Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms

Sara Ahmadian
Dept. of Combinatorics and Optimization
University of Waterloo
Waterloo, Canada
Email: sahmadian@uwaterloo.ca

Ashkan Norouzi-Fard
School of Computer and Communications Science
EPFL
Lausanne, Switzerland
Email: ashkan.norouzifard@epfl.ch

Ola Svensson
School of Computer and Communications Science
EPFL
Lausanne, Switzerland
Email: ola.svensson@epfl.ch

Justin Ward
School of Computer and Communications Science
EPFL
Lausanne, Switzerland
Email: justin.ward@epfl.ch

***Abstract*—Clustering is a classic topic in optimization with k-means being one of the most fundamental such problems. In the absence of any restrictions on the input, the best known algorithm for k-means with a provable guarantee is a simple local search heuristic yielding an approximation guarantee of $9 + \epsilon$, a ratio that is known to be tight with respect to such methods.**

We overcome this barrier by presenting a new primal-dual approach that allows us to (1) exploit the geometric structure of k-means and (2) to satisfy the hard constraint that at most k clusters are selected without deteriorating the approximation guarantee. Our main result is a 6.357-approximation algorithm with respect to the standard LP relaxation. Our techniques are quite general and we also show improved guarantees for the general version of k-means where the underlying metric is not required to be Euclidean and for k-median in Euclidean metrics.

***Keywords*-clustering; primal-dual; k-median; k-means;**

I. Introduction

Clustering problems have been extensively studied in computer science. They play a central role in many areas, including data science and machine learning, and their study has led to the development and refinement of several key techniques in algorithms and theoretical computer science. Perhaps the most widely considered clustering problem is the *k-means* problem: given a set $\mathcal{D}$ of n points in $\mathbb{R}^\ell$ and an integer k, the task is to select a set S of k *cluster centers* in $\mathbb{R}^\ell$, so that $\sum_{j \in \mathcal{D}} c(j, S)$ is minimized, where $c(j, S)$ is the squared Euclidean distance between j and its nearest center in S.

A commonly used heuristic for k-means is Lloyd's algorithm [25], which is based on iterative improvement. However, despite its ubiquity in practice, Lloyd's algorithm has, in general, no worst-case guarantee and may not even converge in polynomial time [3], [29]. To overcome some of these limitations, Arthur and Vassilvitskii [4] proposed a randomized initialization procedure for Lloyd's algorithm, called k-means++, that leads to a $\Theta(\log k)$ expected approximation guarantee in the worst case. Under additional assumptions about the *clusterability* of the input dataset, Ostrovsky et al. [27] showed that this adaptation of Lloyd's algorithm gives a PTAS for k-means clustering. However, under no such assumptions, the best approximation algorithm in the general case has for some time remained a $(9 + \epsilon)$-approximation algorithm based on local search, due to Kanungo et al. [20]. Their analysis also shows that no natural local search algorithm performing a fixed number of swaps can improve upon this ratio. This leads to a barrier for these techniques that are rather far away from the best-known inapproximability result which only says that it is NP-hard to approximate this problem to within a factor better than 1.0013 [21].

While the general problem has resisted improvements, there has been significant progress on the k-means problem under a variety of assumptions. For example, Awasthi, Blum, and Sheffet obtain a PTAS in the special case when the instance has certain stability properties [6] (see also [8]), and there has been a long line of work (beginning with [26]) obtaining better and better PTASes under the assumption that k is constant. Most recently, it has been shown that local search gives a PTAS under the assumption that the dimension ℓ of the dataset is constant [13], [15]. These last results generalize to the case in which

the squared distances are from the shortest path metric on a graph with forbidden minors [13] or from a metric with constant doubling dimension [15]. We remark that the dimension ℓ of a k-means instance may always be assumed to be at most $O(\log n)$ by a standard application of the Johnson-Lindenstrauss transform. But, as the results in [13], [15] exhibit doubly-exponential dependence on the dimension, they do not give any non-trivial implications for the general case. Moreover, such a doubly-exponential dependence is essentially unavoidable, as the problem is APX-hard in the general case [7].

In summary, while k-means is perhaps the most widely used clustering problem in computer science, the only constant-factor approximation algorithm for the general case is based on simple local search heuristics that, for inherent reasons, give guarantees that are rather far from known hardness results. This is in contrast to many other well-studied clustering problems, such as facility location and k-median. Over the past several decades, a toolbox of core algorithmic techniques such as dual fitting, primal-dual and LP-rounding, has been refined and applied to these problems leading to improved approximation guarantees [28], [12], [9], [22], [24], [19], [18], [23], [17]. In particular, the current best approximation guarantees for both facility location (a 1.488-approximation due to Li [22]) and k-median (a 2.675-approximation due to Byrka et al. [10]) are LP-based and give significantly better results than previous local search algorithms [5], [11]. However, such LP-based techniques have not yet been able to attain similar improvements for k-means. One reason for this is that they have relied heavily on the triangle inequality, which does not hold in the case of k-means.

Our results.: In this work, we overcome this barrier by developing new techniques that allow us to exploit the standard LP formulation for k-means. We significantly narrow the gap between known upper and lower bounds by designing a new primal-dual algorithm for the k-means problem. We stress that our algorithm works in the general case that k and ℓ are part of the input, and requires no assumptions on the dataset.

Theorem I.1. *For any $\epsilon > 0$, there is a $(\rho_{mean} + \epsilon)$-approximation algorithm for the k-means problem, where $\rho_{mean} \approx 6.357$. Moreover, the integrality gap of the standard LP is at most ρ_{mean}.*

We now describe our approach and contributions at a high level. Given a k-means instance, we apply standard discretization techniques (e.g., [14]) to obtain an instance of the *discrete k-means* problem, in which we are given a discrete set $\mathcal{F}$ of candidate centers in $\mathbb{R}^\ell$ and must select k centers from $\mathcal{F}$, rather than k arbitrary points in $\mathbb{R}^\ell$. This step incurs an arbitrarily small loss in the approximation guarantee with respect to the original k-means instance. Because our algorithm always returns a set of centers from the discrete set $\mathcal{F}$, all of our results also hold for the *exemplar clustering* problem, in which centers must be chosen from the input points in $\mathcal{D}$. Specifically, we can simply take $\mathcal{F} = \mathcal{D}$.

Using Lagrangian relaxation, we can then consider the resulting discrete problem using the standard linear programming formulation for facility location. This general approach was pioneered in this context by Jain and Vazirani [19] who gave primal-dual algorithms for the k-median problem. In their paper, they first present a *Lagrangian Multiplier Preserving* (LMP) 3-approximation algorithm for the facility location problem. Then they run binary search over the opening cost of the facilities and use the aforementioned algorithm to get two solutions: one that opens more than k facilities and one that opens fewer than k, such that the opening cost of facilities in these solutions are close to each other. These solutions are then combined to obtain a solution that opens exactly k facilities. This step results in losing another factor 2 in the approximation guarantee, which results in a 6-approximation algorithm for k-median. The factor 6 was later improved by Jain, Mahdian, and Saberi [18] who obtained a 4-approximation algorithm for k-median by developing an LMP 2-approximation algorithm for facility location.

Technical contributions.: One can see that the same approach gives a much larger constant factor for the k-means problem since one cannot anymore rely on the triangle inequality. We use two main ideas to overcome this obstacle: (1) we exploit the geometric structure of k-means to obtain an improved LMP-approximation, and (2) we develop a new primal-dual algorithm that opens exactly k facilities while losing only an *arbitrarily small factor*.

For our first contribution, we modify the primal-dual algorithm of Jain and Vazirani [19] into a parameterized version which allows us to regulate the "aggressiveness" of the opening strategy of facilities. By using properties of Euclidean metrics we show that this leads to improved LMP approximation algorithms for k-means.

By the virtue of [2], these results already imply upper bounds on the integrality gaps of the standard LP relaxations, albeit with an *exponential time* rounding algorithm. Our second and more technical contribution is a new primal-dual algorithm that

accomplishes the same task in polynomial time. In other words, we are able to turn an LMP approximation algorithm into an algorithm that opens at most k facilities without deteriorating the approximation guarantee. We believe that this contribution is of independent interest. Indeed, all recent progress on the approximation of k-median beyond long-standing local search approaches [5] has involved reducing the factor 2 that is lost by Jain and Vazirani when two solutions are combined to open exactly k facilities (i.e. in the rounding of a so-called *bipoint* solution) [23], [10]. Here, we show that it is possible to reduce this loss all the way to $(1+\epsilon)$ by fundamentally changing the way in which dual solutions are constructed and maintained.

