempting to make the journey from FPT algorithms back to β-stable instances and use the state-of-the-art approximation algorithm of [24] to solve β-stable instances. Unfortunately the algorithm presented in [24] runs in time $(\log n)^{\tilde{O}(k/\varepsilon)}n^{O(1)}$ and so importing the techniques directly would result in a running time of the form $(\log n)^{\tilde{O}((\beta/\varepsilon)^{-1})}n^{O(1)}$ and not to a polynomial running time for our setting. We must thus go beyond the state-of-the-art techniques for FPT and β-stable k-median instances.

3.2.1 *Local Optima and Stability.*

One particularly important tool when dealing with stable solutions are properties of local optima. A solution S is $(1+\varepsilon)$-locally optimal if $\text{cost}(S')\cdot(1+\varepsilon) \geq \text{cost}(S)$ for any solution S' with $|S \triangle S'| \leq 2$, i.e., that S and S' differ by one center. Local search enumerates over all swaps, terminating when none with sufficient cost improvement can be found, upon which we say that the found solution is locally optimal. The running time for executing such a swap is polynomial. In the following, we will ignore dependencies on the $(1+\varepsilon)$ factor controlling the rate of improvement.

We first introduce a bit of notation. Suppose we are given a locally optimal solution S. We say that a cluster C^* of the optimal solution OPT is *pure* with respect to S, if there exists a cluster C' induced by S such that both clusters agree on all but $O(\varepsilon)\cdot\min(|C^*|,|C'|)$ of the points. We begin by examining structural properties for β-stable solutions.

- For any given constant-factor approximation S, at most $O_\varepsilon(\beta^{-1})$ many clusters are not pure.
- In any locally optimal solution S, we have
$$\sum_{p\in C^*,\ C^* \text{ pure}} d(p,S) \leq \sum_{p\in C^*,\ C^* \text{ pure}} d(p,\text{OPT}) + O(\varepsilon)\cdot\text{opt},$$
that is the cost of the clients in pure clusters is well approximated.

The first property relies on the fact that for every non-pure cluster, S must merge at least two clusters of the optimum. If S is a constant-factor approximation, this cannot happen more than $O_\varepsilon(\beta^{-1})$ many times before the approximation guarantee no longer holds. The second property is implied by the local optimality of S. Specifically, swapping out the center $c' \in S$ serving C' for the center c^* of the pure cluster C^* cannot change the cost by much as these two clusters agree on most of their points.

In a nutshell, a locally optimal solution S approximates all but $O_\varepsilon(\beta^{-1})$ many clusters of the optimal solution very well. In what follows, we will describe how we may extend S to a $(2+\varepsilon)$-approximation overall.

3.2.2 *Identifying the Non-Pure Clusters.*

To find a good approximation for the non-pure clusters, we would like to identify these clusters. With the discussion of the last section, we may assume that we have a locally optimal solution S for which all but $O_\varepsilon(\beta^{-1}) = O_\varepsilon(\log n)$ many clusters are well approximated. Immediate ideas, such as determining the badly approximated clusters by brute force, will not improve over the quasi-polynomial running time. Instead, we look towards fixed parameter tractable (FPT) approaches, parameterized by the number of centers. In fact, we would like an FPT approximation algorithm *parameterized by the number of non-pure clusters.* Starting point is an FPT algorithm based on submodular maximization, initially due to [24], which we first review. However, recall that the running time of this algorithm is $(\log k)^{\tilde{O}(k\varepsilon^{-1})}$ and so will not be enough for our purpose. We thus introduce new ideas to speed things up.

