Achieving Anonymity via Weak Lower Bound Constraints for k-Median and k-Means

Anna Arutyunova
Universität Bonn, Germany

Melanie Schmidt
Universität Köln, Germany

Abstract

We study k-clustering problems with lower bounds, including k-median and k-means clustering with lower bounds. In addition to the point set P and the number of centers k, a k-clustering problem with (uniform) lower bounds gets a number B. The solution space is restricted to clusterings where every cluster has at least B points. We demonstrate how to approximate k-median with lower bounds via a reduction to facility location with lower bounds, for which $O(1)$-approximation algorithms are known.

Then we propose a new constrained clustering problem with lower bounds where we allow points to be assigned *multiple times* (to different centers). This means that for every point, the clustering specifies a set of centers to which it is assigned. We call this *clustering with weak lower bounds*. We give an 8-approximation for k-median clustering with weak lower bounds and an $O(1)$-approximation for k-means with weak lower bounds.

We conclude by showing that at a constant increase in the approximation factor, we can restrict the number of assignments of every point to 2 (or, if we allow fractional assignments, to $1+\epsilon$). This also leads to the first bicritera approximation algorithm for k-means with (standard) lower bounds where bicriteria is interpreted in the sense that the lower bounds are violated by a constant factor.

All algorithms in this paper run in time that is polynomial in n and k (and d for the Euclidean variants considered).

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

Keywords and phrases Clustering with Constraints, lower Bounds, k-Means, Anonymity

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.7

Related Version *Full Version*: https://arxiv.org/abs/2009.03078 [7]

Funding *Anna Arutyunova*: DFG grant RO 5439/1-1.
Melanie Schmidt: DFG grant SCHM 2765/1-1.

Acknowledgements We thank anonymous reviewers for their detailed comments to a previous version.

1 Introduction

We study k-clustering problems with lower bound constraints. Imagine the following approach to publish a reduced version of a large data set: Partition the data into clusters of similar objects, then replace every cluster by one (weighted) point that represents it best. Publish these weighted representatives. For example, it is a fairly natural approach for data that can be modeled as vectors from $\mathbb{R}^d$ to replace a data set by a set of mean vectors, where every mean vector represents a cluster. When representing a cluster by one point, the mean vector minimizes the squared error of the representation. This is a common use case of k-means clustering.

38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021).
Editors: Markus Bläser and Benjamin Monmege; Article No. 7; pp. 7:1–7:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we ask the following: If we want to publish the representatives, it would be very convenient if the clusters were of sufficient size to ensure a certain level of anonymity of the individual data points that they represent. Can we achieve this, say, in the case of k-means clustering or for the related k-median problem?

Using clustering with lower bounds on the cluster sizes to achieve anonymity is an idea posed by Aggarwal et al. [3]. They introduce it in the setting of radii-based clustering, and define the *r-gather problem*: Given a set of points P from a metric space, find a clustering and centers for the clusters such that the maximum distance between a point and its center is minimized and such that every cluster has at least r points. They also define the (k, r)-center problem which is the same problem as the r-gather problem except that the number of clusters is also bounded by the given number k. So the (k, r)-center problem takes the k-center clustering objective but restricts the solution space to clusterings where every cluster has at least r points. Aggarwal et al. [3] give a 2-approximation for both problems.

We pose the same question, but for sum-based objectives such as k-median and k-means. Here instead of the maximum distance between a point and its center, the (squared) distances are added up for all points. For a set of points P from a metric space and a number k, the k-median problem is to find a clustering and centers such that the sum of the distances of every point to its closest center is minimized. For k-means clustering, the distances are squared, the metric is usually Euclidean, and the centers are allowed to come from all of $\mathbb{R}^d$. Now for k-median/k-means clustering with lower bounds, the situation differs in two aspects. We are given an additional parameter B and solutions now satisfy the additional constraint that every cluster has at least B points[1]. To achieve this, points are no longer necessarily assigned to their closest center but the solution now involves an assignment function of points to centers. The objective then is to minimize the (squared) sum of distances from every point to its assigned center. To the best of the authors' knowledge, k-median and k-means with lower bounds have not been studied, but for k-median, an $O(1)$-approximation follows from known work (see below).

For the related (also sum-based) facility location problem, finding solutions with lower bounds on the cluster sizes appeared in very different contexts. Given sets P and F from a finite metric space and *opening costs* for the points in F, the *facility location problem* asks to partition P into clusters and to assign a center from F to each cluster such that the sum of the distances of every point to its cluster plus the sum of the opening costs of open centers is minimized. For facility location with lower bounds, an additional parameter B is given and every cluster has to have at least B points. Karger and Minkoff [20] as well as Guha, Meyerson and Munagala [13] use relaxed versions of facility location with (uniform) lower bounds as subroutines for solving network design problems. This inspired the seminal work of Svitkina [26], who gives a constant-factor approximation algorithm for the facility location problem with (uniform) lower bounds. Ahmadian and Swamy [5] improve the approximation ratio to 82.6. Ahmadian and Swamy [6] state that the algorithm by Svitkina can be adapted for k-median by adequately replacing the first reduction step at the cost of an increase in the approximation factor.

It is often the case that restricting the number of clusters to k instead of having facility costs makes the design of approximation algorithms much more cumbersome, in particular when constraints are involved. For example, the related problem of finding a facility location

[1] In the introduction, we stick to uniform lower bounds since this is what we want for anonymity. In the technical part, we also discuss non-uniform lower bounds.

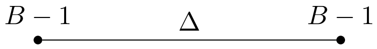

Figure 1 On the difference between lower-bounded clustering and weakly lower-bounded clustering.

solution where every cluster has to satisfy an *upper* bound, usually referred to as *capacitated facility location*, can be 3-approximated (see Aggarwal et al [2]), but finding a constant-factor approximation for capacitated k-median clustering is a long standing open problem [1, 16].

We demonstrate that the situation for *lower* bounds is different. By a relatively straightforward approach that we borrow from the area of approximation algorithms for hierarchical clustering, we show that approximation algorithms for facility location with lower bounds can be converted into approximation algorithms for k-median with lower bounds (at the cost of an increase in the approximation ratio), and this reduction works also for more general k-clustering problems including k-means. This leaves us with two challenges:

1. The resulting approximation algorithm has a very high approximation ratio.
2. For k-means clustering with lower bounds, no bicriteria or true approximation algorithm is known, and the results for standard facility location with lower bounds do not extend to the case for squared Euclidean distances: Both known algorithms for facility location use the triangle inequality an uncontrolled number of times to bound the cost of multiple reassignment steps. Thus the relaxed triangle inequality is not sufficient, as the resulting bound would depend on this number. Also the bicriteria algorithms by Karger and Minkoff [20] and Guha, Meyerson and Munagala [13] require repeated application of the triangle inequality. Thus, k-means with lower bounds needs a new technique.

To tackle these challenges, we define a new variation of lower-bounded clustering that we call *weakly lower-bounded* k-clustering. Here we allow points to be *allocated multiple times*. However a point may not be assigned more than once to the same center. This means that our ‘clustering’ is not a partitioning into subsets, but consists of not necessarily disjoint clusters (whose union is P). Each cluster has to respect the lower bound. To explain this idea, consider Figure 1. There are two locations with $B-1$ points each, and the distance between the two locations is Δ. For clustering with a lower bound of B, we can only open one center, which results in a clustering cost of $(B-1)\Delta$ for k-median (and $\Omega(B\Delta^2)$ for k-means). For clustering with weak lower bound B, we allow to assign points multiple times (but only to different centers). For each allocation, we pay the connection cost. In Figure 1, this allows us to open two centers while assigning one point from every location to the other location. This costs 2Δ for k-median (and $\Omega(\Delta^2)$ for k-means) for the two extra assignments. So even though we pay for more connections, the overall cost is smaller. This means that clustering with weak lower bounds can have an arbitrarily smaller cost than clustering with lower bounds, and in a way, this is a benefit: it means that we potentially pay less for having the lower bounds satisfied. Of course it also means that the gap between the optimal costs of the two problem variants with (standard) lower bounds and weak lower bounds is unbounded. We obtain the following results.