Instead of finding two solutions by binary search as in the framework of [19], we consider a sequence of solutions such that the opening costs and also the dual values of any two consecutive solutions are close in L^∞-norm. We show that this latter property allows us to combine two appropriate, consecutive solutions in the sequence into a single solution that opens exactly k facilities while losing only a factor of $1+\epsilon$ (rather than 2) in the approximation guarantee. Unfortunately, the dual solutions produced by the standard primal-dual algorithm approach are unstable, in the sense that a small change in opening price may result in drastic changes in the value of the dual variables. Thus, we introduce a new primal-dual procedure which instead iteratively transforms a dual solution for one price into a dual solution for another price. By carefully constraining the way in which the dual variables are altered, we show that we can obtain a sequence of "close" solutions that can be combined as desired.

We believe that this technique may be applicable in other settings, as well. An especially interesting open question is whether it is possible combine stronger LMP approximation algorithms, such as the one by Jain, Mahdian, Saberi [18], with our lossless rounding to obtain an improved $(2+\epsilon)$-approximation algorithm for k-median.

Extensions to other problems.: In addition to the standard k-means problem, we show that our results also extend to the following two problems. In the first extension, we consider the Euclidean k-median problem. Here we are given a set $\mathcal{D}$ of n points in $\mathbb{R}^\ell$ and a set $\mathcal{F}$ of m points in $\mathbb{R}^\ell$ corresponding to facilities. The task is to select a set S of at most k facilities from $\mathcal{F}$ so as to minimize $\sum_{j\in\mathcal{D}} c(j,S)$, where $c(j,S)$ is now the (non-squared) Euclidean distance from j to its nearest facility in S. For this problem, no approximation better than the general 2.675-approximation algorithm of Byrka et al. [10] for k-median was known.

Theorem I.2. *For any $\epsilon > 0$, there is a $(\rho_{med}+\epsilon)$-approximation algorithm for the Euclidean k-median problem, where $\rho_{med} \approx 2.633$. Moreover, the integrality gap of the standard LP is at most ρ_{med}.*

In the second extension, we consider a variant of the k-means problem in which each $c(j,S)$ corresponds to the squared distance in an arbitrary (possibly non-Euclidean) metric on $\mathcal{D}\cup\mathcal{F}$. For this problem, the best-known approximation algorithm is a 16-approximation due to Gupta and Tangwongsan [16]. In this paper, we obtain the following improvement:

Theorem I.3. *For any $\epsilon > 0$, there is a $(9+\epsilon)$-approximation algorithm for the k-means problem in general metrics. Moreover, the integrality gap of the standard LP is at most 9.*

We remark that the same hardness reduction as used for k-median [18] immediately yields a much stronger hardness result for the above generalization than what is known for the standard k-means problem: it is hard to approximate the k-means problem in general metrics within a factor $1+8/e-\epsilon \approx 3.94$ for any $\epsilon > 0$.

Outline of paper.: In Section II we review the standard LP formulation that we use, as well as its Lagrangian relaxation. We then in Section III show how to exploit the geometric structure of k-means to give improved LMP guarantees. In Section IV we show the main ideas behind our new rounding approach by giving an algorithm that runs in quasi-polynomial time. These results are then generalized to obtain an algorithm that runs in polynomial time in the full version of this paper[1]. Moreover, in the full version, we discuss the extension of our approach to the other objectives described above.

II. The standard LP relaxation and its Lagrangian relaxation

Here and in the remainder of the paper, we shall consider the *discrete* k-means problem, where we are given a discrete set $\mathcal{F}$ of facilities (corresponding to candidate centers).[1] Henceforth, we will simply refer to the discrete k-means problem as the k-means problem.

Given an instance $(\mathcal{D},\mathcal{F},d,k)$ of the k-means problem or the k-median problem, let $c(j,i)$ denote the connection cost of client j if connected to facility

[1] As discussed in the introduction, it is well-known that a ρ-approximation algorithm for this case can be turned into a $(\rho+\epsilon)$-approximation algorithm for the standard k-means problem, for any constant $\epsilon > 0$ (see e.g., [14]).

i. That is, $c(j,i) = d(j,i)$ in the case of k-median and $c(j,i) = d(j,i)^2$ in the case of k-means. Let $n = |\mathcal{D}|$ and $m = |\mathcal{F}|$.

The standard linear programming (LP) relaxation of these problems has two sets of variables: a variable y_i for each facility $i \in \mathcal{F}$ and a variable x_{ij} for each facility-client pair $i \in \mathcal{F}, j \in \mathcal{D}$. The intuition of these variables is that y_i should indicate whether facility i is opened and x_{ij} should indicate whether client j is connected to facility i. The standard LP relaxation can now be formulated as follows.

$$\begin{aligned}
\min \ & \textstyle\sum_{i\in\mathcal{F}, j\in\mathcal{D}} x_{ij} \cdot c(j,i) \\
\text{s.t.} \ & \textstyle\sum_{i\in\mathcal{F}} x_{ij} \geq 1 \qquad \forall j \in \mathcal{D} && \text{(II.1)} \\
& x_{ij} \leq y_i \qquad \forall j \in \mathcal{D}, i \in \mathcal{F} && \text{(II.2)} \\
& \textstyle\sum_{i\in\mathcal{F}} y_i \leq k && \text{(II.3)} \\
& x, y \geq 0\,. && \text{(II.4)}
\end{aligned}$$

The first set of constraints says that each client should be connected to at least one facility; the second set of constraints enforces that clients can only be connected to opened facilities; and the third constraint says that at most k facilities can be opened. We remark that this is a relaxation of the original problem as we have relaxed the constraint that x and y should take Boolean values to a non-negativity constraint. For future reference, we let OPT_k denote the value of an optimal solution to this relaxation.

A main difficulty for approximating the k-median and the k-means problems is the hard constraint that at most k facilities can be selected, i.e., constraint (II.3) in the above relaxation. A popular way of overcoming this difficulty, pioneered in this context by Jain and Vazirani [19], is to consider the Lagrangian relaxation where we multiply the constraint (II.3) times a Lagrange multiplier λ and move it to the objective. This results, for every $\lambda \geq 0$, in the following relaxation and its dual that we denote by $\mathsf{LP}(\lambda)$ and $\mathsf{DUAL}(\lambda)$, respectively.

$\mathsf{LP}(\lambda)$

$$\begin{aligned}
\min \ & \sum_{i\in\mathcal{F}, j\in\mathcal{D}} x_{ij} \cdot c(j,i) + \lambda \cdot \left(\sum_{i\in\mathcal{F}} y_i - k \right) \\
\text{s.t.} \ & \text{(II.1), (II.2), and (II.4).}
\end{aligned}$$

$\mathsf{DUAL}(\lambda)$

$$\begin{aligned}
\max \ & \sum_{j\in\mathcal{D}} \alpha_j - \lambda \cdot k \\
\text{s.t.} \ & \sum_{j\in\mathcal{D}} [\alpha_j - c(j,i)]^+ \leq \lambda \qquad \forall i \in \mathcal{F} && \text{(II.5)} \\
& \alpha \geq 0.
\end{aligned}$$

Here, we have simplified the dual by noticing that the dual variables $\{\beta_{ij}\}_{i\in\mathcal{F}, j\in\mathcal{D}}$ corresponding to the constraints (II.2) of the primal can always be set $\beta_{ij} = [\alpha_j - c(j,i)]^+$; the notation $[a]^+$ denotes $\max(a, 0)$. Moreover, to see that $\mathsf{LP}(\lambda)$ remains a relaxation, note that any feasible solution to the original LP is a feasible solution to the Lagrangian relaxation of no higher cost. In other words, for any $\lambda \geq 0$, the optimum value of $\mathsf{LP}(\lambda)$ is at most OPT_k.

