Fault Tolerant Clustering Revisited

Nirman Kumar* Benjamin Raichel †

Abstract

In discrete k-center and k-median clustering, we are given a set of points P in a metric space M, and the task is to output a set $C \subseteq \mathsf{P}$, $|C| = k$, such that the cost of clustering P using C is as small as possible. For k-center, the cost is the furthest a point has to travel to its nearest center, whereas for k-median, the cost is the sum of all point to nearest center distances. In the fault-tolerant versions of these problems, we are given an additional parameter $1 \leq \ell \leq k$, such that when computing the cost of clustering, points are assigned to their ℓth nearest-neighbor in C, instead of their nearest neighbor. We provide constant factor approximation algorithms for these problems that are both conceptually simple and highly practical from an implementation stand-point.

1 Introduction

Two of the most common clustering problems are k-center and k-median clustering. In both these problems, the goal is to find the minimum cost partition of a given point set P into k clusters. Each cluster is defined by a point in the set of cluster ***centers***, $C \subseteq \mathsf{P}$, where $|C| = k$. In k-center clustering, the cost is the maximum distance of a point to its assigned cluster center, and in k-median clustering, the cost is the sum of distances of points to their assigned cluster center. In both cases, given a set of cluster centers C, a point is assigned to its closest center in C. Both these problems are NP-hard for most metric spaces. Hochbaum and Shmoys showed that k-center clustering has a 2-approximation algorithm, but for every $\varepsilon > 0$ it cannot be approximated to better than $(2 - \varepsilon)$ unless P=NP [8]. A 2-approximation was also provided by Gonzalez [5], and by Feder and Greene [4]. For k-median, the best known approximation factor is $1 + \sqrt{3} + \varepsilon$. This is a recent result of Li and Svensson [13], but the approximation version of the k-median problem has a long history, and before the result of Li and Svensson, the best known result was by Arya *et al.* [2], that achieved an approximation factor of $(3 + \varepsilon)$ for any $\varepsilon > 0$, using local search. In general metric spaces, k-median is also APX hard. Jain *et al.* showed that k-median is hard to approximate within a factor of $1 + 2/e \approx 1.736$ [9]. In Euclidean spaces, the k-center problem remains APX-hard [4], while k-median admits a PTAS [1, 11, 7].

Fault-Tolerance. As mentioned earlier, in both the k-center and k-median problems, each point is assigned to its closest center. Consider a realistic scenario where k-center clustering is used to decide in which k of n cities, certain facilities (say Sprawlmarts or hospitals) are opened, so that for clients in the n cities, their maximum distance to a facility is minimized. Once the k cities are decided upon, clearly each client goes to its nearest such facility when it requires service. Due to facility downtimes however, sometimes clients may need to go to their second closest, or third

*University of Illinois; `nkumar5@illinois.edu`; `http://www.cs.uiuc.edu/~nkumar5/`.

†University of Illinois; `raichel2@illinois.edu`; `http://www.cs.uiuc.edu/~raichel2/`.

closest facility. Thus, in the fault-tolerant version of the k-center problem, we say that the cost of a client is the distance to its ℓth nearest facility for some fixed $1 \leq \ell \leq k$. The problem now is to find a set of k centers so that the worst case cost is minimized, where in the worst case each client actually goes to its ℓth nearest facility, and the cost of clustering is the maximum distance traveled by any client.

The fault-tolerant k-center problem was first studied by Krumke [12], who gave a 4-approximation algorithm for this problem. Chaudhuri *et al.* provided a 2-approximation algorithm for this problem [3], which is the best possible under standard complexity theoretic assumptions. In both these papers, the version considered, differs slightly from ours in that one only considers points which are not centers when computing the point that has the furthest distance to its ℓth closest center. Khuller *et al.* [10] later considered both versions of the k-center problem. Their first version is the same as ours, i.e. the cost is the maximum distance of any point (including centers) to its ℓth nearest center. They gave a 2-approximation when $\ell < 4$ and a 3-approximation otherwise. Their second version is the same as that of Krumke [12]. For this version, they provided a 2-approximation algorithm matching the result of Chaudhuri *et al.* [3].

For k-median clustering, a fault-tolerant version has been considered by Swamy and Shmoys [14]. In their version, k-centers need to be opened, and in addition there is a fault-tolerance parameter $r \leq k$. The cost for a client is the sum of distances to its r closest facilities. Swamy and Shmoys actually considered a much more general setting for the fault tolerant facility location problem, where the requirement r_j for a client j could be non-uniform. However, for the fault tolerant k-median problem, the algorithm they provided was for a uniform requirement $r_j = r$ for all clients. For this problem, they provided a 4-approximation algorithm. The fault tolerant version we consider is different from the version of Swamy and Shmoys. In our version, the cost for a client is its distance to its ℓth nearest facility (instead of the sum to its l nearest facilities), and we add the cost for all the clients to get the cost of the clustering.

Our Contribution. Our main contribution is in providing and proving the correctness of a natural technique for fault-tolerant clustering. In particular, letting $m = \lfloor k/\ell \rfloor$, we show that given a set of centers which is a constant factor approximation to the optimal m-center (resp. m-median) clustering, one can easily compute a set of k centers whose cost is a constant factor approximation to the optimal fault-tolerant k-center (resp. k-median) clustering. Specifically, in order to turn the non-fault-tolerant solution into a fault-tolerant one, simply add for each point of the m center set, its ℓ nearest neighbors in P. In other words, our main contribution is in proving a relationship between the fault-tolerant and non-fault-tolerant cases, specifically that the non-fault-tolerant solution for m centers is already a near optimal fault-tolerant solution in that, up to a constant factor, it is enough to "reinforce" the current center locations rather than looking for new ones.