Review of the FPT approximation algorithm of [24]. Suppose c^* is a center of cluster C^* in an optimal solution OPT. [24] introduce the notion of a *leader* client $\ell \in C^*$ which minimizes the distance to c^*. Using coresets to first reduce the number of clients to $\tilde{O}(k \log n/\varepsilon^2)$, the leaders of all clusters, as well as the distances $d(\ell, c^*)$ can be identified in FPT-time, i.e. in time $(\log n)^{\tilde{O}(k\varepsilon^{-1})} \cdot n^{O(1)}$. A solution S obtained by simply selecting an arbitrary facility within distance $d(\ell, c^*)$ from ℓ yields a 3-approximation. [24] improve upon this by showing that for any two candidate sets of centers S' and Λ

$$\mathrm{cost}(D, \Lambda) - \mathrm{cost}(D, \Lambda \cup S')$$

is a monotone submodular function.

The first part which consists in guessing the leaders is the computational bottleneck since optimizing a monotone submodular function can be done in polynomial time.

We next describe how we deal with the clusters that are not well approximated. Recall that we first compute a locally optimal solution S. By the properties of local optima, S is already a sufficiently good solution for the pure clusters. For the remaining $O_\varepsilon(\log n)$ non-pure clusters, we only have coarser approximation bounds.

Leader Identification in Polynomial Time via D^2-Sampling. Our next high-level goal is to find leaders of the non-pure clusters and guess their (approximate) distances to the respective optimum centers in time $2^{\beta^{-1}\varepsilon^{-O(1)}} n^{O(1)}$ for β-stable instances.

[24] identify leaders in time $\binom{k \log n}{k}$ by sparsifying the instance via coresets and then determining the distance of leaders to the optimal solution in time $(\log n)^k$. Our overall approach significantly speeds up these two steps by way of distance sampling similar to k-means++ [4].

To identify the leaders faster, we lift the stringent requirement that the leader of cluster C^* is the client closest to the center c^* in the optimal solution. Instead, we consider any client among the $\varepsilon \cdot |C^*|$ closest points to c^* as a valid leader with $\mathrm{avg}_{C^*,\mathrm{OPT}}$ being the maximum distance of any leader of C^* to c^*. We later show that this only induces a minor cost increase in the total approximation bound.

The second relaxation is that we only aim to target clusters that are sufficiently expensive in our locally stable solution S. More precisely, a non-pure cluster C^* with center c^* is *basic cheap* if the total cost of its leaders in S is at most $\varepsilon^{O(1)} \cdot \beta \cdot \mathrm{opt} = \frac{\varepsilon^{O(1)}}{\log n} \cdot \mathrm{opt}$ or $d(c^*, S) \le (1+\varepsilon)\mathrm{avg}_{C^*,\mathrm{OPT}} + \frac{\varepsilon}{|C^*|} \cdot \sum_{p \in C^*}(d(p, c^*) + d(p, S))$. Then we can show that S is already an almost 2-approximation for these clusters modulo some additive slacks, namely

$$\sum_{p \in C^*,\ C^* \text{ basic cheap}} d(p, S) \le 2 \sum_{p \in C^*,\ C^* \text{ basic cheap}} d(p, \mathrm{OPT}) + O(\varepsilon \cdot \mathrm{opt}).$$

. It follows from triangle inequality $d(p, S) \le d(p, c^*) + d(c^*, S)$ for any p; if C^* is basic cheap via the second condition above, it gives a direct upper bound on $d(c^*, S)$, and if it is basic cheap via the first condition, we use the fact that the number of basic cheap clusters (which is less than the number of non-pure) is $\frac{\log n}{\varepsilon^{O(1)}}$ so the sum of their costs in S is $O(\varepsilon \cdot \mathrm{opt})$. This essentially allows us to focus only on non-pure and non-basic-cheap clusters.

We use these two relaxations as follows. Let W be a sample of clients picked independently and proportionately to $\frac{d(p,S)}{\mathrm{cost}(S)}$. With probability $\varepsilon^{O(1)}/\log n$, we are picking a point from a non-pure, non-basic cheap cluster and moreover, conditioned on picking a point from a non-pure, non-basic cheap cluster, with probability at least $\varepsilon^{O(1)}$, the picked point will be a leader. Thus, when picking $O_\varepsilon(\log n)$ samples, we have picked a leader from all but a small fraction of non-pure, non-basic cheap clusters. Thereafter, we can show that with good probability the contribution of the unpicked clusters to the total cost in S is very small. Since we have sampled $O_\varepsilon(\log n)$ clients, we can enumerate all the subsets in polynomial time and focus on a subset that contains all the leaders of the non-basic cheap, non-pure clusters.