If we disregard the constant term $\lambda \cdot k$ in the objective functions, $\mathsf{LP}(\lambda)$ and $\mathsf{DUAL}(\lambda)$ become the standard LP formulation and its dual for the facility location problem where the opening cost of each facility equals λ and the connection costs are defined by $c(\cdot,\cdot)$. Recall that the facility location problem (with uniform opening costs) is defined as the problem of selecting a set $S \subseteq \mathcal{F}$ of facilities to open so as to minimize the opening cost $|S|\lambda$ plus the connection cost $\sum_{j\in\mathcal{D}} c(j,S)$. Jain and Vazirani [19] introduced the following method for addressing the k-median problem motivated by simple economics. On the one hand, if λ is selected to be very small, i.e., it is cheap to open facilities, then a good algorithm for the facility location problem will open many facilities. On the other hand, if λ is selected to be very large, then a good algorithm for the facility location problem will open few facilities. Ideally, we want to use this intuition to find an opening price that leads to the opening of exactly k facilities and thus a solution to the original, constrained problem.

To make this intuition work, we need the notion of *Lagrangian Multiplier Preserving* (LMP) approximations: we say that a ρ-approximation algorithm is LMP for the facility location problem with opening costs λ if it returns a solution $S \subseteq \mathcal{F}$ satisfying

$$\sum_{j\in\mathcal{D}} c(j,S) \leq \rho(\mathrm{OPT}(\lambda) - |S|\lambda)\,,$$

where $\mathrm{OPT}(\lambda)$ denotes the value of an optimal solution to $\mathsf{LP}(\lambda)$ without the constant term $\lambda \cdot k$. The importance of this definition becomes apparent when either $\lambda = 0$ or $|S| \leq k$. In those cases, we can see that the value of the k-median or k-means solution is at most ρ times the optimal value of its relaxation $\mathsf{LP}(\lambda)$, and thus an ρ-approximation

with respect to its standard LP relaxation since $\mathrm{OPT}(\lambda) - k \cdot \lambda \leq \mathrm{OPT}_k$ for any $\lambda \geq 0$. For further explanations and applications of this technique, we refer the reader to the excellent text books [30] and [31].

III. An improved LMP approximation for k-means

In this section we show how to exploit the Euclidean structure of k-means to achieve better approximation guarantees. Our LMP approximation algorithm builds upon the primal-dual algorithm for the facility location problem by Jain and Vazirani [19]. We refer to their algorithm as the JV algorithm. The main modification to their algorithm is that we allow for a more "aggressive" opening strategy of facilities. The amount of aggressiveness is measured by the parameter δ: we devise an algorithm JV(δ) for each parameter $\delta \geq 0$, where a smaller δ results in a more aggressive opening strategy. We first describe JV(δ) and we then optimize δ for the considered objectives to obtain the claimed approximation guarantees.

We remark that the result in [2] (non-constructively) upper bounds the integrality gap of the standard LP relaxation of k-median in terms of the LMP approximation guarantee of JV. This readily generalizes to the k-means problem and JV(δ). Consequently, our guarantees presented here for k-means (and in Section **??** for the other objectives) upper bound the integrality gaps as the theorems state in the introduction.

A. Description of JV(δ)

As alluded to above, the algorithm is a modification of JV, and Remark III.2 below highlights the difference. The algorithm consists of two phases: the dual-growth phase and the pruning phase.

Dual-growth phase: In this stage, we construct a feasible dual solution α to DUAL(λ). Initially, we set $\alpha = \mathbf{0}$ and let $A = \mathcal{D}$ denote the set of active clients (which is all clients at first). We then repeat the following until there are no active clients, i.e., $A = \emptyset$: increase the dual-variables $\{\alpha_j\}_{j \in A}$ corresponding to the active clients at a uniform rate until one of the following events occur (if several events happen at the same time, break ties arbitrarily):

> Event 1: A dual constraint $\sum_{j \in \mathcal{D}} [\alpha_j - c(j,i)]^+ \leq \lambda$ becomes tight for a facility $i \in \mathcal{F}$. In this case we say that facility i is *tight* or *temporarily opened*. We update A by removing the active clients with a *tight edge* to i, that is, a client $j \in A$ is removed if $\alpha_j - c(j,i) \geq 0$. For future reference, we say that facility i is the *witness* of these removed clients.
>
> Event 2: An active client $j \in A$ gets a tight edge, i.e., $\alpha_j - c(j,i) = 0$, to some already tight facility i. In this case, we remove j from A and let i be its witness.

This completes the description of the dual-growth phase. Before proceeding to the pruning phase, let us remark that the constructed α is indeed a feasible solution to DUAL(λ) by design. It is clear that α is non-negative. Now consider a facility $i \in \mathcal{F}$ and its corresponding dual constraint $\sum_{j \in \mathcal{D}} [\alpha_j - c(j,i)]^+ \leq \lambda$. On the one hand, the constraint is clearly satisfied if it never becomes tight during the dual-growth phase. On other hand, if it becomes tight, then all clients with a tight edge to it are removed from the active set of clients by Event 1. Moreover, if any client gets a tight edge to i in subsequent iterations it gets immediately removed from the set of active clients by Event 2. Therefore the left-hand-side of the constraint will never increase (nor decrease) after it becomes tight so the constraint remains satisfied. Having proved that α is a feasible solution to DUAL(λ), let us now describe the pruning phase.

Pruning phase: After the dual-growth phase (too) many facilities are temporarily opened. The pruning phase will select a subset of these facilities to open. In order to formally describe this process, we need the following notation. For a client j, let $N(j) = \{i \in \mathcal{F} : \alpha_j - c(j,i) > 0\}$ denote the facilities to which client j contributes to the opening cost. Similarly, for $i \in \mathcal{F}$, let $N(i) = \{j \in \mathcal{D} : \alpha_j - c(j,i) > 0\}$ denote the clients with a positive contribution toward i's opening cost. For a temporarily opened facility i, let

$$t_i = \max_{j \in N(i)} \alpha_j \,,$$

and by convention let $t_i = 0$ if $N(i) = \emptyset$ (this convention will be useful in future sections and will only be used when the opening cost λ of facilities are set to 0). Note that, if $N(i) \neq \emptyset$, then t_i equals the "time" that facility i was temporarily opened in the dual-growth phase. A crucial property of t_i that follows from the construction of α is the following.

Claim III.1. *For a client j and its witness i, $\alpha_j \geq t_i$. Moreover, for any $j' \in N(i)$ we have $t_i \geq \alpha_{j'}$.*

For the pruning phase, it will be convenient to define the *client-facility graph* G and the *conflict graph* H. The vertex set of G consist of all the clients and all the facilities i such that $\sum_{j \in \mathcal{D}} [\alpha_j - c(j,i)]^+ = \lambda$ (i.e., the tight or temporarily open facilities). There is an edge between facility i and client j if $i \in N(j)$.

The conflict graph H is defined based on the client-facility graph G and t as follows:

- The vertex set consists of all facilities in G.
- There is an edge between two facilities i and i' if some client j is adjacent to both of them in G and $c(i,i') \leq \delta \min(t_i, t_{i'})$.

The pruning phase now finds a (inclusion-wise) maximal independent set IS of H and opens those facilities; clients are connected to the closest facility in IS.

Remark III.2. *The difference between the original algorithm JV and our modified JV(δ) is the additional condition $c(i,i') \leq \delta \min(t_i, t_{i'})$ in the definition of the conflict graph. Notice that if we select a smaller δ we will have fewer edges in H. Therefore a maximal independent set will likely grow in size, which results in a more "aggressive" opening strategy. Adjusting δ will allow us to achieve better* LMP *approximation guarantees.*

B. Analysis of JV(δ)

We start with some intuition that illustrates our approach. From the standard analysis of JV (and our analysis of k-means in general metrics presented in Section **??**), it is clear that the bottleneck for the approximation guarantee comes from the connection-cost analysis of clients that need to do a "3-hop" as illustrated in the left part of Figure 1: client j is connected to open facility i_2 and the squared-distance is bounded by the path $j - i_1 - j_1 - i_2$. Moreover, this analysis is tight when considering $\mathsf{JV} = \mathsf{JV}(\infty)$. Our strategy will now be as follows: Select δ to be a constant smaller than 4. This means that in the configurations of Figure 1, we will also open i_2 if the distance between i_1 and i_2 is close to 2. Therefore, if we do not open i_2, the distance between i_1 and i_2 is less than 2 (as in the right part of Figure 1) which allows us to get an approximation guarantee better than 9. However, this might result in a client contributing to the opening cost of many facilities in IS. Nonetheless, by using the properties of Euclidean metrics, we show that even in this case, we are able to achieve an LMP approximation guarantee with ratio better than 9.