- We design an 8-approximation algorithm for weakly lower-bounded k-median and an $O(1)$-approximation algorithm for weakly lower-bounded k-means. The algorithms are conceptually simpler than their counterparts for lower-bounded facility location.
- Then we show that we can adapt the solutions such that every point is assigned to *only two centers* at the cost of a constant factor increase in the approximation ratio. We say that a solution has b-weak lower bounds if every point is assigned to at most b centers, so our results satisfy 2-weak lower bounds.

- Furthermore, we show that for $\epsilon \in (0, 1)$ we can also get $O(1/\epsilon)$-approximate solutions that satisfy $(1 + \epsilon)$-weak lower bounds if we allow fractional assignments of points.
- Finally, we show that our result on 2-weak lower bounds also implies a $(O(1), O(1))$-bicriteria approximation result for lower bounds, where the lower bounds are satisfied only to an extent of $B/O(1)$. Applying this result to squared Euclidean distances yields a bicriteria approximation for k-means with lower bounds, which is the first to the best of the authors' knowledge.
- Our results also extend to non-uniform lower bounds.

Recall our anonymization goal. When using weakly lower-bounded clustering, we still get the number of clusters that we desire and we also fully satisfy the anonymity requirement. We achieve this by distorting the data slightly by allowing data points to influence two clusters. In the fractional case, we get a solution where every data point is assigned to one main cluster and then contributes an ϵ-connection to a different cluster. By this small disturbance of the data set, we can meet the anonymity lower bound requirement for all clusters.

Techniques. The proof that k-clustering can be reduced to facility location builds upon a known nesting technique from the area of approximation algorithms for hierarchical clustering and is relatively straightforward. Our conceptional contribution is the definition of weakly lower-bounded clustering as a means to achieve anonymity. To obtain constant-factor approximations for weakly lower-bounded clustering, the idea is to incorporate an estimate for the cost of establishing lower bounds via facility costs, approximate a k-clustering problem with facility costs and then enforce lower bounds on a solution by connecting the closest $B - \ell$ points to a center which previously only had ℓ points. Similar ideas are present in the literature, which we adapt to our new problem formulation.

The main technical contribution in our paper is the proof that a solution assigning points to arbitrarily many centers can be converted into a solution where every point is assigned at most twice (or $(1 + \epsilon)$-times, respectively), not only for k-median, but also for k-means. The latter means that the proof cannot use subsequent reassignment steps as it is the case in previous algorithms but has to carefully ensure that points are only reassigned once. We can also bypass this problem in the construction of a bicriteria algorithm. Previous bicriteria algorithms for lower bounds do not extend to k-means due to using multiple reassignments.

Related work. Approximation algorithms for clustering have been studied for decades. The unconstrained k-center problem can be 2-approximated [11, 14] and this is tight under P $\neq$ NP [15]. The (k, r)-center problem we discussed above is introduced and 2-approximated in [3]. We also call this problem k-center with lower bounds. McCutchen and Khuller [25] study k-center with lower bounds in a streaming setting and provide a $(6 + \epsilon)$-approximation. One can also consider *non-uniform* lower bounds, i.e., every center has an individual lower bound that has to be satisfied if the center is opened. This variant is studied by Ahmadian and Swamy in [6] and they give a 3-approximation (for the slightly more general *k-supplier* problem with non-uniform lower bounds).

The facility location problem has a rich history of approximation algorithms and the currently best algorithm due to Li [22], achieving an approximation ratio of 1.488, is very close to the best known lower bound of 1.463 [12]. Bicriteria approximation algorithms for facility location with lower bounds are developed by Karger and Minkoff [20] and Guha, Meyerson and Munagala [13]. Svitkina [26] gives the first $O(1)$-approximation algorithm. The core of the algorithm is a reduction to facility location with capacities, embedded in a long chain of pre- and postprocessing steps. Ahmadian and Swamy [5] improve the approximation guarantee to 82.6. For the case of non-uniform lower bounds, Li [23] gives an $O(1)$-approximation algorithm. Although we did not discuss this in the introduction because it is less relevant to the anonymity motivation, this result also implies an $O(1)$-approximation for k-median with non-uniform lower bounds, as we show in the full version of this paper [7, Appendix A].

The k-median and k-means problems are APX-hard with the best known lower bounds being $1+2/e$ [18] and 1.0013 [8, 21]. The k-median problem can be $(2.675+\epsilon)$-approximated [9] and the best known approximation ratio for the k-means problem is $6.357+\epsilon$ [4]. To the best of the authors' knowledge, k-median and k-means with lower bounds have not been studied before. For the k-median problem, $O(1)$-approximations follow relatively easy from the work on facility location as outlined in the full version [7, Appendix A] and there is a possible adaptation of the algorithm by Svitkina as mentioned above. The authors are neither aware of an approximation algorithm or bicriteria algorithm for facility location with lower bounds that works for squared metrics, nor of one for k-means with lower bounds. We propose a bicriteria result that is applicable to k-means.

Finding a polynomial constant-factor approximation algorithm for the k-median problem with *upper bounds*, i.e., with capacities, is a long standing open problem. Recently, efforts have been made to obtain FPT approximation algorithms for the problem [1, 10].

2 Preliminaries

A k-clustering problem gets a finite set of input points P, a possibly infinite set of possible centers F, and a number $k \in \mathbb{N}$ and asks for a set of centers $C \subset F$ with $|C| \leq k$ and a mapping $a : P \to C$ such that

$$\mathrm{cost}(P, C, a) = \mathrm{cost}(C, a) = \sum_{x \in P} d(x, a(x))$$

is minimized, where $d : (P \cup F) \times (P \cup F) \to \mathbb{R}^+$ is a distance function that is symmetric and satisfies that $d(x, y) = 0$ iff $x = y$. For the *generalized k-median problem*, the distance d satisfies the α-relaxed triangle inequality, i.e., for all $x, y, z \in P \cup F$, it holds that $d(x, y) \leq \alpha d(x, z) + \alpha d(y, z)$.

We define the *k-median problem* as a generalized k-median problem with $P = F$ (finite) and $\alpha = 1$, and the *k-means problem* by setting $F = \mathbb{R}^d$ and $P \subset F$, and choosing d as the squared Euclidean distance, for which $\alpha = 2$. For these two problems, choosing the mapping $a : P \to C$ is always optimally done by assigning every point to (one of) its closest center(s). A *generalized facility location problem* has the same input as a generalized k-median problem except that it gets facility costs $f : F \to \mathbb{R}$ instead of a number k. The goal is to find a set of centers $C \subset F$ without cardinality constraint that minimizes $\sum_{x \in P} d(x, a(x)) + \sum_{c \in C} f(c)$. We use the term facility location not only if d is a metric but also in the case of a distance function satisfying the α-relaxed triangle inequality, analogously to the generalized k-median problem defined above.