For fault-tolerant k-center we prove that if one applies this post-processing technique to any c-approximate solution to the non-fault-tolerant problem with m centers, then one is guaranteed a $(1 + 2c)$-approximation to the optimal fault-tolerant clustering. Similarly, for fault-tolerant k-median we show this post processing technique leads to a $(1 + 4c)$-approximation.

Our second main result is that using the algorithm of Gonzalez [5] for the initial m-center solution, gives a tighter approximation ratio guarantee. Specifically, we get a 3-approximation when $\ell | k$, and a 4-approximation otherwise, for fault-tolerant k-center. Additionally, on the median side, to the best of our knowledge, we are the first to consider this particular variant of fault-tolerant k-median clustering.

The approximation ratios of our algorithms are reasonable but not optimal. However, the authors feel that the algorithms more than make up for this in their conceptual simplicity and practicality from an implementation stand-point. Notably, if one has an existing implementation of an m-center or an m-median clustering approximation algorithm, one can immediately turn it into a fault-tolerant clustering algorithm for k centers with this technique.

Organization. In Section 2 we set up notation and formally define our variants for the fault-tolerant k-center and k-median problems. In Section 3 we review the algorithm of Gonzalez [5], and present our algorithms for the fault-tolerant k-center and k-median problems. In Section 4 we analyze the approximation ratios of our algorithm. We conclude in Section 5.

2 Preliminaries

2.1 Notation

We are given a set of n points $\mathsf{P} = \{\mathsf{p}_1, \ldots, \mathsf{p}_n\}$ in a metric space M. Let $\mathsf{d}(\mathsf{p}, \mathsf{p}')$ denote the distance between the points p and p' in M. For a point $\mathsf{p} \in M$, and a number $x \geq 0$, let $\mathsf{ball}(\mathsf{p}, x)$ denote the closed ball of radius x with center p. For a point $\mathsf{p} \in M$, a subset $S \subseteq \mathsf{P}$, and an integer $1 \leq i \leq |S|$, let $\mathsf{d}_i(\mathsf{p}, S)$ denote the radius of the smallest (closed) ball with center p that contains at least i points in the set S. Let $\mathsf{nn}_i(\mathsf{p}, S)$ denote the ith nearest neighbor of p in S, i.e. the point in S such that $\mathsf{d}(\mathsf{p}, \mathsf{nn}_i(\mathsf{p}, S)) = \mathsf{d}_i(\mathsf{p}, S)$.[1] Let $\mathsf{NN}_i(\mathsf{p}, S) = \cup_{j=1}^{i} \{\mathsf{nn}_j(\mathsf{p}, S)\}$ be the set of i nearest neighbors of p in S. By definition, for $1 \leq i \leq |S|$, $|\mathsf{NN}_i(\mathsf{p}, S)| = i$. The following is an easy observation.

Observation 2.1 *For any fixed $Q \subseteq \mathsf{P}$ and integer $1 \leq i \leq |Q|$, the function $\mathsf{d}_i(\cdot, Q)$ is a 1-Lipschitz function of its argument, i.e., for any $\mathsf{p}, \mathsf{q} \in M$, $\mathsf{d}_i(\mathsf{p}, Q) \leq \mathsf{d}_i(\mathsf{q}, Q) + \mathsf{d}(\mathsf{p}, \mathsf{q})$.*

2.2 Problem Definitions

Problem 2.2 (Fault-tolerant k-center) *Let P be a set of n points in M, and let k and ℓ be two given integer parameters such that $1 \leq \ell \leq k \leq n$. For a subset $C \subseteq \mathsf{P}$, we define the cost function $\mu(\mathsf{P}, C)$ as,*

$$\mu(\mathsf{P}, C) = \max_{\mathsf{p} \in \mathsf{P}} \mathsf{d}_\ell(\mathsf{p}, C).$$

The fault-tolerant k-center problem, denoted $\mathrm{FTC}(\mathsf{P}, k, \ell)$, is to compute a set C^* with $|C^*| = k$ such that,

$$\mu(\mathsf{P}, C^*) = \min_{C \subseteq \mathsf{P}, |C| = k} \mu(\mathsf{P}, C).$$

For a given instance of $\mathrm{FTC}(\mathsf{P}, k, \ell)$, we call C^* the optimum solution and we let r_{opt} denote its cost, i.e. $r_{\mathsf{opt}} = \mu(\mathsf{P}, C^*)$. The classical ***k-center*** clustering problem on a point set P is $\mathrm{FTC}(\mathsf{P}, k, 1)$, and is referred to as the ***non-fault-tolerant*** k-center problem.

Problem 2.3 (Fault-tolerant k-median) *Let P be set of n points in M, and let k and ℓ be two given integer parameters such that $1 \leq \ell \leq k \leq n$. For a subset $C \subseteq \mathsf{P}$, we define the cost function $\mu(\mathsf{P}, C)$ as,*

$$\mu(\mathsf{P}, C) = \sum_{\mathsf{p} \in \mathsf{P}} \mathsf{d}_\ell(\mathsf{p}, C).$$

[1] In case of non unique distances, we use the standard technique of lexicographic ordering of the pairs $(\mathsf{d}(\mathsf{p}, \mathsf{p}_j), j)$ to ensure that the 1st, 2nd, $\ldots$, $|S|$th, nearest-neighbors of p are all unique.