Specifically, define δ_{mean} to be the constant larger than 2 that minimizes

$$\rho_{\mathsf{mean}}(\delta) = \max\left\{(1+\sqrt{\delta})^2, \frac{1}{\delta/2-1}\right\},$$

which will be our approximation guarantee. It can be verified that $\delta_{\mathsf{mean}} \approx 2.3146$ and $\rho_{\mathsf{mean}} \approx 6.3574$. Let also $c(j,i) = d(j,i)^2$ where d is the underlying Euclidean metric. The proof uses the following basic facts about squared-distances in Euclidean metrics: given $x_1, x_2, \ldots, x_s \in \mathbb{R}^\ell$, we have that $\min_{y\in\mathbb{R}^\ell} \sum_{i=1}^s \|x_i - y\|_2^2$ is attained by the *centroid* $\mu = \frac{1}{s}\sum_{i=1}^s x_i$ and in addition we have the identity $\sum_{i=1}^s \|x_i - \mu\|_2^2 = \frac{1}{2s}\sum_{i=1}^s \sum_{j=1}^s \|x_i - x_j\|_2^2$.

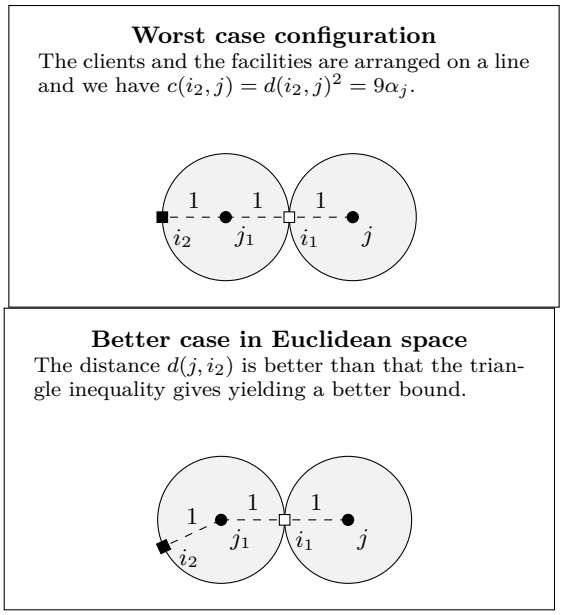

Figure 1. The intuition how we improve the guarantee in the Euclidean case. In both cases, we have $\alpha_j = \alpha_{j_1} = 1$. Moreover, $i_1 \notin \mathsf{IS}, i_2 \in \mathsf{IS}$ and we are interested in bounding $c(j,i_2)$ as a function of α_j.

Theorem III.3. *Let d be a Euclidean metric on $\mathcal{D} \cup \mathcal{F}$ and suppose that $c(j,i) = d(j,i)^2$ for every $i \in \mathcal{F}$ and $j \in \mathcal{D}$. Then, for any $\lambda \geq 0$, Algorithm JV(δ_{mean}) constructs a solution α to DUAL(λ) and returns a set IS of opened facilities such that*

$$\sum_{j\in\mathcal{D}} c(j,\mathsf{IS}) \leq \rho_{\mathsf{mean}} \cdot \Big(\sum_{j\in\mathcal{D}} \alpha_j - \lambda|\mathsf{IS}|\Big).$$

Proof: To simplify notation, we use δ instead of δ_{mean} throughout the proof. Consider any client $j \in \mathcal{D}$. We shall prove that

$$\begin{aligned}\frac{c(j,\mathsf{IS})}{\rho_{\mathsf{mean}}} &\leq \alpha_j - \sum_{i\in N(j)\cap\mathsf{IS}} (\alpha_j - c(j,i)) \\ &= \alpha_j - \sum_{i\in\mathsf{IS}} [\alpha_j - c(j,i)]^+ .\end{aligned}$$

The statement then follows by summing up over all clients and noting that any facility $i \in \mathsf{IS}$ was temporarily opened and thus we have $\sum_{j\in\mathcal{D}}[\alpha_j - c(i,j)]^+ = \lambda$. A difference compared to the standard analysis of JV is that in our algorithm we may open several facilities in $N(j)$, i.e., client j may contribute to the opening of several facilities. We divide

our analysis into the three cases $|N(j) \cap \mathsf{IS}| = 1$, $|N(j) \cap \mathsf{IS}| > 1$, and $|N(j) \cap \mathsf{IS}| = 0$. For brevity, let S denote $N(j) \cap \mathsf{IS}$ and $s = |S|$.

Case $s = 1$: If we let i^* be the unique facility in S,

$$\begin{aligned}\frac{c(j,\mathsf{IS})}{\rho_{\mathsf{mean}}} &\leq c(j,\mathsf{IS}) \leq c(j,i^*)\\ &= \alpha_j - (\alpha_j - c(j,i^*))\\ &= \alpha_j - \sum_{i \in N(j)\cap \mathsf{IS}} (\alpha_j - c(j,i))\,.\end{aligned}$$

Case $s > 1$: In this case, there are multiple facilities in IS that j is contributing to. We need to show that $\alpha_j - \sum_{i\in S}(\alpha_j - c(j,i)) \geq \frac{1}{\rho_{\mathsf{mean}}} c(j,\mathsf{IS})$.

The sum $\sum_{i\in S} c(j,i)$ is the sum of square distances from j to facilities in S which is at least the sum of square distances of these facilities from their centroid μ, i.e., $\sum_{i\in S} c(j,i) \geq \sum_{i\in S} c(i,\mu)$. Moreover, by the identity, $\sum_{i\in S} c(i,\mu) = \frac{1}{2s}\sum_{i\in S}\sum_{i'\in S} c(i,i')$, we get

$$\sum_{i\in S} c(j,i) \geq \frac{1}{2s}\sum_{i\in S}\sum_{i'\in S} c(i,i')\,.$$

As there is no edge between any pair of distinct facilities i and i' in $S \subseteq \mathsf{IS}$, we must have

$$c(i,i') > \delta \cdot \min(t_i, t_{i'}) \geq \delta \cdot \alpha_j,$$

where the last inequality follows because j is contributing to both i and i' and hence $\min(t_i, t_{i'}) \geq \alpha_j$. By the above,

$$\begin{aligned}\sum_{i\in S} c(j,i) &\geq \frac{\sum_{i\in S}\sum_{i'\in S} c(i,i')}{2s}\\ \geq \frac{\sum_{i\in S}\sum_{i'\neq i\in S} \delta\cdot\alpha_j}{2s} &= \delta\cdot\frac{s-1}{2}\cdot\alpha_j\,.\end{aligned}$$

Hence,

$$\begin{aligned}\sum_{i\in S}(\alpha_j - c(j,i)) &\leq \left(s - \delta\cdot\frac{s-1}{2}\right)\alpha_j\\ &= \left(s\big(1-\tfrac{\delta}{2}\big) + \tfrac{\delta}{2}\right)\alpha_j\,.\end{aligned}$$

Now, since $\delta \geq 2$ the above upper bound is a non-increasing function of s. Therefore, since $s \geq 2$ we always have

$$\sum_{i\in S}(\alpha_j - c(j,i)) \leq \big(2 - \tfrac{\delta}{2}\big)\alpha_j\,. \tag{III.1}$$

We also know that $\alpha_j > c(j,i)$ for any $i \in S$. Therefore, $\alpha_j > c(j,\mathsf{IS})$ and, since $\delta \geq 2$:

$$\big(\tfrac{\delta}{2} - 1\big)c(j,\mathsf{IS}) \leq \big(\tfrac{\delta}{2} - 1\big)\alpha_j\,. \tag{III.2}$$

Combining Inequalities (III.1) and (III.2),

$$\begin{aligned}\sum_{i\in S}(\alpha_j - c(j,i)) + \big(\tfrac{\delta}{2} - 1\big)c(j,\mathsf{IS}) \leq\\ \big(2 - \tfrac{\delta}{2}\big)\alpha_j + \big(\tfrac{\delta}{2} - 1\big)\alpha_j = \alpha_j\,.\end{aligned}$$

We conclude the analysis of this case by rearranging the above inequality and recalling that $\rho_{\mathsf{mean}} \geq \frac{1}{\delta/2-1}$.