We study generalized k-median and generalized facility location problems under side constraints which means that the choice of the mapping a is restricted. The side constraints that we study are versions of lower bounded clustering, i.e., they demand that every center gets a minimum number of points that are assigned to it. For clustering with (uniform) lower bounds, the input contains a number B and every cluster in the solution has to have at least B points. Non-uniform lower bounds are meaningful in the case of a finite set F and then, non-uniform lower bounds are given via a function $B : F \to \mathbb{N}$. If any points are assigned to a center $c \in F$ in a feasible solution, then it has to be at least $B(c)$ points.

When adding constraints, there is a subtle detail in the definition of generalized k-median problems for the case $P = F$: The question whether the center of a cluster has to be part of the cluster. Notice that without constraints, this makes no difference because assigning a center to a different center than itself cannot be beneficial. When we add lower bounds, this can change. We assume that choosing a center outside of the cluster is allowed and specifically say when the solution is such that centers are members of their clusters.

Our new problem variant called *weakly lower-bounded generalized k-median* is defined as follows. Given an instance of the same form as for the unconstrained generalized k-median problem plus lower bounds $B : F \to \mathbb{N}$, the goal is to compute a set of at most k centers $C \subset F$ and an assignment $a\colon P \to \mathcal{P}(C)$ such that the lower bound is satisfied, i.e., $|\{x \in P \mid c \in a(x)\}| \geq B(c)$ for all $c \in C$ and every point is assigned at least once. If a point is assigned multiple times the distance of the point to all assigned centers is paid by the solution. The total cost of a solution is given by

$$\mathrm{cost}(C, a) = \sum_{x \in P} \sum_{c \in a(x)} d(x, c).$$

If a solution of a weakly lower-bounded clustering problem satisfies that every point is assigned to at most b centers, then we say that the solution satisfies b-weak lower bounds.

3 Reducing Lower-Bounded k-Clustering to Facility Location

In this section, we observe that by using a known technique from the area of approximation algorithms for hierarchical clustering, we can turn an approximation algorithm for generalized facility location with lower bounds into an algorithm for generalized k-median with lower bounds. The technique is called *nesting*. Given two solutions S_1 and S_2 for the same generalized facility location problem with different number of centers $k_1 > k_2$, nesting describes how to find a solution S with k_2 centers which has a cost bounded by a constant times the costs of S_1 and S_2 and which is *hierarchically compatible* with S_1, i.e., the clusters in S result from merging clusters in S_1. We use this by computing a solution S_1 with an approximation algorithm for generalized facility location satisfying the lower bounds and a solution S_2 for unconstrained generalized k-median and then combining them via a nesting step. The resulting solution S has at most k centers and the clusters result from clusters that satisfy the lower bound – thus they satisfy the lower bound as well. For uniform lower bounds, the execution of this plan is very straightforward, for non-uniform lower bounds we have to be a bit more careful and adjust the nesting appropriately. Since most of this section follows relatively straightforwardly from known work, we defer the details to the full version [7, Appendix A]. Although the reduction is applicable to generalized k-median, this only helps to obtain constant-factor approximations for k-median because no approximation algorithms for generalized facility location with lower bounds are known for $\alpha > 1$. We get the following statement from combining the (adjusted) nesting results from Lin et al. [24] and the approximation algorithms for facility location with uniform lower bounds by Ahmadian and Swamy [5] and non-uniform lower bounds by Li [23].

▶ **Corollary 1.** *There exist polynomial-time $O(1)$-approximation algorithms for the k-median problem with uniform and non-uniform lower bounds.*

As a final note we observe that the crucial property of lower bound constraints we use here is *mergeability*: If a uniform lower bound is satisfied for a solution, then merging clusters results in a solution that is still feasible. This is in stark contrast to for example capacitated clustering. Our reduction works for mergeable constraints in general.

4 Generalized k-Median with Weak Lower Bounds

Now we consider a relaxed version of generalized k-median with lower bounds where points in P can be assigned multiple times. This relaxation does make sense since we have lower bounds on the centers, so it can be more valuable to assign points to multiple centers to satisfy the lower bounds instead of closing the respective centers. To see this we refer to Figure 1. We call this problem *generalized k-median with weak lower bounds.*

For ease of presentation, it is sensible to assume that F is finite. We observe that we can always set $F = P$ at a constant increase in the cost function if we are given a uniform lower bound. In particular, we assume in this section that $F = P$ holds for k-means.

▶ **Lemma 2.** *Let P be a point set and F be a possibly infinite set of centers. Let $a : P \to F$ be a mapping and define $a'(x) = \arg\min_{y \in P} d(y, a(x))$. Then it holds that*

$$\sum_{x\in P} d(x, a'(x)) \leq 2\alpha \cdot \sum_{x\in P} d(x, a(x)).$$

Proof. The lemma follows from the relaxed triangle inequality:

$$\sum_{x\in P} d(x, a'(x)) \leq \alpha \sum_{x\in P} \big(d(x, a(x)) + d(a(x), a'(x))\big) \leq 2\alpha \cdot \sum_{x\in P} d(x, a(x)). \qquad ◀$$

To achieve anonymity it is enough to have a uniform lower bound. However if we assume $F = P$ from the beginning, then our results also hold for *non-uniform lower bounds*, so we consider this more general case in this section.

For standard k-median/k-means with weak lower bounds we give an 8-approximate algorithm and an $O(1)$-approximate algorithm respectively. Furthermore we show that a solution to generalized k-median with weak lower bounds can be transformed into a solution to generalized k-median with *2-weak lower bounds* in polynomial time. We show that this transformation increases the cost only by a factor of $\alpha(\alpha + 1)$. We combine this with the approximation algorithm for standard k-median/k-means with weak lower bounds and obtain an approximation algorithm for standard k-median/k-means with 2-weak lower bounds. If we allow fractional assignments we show how to obtain a solution which assigns every point by an amount of at most $1 + \epsilon$ for arbitrary $\epsilon \in (0, 1)$, losing $\lceil \frac{1}{\epsilon} \rceil \alpha(\alpha + 1) + 1$ in the approximation factor.

Computing a solution. To approximate generalized k-median with weak non-uniform lower bounds, we reduce this problem to generalized k-median with center costs. In this variant of generalized k-median, the input contains both a number k *and* center opening costs $f : F \to \mathbb{R}^+$. The objective is then

$$\text{cost}^f(C, a) = \sum_{x\in P} d(x, a(x)) + \sum_{c\in C} f(c)$$

while the solution space is constrained to center sets of size at most k as for generalized k-median. The reduction that we use works by introducing a center cost of

$$f(c) = \sum_{p\in D_c} d(p, c) \tag{1}$$

for every point $c \in F$. This cost is paid if c becomes a center. Here D_c is the set consisting of the $B(c)$ nearest points in P to c. The idea for this reduction is adapted from the bicriteria algorithm for lower-bounded facility location presented by Guha, Meyerson and Munagala [13] and Karger, Minkoff [20].

Note that for a center c in a feasible solution (C,a) to generalized k-median with weak lower bounds, the term $\sum_{p\in D_c} d(p,c)$ is a lower bound on the assignment cost caused by c. This leads to the following lemma.

▶ **Lemma 3.** *Let OPT' be an optimal solution to the generalized k-median problem with center costs as defined in* (1) *and $OPT = (O,h)$ be an optimal solution to generalized k-median with weak lower bounds. It holds that* $\mathrm{cost}^f(OPT') \le 2\,\mathrm{cost}(OPT)$.