The fault-tolerant k-median problem, denoted $\mathrm{FTM}(\mathsf{P}, k, \ell)$*, is to compute a set* C^* *with* $|C^*| = k$ *such that,*

$$\mu(\mathsf{P}, C^*) = \min_{C \subseteq \mathsf{P}, |C|=k} \mu(\mathsf{P}, C) \,.$$

For a given instance of $\mathrm{FTM}(\mathsf{P}, k, \ell)$, we call C^* the optimum solution and we let σ_{opt} denote its cost, i.e. $\sigma_{\mathsf{opt}} = \mu(\mathsf{P}, C^*)$. The classical ***$k$-median*** clustering problem on a point set P is $\mathrm{FTM}(\mathsf{P}, k, 1)$, and is referred to as the ***non-fault-tolerant*** k-median problem.

3 Algorithms

Our algorithms for both problems, $\mathrm{FTC}(\mathsf{P}, k, \ell)$ and $\mathrm{FTM}(\mathsf{P}, k, \ell)$, have the same structure. In the first step they run an approximation algorithm for the non-fault-tolerant version of the respective problem, for $m = \lfloor k/\ell \rfloor$ centers, and in the second step, the solution output by the first step is added to in a straightforward manner described below. Notice that for either fault-tolerant problem, any approximation algorithm for the non-fault-tolerant version can be used in the first step. In particular, we prove that if the chosen algorithm for this first step is a c-approximation algorithm for the non-fault-tolerant problem for m centers, then the set we output at the end of step two will be a $(1 + 2c)$-approximation (resp. $(1 + 4c)$-approximation) for the fault-tolerant k-center (resp. k-median) problem with k centers.

Natural choices to use for our non-fault-tolerant m-median algorithm include the local search algorithm of Arya *et al.* [2], which is favored for its combinatorial nature, and simplicity of implementation, or the recent algorithm by Li and Svensson [13], which facilitates a slight improvement in the approximation factor. For the algorithms of Arya *et al.* and that of Li and Svennson we refer the reader to the respective papers, as knowledge of these algorithms is not required for understanding our algorithm. We let $\mathcal{A}_{\mathsf{m}}(\mathsf{P}, m)$ denote the chosen approximation algorithm for m-median.

Similarly, we let $\mathcal{A}_{\mathsf{c}}(\mathsf{P}, m)$ denote the chosen approximation algorithm for non-fault-tolerant m-center. Perhaps the most natural choice for our m-center algorithm is the 2-approximation algorithm by Gonzalez [5]. In fact, in Section 4.2.1 we show that this particular choice leads to a simpler analysis than the general case, and produces a much tighter approximation ratio guarantee. Since knowledge of the algorithm of Gonzalez is needed for this analysis, we briefly review this algorithm below in Section 3.2.

3.1 Fault-tolerant algorithms

We now describe the algorithms for fault-tolerant k-center and fault-tolerant k-median, that is $\mathrm{FTC}(\mathsf{P}, k, \ell)$ and $\mathrm{FTM}(\mathsf{P}, k, \ell)$.

For the problem $\mathrm{FTC}(\mathsf{P}, k, \ell)$ (resp. $\mathrm{FTM}(\mathsf{P}, k, \ell)$) first run the algorithm $\mathcal{A}_{\mathsf{c}}(\mathsf{P}, m)$ (resp. $\mathcal{A}_{\mathsf{m}}(\mathsf{P}, m)$). Let $Q \subseteq \mathsf{P}$ denote the set of m centers output, and let $Q = \{\mathsf{q}_1, \ldots, \mathsf{q}_m\}$. Then the set of centers we output for our fault-tolerant solution is, $C = \bigcup_{i=1}^{m} \mathsf{NN}_\ell(\mathsf{q}_i, \mathsf{P})$. That is, we take the ℓ nearest neighbors of each point q_i in P, for $i = 1, \ldots, m$. We only use this set C in the analysis. If however C has less than k points, we can throw in $k - |C|$ additional points chosen arbitrarily from $\mathsf{P} \setminus C$, since adding additional centers can only decrease the cost of our solution.

Let $\mathcal{A}_{\mathsf{fc}}(\mathsf{P}, k, \ell)$ and $\mathcal{A}_{\mathsf{fm}}(\mathsf{P}, k, \ell)$ denote these algorithms for $\mathrm{FTC}(\mathsf{P}, k, \ell)$ and $\mathrm{FTM}(\mathsf{P}, k, \ell)$, respectively.

3.2 The algorithm of Gonzalez

We now describe the 2-approximation algorithm for the m-center problem, due to Gonzalez [5]. Gonzalez's algorithm builds a solution set C iteratively. To kick-start the iteration, we let $C = \{\mathsf{p}\}$ where $\mathsf{p} \in \mathsf{P}$ is an arbitrary point. Until m points have been accumulated, the algorithm repeatedly looks for the furthest point in P to the current set C, and adds the found point to C. More formally, at each step we compute $\arg\max_{\mathsf{q}\in\mathsf{P}} \mathsf{d}(\mathsf{q}, C)$, and add it to C.

This algorithm is not only simple from a conceptual stand-point, but also in regards to implementation and running time. Indeed, by just maintaining for each point in P, its current nearest center among C, the above algorithm can be implemented in $O(n)$ time per iteration, for a total time of $O(nm)$. As mentioned earlier, the result of Hochbaum and Shmoys [8] implies that the approximation factor for this algorithm for general metric spaces, is the best possible.