Using a constant number of potential guesses per leader, we then show how to get a sufficiently good approximation of the distance ρ from each leader to its closest optimum center. Specifically, the cost of the leader in solution S is a rough approximation, so we can interpolate around it.

Existence of a Good Solution. We now use W from last section to identify a modified solution from S that is $(2 + O(\varepsilon))$-approximate. Let $\hat{OPT}$ denote the set of centers of non-pure, non-basic cheap clusters that have a leader in W. The cost of all the other clusters, i.e., the non-pure, non-basic cheap clusters that do not have a leader in W is nearly optimal. We thus wish to add to S the optimal centers of $\hat{\mathrm{OPT}}$ and remove an equally sized set $S_0 \subseteq S$. An important constraint on S_0 is that it contains no center c such that $c = \operatorname*{argmin}_{c' \in S} d(c^*, c')$ for any center $c^* \in \mathrm{OPT} - \hat{\mathrm{OPT}}$. Intuitively, we do not wish to remove any centers that are serving the majority of points from a pure cluster, or centers that are used to serve basic cheap clusters. We show that such an S_0 exists such that $M_O := S - S_0 \cup \hat{\mathrm{OPT}}$ is a $(2+\varepsilon)$-approximation: If a cluster of OPT is pure, basic cheap, or does have a leader in W, there is one center in S serving their clients up to a $2+\varepsilon$ factor; for the remaining, we would have their optimum center (in $\hat{\mathrm{OPT}}$).

So our goal from now on is to approximate the solution M_O, where $|S_0| = \hat{\mathrm{OPT}} = O_\varepsilon(\log n)$, and this final goal has two key components: (1) removing some existing centers from S and (2) adding the centers from $\hat{\mathrm{OPT}}$. The issue is that stated as such this seems as hard as solving the k-median problem. We first address (1), and we will address (2) by making use of the leaders we have correctly guessed for the centers in $\hat{\mathrm{OPT}}$ at the previous steps.

Center Removal. Let us first try to address (1). Remember we start with solution S, and the leader and radii guesses from the previous steps. The key issue with (1) is that the clients in some of the clusters we would like to remove (i.e., served by S_0 in solution S) are actually served by a center in $\hat{\mathrm{OPT}}$, but we do not know which are these centers yet! We will thus make use of some approximate loose bound on their cost in solution $S - S_0 \cup \hat{\mathrm{OPT}}$ by introducing *dummy* centers Λ. For each leader ℓ and guessed distance/radii ρ, we will introduce a dummy center at distance ρ from ℓ and at distance $d(p, \ell) + \rho$ from any other vertex p.

If we promise to open a center in the ball of radii ρ around ℓ – which solution $S - S_0 \cup \hat{\mathrm{OPT}}$ does – then we will guarantee that there will be a center not further away than the dummy centers in the final solution. We now work with this solution $S \cup \Lambda$ which provides a worst-case fall back on where the centers in $\hat{\mathrm{OPT}}$ could be. We can show that $M_D := S \cup \Lambda - S_0$ is a 3-approximate solution.