Proof. For $p \in P$ let $c_p = \mathrm{argmin}\{d(p,c) \mid c \in h(p)\}$ be the closest center to which p is assigned in OPT. We define $h'(p) = c_p$ for all $p \in P$ and obtain a feasible solution (O,h') to the generalized k-median problem with center cost. Furthermore we have

$$\begin{aligned}\mathrm{cost}^f(OPT') \le \mathrm{cost}^f(O,h') &= \sum_{c\in O} f(c) + \sum_{p\in P} d(p,h'(p))\\ &= \sum_{c\in O}\sum_{p\in D_c} d(p,c) + \sum_{p\in P} d(p,h'(p))\\ &\le 2\sum_{p\in P}\sum_{c\in h(p)} d(p,c)\\ &= 2\,\mathrm{cost}(OPT).\end{aligned}$$

The second inequality follows from the fact that $\sum_{c\in O}\sum_{p\in D_c} d(p,c)$ and $\sum_{p\in P} d(p,h'(p))$ are both lower bounds on the assignment cost of OPT. ◀

Let (C,a) be a solution for the generalized k-median problem with center costs. To turn it into a solution for generalized k-median with weak lower bounds we have to modify the assignment. Let $c \in C$ and $n_c = |a^{-1}(c)|$. We additionally assign $m_c = \max\{0, B(c) - n_c\}$ points to c to satisfy the lower bound. Let $S_c \subset D_c$ be the set of points in D_c which are not assigned to c. We choose m_c points from S_c and assign them to c. This is feasible since we are allowed to assign points multiple times. Let (C,a') be the corresponding solution.

▶ **Lemma 4.** *It holds that* $\mathrm{cost}(C,a') \le \mathrm{cost}^f(C,a)$.

Proof. The additional assignment cost for each center $c \in C$ can be upper bounded by $\sum_{p\in D_c} d(p,c)$. We obtain

$$\begin{aligned}\mathrm{cost}(C,a') \le &\sum_{c\in C}\sum_{p\in D_c} d(p,c) + \sum_{p\in P} d(p,a(p))\\ = &\,\mathrm{cost}^f(C,a).\end{aligned}$$

◀

Lemma 3 and Lemma 4 imply the following corollary.

▶ **Corollary 5.** *Given a γ-approximation for the generalized k-median problem with center costs, we get a 2γ-approximation for the generalized k-median problem with weak lower bounds in polynomial time.*

For k-median, we combine Corollary 5 with Corollary 5.5 from [27] which shows that an algorithm by Jain et al. [17] can be used to obtain a 4-approximation for the k-median problem with center costs. This gives an 8-approximation for k-median with weak lower bounds. For k-means, we use the algorithm by Jain and Vazirani [19] which was originally designed for k-median. However, as outlined in the journal version [19], it can be used for k-means when $F = P$, and also for k-median with center costs. The two extensions are not conflicting and can both be applied to obtain an $O(1)$-approximation for k-means with center costs for the case $F = P$.

4.1 Reducing the Number of Assignments per Client

We see that the solution for standard k-median/k-means with weak lower bounds computed above can assign a point to all centers in the worst case. The number of assigned centers per point cannot be bounded by a constant. This may not be desirable in the context of publishing anonymized representatives since the distortion of the original data set is not bounded.

However, we show that any solution to the generalized k-median problem with weak lower bounds can be transformed into a solution assigning every point at most twice. This increases the cost by a factor of $\alpha(\alpha+1)$. Recall that α is the constant appearing in the relaxed triangle inequality. This leads to the following theorem.

▶ **Theorem 6.** *Given a solution (C,a) to generalized k-median with weak lower bounds, we can compute a solution $(\widetilde{C},\widetilde{a})$ to generalized k-median with 2-weak lower bounds (assigning every point at most twice) in polynomial time such that* $\mathrm{cost}(\widetilde{C},\widetilde{a}) \leq \alpha(\alpha+1)\,\mathrm{cost}(C,a)$.

Reassignment process. We start by setting $\widetilde{C}=C$ and $\widetilde{a}=a$ and modify both $\widetilde{C}$ and $\widetilde{a}$ until we obtain a feasible solution to generalized k-median with 2-weak lower bounds. During the process, the centers in $\widetilde{C}$ are called *currently open*, and when a center is deleted from $\widetilde{C}$, we say it is *closed*. The centers are processed in an arbitrary but fixed order, i.e., we assume that $C=\{c_1,\dots,c_{k'}\}$ for some $k'\leq k$ and process them in order $c_1,\dots,c_{k'}$. We say that c_i is *smaller* than c_j if $i<j$.

Let $c=c_i$ be the currently processed center. By P_c, we denote the set of points assigned to c under $\widetilde{a}$. We divide P_c into three sets $P_c^1=\{q\in P_c \mid |\widetilde{a}(q)|=1\}$, $P_c^2=\{q\in P_c \mid |\widetilde{a}(q)|=2\}$ and $P_c^3=\{q\in P_c \mid |\widetilde{a}(q)|\geq 3\}$. Furthermore with $C(P_c^3)$ we denote all centers which are connected to at least one point in P_c^3 under $\widetilde{a}$.

If P_c^3 is empty, we are done and proceed with the next center in $\widetilde{C}$. Otherwise we need to empty P_c^3. Observe that points in P_c^3 are assigned to multiple centers, so if we delete the connection between one of these points and c, the point is still served by some other center. However, doing so may violate the lower bound at c. So we have to replace this connection.

As long as P_c^3 is non-empty, we do the following. We pick a center $d=\min C(P_c^3)\backslash\{c\}$ and a point $x\in P_c^3$ connected to d. We want to assign a point y from P_d^1 to c to free x. For technical reasons, we restrict the choice of y: We exclude all points from the subset $\overline{P_d^1}:=\{q\in P_d^1 \mid |a(q)|\geq 3 \text{ and } a(q)\cap\{c_1,\dots,c_{i-1}\}\cap\widetilde{C}\neq\emptyset\}$, i.e., all points which were assigned to at least 3 centers under the initial assignment a, and where one of these at least 3 centers is still open *and* smaller than c.

If $P_d^1\backslash\overline{P_d^1}$ is non-empty, we pick a point $y\in P_d^1\backslash\overline{P_d^1}$ arbitrarily. We set $\widetilde{a}(y)=\{d,c\}$ and $\widetilde{a}(x)=\widetilde{a}(x)\backslash\{c\}$. So x is no longer connected to c, but to satisfy the lower bound at c we replace x by y (Figure 2). If $P_d^1\backslash\overline{P_d^1}$ is empty, our replacement plan does not work. Instead, we close d. This means that x is now assigned to one center less, and, if this happens repeatedly, x will at some point no longer be in P_c^3. Since we close d, all points in P_d^1 have

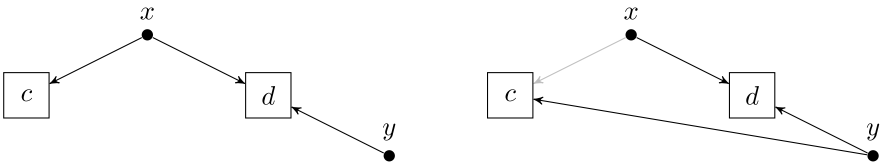

Figure 2 Connection between $x\in P_c^3$ and c is deleted. A point $y\in P_d^1$ replaces x.