4 Results and Analysis

We now present our results and their proofs. Our first result, is that using a factor c-approximation algorithm for $\mathcal{A}_{\mathsf{m}}(\mathsf{P}, m)$ in the algorithm $\mathcal{A}_{\mathsf{fm}}(\mathsf{P}, k, \ell)$ gives a $(1 + 4c)$-approximation algorithm for the problem $\mathrm{FTM}(\mathsf{P}, k, \ell)$. The structure of the k-center problem allows us to use a nearly identical analysis except with one simplification, yielding an improved $(1 + 2c)$-approximation algorithm for the problem $\mathrm{FTC}(\mathsf{P}, k, \ell)$. Our second result, shows that if one uses the algorithm of Gonzalez [5] for the subroutine $\mathcal{A}_{\mathsf{c}}(\mathsf{P}, m)$, then one can guarantee a tighter approximation ratio of 4 (or 3 if $l|k$), as opposed to the 5 guaranteed by our first result.

4.1 Analysis for fault-tolerant k-median

Theorem 4.1 *For a given point set* P *in a metric space* M *with* $|\mathsf{P}| = n$*, the algorithm* $\mathcal{A}_{\mathsf{fm}}(\mathsf{P}, k, \ell)$ *achieves a* $(1 + 4c)$*-approximation to the optimal solution of* $\mathrm{FTM}(\mathsf{P}, k, \ell)$*, where* c *is the approximation guarantee of the subroutine* $\mathcal{A}_{\mathsf{m}}(\mathsf{P}, m)$*, where* $m = \lfloor k/\ell \rfloor$*.*

As a corollary we have,

Corollary 4.2 *There is a* 12*-approximation algorithm for the problem* $\mathrm{FTM}(\mathsf{P}, k, \ell)$*.*

Proof: We use the $(1 + \sqrt{3} + \varepsilon)$-approximation algorithm of Li and Svennson [13] with a small enough ε, for the subroutine $\mathcal{A}_{\mathsf{m}}(\mathsf{P}, m)$. The result follows by appealing to Theorem 4.1. ■

Proof of Theorem 4.1

We refer the reader to Section 2 for notation already introduced. We need some more notation. For a given instance of $\mathrm{FTM}(\mathsf{P}, k, \ell)$, let $C^* = \{w_1, w_2, \ldots, w_k\}$ be an optimal set of centers, and let σ_{opt} be its cost, i.e, $\sigma_{\mathsf{opt}} = \sum_{\mathsf{p}\in\mathsf{P}} \mathsf{d}_\ell(\mathsf{p}, C^*)$. Let $C = \{c_1, \ldots, c_k\}$ be the set of centers returned by $\mathcal{A}_{\mathsf{fm}}(\mathsf{P}, k, \ell)$, and σ_{alg} its cost.

Let $m = \lfloor k/\ell \rfloor$, and let σ_{med} denote the cost of the optimum m-median clustering on P, i.e., the optimum for the problem $\mathrm{FTM}(\mathsf{P}, m, 1)$. When $\mathcal{A}_{\mathsf{fm}}(\mathsf{P}, k, \ell)$ is run, it makes a subroutine call to $\mathcal{A}_{\mathsf{m}}(\mathsf{P}, m)$. Let $Q = \{\mathsf{q}_1, \ldots, \mathsf{q}_m\}$ be the set of centers returned by this subroutine call. We know that Q is a c-approximation to the optimal solution to $\mathrm{FTC}(\mathsf{P}, m, 1)$.

Notice that, C includes $\bigcup_{i=1}^m \mathsf{NN}_\ell(\mathsf{q}_i, \mathsf{P})$. We assume that the set C has exactly k points. As mentioned earlier, we only require that C includes $\bigcup_{i=1}^m \mathsf{NN}_\ell(\mathsf{q}_i, \mathsf{P})$ in our analysis, and if $|\bigcup_{i=1}^m \mathsf{NN}_\ell(\mathsf{q}_i, \mathsf{P})| < k$, we can always add additional points. This can only decrease the cost of clustering.

Proving the following two claims will immediately imply $\sigma_{\mathsf{alg}} \leq (1+4c)\sigma_{\mathsf{opt}}$.

Claim 4.3 *We have that,* $\sigma_{\mathsf{alg}} \leq \sigma_{\mathsf{opt}} + 2c\sigma_{\mathsf{med}}$.

Claim 4.4 *We have that,* $\sigma_{\mathsf{med}} \leq 2\sigma_{\mathsf{opt}}$.

Proof of Claim 4.3: Let $\mathsf{p} \in \mathsf{P}$, and let $\mathsf{q} = \mathsf{nn}_1(\mathsf{p}, Q)$. By Observation 2.1, $\mathsf{d}_\ell(\mathsf{p}, C) \leq \mathsf{d}(\mathsf{p}, \mathsf{q}) + \mathsf{d}_\ell(\mathsf{q}, C)$. As $\mathsf{NN}_\ell(\mathsf{q}, \mathsf{P}) \subseteq C$, we have that $\mathsf{d}_\ell(\mathsf{q}, \mathsf{P}) = \mathsf{d}_\ell(\mathsf{q}, C)$. Again by Observation 2.1, $\mathsf{d}_\ell(\mathsf{q}, \mathsf{P}) \leq \mathsf{d}(\mathsf{q}, \mathsf{p}) + \mathsf{d}_\ell(\mathsf{p}, \mathsf{P})$. Combining the two inequalities gives, $\mathsf{d}_\ell(\mathsf{p}, C) \leq 2\mathsf{d}(\mathsf{p}, \mathsf{q}) + \mathsf{d}_\ell(\mathsf{p}, \mathsf{P}) = 2\mathsf{d}_1(\mathsf{p}, Q) + \mathsf{d}_\ell(\mathsf{p}, \mathsf{P})$. Thus,

$$\sigma_{\mathsf{alg}} = \sum_{\mathsf{p} \in \mathsf{P}} \mathsf{d}_\ell(\mathsf{p}, C) \leq \sum_{\mathsf{p} \in \mathsf{P}} (2\mathsf{d}_1(\mathsf{p}, Q) + \mathsf{d}_\ell(\mathsf{p}, \mathsf{P})) \leq 2c\sigma_{\mathsf{med}} + \sigma_{\mathsf{opt}}, \tag{1}$$

as Q is a c-approximate m-median solution, $\mathsf{d}_\ell(\mathsf{p}, \mathsf{P}) \leq \mathsf{d}_\ell(\mathsf{p}, C^*)$, and $\sigma_{\mathsf{opt}} = \sum_{\mathsf{p} \in \mathsf{P}} \mathsf{d}_\ell(\mathsf{p}, C^*)$. ■

The following is required to prove Claim 4.4, but is interesting in its own right.