Case $s = 0$: Here, we claim that there exists a tight facility i such that

$$d(j,i) + \sqrt{\delta t_i} \leq (1+\sqrt{\delta})\sqrt{\alpha_j}\,. \tag{III.3}$$

To see that such a facility i exists, consider the witness $w(j)$ of j. By Claim III.1, we have $\alpha_j \geq t_{w(j)}$ and since j has a tight edge to its witness $w(j)$, $\alpha_j \geq c(j,w(j) = d(j,w(j))^2$; or, equivalently, $\sqrt{\alpha_j} \geq \sqrt{t_{w(j)}}$ and $\sqrt{\alpha_j} \geq d(j,w(j))$ which implies that there is a tight facility, namely $w(j)$, satisfying (III.3).

Since IS is a maximal independent set of H, either $i \in \mathsf{IS}$, in which case $d(j,\mathsf{IS}) \leq d(j,i)$, or there is an $i' \in \mathsf{IS}$ such that the edge (i',i) is in H, in which case

$$d(j,\mathsf{IS}) \leq d(j,i) + d(i,i') \leq d(j,i) + \sqrt{\delta t_i}\,,$$

where the second inequality follows from $d(i,i')^2 = c(i,i') \leq \delta\min(t_i, t_{i'})$ by the definition of H. In any case, we have by (III.3)

$$d(j,\mathsf{IS}) \leq (1+\sqrt{\delta})\sqrt{\alpha_j}\,.$$

Squaring both sides and recalling that $\rho_{\mathsf{mean}} \geq (1+\sqrt{\delta})^2$ completes the last case and the proof of the theorem. ∎

IV. Quasi-polynomial time algorithm

In this section, we present a quasi-polynomial time approach that turns the LMP approximation algorithm presented in the previous section into an approximation algorithm for k-means, i.e., into an algorithm finding a solution that satisfies the strict constraint that at most k facilities are opened. This is achieved by only deteriorating the approximation guarantee by an arbitrarily small factor regulated by ϵ. We also introduce several of the ideas used in the polynomial time approach. Although the results obtained in this section are weaker (quasi-polynomial instead of polynomial), we believe that the easier quasi-polynomial algorithm serves as a good starting point before reading the more complex polynomial time algorithm. Let $\rho = \rho_{\mathsf{mean}}$ denote the approximation guarantee and $\delta = \delta_{\mathsf{mean}}$ denote the parameter to our algorithm. Throughout this section we fix $\epsilon > 0$ to be a small constant, and we assume for notational convenience and without

loss of generality that $n \gg 1/\epsilon$. We shall also assume that the distances satisfy the following:

Lemma IV.1. *By losing a factor $(1 + 100/n^2)$ in the approximation guarantee, we can assume that the squared-distance between any client j and any facility i satisfies: $1 \le d(i,j)^2 \le n^6$, where $n = |\mathcal{D}|$.*

The proof follows by standard discretization techniques and is presented in the full version of this paper[1].

Our algorithm will produce a $(\rho + O(\epsilon))$-approximate solution. In the algorithm, we consider separately the two phases of the primal-dual algorithm from Section III-B. Suppose that the first phase produces a set of values $\alpha = \{\alpha_j\}_{j \in \mathcal{D}}$ satisfying the following definition:

Definition IV.2. *A feasible solution α of $\mathsf{DUAL}(\lambda)$ is* good *if for every $j \in \mathcal{D}$ there exists a tight facility i such that $(1 + \sqrt{\delta} + \epsilon)\sqrt{\alpha_j} \ge d(j,i) + \sqrt{\delta t_i}$.*

Recall that for a dual solution α, t_i is defined to be the largest α-value out of all clients that are contributing to a facility i: $t_i = \max_{j \in N(i)} \alpha_j$ where $N(i) = \{j \in \mathcal{D} : \alpha_j - d(i,j)^2 > 0\}$.

As the condition of Definition IV.2 relaxes (III.3) by a tiny amount (regulated by ϵ), our analysis in Section III shows that as long as the first stage of the primal-dual algorithm produces an α that is good, the second stage will find a set of facilities IS such that $\sum_{j \in \mathcal{D}} d(j,\mathsf{IS})^2 = \sum_{j \in \mathcal{D}} c(j,\mathsf{IS}) \le (\rho + O(\epsilon))\big(\sum_{j \in \mathcal{D}} \alpha_j - \lambda|\mathsf{IS}|\big)$. If we could somehow find a value λ such that the second stage opened *exactly* k facilities, then we would obtain a $(\rho + O(\epsilon))$-approximation algorithm. In order to accomplish this, we first enumerate all potential values $\lambda = 0, 1 \cdot \epsilon_z, 2 \cdot \epsilon_z, \ldots, L \cdot \epsilon_z$, where ϵ_z is a small step size and L is large enough to guarantee that we eventually find a solution of size at most k (for a precise definition of L and ϵ_z, see (IV.1) and (IV.2)). Specifically, in Section IV-A, we give an algorithm that in time $n^{O(\epsilon^{-1} \log n)}$ generates a quasi-polynomial-length sequence of solutions $\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(L)}$, where $\alpha^{(\ell)}$ is a good solution to $\mathsf{DUAL}(\ell \cdot \epsilon_z)$. We shall ensure that each consecutive set of values $\alpha^{(\ell)}, \alpha^{(\ell+1)}$ are *close* in the following sense:

Definition IV.3. *Two solutions α and α' are* close *if $|\alpha'_j - \alpha_j| \le \frac{1}{n^2}$ for all $j \in \mathcal{D}$.*

Unfortunately, it may be the case that for a good solution $\alpha^{(\ell)}$ to $\mathsf{DUAL}(\lambda)$, the second stage of our algorithm opens more than k facilities, while for the next good solution $\alpha^{(\ell+1)}$ to $\mathsf{DUAL}(\lambda + \epsilon_z)$, it opens fewer than k facilities. In order to obtain a solution that opens *exactly* k facilities, we must somehow interpolate between consecutive solutions in our sequence. In Section IV-B we describe an algorithm that accomplishes this task. Specifically, for each pair of consecutive solutions $\alpha^{(\ell)}, \alpha^{(\ell+1)}$ we show that, since their α-values are nearly the same, we can control the way in which a maximal independent set in the associated conflict graphs changes. Formally, we show how to maintain a sequence of approximate integral solutions with cost bounded by $\alpha^{(\ell)}$ and $\alpha^{(\ell+1)}$, in which the number of open facilities decreases by at most one in each step. This ensures that some solution indeed opens exactly k facilities and it will be found in time $n^{O(\epsilon^{-1} \log n)}$.

A. Generating a sequence of close, good solutions

We first describe our procedure for generating a close sequence of good solutions. Select the following parameters

$$\epsilon_z = n^{-3-10\log_{1+\epsilon} n}, \tag{IV.1}$$

$$L = 4n^7 \cdot \epsilon_z^{-1} = n^{O(\epsilon^{-1} \log n)}. \tag{IV.2}$$

We also use the notion of *buckets* that partition the real line:

Definition IV.4. *For any value $v \in \mathbb{R}$, let $B(v) = 0$, if $v < 1$, and $B(v) = 1 + \lfloor \log_{1+\epsilon}(v) \rfloor$ if $v \ge 1$. We say that $B(v)$ is the index of the* bucket containing *v.*

The buckets will be used to partition the α-values of the clients. As, in every constructed solution α, each client will have a tight edge to a facility, Lemma IV.1 implies that α_j will always be at least 1. Therefore, the definition gives the property that the α-values of any two clients j and j' in the same bucket differ by at most a factor of $1 + \epsilon$. In other words, the buckets will be used to classify the clients according to similar α-values.