Algorithm 1 Reducing the number of assigned centers per point to two.

```
1  define an ordering on the centers $c_1 \leq c_2 \ldots \leq c_{k'}$
2  set $\widetilde{C} := C$ and $\widetilde{a} := a$
3  for all $c \in C$
4      $P_c := \{q \in P \mid c \in \widetilde{a}(q)\}$
5      $P_c^3 := \{q \in P_c \mid |\widetilde{a}(q)| \geq 3\}$,  $P_c^i := \{q \in P_c \mid |\widetilde{a}(q)| = i\}$ for $i = 1, 2$
6      $C(P_c^3) := \bigcup_{q \in P_c^3} \widetilde{a}(q)$
7  for $i = 1$ to $l$ do
8      while $P_{c_i}^3 \neq \emptyset$ do
9          $d = \min C(P_{c_i}^3) \backslash \{c_i\}$
10         $\overline{P_d^1} = \{q \in P_d^1 \mid |a(q)| \geq 3$ and $a(q) \cap \{c_1, \ldots, c_{i-1}\} \cap \widetilde{C} \neq \emptyset\}\}$
11         if $P_d^1 \backslash \overline{P_d^1} = \emptyset$ then
12             for all $q \in P_d^1$
13                 let $e = \min(a(q) \cap \widetilde{C})$
14                 set $\widetilde{a}(q) = \{e\}$
15             delete $d$ from $\widetilde{C}$ and all connections to $d$ in $\widetilde{a}$
16         else
17             pick $x \in P_{c_i}^3$ connected to $d$ and $y \in P_d^1 \backslash \overline{P_d^1}$
18             set $\widetilde{a}(x) = \widetilde{a}(x) \backslash \{c_i\}$, $\widetilde{a}(y) = \{c_i, d\}$
```

to be reassigned because they are only connected to d. For each $q \in P_d^1$, we reassign q to the smallest currently open center in $a(q)$. Notice that such a center exists and is smaller than c because $P_d^1 = \overline{P_d^1}$ and for every $q \in \overline{P_d^1}$, there is at least one center in $a(q) \cap \widetilde{C}$ which is smaller than c.

The entire process is described in Algorithm 1. It satisfies the following invariants.

▶ **Lemma 7.** *Algorithm 1 computes a feasible solution $(\widetilde{C}, \widetilde{a})$ to generalized k-median with 2-weak lower bounds. Furthermore the following properties hold during all steps of the algorithm.*

1. *The algorithm never establishes connections for points currently assigned more than once.*
2. *For any center $c \in C$, P_c does not change before c is processed or closed.*
3. *If a connection between $x \in P$ and the currently processed center $c \in \widetilde{C}$ is deleted by the algorithm, we have from this time on $x \notin P_c^3$ until termination. Moreover P_c^3 remains empty after c is processed.*
4. *While the algorithm processes $c \in C$ we always have $c < \min C(P_c^3) \backslash \{c\}$. Moreover all currently open centers which are smaller than c remain open until termination.*
5. *If the algorithm establishes a new connection in Line 14 or Line 18 it remains until termination.*

Proof. The process terminates: For every iteration of the while loop starting in Line 8, either a point is deleted from $P_{c_i}^3$ or there is at least one point $x \in P_{c_i}^3$ for which $|\widetilde{a}(x)|$ is reduced by one. Furthermore $|\widetilde{a}(x)|$ does never increase for any $x \in P_{c_i}^3$.

The final solution satisfies lower bounds: Every time we delete a connection between a point and a center it either happens because the center is closed or we replace this connection by assigning a new point to it. So the lower bounds are satisfied at all open centers.

All points stay connected to a center: Assume that the algorithm deletes the connection between a point p and the center d it is exclusively assigned to. This only happens if at this time d is closed by the algorithm. Then p is assigned to another center as defined in Line 14.

We conclude that the solution is feasible.

Property 1: The algorithm establishes connections in Line 14 and Line 18 which always involve a point currently assigned once.

Property 2: Let $c \in C$. Connections are only changed for the center that is currently processed or for a smaller center which has been processed already. Thus, the algorithm does not add or delete any connections involving c before c is processed or closed.

Property 3: Assume that after the connection between $x \in P_c^3$ and c is deleted by the algorithm, x is again part of P_c^3. That would require that the algorithm establishes a new connection for a point which is connected more than once, which does not happen by Property 1. For the same reason P_c^3 remains empty after c is processed by the algorithm.

Property 4: Assume c is currently processed by the algorithm and $d = \min C(P_c^3)\backslash\{c\}$. We know that at this time P_d^3 is non-empty, which is by Property 3 only possible if d is processed after c. Thus we have $c < d$. This also means that centers can only be closed by the algorithm if they are not processed so far.

Property 5: If a connection is deleted, the respective point is either connected to more than two centers or to a center which is closed at this time. A connection in Line 14 or Line 18 is established by the algorithm between a point which is at this time assigned exactly once and a center which is already processed or currently processed by the algorithm. Thus the point is from this time on never assigned to more than two centers and the center remains open until termination by Property 4. So the necessary conditions for a deletion of this connection are never fulfilled. ◀

We now want to bound the cost of new connections created by the algorithm by the cost of the original solution. Notice that only Line 18 generates new connections, Line 14 re-establishes connections that were originally present. So let N_c be the set of all points newly assigned to c by the algorithm in Line 18 while center c is processed. For $y \in N_c$ let d_y be the respective center in Line 9 of Algorithm 1 and x_y the point in Line 17 contained in P_c^3 and connected to d_y. Using the α-relaxed triangle inequality, we obtain the following upper bound.

$$\begin{aligned} d(y,c) &\le \alpha(d(y,x_y) + d(x_y,c)) \le \alpha\Big(\alpha\big(d(y,d_y) + d(d_y,x_y)\big) + d(x_y,c)\Big) \\ &= \alpha^2\big(d(y,d_y) + d(d_y,x_y)\big) + \alpha d(x_y,c). \end{aligned} \tag{2}$$

We can apply (2) to all $c \in \widetilde{C}$ and all $y \in N_c$. This yields the following upper bound on the cost of the final solution $(\widetilde{C}, \widetilde{a})$.

$$\begin{aligned} \mathrm{cost}(\widetilde{C},\widetilde{a}) &= \sum_{c\in\widetilde{C}} \sum_{\substack{y\in P:\\ c\in\widetilde{a}(y)}} d(y,c) = \sum_{c\in\widetilde{C}} \Big(\sum_{y\in P_c\backslash N_c} d(y,c) + \sum_{y\in N_c} d(y,c)\Big) \\ &\le \sum_{c\in\widetilde{C}} \Big(\sum_{y\in P_c\backslash N_c} d(y,c) + \sum_{y\in N_c} \alpha^2(d(y,d_y) + d(d_y,x_y)) + \alpha d(x_y,c)\Big). \end{aligned} \tag{3}$$

Expression (3) is what we want to pay for. We show in Observation 8 below that all involved distances contribute to the original cost as well. So in principle, we can bound each summand by a term in the original cost. But what we need to do is to bound the number of times that each term in the original cost gets charged. To organize the counting, we count how many times a specific tuple of a point z and a center f occurs as $d(z,f)$ in (3). Since it is important at which position a tuple appears, we give names to the different occurrences (also see Figure 3).

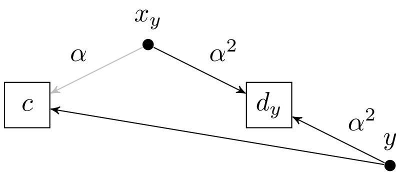

Figure 3 Bounding the distance between y and c. The respective distances appear with a factor of α or α^2. Tuple (x_y, c) is of Type 1 and $(x_y, d_y), (y, d_y)$ are of Type 2.

We say that that a tuple appears as a tuple of Type 0 if it appears as $d(y, c)$ in (3), as tuple of Type 1 if it appears as $d(x_y, c)$, and as tuple of Type 2 if it appears as $d(y, d_y)$ or $d(d_y, x_y)$. We distinguish the latter type further by calling a tuple occurring as $d(y, d_y)$ a tuple of Type 2.1 and a tuple occurring as $d(x_y, d_y)$ a tuple of Type 2.2. We say that $(y, d_y), (d_y, x_y)$ and (x_y, c) *contribute* to the cost of (y, c), where by the *cost* of (y, c) we mean the upper bound on $d(y, c)$ in (2) which we want to pay for.