Lemma 4.5 *Let M be any metric space. Let $\mathsf{X} \subseteq M$ with $|\mathsf{X}| = t$. Then for any integer $1 \leq h \leq t$, and any finite set $\mathsf{Y} \subseteq M$, there exists a subset $S \subseteq \mathsf{Y}$, such that (A) $|S| \leq t/h$, and, (B) $\forall \mathsf{y} \in \mathsf{Y}$, $\mathsf{d}_1(\mathsf{y}, S) \leq 2\mathsf{d}_h(\mathsf{y}, \mathsf{X})$.*

Proof: We give an algorithm to construct such a subset $S \subseteq \mathsf{Y}$. This subset is constructed by iteratively scooping out the points of the minimum radius ball containing h points from X, adding the center to S, and repeating. Formally, let $W_0 = \emptyset$, and for $i = 1, \ldots, \lfloor t/h \rfloor$, define iteratively, $\mathsf{X}_i = \mathsf{X} \setminus \left(\bigcup_{j=0}^{i-1} W_j \right)$, $\mathsf{y}_i = \arg\min_{\mathsf{v} \in \mathsf{Y}} \mathsf{d}_h(\mathsf{v}, \mathsf{X}_i)$, and, $W_i = \mathsf{NN}_h(\mathsf{y}_i, \mathsf{X}_i)$. We prove that $S = \bigcup_{i=1}^{\lfloor t/h \rfloor} \{\mathsf{y}_i\}$, is the desired subset of points.

First, clearly $|S| \leq t/h$. Let $\mathsf{y} \in \mathsf{Y}$, and let $\mathsf{b} = \mathsf{ball}(\mathsf{y}, x)$, where $x = \mathsf{d}_h(\mathsf{y}, \mathsf{X})$. Let W_i be the first subset, i.e. the one with smallest index i, such that there exists some point $\mathsf{w} \in \mathsf{b} \cap W_i$. Such a point must exist, since fewer than h points are in $\mathsf{X} \setminus \left(\bigcup_{j=1}^{\lfloor t/h \rfloor} W_j \right)$, while $|\mathsf{b} \cap \mathsf{X}| \geq h$. Clearly $\mathsf{b} \cap \mathsf{X} \subseteq \mathsf{X}_i$, as i is the minimum index such that $\mathsf{b} \cap W_i \neq \emptyset$. As such we have, $\mathsf{d}_h(\mathsf{y}, \mathsf{X}) = \mathsf{d}_h(\mathsf{y}, \mathsf{X}_i)$. Let $r_i = \mathsf{d}_h(\mathsf{y}_i, \mathsf{X}_i)$, be the radius of the ball that scooped out W_i. Clearly $r_i \leq x$, as

$$x = \mathsf{d}_h(\mathsf{y}, \mathsf{X}) = \mathsf{d}_h(\mathsf{y}, \mathsf{X}_i) \geq r_i = \arg\min_{\mathsf{v} \in \mathsf{Y}} \mathsf{d}_h(\mathsf{v}, \mathsf{X}_i) \,.$$

Now, since $\mathsf{w} \in \mathsf{b} \cap W_i$, $\mathsf{d}(\mathsf{y}_i, \mathsf{w}) \leq r_i = \mathsf{d}_h(\mathsf{y}_i, \mathsf{X}_i)$. By the triangle inequality,

$$\mathsf{d}_1(\mathsf{y}, S) \leq \mathsf{d}(\mathsf{y}, \mathsf{y}_i) \leq \mathsf{d}(\mathsf{y}, \mathsf{w}) + \mathsf{d}(\mathsf{w}, \mathsf{y}_i) \leq x + r_i \leq 2x = 2\mathsf{d}_h(\mathsf{y}, \mathsf{X}) \,.$$

■

Proof of Claim 4.4: We use Lemma 4.5 with $\mathsf{Y} = \mathsf{P}$, $\mathsf{X} = C^*$, $t = |C^*| = k$ and $h = \ell$. Let S be the subset of P guaranteed by Lemma 4.5. Now $|S| \leq k/\ell$, and as such $|S| \leq m$. We have,

$$\sigma_{\mathsf{med}} \leq \sum_{\mathsf{p} \in \mathsf{P}} \mathsf{d}_1(\mathsf{p}, S) \leq \sum_{\mathsf{p} \in \mathsf{P}} 2\mathsf{d}_\ell(\mathsf{p}, C^*) = 2\sigma_{\mathsf{opt}}. \tag{2}$$

The first inequality follows since σ_{med} is the cost of the optimum m-median clustering of P, while $\sum_{\mathsf{p} \in \mathsf{P}} \mathsf{d}_1(\mathsf{p}, S)$ is the cost of a $|S|$-median clustering of P by the set of centers $S \subseteq \mathsf{P}$ with $|S| \leq m$. The second inequality follows from Lemma 4.5. ■

This concludes the proof of Theorem 4.1.