Equipped with this we proceed to our 3-step center removal. We know that we have at most $|S_0| = O_\varepsilon(\log n)$ centers we need to remove. Our first procedure aims at removing the centers of clusters of high cost in S_0. Start with solution $S \cup \Lambda$. For $O_\varepsilon(\log n)$ times, we sample a center $c \in S$ proportionally to the cost of its cluster in solution $S \cup \Lambda$, remove c from S with probability $1/2$, and repeat on the resulting solution. Let Q be the centers removed from S. Then with probability at least $2^{-O_\varepsilon(\log n)}$ the procedure succeeds by either obtaining $Q = S_0$ or ensuring that the total cost of the clusters of $S_0 - Q$ in the solution is at most $O(\varepsilon \cdot \mathrm{opt})$. This indeed follows from the fact that, if we only removed centers $Q \subseteq S_0$ up to a certain step, the solution is never more than a $(3+O(\varepsilon))$-approximation (since $S \cup \Lambda - S_0$ is a $(3+O(\varepsilon))$-approximation) and so the probability of sampling a new center of $S_0 - Q$ is $\Omega(\varepsilon)$, unless the total cost of the clusters of $S_0 - Q$ is small. Therefore, repeating the process $n^{O_\varepsilon(1)}$ times, we have that one of the runs succeeds with high probability.

Thus, after this step, we have removed a set Q of centers of S_0 and the total cost of the clusters of $S_0 - Q$ is $O(\varepsilon \cdot \mathrm{opt})$. Following this procedure, we only have to remove inexpensive clusters. While here it is not possible to identify the correct ones, making a mistake is not too costly. For any $c \in S_0 - Q$, we can now think of its whole cluster as being assigned entirely to a unique center in $S - S_0 \cup \widehat{\mathrm{OPT}}$ since by triangle inequality this would only increase its cost by $O(\varepsilon \cdot \mathrm{opt})$ in total. We can thus think of the remaining centers $S_0 - Q$ as either being assigned to $\widehat{\mathrm{OPT}}$ (we call this set $\mathcal{R}$), or to a center in $S - S_0$ (we call this set $\mathcal{U}$). Suppose we know how many centers fall into these two types, a figure we determine by trying all possibilities. The next step would be ideally to remove $\mathcal{U}$. We instead remove a *proxy* $\widetilde{\mathcal{U}}$ of $\mathcal{U}$, $|\widetilde{\mathcal{U}}| = |\mathcal{U}|$. A natural starting point to find a center for reassignment is to do so by greedily removing the center c for which the reassignment cost is minimized. However, we must be careful. First, the reassignment costs are dynamic, that is when removing a center, it affects reassignment costs of subsequently removed ones. Second, if a center c was used to bound the reassigment cost of an deleted center, we cannot delete it later in order to prevent the error from adding up. We design a procedure that produces a $\widetilde{\mathcal{U}}$ satisfying all these constraints. We remark that $\widetilde{\mathcal{U}} \cap \mathcal{R} = \emptyset$, hence removing $\mathcal{R}$ is still a valid option in the later stages.

Thus, for the purpose of analysis (the algorithm does not know $\mathcal{R}$), we obtain the solutions $M'_D = S - (Q \cup \widetilde{\mathcal{U}} \cup \mathcal{R}) \cup \Lambda$ and $M'_O = S - (Q \cup \widetilde{\mathcal{U}} \cup \mathcal{R}) \cup \widehat{\mathrm{OPT}}$. Moreover, the costs satisfy $\mathrm{cost}(M'_O) + \mathrm{cost}(M'_D) \le (4+\varepsilon)\mathrm{OPT}$, which is the same as summed up costs of M_O and M_D.

Improvement via Submodular Maximization. We can now use submodular maximization techniques (specifically algorithms to maximize a submodular function over a partition matroid constraint) in a similar but more general setting as in the work of [24]. In more detail, we wish to find the best $|\widehat{\mathrm{OPT}}|$ centers X to open (instead of the dummy centers Λ), one at the prescribed distance from each leader, to improve the cost of M'_D as much as possible. This is almost identical to the work of [24] except that we also need to close $|\mathcal{R}|$ other centers $\widetilde{\mathcal{R}}$ in $S - Q - \widetilde{\mathcal{U}}$. Ideally, we would like to open $\widehat{\mathrm{OPT}}$ and close $\mathcal{R}$. We show that selecting a good enough pair $(X, \widetilde{\mathcal{R}})$ can be reduced to a submodular maximization problem under a partition matroid constraint. In particular, we carefully define our submodular function so that its value, for a given subset X of centers, is calculated by optimally selecting the set $\widetilde{\mathcal{R}}$ to close (for the choice X). Then we can use state-of-the-art algorithms to optimize this and *simultaneously* approximate the centers $\widehat{\mathrm{OPT}}$ to open and the set $\mathcal{R}$ to close.