▶ **Observation 8.** *If a tuple (z, f), $z \in P, f \in C$, occurs as Type 0, 1 or 2, then $f \in a(z)$, so in particular, $d(z, f)$ occurs as a term in the cost of the original solution.*

Proof. For a center c the set $P_c \backslash N_c$ consists of points which are assigned to c by the initial assignment a or assigned to c while c is not processed by the algorithm. The latter can only happen if a connection is reestablished in Line 14 which requires that the connection was already present in (C, a). So Type 0 tuples satisfy the statement.

For Type 1 and 2 tuples, consider $y \in N_c$ for some center c and the respective tuples $(x_y, c), (y, d_y), (x_y, d_y)$. Notice that both y and x_y are connected to d_y the step before y is assigned to c. By Property 4 of Lemma 7 we have $c < d_y$. Thus we know by Property 2 of Lemma 7 that P_{d_y} is not changed by the algorithm at least until y is assigned to c. So $d_y \in a(y)$ and $d_y \in a(x_y)$ which proves that Type 2 tuples satisfy the statement. Moreover it holds that $c \in a(x_y)$ since there is a time where $x_y \in P_c^3$. This can, by Property 1 of Lemma 7, only happen if the connection between x_y and c is already part of (C, a). Thus, Type 1 tuples satisfy the statement. ◀

As indicated above, a tuple (z, f) can contribute to the cost of multiple tuples. Notice that a tuple occurs at most once as a tuple of Type 0 in (3). To bound the cost of $(\widetilde{C}, \widetilde{a})$ we bound the number of times a tuple appears as Type 1 or Type 2 tuple in (3).

▶ **Lemma 9.** *For all $z \in P, f \in C$, the tuple (z, f) can appear in* (3) *at most once as a tuple of Type 1 and at most once as a tuple of Type 2.*

Proof. In the following, the tuple whose cost the tuple (z, f) contributes to will always be named (y, c), and we denote the time at which y is newly assigned to c by t.

Type 1: Assume (z, f) contributes to the cost of (y, c) as a tuple of Type 1. Then $f = c$. Notice that at the time step before t we must have $z \in P_c^3$ and afterwards, z is never again contained in P_c^3 by Property 3 of Lemma 7. Thus the pair (z, c) can never again be responsible for any reassignment to c, i.e., $(z, c) = (z, f)$ does not contribute to any further cost as a tuple of Type 1.

Type 2.1: Assume that (z, f) contributes to the cost of (y, c) as a tuple of Type 2.1. Then $z = y$. At the time step before t, we have $y \in P_f^1$, $f \in C(P_c^3)$, and at time t, we have $y \in P_c^2 \cap P_f^2$. By Property 5 of Lemma 7, newly established connections are never deleted, so after time t, it always holds that $y \in P_c$. So even if y is in P_f at a later time, it cannot be in P_f^1 since it is also connected to c. So $(y, f) = (z, f)$ does not contribute to any further cost as tuple of Type 2.1. Furthermore by Property 1 of Lemma 7 we know that y is always assigned to fewer than three centers after t which means that (y, f) does not contribute as tuple of Type 2.2 to the cost of any connection established by the algorithm after t either.

Type 2.2: Finally we consider the case where (z, f) contributes to the cost of (y, c) as a tuple of Type 2.2. At time t, the algorithm processes c. By the way the algorithm chooses f and z, we know that $z \in P_c^3$ (at the beginning of the process, i.e., before t) and $f = \min C(P_c^3)\backslash\{c\}$. After t, Property 3 of Lemma 7 implies $z \notin P_c^3$, which means that as a tuple of Type 2.2, it can never again contribute to the cost of any tuple containing c. Assume instead that it contributes (as Type 2.2) to the cost of a tuple (y', c') for a center $c' \neq c$, and some point $y' \in P$. This is supposed to happen after t, so y' is newly assigned to c' at some time $t' > t$. Before c' is processed, we must always have $z \in P_{c'}^3$ by Property 1 and 2 of Lemma 7. So in particular, at time $t < t'$ we have $c' \in C(P_c^3)\backslash\{c\}$. Moreover we know that at some time while c' is processed by the algorithm we have $f = \min C(P_{c'}^3)\backslash\{c'\}$. Using Property 4 of Lemma 7 we conclude that $c' < f$. Which is a contradiction since the algorithm chose f and not c' at time t, i.e., $f = \min C(P_c^3)\backslash\{c\}$ must hold. Thus, (z, f) cannot contribute to the cost of (y', c') as a tuple of Type 2.2.

It is left to show that (z, f) cannot contribute to the cost of any (y', c') as a tuple of Type 2.1 at some time $t' > t$. For a contribution as Type 2.1, we would have $z = y'$ and $y' \in P_f^1$. We show that in this case y' is in fact contained in $\overline{P_f^1}$. Remember that at time t we have $y' = z \in P_c^3$ and that this only happens if $|a(y')| \geq 3$ by Property 1 of Lemma 7. Moreover c is still open by Property 4 of Lemma 7 and is smaller than c'. Thus $c \in a(y') \cap \{e \mid e < c'\} \cap \widetilde{C}$, which proves $y' \in \overline{P_f^1}$. Therefore the algorithm does not assign y' to c' (see Lines 11-15) and (z, f) does not contribute as tuple of Type 2.1 to the cost of any connection established by the algorithm after t. ◀

We now know that a tuple only appears at most once as any of the three tuple types. For the final counting, we define $T0$, $T1$ and $T2$ as the sets of all tuples of Type 0, 1 and 2, respectively. We could already prove a bound on the cost now, but to make it slightly smaller and prove Theorem 6, we need one final statement.

▶ **Lemma 10.** *The set $T0 \cap T1 \cap T2$ is empty.*

Proof. Let $(z, f) \in T0 \cap T1 \cap T2$. Since (z, f) is of Type 0, the point z must be connected to f in the final assignment $\widetilde{a}$. We distinguish whether the connection between z and f was deleted at some point by the algorithm or not. If it is not deleted, (z, f) cannot be of Type 1 since this would require that z is temporarily not assigned to f. Otherwise the connection between z and f was deleted while f was processed and later reestablished by the algorithm in Line 14.

By assumption the tuple is also of Type 2. Assume it is of Type 2.1 and contributes to the cost of a tuple (y, c) with $z = y$. We know that $c < f$ by Property 4 of Lemma 7. Consider the time when z is newly assigned to c. The step before we have $z \in P_f^1$. On the other hand while f is processed we have $z \in P_f^3$ in contradiction to Property 1 of Lemma 7.

Assume finally that (z, f) is of Type 2.2 and contributes to the cost of a tuple (y, c). Again we have $c < f$. Consider the time y is newly assigned to c. The step before we have $z \in P_c^3$ and, by Property 1 and 2 of Lemma 7, also $z \in P_f^3$. At the time the connection between z and f is reestablished by the algorithm, both centers are contained in $a(z) \cap \widetilde{C}$. This is a contradiction to $c < f = \min(a(z) \cap \widetilde{C})$. This completes the proof. ◀

Proof of Theorem 6. Slightly abusing the notation we write $d(e)$ for a tuple $e = (z, f)$ by which we mean the distance $d(z, f)$. Combining Lemma 9 and 10 we obtain

$$\begin{aligned}
\operatorname{cost}(\widetilde{C}, \widetilde{a}) &\leq \sum_{c \in \widetilde{C}} \Big(\sum_{y \in P_c \setminus N_c} d(y, c) + \sum_{y \in N_c} \alpha^2 (d(y, d_y) + d(d_y, x_y)) + \alpha d(x_y, c) \Big) && (3) \\
&= \sum_{e \in T0} d(e) + \alpha^2 \sum_{e \in T2} d(e) + \alpha \sum_{e \in T1} d(e) && (4) \\
&\leq (\alpha^2 + \alpha) \operatorname{cost}(C, a). && (5)
\end{aligned}$$

By Lemma 9 we know that a tuple only appears at most once as any of the three tuple types. We replace (3) by summing up the cost of all tuples in T_i for $i = 0, 1, 2$ with the respective factor for each type and obtain (4).