We now describe a procedure QuasiSweep that takes as input a good dual solution α^{in} of $\mathsf{DUAL}(\lambda)$ and outputs a good dual solution α^{out} of $\mathsf{DUAL}(\lambda + \epsilon_z)$ such that α^{in} and α^{out} are close. In order to generate the desired close sequence of solutions, we first define an initial solution for $\mathsf{DUAL}(0)$ by $\alpha_j = \min_{i \in \mathcal{F}} d(i,j)^2$ for $j \in \mathcal{D}$. Then, for $0 \le \ell < L$, we call QuasiSweep with $\alpha^{\mathsf{in}} = \alpha^{(\ell)}$ to generate the next solution $\alpha^{(\ell+1)}$ in our sequence. We shall show that each $\alpha^{(\ell)}$ is a feasible dual solution of $\mathsf{DUAL}(\ell \cdot \epsilon_z)$, and that the following invariant holds throughout the generation of our sequence:

Invariant 1. *In every solution $\alpha = \alpha^{(\ell)}$, $(0 \le \ell \le L)$, every client $j \in \mathcal{D}$ has a tight edge to a tight facility $w(j) \in \mathcal{F}$ (its witness) such that $B(t_{w(j)}) \le B(\alpha_j)$.*

Note that this implies that each solution in our sequence is good. Indeed, consider a dual solution α that satisfies Invariant 1. Then, for any client j, we have some i $(= w(j))$ such that $\sqrt{\alpha_j} \geq d(i,j)$ (since j has a tight edge to $w(j)$) and $\sqrt{(1+\epsilon)\delta\alpha_j} \geq \sqrt{\delta t_i}$ where we used that $B(\alpha_j) \geq B(t_i)$ implies $(1+\epsilon)\alpha_j \geq t_i$. Hence, $(1+\sqrt{\delta}+\epsilon)\sqrt{\alpha_j} \geq \left(1+\sqrt{(1+\epsilon)\delta}\right)\sqrt{\alpha_j} \geq d(i,j)+\sqrt{\delta t_i}$, and so α is good (here, for the first inequality we have used that $\sqrt{1+\epsilon} \leq 1+\epsilon/2$ and $\sqrt{\delta} \leq 2$). We observe that our initial solution $\alpha^{(0)}$ has $t_i = 0$ for all $i \in \mathcal{F}$, and so Invariant 1 holds trivially. In our following analysis, we will show that each call to Sweep preserves Invariant 1.

1) Description of QuasiSweep*:* We now formally describe the procedure QuasiSweep that, given the last previously generated solution α^{in} in our sequences produces the solution α^{out} returned next.

We initialize the algorithm by setting $\alpha_j = \alpha_j^{\text{in}}$ for each $j \in \mathcal{D}$ and by increasing the opening prices of each facility from λ to $\lambda + \epsilon_z$. At this point, no facility is tight and therefore the solution α is not a good solution of $\mathsf{DUAL}(\lambda+\epsilon_z)$. We now describe how to modify α to obtain a solution α^{out} satisfying Invariant 1 (and hence into a good solution). The algorithm will maintain a current set A of active clients and a current threshold θ. Initially, $A = \emptyset$, and $\theta = 0$. We slowly increase θ and whenever $\theta = \alpha_j$ for some client j, we add j to A. While $j \in A$, we increase α_j at the same rate as θ. We remove a client j from A, whenever the following occurs:

> j has a tight edge to some tight facility i with $B(\alpha_j) \geq B(t_i)$. In this case, we say that i is the *witness* of j.

Note that if a client j satisfies this condition when it is added to A, then we remove j from A immediately after it is added. In this case, α_j is not increased.

Increasing the α-values for clients in A, may cause the contributions to some facility i to exceed the opening cost $\lambda+\epsilon_z$. To prevent this from happening, we also decrease every value α_j with $B(\alpha_j) > B(\theta)$ at a rate of $|A|$ times the rate that θ is increasing. Observe that while there exists any such $j \in N(i)$, the total contribution of the clients toward opening this i cannot increase, and so i cannot become tight. It follows that once any facility i becomes tight, $B(\alpha_j) \leq B(\theta)$ for every $j \in N(i)$ and so i is presently a witness for all clients $j \in N(i) \cap A$. At this moment all such clients in $N(i) \cap A$ will be removed from A and their α-values will not subsequently be changed. Thus, i remains tight until the end of QuasiSweep. Moreover, observe any other client j' that is added to A later will immediately be removed from A as soon as it has a tight edge to i. Thus, neither t_i nor the total contribution to i change throughout the remainder of QuasiSweep. In particular, i remains a witness for all such clients j for the remainder of QuasiSweep.

We stop increasing θ once every client j has been added and removed from A. The procedure QuasiSweep then terminates and outputs $\alpha^{\text{out}} = \alpha$. As we have just argued, the contributions to any tight facility can never increase, and every client that is removed from j will have a witness through the rest of QuasiSweep (in particular, in α^{out}). Thus, α^{out} is a feasible solution of $\mathsf{DUAL}(\lambda+\epsilon_z)$ in which every client j has a witness $w(j)$, i.e., j has a tight edge to the tight facility $w(j)$ and $B(t_{w(j)}) \leq B(\alpha_j)$. Hence, the output of Sweep always satisfies Invariant 1.

This completes the description of QuasiSweep. We now show that the produced sequence of solutions is close and to analyze the running time.

2) Closeness and running time: We begin by showing that QuasiSweep produces a close sequence of solutions.

Lemma IV.5. *For each client $j \in \mathcal{D}$, we have $|\alpha_j^{\text{in}} - \alpha_j^{\text{out}}| \leq 1/n^2$.*

The proof of this lemma is available in the full version of this paper [1].

For the sake of clarity, we have presented the QuasiSweep procedure in a continuous fashion. We show in the full version of this paper [1] how to implement QuasiSweep as a discrete algorithm running in polynomial time. We conclude the analysis of this section by noting that, as Sweep is repeated $L = n^{O(\epsilon^{-1}\log n)}$ times, the total running time for producing the sequence $\alpha^{(0)}, \alpha^{(1)}, \ldots, \alpha^{(L)}$ is $n^{O(\epsilon^{-1}\log n)}$.

B. Finding a solution of size k

In this section we describe our algorithm for finding a solution of k facilities given a close sequence $\alpha^{(0)}, \alpha^{(1)} \ldots, \alpha^{(L)}$, where $\alpha^{(\ell)}$ is a good solution to $\mathsf{DUAL}(\epsilon_z \cdot \ell)$.

We associate with each dual solution $\alpha^{(\ell)}$ a client-facility graph and a conflict graph that are defined in exactly the same way as in Section III-A: that is, the graph $G^{(\ell)}$ is a bipartite graph with all of $\mathcal{D}$ on one side and every tight facility in $\alpha^{(\ell)}$ on the other and $G^{(\ell)}$ contains the edge (j,i) if and only if $\alpha_j^{(\ell)} > c(j,i)$. Given each $G^{(\ell)}$, recall that $H^{(\ell)}$ is then a graph consisting of the facilities present in $G^{(\ell)}$, which contains an edge (i,i') if i and i' are both adjacent to some client j in $G^{(\ell)}$ and $c(i,i') \leq \delta \min(t_i^{(\ell)}, t_{i'}^{(\ell)})$, where for each i, we have $t_i^{(\ell)} = \max\{\alpha_j^{(\ell)} : \alpha_j^{(\ell)} > c(j,i)\}$ (and again we adopt