ACKNOWLEDGMENTS

Fabrizio Grandoni is supported in part by the SNF Grant 200021_200731 / 1. Euiwoong Lee is supported in part by NSF 2236669. Chris Schwiegelshohn is Supported in part by the Independent Research Fund Denmark (DFF) under grant No 1051-00106B. Ola Svensson is supported in part by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00054.

REFERENCES

[1] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. 2020. Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms. *SIAM J. Comput.* 49, 4 (2020). https://doi.org/10.1137/18M1171321

[2] Aaron Archer, Ranjithkumar Rajagopalan, and David B. Shmoys. 2003. Lagrangian Relaxation for the k-Median Problem: New Insights and Continuity Properties. In *Algorithms - ESA 2003, 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings (Lecture Notes in Computer Science, Vol. 2832)*, Giuseppe Di Battista and Uri Zwick (Eds.). Springer, 31–42. https://doi.org/10.1007/978-3-540-39658-1_6

[3] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. 1998. Approximation Schemes for Euclidean *k*-Medians and Related Problems. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998.* 106–113. https://doi.org/10.1145/276698.276718

[4] David Arthur and Sergei Vassilvitskii. 2007. k-means++: the advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007.* 1027–1035. http://dl.acm.org/citation.cfm?id=1283383.1283494

[5] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. 2004. Local Search Heuristics for k-Median and Facility Location Problems. *SIAM J. Comput.* 33, 3 (2004), 544–562. https://doi.org/10.1137/S0097539702416402

[6] Pranjal Awasthi, Avrim Blum, and Or Sheffet. 2010. Stability Yields a PTAS for k-Median and k-Means Clustering. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA.* 309–318. https://doi.org/10.1109/FOCS.2010.36

[7] Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. 2015. The Hardness of Approximation of Euclidean k-Means. In *31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, The Netherlands (LIPIcs, Vol. 34)*, Lars Arge and János Pach (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 754–767. https://doi.org/10.4230/LIPICS.SOCG.2015.754

[8] Yair Bartal. 1996. Probabilistic Approximations of Metric Spaces and Its Algorithmic Applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996.* IEEE Computer Society, 184–193. https://doi.org/10.1109/SFCS.1996.548477

[9] Yair Bartal. 1998. On Approximating Arbitrary Metrices by Tree Metrics. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, Jeffrey Scott Vitter (Ed.). ACM, 161–168. https://doi.org/10.1145/276698.276725

[10] Jaroslaw Byrka and Karen Aardal. 2010. An Optimal Bifactor Approximation Algorithm for the Metric Uncapacitated Facility Location Problem. *SIAM J. Comput.* 39, 6 (2010), 2212–2231. https://doi.org/10.1137/070708901

[11] Jaroslaw Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. 2017. An Improved Approximation for *k*-Median and Positive Correlation in Budgeted Optimization. *ACM Trans. Algorithms* 13, 2 (2017), 23:1–23:31. https://doi.org/10.1145/2981561

[12] Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto Guha. 1998. Rounding via Trees: Deterministic Approximation Algorithms for Group Steiner Trees and *k*-Median. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, Jeffrey Scott Vitter (Ed.). ACM, 114–123. https://doi.org/10.1145/276698.276719

[13] Moses Charikar and Sudipto Guha. 2005. Improved Combinatorial Algorithms for Facility Location Problems. *SIAM J. Comput.* 34, 4 (2005), 803–824. https://doi.org/10.1137/S0097539701398594

[14] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. 1999. A Constant-Factor Approximation Algorithm for the k-Median Problem (Extended Abstract). In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton (Eds.). ACM, 1–10. https://doi.org/10.1145/301250.301257

[15] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. 2002. A Constant-Factor Approximation Algorithm for the k-Median Problem. *J. Comput. Syst. Sci.* 65, 1 (2002), 129–149.