Finally by Observation 8 the cost $d(e)$ for $e \in T_0 \cup T_1 \cup T_2$ occurs as a term in the original solution and $T_0 \cap T_1 \cap T_2 = \emptyset$ by Lemma 10, which proves (5). ◀

So it is possible to reduce the number of assignments per point to two at a constant factor increase in the approximation factor. We can go even further and allow points to be fractionally assigned to centers which poses the question if it is possible to bound the assigned amount by a number smaller than two. Indeed we can prove for every $\epsilon \in (0, 1)$ that we can modify a solution to generalized k-median with weak lower bounds such that every point is assigned by an amount of at most $1 + \epsilon$ and the cost increases by a factor of $\mathcal{O}(\frac{1}{\epsilon}\alpha^2)$. Note that even if we allow fractional assignments of points to centers, the centers remain either open or closed, which differentiates our result from a truly fractional solution, where it is also allowed to open centers fractionally. Furthermore, the new assignment assigns every point to at most two centers. It is assigned by an amount of one to one center and potentially by an additional amount of ϵ to a second center.

Since we consider fractional assignments we modify our notation and denote with $\widetilde{a}_x^c \in [0, 1]$ the amount by which $x \in P$ is assigned to $c \in \widetilde{C}$, where $\widetilde{C}$ is the set of centers. Let $\widetilde{a}_x = \sum_{c \in \widetilde{C}} \widetilde{a}_x^c$ be the amount by which $x \in P$ is assigned to $\widetilde{C}$. The assignment $\widetilde{a}$ is feasible if $\widetilde{a}_x \geq 1$ for all $x \in P$ and $\sum_{x \in P} \widetilde{a}_x^c \geq B(c)$ for all $c \in \widetilde{C}$, and its cost is

$$\operatorname{cost}(\widetilde{C}, \widetilde{a}) = \sum_{c \in \widetilde{C}} \sum_{x \in P} \widetilde{a}_x^c d(x, c).$$

We omit the proof of the following theorem as it is similar to the proof of Theorem 6, but to satisfy lower bounds we can only assign an amount of ϵ from points which are already assigned once. Therefore we consider suitable sets with $\lceil \frac{1}{\epsilon} \rceil$ points, which leads to the increase of $\mathcal{O}(\frac{1}{\epsilon})$ in the approximation factor. For more details we refer to [7, Appendix C].

▶ **Theorem 11.** *Given $0 < \epsilon < 1$ and a solution (C, a) to generalized k-median with weak lower bounds, we can compute a solution $(\widetilde{C}, \widetilde{a})$ to generalized k-median with $(1 + \epsilon)$-weak lower bounds, i.e., $\widetilde{a}_x \leq 1 + \epsilon$ for all $x \in P$ in polynomial time such that* $\operatorname{cost}(\widetilde{C}, \widetilde{a}) \leq (\lceil \frac{1}{\epsilon} \rceil \alpha(\alpha + 1) + 1) \operatorname{cost}(C, a)$.

On pages 7-8 we reduce generalized k-median with weak lower bounds to generalized k-median with center cost and obtain an 8 or $O(1)$-approximation for k-median or k-means with weak lower bounds, respectively. We combine this with Theorem 6 to get a solution with 2-weak lower bounds whose cost is a constant factor away from the problem with weak lower bounds. Since weak lower bounds are a relaxation of 2-weak lower bounds, we get:

Algorithm 2 A $(\beta, \gamma \max\{\frac{\alpha\beta}{1-\beta}+1, \frac{\alpha^2\beta}{1-\beta}\})$-bicriteria approximation algorithm to generalized k-median with lower bounds.

```
Input  : γ-approximate solution $(C, a)$ to generalized $k$-median with 2-weak lower
         bounds, $C = \{c_1, \ldots, c_{k'}\}$
Output : Bicriteria solution $(C', a')$ to generalized $k$-median with lower bounds.
set $C' = \emptyset$, $a'(x) = \perp$ for all $x \in P$
$N = P$
for $i = 1$ to $k'$ do
    $A_i = \{x \in P \mid c_i \in a(x)\}$
    $B_i = \{x \in A_i \mid a(x) \subset \{c_1 \ldots, c_i\}\} \cap N$
    if $A_i \cap N \geq \beta B(c_i)$ then
        set $a'(x) = c_i$ for all $x \in A_i \cap N$
        $N = N \backslash A_i$
        $C' = C' \cup \{c_i\}$
    else
        set $a'(x) = \arg\min_{c \in C'} d(x, c)$ for all $x \in B_i$
```

▶ **Corollary 12.** *Let OPT be an optimal solution to k-median/k-means with 2-weak lower bounds. We can compute a solution (C, a) in polynomial time for*

1. *k-median with 2-weak lower bounds with* $\text{cost}(C, a) \leq 16 \,\text{cost}(OPT)$
2. *k-means with 2-weak lower bounds with* $\text{cost}(C, a) \leq O(1) \,\text{cost}(OPT)$.

Combining the results from Section 4 with Theorem 11 we obtain:

▶ **Corollary 13.** *Let OPT be an optimal solution to k-median/k-means with $(1+\epsilon)$-weak lower bounds. We can compute a solution (C, a) in polynomial time for*

1. *k-median with $(1+\epsilon)$-weak lower bounds with* $\text{cost}(C, a) \leq (16\lceil \frac{1}{\epsilon} \rceil + 8) \,\text{cost}(OPT)$
2. *k-means with $(1+\epsilon)$-weak lower bounds with* $\text{cost}(C, a) \leq O(\frac{1}{\epsilon}) \,\text{cost}(OPT)$.

4.2 A Bicriteria Algorithm to Generalized k-Median with Lower Bounds

A (β, δ)-bicriteria solution for generalized k-median with lower bounds consists of at most k centers $C' \subset F$ and an assignment $a' \colon P \to C$ such that at least $\beta B(c)$ points are assigned to $c \in C'$ by a' and $\text{cost}(C', a') \leq \delta \,\text{cost}(OPT)$. Here OPT denotes an optimal solution to generalized k-median with lower bounds.

Given a $\beta \geq \frac{1}{2}$ and a γ-approximate solution to generalized k-median with 2-weak lower bounds (C, a), we can compute a $(\beta, \gamma \max\{\frac{\alpha\beta}{1-\beta}+1, \frac{\alpha^2\beta}{1-\beta}\})$-bicriteria solution in the following way. Let $C = \{c_1, \ldots, c_{k'}\}$ for some $k' \leq k$. We process the centers in order $c_1, \ldots, c_{k'}$ and decide if they are open or closed. We say that c_i is *smaller* than c_j if $i < j$. If we decide that a center c is open we directly assign at least $\lceil \beta B(c) \rceil$ points to c. In the beginning all points are unassigned.