the convention that $t_i^{(\ell)} = 0$ if $\alpha_j^{(\ell)} \leq c(j,i)$ for all $j \in \mathcal{D}$). Thus, we have a sequence $G^{(0)}, \ldots, G^{(L)}$ of client-facility graphs and a sequence $H^{(0)}, \ldots, H^{(L)}$ of conflict graphs obtained from our sequence of dual solutions. The main goal of this section is to give a corresponding sequence of maximal independent sets of the conflict graphs so that the size of the solution (independent set) never decreases by more than 1 in this sequence. Unfortunately, this is not quite possible. Instead, starting with a maximal independent set $\mathsf{IS}^{(\ell)}$ of $H^{(\ell)}$, we shall slowly transform it into a maximal independent set $\mathsf{IS}^{(\ell+1)}$ of $H^{(\ell+1)}$ by considering maximal independent sets in a sequence of polynomially many intermediate conflict graphs $H^{(\ell)} = H^{(\ell,0)}, H^{(\ell,1)}, \ldots, H^{(\ell,p_\ell)} = H^{(\ell+1)}$. We shall refer to these independent sets as $\mathsf{IS}^{(\ell)} = \mathsf{IS}^{(\ell,0)}, \mathsf{IS}^{(\ell,1)}, \ldots, \mathsf{IS}^{(\ell,p_\ell)} = \mathsf{IS}^{(\ell+1)}$. This interpolation will allow us to ensure that the size of our independent set decreases by at most 1 throughout this sequence. It follows that at some point we find a solution IS of size exactly k: on the one hand, since $H^{(0)}$ contains all facilities and no edges we have $\mathsf{IS}^{(0)} = \mathcal{F}$, which by assumption is strictly greater than k. On the other hand, we must have $|\mathsf{IS}^{(L)}| \leq 1$. Indeed, as $\alpha^{(L)}$ is a good dual solution of $\mathsf{DUAL}(L\epsilon_z) = \mathsf{DUAL}(4n^7)$, we claim $H^{(L)}$ is a clique. First, note that any tight facility i in $\alpha^{(L)}$ has $t_i \geq \frac{L\epsilon_z}{n} = 4n^6$ which means that all clients have a tight edge to i when i becomes tight (since the maximum squared facility-client distance is n^6 by Lemma IV.1). Second, any two facilities i, i' have $d(i,i')^2 \leq 4n^6$ using the triangle inequality and facility-client distance bound. Combining these two insights, we can see that $H^{(L)}$ is a clique and so $|\mathsf{IS}^{(L)}| \leq 1$.

It remains to describe and analyze the procedure QuasiGraphUpdate that will perform the interpolation between two conflict graphs $H^{(\ell)}$ and $H^{(\ell+1)}$ when given a maximal independent set $\mathsf{IS}^{(\ell)}$ of $H^{(\ell)}$ so that $|\mathsf{IS}^{(\ell)}| > k$. We run this procedure at most L times starting with $H^{(0)}, H^{(1)}$, and $\mathsf{IS}^{(0)} = \mathcal{F}$ until we find a solution of size k.

1) Description of QuasiGraphUpdate*:* Denote the input by $H^{(\ell)}, H^{(\ell+1)}$, and $\mathsf{IS}^{(\ell)}$ (the maximal independent set of $H^{(\ell)}$ of size greater than k). Although we are interested in producing a sequence of conflict graphs, it will be helpful to think of a process that alters some "hybrid" client-facility graph G, then uses G and the corresponding opening times t to construct a new conflict graph H after each alteration. To ease the description of this process, we duplicate each facility that appears both in $G^{(\ell)}$ and $G^{(\ell+1)}$ so as to ensure that these sets are disjoint. Let $\mathcal{V}^{(\ell)}$ and $\mathcal{V}^{(\ell+1)}$ denote the (now disjoint) sets of facilities in $G^{(\ell)}$ and $G^{(\ell+1)}$, respectively. Note that the duplication of facilities does not alter the solution space of the considered instance, as one may assume that at most one facility is opened at each location. Note that our algorithm will also satisfy this property, since $d(i,i')^2 = 0$ for any pair of co-located facilities i, i'.

Initially, we let G be the client-facility graph with bipartition $\mathcal{D}$ and $\mathcal{V}^{(\ell)} \cup \mathcal{V}^{(\ell+1)}$ that has an edge from client j to facility $i \in \mathcal{V}^{(\ell)}$ if (j,i) is present in $G^{(\ell)}$ and to $i \in \mathcal{V}^{(\ell+1)}$ if (j,i) is present in $G^{(\ell+1)}$. The opening time t_i of facility i is now naturally set to $t_i^{(\ell)}$ if $i \in \mathcal{V}^{(\ell)}$ and to $t_i^{(\ell+1)}$ if $i \in \mathcal{V}^{(\ell+1)}$. Informally, G is the union of the two client-facility graphs $G^{(\ell)}$ and $G^{(\ell+1)}$ where the client vertices are shared. We then generate[2] the conflict graph $H^{(\ell,1)}$ from G and t. As the induced subgraph of $H^{(\ell,1)}$ on vertex set $\mathcal{V}^{\ell}$ equals $H^{(\ell)} = H^{(\ell,0)}$, we have that $\mathsf{IS}^{(\ell)}$ is also an independent set of $H^{(\ell,1)}$. We obtain a maximal independent set $\mathsf{IS}^{(\ell,1)}$ of $H^{(\ell,1)}$ by greedily extending $\mathsf{IS}^{(\ell)}$. Clearly, the independent set can only increase so we still have $|\mathsf{IS}^{(\ell,1)}| > k$.

To produce the remaining sequence, we iteratively perform changes, but construct and output a new conflict graph and maximal independent set after *each* such change. Specifically, we remove from G each facility $i \in \mathcal{V}^{(\ell)}$, one by one. At the end of the procedure (after $|\mathcal{V}^{(\ell)}|$ many steps), we have $G = G^{(\ell+1)}$ and so $H^{(\ell,p_\ell)} = H^{(\ell+1)}$. Note that at each step, our modification to G results in removing a single facility i from the associated conflict graph. Thus, if $\mathsf{IS}^{(\ell,s)}$ is an independent set in $H^{(\ell,s)}$ before a modification, then $\mathsf{IS}^{(\ell,s)} \setminus \{i\}$ is an independent set in $H^{(\ell,s+1)}$. We obtain a maximal independent set $\mathsf{IS}^{(\ell,s+1)}$ of $H^{(\ell,s+1)}$ by greedily extending $\mathsf{IS}^{(\ell,s)} \setminus \{i\}$. Then, for each step s, we have $|\mathsf{IS}^{(\ell,s+1)}| \geq |\mathsf{IS}^{(\ell,s)}| - 1$, as required.

2) Analysis: The total running time is $n^{O(\epsilon^{-1} \log n)}$ since the number of steps L (and the number of dual solutions in our sequence) is $n^{O(\epsilon^{-1} \log n)}$ and each step runs in polynomial time since it involves the construction of at most $O(|\mathcal{F}|)$ conflict graphs and maximal independent sets.

We proceed to analyze the approximation guarantee. Consider the first time that we produce some maximal independent set IS of size exactly k. Suppose that when this happened, we were moving between two solutions $\alpha^{(\ell)}$ and $\alpha^{(\ell+1)}$, i.e., $\mathsf{IS} = \mathsf{IS}^{(\ell,s)}$ is a maximal independent set of $H^{(\ell,s)}$ for some $1 \leq s \leq p_\ell$. That we may assume that $s \geq 1$ follows

[2]Recall that a conflict graph is defined in terms of a client-facility graph G and t: the vertices are the facilities in G, and two facilities i and i' are adjacent if there is some client j that is adjacent to both of them in G and $d(i,i')^2 \leq \delta \min(t_i, t_{i'})$.

from $|\mathsf{IS}^{(0)}| > k$ and $\mathsf{IS}^{(\ell-1,p_\ell)} = \mathsf{IS}^{(\ell)} = \mathsf{IS}^{(\ell,0)}$ (recall that IS was selected to be the *first* independent set of size k).

To ease notation, we let $H = H^{(\ell,s)}$ and denote by G the "hybrid" client-facility graph that generated H. In order to analyze the cost of IS, let us form a hybrid solution α by setting $\alpha_j = \min(\alpha_j^{(\ell)}, \alpha_j^{(\ell+1)})$ for each client $j \in \mathcal{D}$. Note that $\alpha \leq \alpha^{(\ell)}$ is a feasible solution of $\mathsf{DUAL}(\lambda)$ where $\lambda = \ell \cdot \epsilon_z$ and, since $\alpha^{(\ell)}$ and $\alpha^{(\ell+1)}$ are close, $\alpha_j \geq \alpha_j^{(\ell)} - \frac{1}{n^2}$ and $\alpha_j \geq \alpha_j^{(\ell+1)} - \frac{1}{n^2}$ for all j. For each client j, we define a set of facilities $S_j \subseteq \mathsf{IS}$ to which j contributes, as follows. For all $i \in \mathsf{IS}$, we have $i \in S_j$ if $\alpha_j > d(j,i)^2$. Note that S_j is a subset of j's neighborhood in G and therefore for all $i \in S_j$

$$\alpha_j = \min(\alpha_j^{(\ell)}, \alpha_j^{(\ell+1)}) \leq t_i = \begin{cases} t_i^{(\ell)} & \text{if } i \in \mathcal{V}^{(\ell)} \\ t_i^{(\ell+1)} & \text{if } i \in \mathcal{V}^{(\ell+1)} \end{cases}$$

Using the fact that $\alpha^{(\ell+1)}$ is a good dual solution, we can bound the total service cost of all clients in the integral solution IS. Let us first proceed separately for those clients with $|S_j| > 0$. Let $\mathcal{D}_0 = \{j \in \mathcal{D} : |S_j| = 0\}$, and $\mathcal{D}_{>0} = \mathcal{D} \setminus \mathcal{D}_0$. We remark that the analysis is now very similar to the proof of Theorem III.3. We define $\beta_{ij} = [\alpha_j - d(i,j)^2]^+$ and similarly $\beta_{ij}^{(\ell)} = [\alpha_j^{(\ell)} - d(i,j)^2]^+$ and $\beta_{ij}^{(\ell+1)} = [\alpha_j^{(\ell+1)} - d(i,j)^2]^+$.