4.2 Analysis for fault-tolerant k-center

We now present the analogues result to Theorem 4.1 for fault-tolerant k-center. By following the proof nearly verbatim from the previous section one sees that similar to $\mathcal{A}_{\mathsf{fm}}(\mathsf{P}, k, \ell)$, $\mathcal{A}_{\mathsf{fc}}(\mathsf{P}, k, \ell)$ also provides a $(1 + 4c)$-approximation. However, in this case we will actually get a $(1 + 2c)$-approximation, since now an improved and simpler version of Claim 4.3 holds.

As a quick note on notation, here r_{alg}, r_{opt}, and r_{cen} will play the analogues role for center as σ_{alg}, σ_{opt}, and σ_{med} played for median.

Claim 4.6 *We have that,* $r_{\mathsf{alg}} \leq r_{\mathsf{opt}} + 2cr_{\mathsf{cen}}$.

Proof of Claim 4.6: Let $\mathsf{p} \in \mathsf{P}$, and let $\mathsf{q} = \mathsf{nn}_1(\mathsf{p}, Q)$. By Observation 2.1, $\mathsf{d}_\ell(\mathsf{p}, C) \leq \mathsf{d}(\mathsf{p}, \mathsf{q}) + \mathsf{d}_\ell(\mathsf{q}, C) = \mathsf{d}(\mathsf{p}, \mathsf{q}) + \mathsf{d}_\ell(\mathsf{q}, \mathsf{P})$, where the equality follows since $\mathsf{NN}_\ell(\mathsf{q}, \mathsf{P}) \subseteq C$. Thus,

$$\begin{aligned} r_{\mathsf{alg}} &= \max_{\mathsf{p} \in \mathsf{P}} \mathsf{d}_\ell(\mathsf{p}, C) \leq \max_{\mathsf{p} \in \mathsf{P}} (\mathsf{d}_1(\mathsf{p}, Q) + \mathsf{d}_\ell(\mathsf{q}, \mathsf{P})) \\ &\leq 2cr_{\mathsf{cen}} + r_{\mathsf{opt}}, \end{aligned} \tag{3}$$

as Q is a c-approximate m-center solution, $\mathsf{d}_\ell(\mathsf{q}, \mathsf{P}) \leq \mathsf{d}_\ell(\mathsf{q}, C^*)$, and $r_{\mathsf{opt}} = \max_{\mathsf{p} \in \mathsf{P}} \mathsf{d}_\ell(\mathsf{p}, C^*)$. ■

Theorem 4.7 *For a given point set* P *in a metric space* M *with* $|\mathsf{P}| = n$*, the algorithm* $\mathcal{A}_{\mathsf{fc}}(\mathsf{P}, k, \ell)$ *achieves a* $(1 + 2c)$*-approximation to the optimal solution of* FTC(P, k, ℓ)*, where* c *is the approximation guarantee of the subroutine* $\mathcal{A}_{\mathsf{c}}(\mathsf{P}, m)$*, where* $m = \lfloor k/\ell \rfloor$.

Proof: As stated above, the proof of this theorem is very similar to the proof of Theorem 4.1. In fact, we can repeat the proof of Theorem 4.1 almost word for word, except that we need to replace the sum function $\sum_{\mathsf{p} \in \mathsf{P}}$ by the max function, $\max_{\mathsf{p} \in \mathsf{P}}$. More specifically, this needs to be done for Eq. (2) in the proof of Claim 4.4, and to replace Eq. (1) from Claim 4.3 we instead use the improved Eq. (3) from Claim 4.6. As the proof can be reconstructed step-by-step from the detailed proof of Theorem 4.1 by making these modifications, we omit it for the sake of brevity. ■

4.2.1 A tighter analysis when using Gonzalez's algorithm as a subroutine

If we use a 2-approximation algorithm for the subroutine $\mathcal{A}_{\mathsf{c}}(\mathsf{P}, m)$, Theorem 4.7 implies that $\mathcal{A}_{\mathsf{fc}}(\mathsf{P}, k, \ell)$ is a 9-approximation algorithm. Here we present a tighter analysis for the case when we use the 2-approximation algorithm of Gonzalez [5] (see also Section 3.2) for the subroutine $\mathcal{A}_{\mathsf{c}}(\mathsf{P}, m)$.

See Section 2 for definitions and notation introduced previously. Some more notation is needed. Let $C^* = \{w_1, w_2, \ldots, w_k\}$ be an optimal set of centers. Its cost, r_{opt}, is $\max_{\mathsf{p} \in \mathsf{P}} \mathsf{d}_\ell(\mathsf{p}, C^*)$. Let $C = \{c_1, \ldots, c_k\}$ be the set of centers returned by $\mathcal{A}_{\mathsf{fc}}(\mathsf{P}, k, \ell)$, and let r_{alg} be its cost.

Let $m = \lfloor k/\ell \rfloor$, where for now we assume $\ell | k$, i.e, $m = k/\ell$. As we show later, this assumption can be removed. When $\mathcal{A}_{\mathsf{fc}}(\mathsf{P}, k, \ell)$ is run, it makes a subroutine call to $\mathcal{A}_{\mathsf{c}}(\mathsf{P}, m)$. As mentioned, in this section we require this subroutine to be the algorithm of Gonzalez [5]. Let $Q = \{\mathsf{q}_1, \ldots, \mathsf{q}_m\}$ be the set of centers returned by this subroutine call. Additionally, let $r_i = \mathsf{d}(\mathsf{q}_i, Q_{i-1})$ for $2 \leq i \leq m$, where $Q_{i-1} = \{\mathsf{q}_1, \ldots, \mathsf{q}_{i-1}\}$. We assume $m > 1$, as the $m = 1$ case is easier.