Consider center c_i. Let A_i be the set of all points assigned to c_i under a. We know that $|A_i| \geq B(c_i)$. If at least $\lceil \beta B(c_i) \rceil$ points in A_i are not assigned so far, c_i remains open and all currently unassigned points from A_i are assigned to c_i (Figure 4). If less than $\lceil \beta B(c_i) \rceil$ points from A_i are unassigned, the center is closed.

Let C' denote the centers from $\{c_1, \ldots, c_{i-1}\}$ which are open and B_i the set of unassigned points from A_i which are not connected to any center larger than c_i under a. To guarantee that all points are assigned at the end, we have to care about points in B_i. By assumption there are at most $\lfloor \beta B(c_i) \rfloor$ such points. We simply assign a point $p \in B_i$ to the nearest center $\arg\min_{c \in C'} d(c, p)$ in C'.

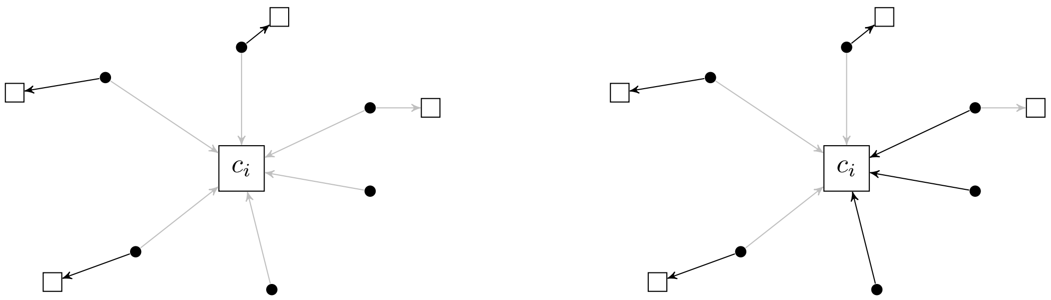

Figure 4 Shows the case where A_i contains at least $\lceil \beta B(c_i) \rceil$ unassigned points. The three points on the left are already assigned to other centers and the three points on the right are newly assigned to c_i. The gray connections come from a.

The whole procedure is described in Algorithm 2. For the proof of the claimed approximation factor we refer to [7, Appendix D].

▶ **Theorem 14.** *Given a γ-approximate solution (C, a) to generalized k-median with 2-weak lower bounds and a fixed $\beta \in [0.5, 1)$, Algorithm 2 computes a $(\beta, \gamma \max\{\frac{\alpha\beta}{1-\beta} + 1, \frac{\alpha^2\beta}{1-\beta}\})$-bicriteria solution to generalized k-median with lower bounds in polynomial time. In particular, there exists a polynomial-time $(\frac{1}{2}, O(1))$-bicriteria approximation algorithm for k-means with lower bounds.*

References

1 Marek Adamczyk, Jaroslaw Byrka, Jan Marcinkowski, Syed Mohammad Meesum, and Michal Wlodarczyk. Constant-factor FPT approximation for capacitated k-median. In *Proceedings of the 27th Annual European Symposium on Algorithms (ESA)*, pages 1:1–1:14, 2019. doi:10.4230/LIPIcs.ESA.2019.1.

2 Ankit Aggarwal, Anand Louis, Manisha Bansal, Naveen Garg, Neelima Gupta, Shubham Gupta, and Surabhi Jain. A 3-approximation algorithm for the facility location problem with uniform capacities. *Mathematical Programming*, 141(1-2):527–547, 2013. doi:10.1007/s10107-012-0565-4.

3 Gagan Aggarwal, Rina Panigrahy, Tomás Feder, Dilys Thomas, Krishnaram Kenthapadi, Samir Khuller, and An Zhu. Achieving anonymity via clustering. *ACM Transactions on Algorithms (TALG)*, 6(3):49:1–49:19, 2010. doi:10.1145/1798596.1798602.

4 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and Euclidean k-median by primal-dual algorithms. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 61–72, 2017. doi:10.1109/FOCS.2017.15.

5 Sara Ahmadian and Chaitanya Swamy. Improved approximation guarantees for lower-bounded facility location. In *Proceedings of the 10th International Workshop on Approximation and Online Algorithms (WAOA)*, pages 257–271, 2012. doi:10.1007/978-3-642-38016-7_21.

6 Sara Ahmadian and Chaitanya Swamy. Approximation algorithms for clustering problems with lower bounds and outliers. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, (ICALP)*, pages 69:1–69:15, 2016. doi:10.4230/LIPIcs.ICALP.2016.69.

7 Anna Arutyunova and Melanie Schmidt. Achieving anonymity via weak lower bound constraints for k-median and k-means. arXiv:2009.03078v2.

8 Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of Euclidean k-means. In *Proceedings of the 31st International Symposium on Computational Geometry (SoCG)*, pages 754–767, 2015. doi:10.4230/LIPIcs.SOCG.2015.754.

9 Jaroslaw Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Transaction on Algorithms (TALG)*, 13(2):23:1–23:31, 2017. doi:10.1145/2981561.

10 Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT approximations for k-median and k-means. In *Proceedings of the 46th International Colloqium on Automata, Languages, and Programming (ICALP)*, pages 42:1–42:14, 2019. doi:10.4230/LIPIcs.ICALP.2019.42.

11 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science (TCS)*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.

12 Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of Algorithms*, 31(1):228–248, 1999. doi:10.1006/jagm.1998.0993.

13 Sudipto Guha, Adam Meyerson, and Kamesh Munagala. Hierarchical placement and network design problems. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 603–612, 2000. doi:10.1109/SFCS.2000.892328.

14 Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986. doi:10.1145/5925.5933.

15 Wen-Lian Hsu and George L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics (DAM)*, 1(3):209–215, 1979. doi:10.1016/0166-218X(79)90044-1.

16 Tanmay Inamdar and Kasturi Varadarajan. Capacitated sum-of-radii clustering: An FPT approximation. In *Proceedings of the 28th Annual European Symposium on Algorithms (ESA)*, volume 173 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 62:1–62:17, 2020. doi:10.4230/LIPIcs.ESA.2020.62.

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

18 Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 731–740, 2002. doi:10.1145/509907.510012.

19 Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001. doi:10.1145/375827.375845.

20 David R. Karger and Maria Minkoff. Building Steiner trees with incomplete global knowledge. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 613–623, 2000. doi:10.1109/SFCS.2000.892329.

21 Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *Information Processing Letters (IPL)*, 120:40–43, 2017. doi:10.1016/j.ipl.2016.11.009.

22 Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation*, 222:45–58, 2013. doi:10.1016/j.ic.2012.01.007.

23 Shi Li. On facility location with general lower bounds. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2279–2290, 2019. doi:10.1137/1.9781611975482.138.

24 Guolong Lin, Chandrashekhar Nagarajan, Rajmohan Rajaraman, and David P. Williamson. A general approach for incremental approximation and hierarchical clustering. *SIAM Journal on Computing (SICOMP)*, 39(8):3633–3669, 2010. doi:10.1137/070698257.

25 Richard Matthew McCutchen and Samir Khuller. Streaming algorithms for k-center clustering with outliers and with anonymity. In *Proceedings of the 11th International Workshop on Approximation, Randomization and Combinatorial Optimization (APPROX)*, pages 165–178, 2008. doi:10.1007/978-3-540-85363-3_14.

26 Zoya Svitkina. Lower-bounded facility location. *ACM Transactions on Algorithms (TALG)*, 6(4):69, 2010. doi:10.1145/1824777.1824789.

27 Jens Vygen. Lecture notes – approximation algorithms for facility location problems, 2004/2005. accessed May 8th, 2019. URL: http://gett.or.uni-bonn.de/~vygen/files/fl.pdf.