Lemma IV.6. *For any $j \in \mathcal{D}_{>0}$, $d(j,\mathsf{IS})^2 \leq \rho \cdot \left(\alpha_j - \sum_{i \in S_j} \beta_{ij}\right)$.*

Proof: Consider some $j \in \mathcal{D}_{>0}$ and first suppose that $|S_j| = 1$. Then, if we let $S_j = \{i\}$, $\alpha_j = \beta_{ij} + d(j,i)^2 \geq \beta_{ij} + d(j,\mathsf{IS})^2$ just as in "Case $s = 1$" of Theorem III.3. Next, suppose that $|S_j| = s > 1$. In other words, j is contributing to multiple facilities in IS. By construction we have $\alpha_j \leq \min(t_i, t_{i'})$ for any two facilities $i, i' \in S_j$. Thus, $\alpha_j - \sum_{i \in S_j} \beta_{ij} \geq \frac{1}{\rho} d(j,\mathsf{IS})^2$ by the exact same arguments as in "Case $s > 1$" of Theorem III.3. ∎

Next, we bound the total service cost of all those clients that do not contribute to any facility in IS. The proof is very similar to "Case $s = 0$" in the proof of Theorem III.3. The proof of the following two lemmas are available in the full version of this paper [1].

Lemma IV.7. *For every $j \in \mathcal{D}_0$, $d(j,\mathsf{IS})^2 \leq (1 + 5\epsilon)\rho \cdot \alpha_j$.*

One difference compared to the analysis in Section III-B is that not all opened facilities are fully paid for. However, they are almost paid for:

Lemma IV.8. *For any $i \in \mathsf{IS}$, $\sum_{j \in \mathcal{D}} \beta_{ij} \geq \lambda - \frac{1}{n}$.*

We now combine the above lemmas to bound the approximation guarantee of the found solution. Recall that OPT_k denotes the optimum value of the standard LP-relaxation (see Section II).

Theorem IV.9. $\sum_{j \in \mathcal{D}} d(j,\mathsf{IS})^2 \leq (1+6\epsilon)\rho \cdot \mathrm{OPT}_k$.

Proof: From Lemmas IV.6 and IV.7 we have:

$$\sum_{j \in \mathcal{D}} d(j,\mathsf{IS})^2 \leq (1+5\epsilon)\rho \sum_{j \in \mathcal{D}} \left(\alpha_j - \sum_{i \in S_j} \beta_{ij}\right).$$

By Lemma IV.8 (note that by definition, $\sum_{i \in \mathsf{IS}} \beta_{ij} = \sum_{i \in S_j} \beta_{ij}$),

$$\begin{aligned} \sum_{j \in \mathcal{D}} \left(\alpha_j - \sum_{i \in S_j} \beta_{ij}\right) &\leq \sum_{j \in \mathcal{D}} \alpha_j - |\mathsf{IS}| \left(\lambda - \tfrac{1}{n}\right) \\ &= \sum_{j \in \mathcal{D}} \alpha_j - k \cdot \lambda + \tfrac{k}{n} \leq \mathrm{OPT}_k + 1\,, \end{aligned}$$

where the last inequality follows from $k \leq n$ and, as α is a feasible solution to $\mathsf{DUAL}(\lambda)$, $\sum_{j \in \mathcal{D}} \alpha_j - k \cdot \lambda \leq \mathrm{OPT}_k$. The statement now follows from $\mathrm{OPT}_k \geq \sum_{j \in \mathcal{D}} \min_{i \in \mathcal{F}} d(i,j)^2 \geq n$ and $n \gg 1/\epsilon$, which imply that $\mathrm{OPT}_k + 1 \leq (1+\epsilon)\mathrm{OPT}_k$. ∎

We have thus proved that our quasi-polynomial algorithm produces a $(\rho + O(\epsilon))$-approximate solution which implies Theorem I.1. The quasi-polynomial algorithms for the other considered problems are the same except for the selection of δ and ρ, and that in the k-median problem the connection costs are the (non-squared) distances.

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments. This research is supported by ERC Starting Grant 335288-OptApprox.

References

[1] S. Ahmadian, A. Norouzi-Fard, O. Svensson, and J. Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *CoRR*, abs/1612.07925, 2016.

[2] A. Archer, R. Rajagopalan, and D. B. Shmoys. Lagrangian relaxation for the k-median problem: New insights and continuity properties. In *Proc. 11th ESA*, pages 31–42, 2003.

[3] D. Arthur and S. Vassilvitskii. How slow is the k-means method? In *Proc. 22nd SoCG*, pages 144–153, 2006.

[4] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proc. 18th SODA*, pages 1027–1035, 2007.

[5] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

[6] P. Awasthi, A. Blum, and O. Sheffet. Stability yields a PTAS for k-median and k-means clustering. In *Proc. 51st FOCS*, pages 309–318, 2010.

[7] P. Awasthi, M. Charikar, R. Krishnaswamy, and A. K. Sinop. The hardness of approximation of euclidean k-means. In *Proc. 31st SoCG*, pages 754–767, 2015.

[8] M.-F. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Proc. 20th SODA*, pages 1068–1077, 2009.

[9] J. Byrka and K. Aardal. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *SIAM J. Comput.*, 39(6):2212–2231, 2010.

[10] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proc. 26th SODA*, pages 737–756, 2015.

[11] M. Charikar and S. Guha. Improved combinatorial algorithms for facility location problems. *SIAM J. Comput.*, 34(4):803–824, 2005.

[12] F. A. Chudak and D. B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. *SIAM J. Comput.*, 33(1):1–25, 2004.

[13] V. Cohen-Addad, P. N. Klein, and C. Mathieu. The power of local search for clustering. *CoRR*, abs/1603.09535, 2016.

[14] D. Feldman, M. Monemizadeh, and C. Sohler. A PTAS for k-means clustering based on weak coresets. In J. Erickson, editor, *Proc. 23rd SoCG*, pages 11–18, 2007.

[15] Z. Friggstad, M. Rezapour, and M. R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. *CoRR*, abs/1603.08976, 2016.

[16] A. Gupta and K. Tangwongsan. Simpler analyses of local search algorithms for facility location. *CoRR*, abs/0809.2554, 2008.

[17] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *J. ACM*, 50:795–824, 2003.

[18] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proc. 34th STOC*, pages 731–740, 2002.

[19] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

[20] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004.

[21] E. Lee, M. Schmidt, and J. Wright. Improved and simplified inapproximability for k-means. *CoRR*, abs/1509.00916, 2015.

[22] S. Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Inf. Comput.*, 222:45–58, 2013.

[23] S. Li and O. Svensson. Approximating k-median via pseudo-approximation. *SIAM J. Comput.*, 45(2):530–547, 2016.

[24] J. Lin and J. S. Vitter. Approximation algorithms for geometric median problems. *Inf. Process. Lett.*, 44:245–249, 1992.

[25] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theor.*, 28(2):129–137, Sept. 2006.

[26] J. Matoušek. On approximate geometric k-clustering. *Discrete & Computational Geometry*, 24(1):61–84, 2000.

[27] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the k-means problem. *J. ACM*, 59(6):28:1–28:22, Jan. 2013.

[28] D. B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *Proc. 29th STOC*, pages 265–274, 1997.

[29] A. Vattani. k-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, 45(4):596–616, 2011.

[30] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[31] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.