The following is easy to see, and is used in the correctness proof for the algorithm of Gonzalez. See [6] for an exposition.

Lemma 4.8 *For* $i \neq j$*,* $\mathsf{d}(\mathsf{q}_i, \mathsf{q}_j) \geq r_m$.

Lemma 4.9 *For any* q_i*,* $\mathsf{NN}_\ell(\mathsf{q}_i, C^*) \subseteq \mathsf{ball}(\mathsf{q}_i, r_{\mathsf{opt}})$ *and* $\mathsf{NN}_\ell(\mathsf{q}_i, \mathsf{P}) \subseteq \mathsf{ball}(\mathsf{q}_i, r_{\mathsf{opt}})$.

Proof: The first claim follows since $\mathsf{q}_i \in \mathsf{P}$ and so $\mathsf{d}_\ell(\mathsf{q}_i, C^*) \le r_{\mathsf{opt}}$. As $C^* \subseteq \mathsf{P}$, the second claim follows. ■

Lemma 4.10 *We have that,* $r_{\mathsf{alg}} \le r_m + r_{\mathsf{opt}}$.

Proof: As in Gonzalez's algorithm, we have $r_m = \max_{\mathsf{p}\in\mathsf{P}} \mathsf{d}(\mathsf{p}, Q_{m-1})$, and so $\mathsf{d}(\mathsf{p}, Q) \le r_m$ for any $\mathsf{p} \in \mathsf{P}$. Consider any point $\mathsf{p} \in \mathsf{P}$, and let $\mathsf{q} = \mathsf{nn}_1(\mathsf{p}, Q)$. By how $\mathcal{A}_{\mathsf{fc}}(\mathsf{P}, k, \ell)$ is defined, $\mathsf{NN}_\ell(\mathsf{q}, \mathsf{P}) \subseteq C$, and so $\mathsf{d}_\ell(\mathsf{q}, C) = \mathsf{d}_\ell(\mathsf{q}, \mathsf{P}) \le \mathsf{d}_\ell(\mathsf{q}, C^*) \le r_{\mathsf{opt}}$. By Observation 2.1 we have, $\mathsf{d}_\ell(\mathsf{p}, C) \le \mathsf{d}(\mathsf{p}, \mathsf{q}) + \mathsf{d}_\ell(\mathsf{q}, C) \le r_m + r_{\mathsf{opt}}$. ■

Lemma 4.11 *If* $r_{\mathsf{alg}} > 3r_{\mathsf{opt}}$*, then for any* $1 \le i \ne j \le m$*,* $\mathsf{ball}(\mathsf{q}_i, r_{\mathsf{opt}})$ *and* $\mathsf{ball}(\mathsf{q}_j, r_{\mathsf{opt}})$ *are disjoint and each contains at least* ℓ *centers from* C^**.*

Proof: Let q_i and q_j be any two distinct centers in Q. By Lemma 4.8 and Lemma 4.10, $\mathsf{d}(\mathsf{q}_i, \mathsf{q}_j) \ge r_m \ge r_{\mathsf{alg}} - r_{\mathsf{opt}} > 2r_{\mathsf{opt}}$, which implies that, $\mathsf{ball}(\mathsf{q}_i, r_{\mathsf{opt}}) \cap \mathsf{ball}(\mathsf{q}_j, r_{\mathsf{opt}}) = \emptyset$. Each ball contains ℓ centers from C^* by Lemma 4.9. ■

Lemma 4.12 *We have that,* $r_{\mathsf{alg}} \le 3r_{\mathsf{opt}}$.

Proof: Suppose otherwise that $r_{\mathsf{alg}} > 3r_{\mathsf{opt}}$. By Lemma 4.11, for $i = 1, \ldots, m$, $|\mathsf{ball}(\mathsf{q}_i, r_{\mathsf{opt}}) \cap C^*| \ge \ell$, and for $1 \le i < j \le m$, $\mathsf{ball}(\mathsf{q}_i, r_{\mathsf{opt}}) \cap \mathsf{ball}(\mathsf{q}_j, r_{\mathsf{opt}}) = \emptyset$. Assign all points in $C^* \cap \mathsf{ball}(\mathsf{q}_i, r_{\mathsf{opt}})$ to q_i. Notice, q_i is the unique point from Q within distance r_{opt} for any point assigned to it. Now $|Q| = m = k/\ell$, and each point in Q gets at least ℓ points of C^* assigned to it uniquely. As such, there are at least $m\ell = k$ points of C^* assigned to some point of Q. Since $|C^*| = k$, it follows that each center in C^* gets assigned to a center in Q within distance r_{opt}. For $\mathsf{p} \in \mathsf{P}$, let v be its closest center in C^*. Let q be v's center from Q in distance $\le r_{\mathsf{opt}}$. We have $\mathsf{d}(\mathsf{p}, \mathsf{q}) \le \mathsf{d}(\mathsf{p}, \mathsf{v}) + \mathsf{d}(\mathsf{v}, \mathsf{q}) \le r_{\mathsf{opt}} + r_{\mathsf{opt}} = 2r_{\mathsf{opt}}$, by the triangle inequality. As $\mathsf{NN}_\ell(\mathsf{q}, \mathsf{P}) \subseteq C$, we have that $\mathsf{d}_\ell(\mathsf{q}, C) = \mathsf{d}_\ell(\mathsf{q}, \mathsf{P}) \le \mathsf{d}_\ell(\mathsf{q}, C^*) \le r_{\mathsf{opt}}$. By Observation 2.1, we have that, $\mathsf{d}_\ell(\mathsf{p}, C) \le \mathsf{d}_\ell(\mathsf{q}, C) + \mathsf{d}(\mathsf{p}, \mathsf{q}) \le r_{\mathsf{opt}} + 2r_{\mathsf{opt}} = 3r_{\mathsf{opt}}$. This implies $r_{\mathsf{alg}} \le 3r_{\mathsf{opt}}$, a contradiction. ■