[16] Moses Charikar and Shi Li. 2012. 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 (Lecture Notes in Computer Science, Vol. 7391)*, Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer (Eds.). Springer, 194–205. https://doi.org/10.1007/978-3-642-31594-7_17

[17] Fabián A. Chudak and David B. Shmoys. 2003. Improved Approximation Algorithms for the Uncapacitated Facility Location Problem. *SIAM J. Comput.* 33, 1 (2003), 1–25. https://doi.org/10.1137/S0097539703405754

[18] Vincent Cohen-Addad, Karthik C. S., and Euiwoong Lee. 2021. On Approximability of Clustering Problems Without Candidate Centers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2635–2648.

[19] Vincent Cohen-Addad, Hossein Esfandiari, Vahab S. Mirrokni, and Shyam Narayanan. 2022. Improved approximations for Euclidean k-means and k-median, via nested quasi-independent sets. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, Stefano Leonardi and Anupam Gupta (Eds.). ACM, 1621–1628. https://doi.org/10.1145/3519935.3520011

[20] Vincent Cohen-Addad, Andreas Emil Feldmann, and David Saulpic. 2019. Near-Linear Time Approximations Schemes for Clustering in Doubling Metrics. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, David Zuckerman (Ed.). IEEE Computer Society, 540–559. https://doi.org/10.1109/FOCS.2019.00041

[21] Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, and Chris Schwiegelshohn. 2023. Breaching the 2 LMP Approximation Barrier for Facility Location with Applications to k-Median. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, Nikhil Bansal and Viswanath Nagarajan (Eds.). SIAM, 940–986. https://doi.org/10.1137/1.9781611977554.CH37

[22] Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn, and Ola Svensson. 2025. A $(2+\varepsilon)$-Approximation Algorithm for Metric k-Median. arXiv:2503.10972 [cs.DS] https://arxiv.org/abs/2503.10972

[23] Vincent Cohen-Addad, Anupam Gupta, Lunjia Hu, Hoon Oh, and David Saulpic. 2022. An Improved Local Search Algorithm for k-Median. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, Joseph (Seffi) Naor and Niv Buchbinder (Eds.). SIAM, 1556–1612. https://doi.org/10.1137/1.9781611977073.65

[24] Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. 2019. Tight FPT Approximations for k-Median and k-Means. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece (LIPIcs, Vol. 132)*, Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 42:1–42:14. https://doi.org/10.4230/LIPICS.ICALP.2019.42

[25] Vincent Cohen-Addad and Karthik C. S. 2019. Inapproximability of Clustering in Lp Metrics. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, David Zuckerman (Ed.). IEEE Computer Society, 519–539. https://doi.org/10.1109/FOCS.2019.00040

[26] Vincent Cohen-Addad and Chris Schwiegelshohn. 2017. On the Local Structure of Stable Clustering Instances. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, Chris Umans (Ed.). IEEE Computer Society, 49–60. https://doi.org/10.1109/FOCS.2017.14

[27] Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. 2019. Local Search Yields a PTAS for k-Means in Doubling Metrics. *SIAM J. Comput.* 48, 2 (2019), 452–480. https://doi.org/10.1137/17M1127181

[28] Teofilo F. Gonzalez. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theor. Comput. Sci.* 38 (1985), 293–306. https://doi.org/10.1016/0304-3975(85)90224-5

[29] Kishen N. Gowda, Thomas W. Pensyl, Aravind Srinivasan, and Khoa Trinh. 2023. Improved Bi-point Rounding Algorithms and a Golden Barrier for k-Median. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, Nikhil Bansal and Viswanath Nagarajan (Eds.). SIAM, 987–1011. https://doi.org/10.1137/1.9781611977554.CH38

[30] Fabrizio Grandoni, Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Rakesh Venkat. 2022. A refined approximation for Euclidean k-means. *Inf. Process. Lett.* 176 (2022), 106251. https://doi.org/10.1016/J.IPL.2022.106251

[31] Sudipto Guha and Samir Khuller. 1999. Greedy Strikes Back: Improved Facility Location Algorithms. *J. Algorithms* 31, 1 (1999), 228–248. https://doi.org/10.1006/jagm.1998.0993

[32] Anupam Gupta and Kanat Tangwongsan. 2008. Simpler Analyses of Local Search Algorithms for Facility Location. *CoRR* abs/0809.2554 (2008). arXiv:0809.2554 http://arxiv.org/abs/0809.2554

[33] Venkatesan Guruswami and Piotr Indyk. 2003. Embeddings and non-approximability of geometric problems. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003, Baltimore, Maryland, USA*. 537–538. http://dl.acm.org/citation.cfm?id=644108.644198

[34] Dorit S. Hochbaum and David B. Shmoys. 1986. A unified approach to approximation algorithms for bottleneck problems. *J. ACM* 33, 3 (1986), 533–550. https://doi.org/10.1145/5925.5933

[35] Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. 2003. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM* 50, 6 (2003), 795–824. https://doi.org/10.1145/950620.950621

[36] Kamal Jain, Mohammad Mahdian, and Amin Saberi. 2002. A new greedy approach for facility location problems. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*. 731–740. https://doi.org/10.1145/509907.510012

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

[38] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. 2004. A local search approximation algorithm for k-means clustering. *Comput. Geom.* 28, 2-3 (2004), 89–112. https://doi.org/10.1016/j.comgeo.2004.03.003

[39] Madhukar R. Korupolu, C. Greg Plaxton, and Rajmohan Rajaraman. 2000. Analysis of a Local Search Heuristic for Facility Location Problems. *J. Algorithms* 37, 1 (2000), 146–188. https://doi.org/10.1006/jagm.2000.1100

[40] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. 2010. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM* 57, 2 (2010), 5:1–5:32. https://doi.org/10.1145/1667053.1667054

[41] Euiwoong Lee, Melanie Schmidt, and John Wright. 2017. Improved and simplified inapproximability for k-means. *Inf. Process. Lett.* 120 (2017), 40–43. https://doi.org/10.1016/J.IPL.2016.11.009

[42] Shi Li. 2013. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Inf. Comput.* 222 (2013), 45–58. https://doi.org/10.1016/j.ic.2012.01.007

[43] Shi Li and Ola Svensson. 2013. Approximating k-median via pseudo-approximation. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. 901–910. https://doi.org/10.1145/2488608.2488723

[44] Shi Li and Ola Svensson. 2016. Approximating k-Median via Pseudo-Approximation. *SIAM J. Comput.* 45, 2 (2016), 530–547. https://doi.org/10.1137/130938645

[45] Mohammad Mahdian, Yinyu Ye, and Jiawei Zhang. 2006. Approximation Algorithms for Metric Facility Location Problems. *SIAM J. Comput.* 36, 2 (2006), 411–432. https://doi.org/10.1137/S0097539703435716

[46] Ramgopal R. Mettu and C. Greg Plaxton. 2003. The Online Median Problem. *SIAM J. Comput.* 32, 3 (2003), 816–832. https://doi.org/10.1137/S0097539701383443

[47] Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. 2012. The effectiveness of Lloyd-type methods for the k-means problem. *J. ACM* 59, 6 (2012), 28. https://doi.org/10.1145/2395116.2395117

[48] David B. Shmoys, Éva Tardos, and Karen Aardal. 1997. Approximation Algorithms for Facility Location Problems (Extended Abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, Frank Thomson Leighton and Peter W. Shor (Eds.). ACM, 265–274. https://doi.org/10.1145/258533.258600

Received 2024-11-04; accepted 2025-02-01