Theorem 4.13 *For a given instance of* FTC(P, k, ℓ)*, when using the algorithm of Gonzalez [5] for the subroutine* $\mathcal{A}_{\mathsf{c}}(\mathsf{P}, m)$*, the algorithm* $\mathcal{A}_{\mathsf{fc}}(\mathsf{P}, k, \ell)$ *achieves a* 4*-approximation to the optimal solution to* FTC(P, k, ℓ)*, and a 3-approximation when* $\ell | k$*.*

Proof: The $\ell | k$ case follows from Lemma 4.12. If ℓ does not divide k, the proof of Lemma 4.12 needs to be changed as follows. Suppose, $k = \ell * m + r$ for some integer $0 < r < \ell$. Let $k' = \ell * m$. As in the proof of Lemma 4.12, it follows from Lemma 4.11, that if $r_{\mathsf{alg}} > 3r_{\mathsf{opt}}$, then at least k' centers from C^* will be within distance at most r_{opt} to a center in Q. Therefore, there are at most $k - k' = r$ centers from C^*, that are not within r_{opt} to some point in Q. However, each such center needs $\ell > r$ centers from C^*, to be within distance r_{opt}, and so each such center must be within distance r_{opt} from one of the centers of C^* that is near a center in Q, i.e. within distance r_{opt} to some center in Q. Hence, by the triangle inequality, each center in C^*, has a center of Q within distance at most $2r_{\mathsf{opt}}$. Repeating the argument of Lemma 4.12, with this different upper bound, we get that $r_{\mathsf{alg}} \le 4r_{\mathsf{opt}}$. ■

5 Conclusions

In this paper we investigated fault-tolerant variants of the k-center and k-median clustering problems. Our algorithm achieves a $(1+2c)$-approximation (resp. $(1+4c)$-approximation) factor, where

c is the approximation factor for the non-fault-tolerant m-center (resp. m-median) algorithm that we use as a subroutine. Using a better analysis for the case of fault-tolerant k-center, when we use Gonzalez's algorithm as a subroutine, we showed that our algorithm has a tighter approximation ratio of 4. For fault-tolerant k-median, we get a $(5 + 4\sqrt{3} + \varepsilon) \approx 12$-approximation algorithm, by using the recent algorithm of Li and Svensson as a subroutine [13]. We can see several questions for future research.

- The best known approximation factor for the fault-tolerant k-center problem is 2 by Chaudhuri *et al.* [3] and Khuller *et al.* [10]. Their techniques are based on the work of Hochbaum and Shmoys [8] and Krumke [12]. Our algorithm, which leads to a 4-approximation for fault-tolerant k-center is based on the 2-approximation to k-center by Gonzalez [5]. Can the algorithm or its analysis be improved to get a factor 2-approximation? Also, can we deal with the second variant of fault-tolerant k-center in the work of Khuller *et al.*– which also happens to be the version considered by Krumke and Chaudhuri *et al.*?

- The fault-tolerant k-median variant that we investigate, is very different from the work of Swamy and Shmoys [14], but their techniques are more technically involved. As we show, we reduce the fault-tolerant version to the non-fault-tolerant version for a smaller number of centers. An important question that arises is the following: Can the version considered by Swamy and Shmoys be reduced to the non-fault-tolerant version, or some variant thereof, i.e., can we use some simpler problem as an oracle to get a fault-tolerant k-median algorithm, for the version of Swamy and Shmoys?

Acknowledgements

We would like to thank Sariel Har-Peled for useful discussions, and in particular for the discussion that led us to think about this problem.

References

[1] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-median and related problems. In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 106–113, 1998.

[2] V. Arya, N. Garg, R. Khandekar, K. Munagala, and V. Pandit. Local search heuristic for k-median and facility location problems. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, pages 21–29, 2001.

[3] S. Chaudhuri, N. Garg, and R. Ravi. The p-neighbor k-center problem. *Inf. Proc. Lett.*, 65(3):131–134, 1998.

[4] T. Feder and D. H. Greene. Optimal algorithms for approximate clustering. In *Proc. 20th Annu. ACM Sympos. Theory Comput.*, pages 434–444, 1988.

[5] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[6] S. Har-Peled. *Geometric Approximation Algorithms.* Amer. Math. Soc., 2011.

[7] S. Har-Peled and S. Mazumdar. Coresets for k-means and k-median clustering and their applications. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 291–300, 2004.

[8] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-cener problem. *Math. of Oper. res.*, 10(2):180–184, 1985.

[9] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proc. 34th Annu. ACM Sympos. Theory Comput.*, pages 731–740, 2002.

[10] S. Khuller, R. Pless, and Y. J. Sussmann. Fault tolerant k-center problems. *Theor. Comput. Sci.*, 242(1-2):237–245, 2000.

[11] S. G. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the Euclidean κ-median problem. In *Proc. 7th Annu. European Sympos. Algorithms*, pages 378–389, 1999.

[12] S. O. Krumke. On a generalization of the p-center problem. *Inf. Proc. Lett.*, 56:67–71, 1995.

[13] S. Li and O. Svensson. Approximating k-median via pseudo-approximation. In *Proc. 45th Annu. ACM Sympos. Theory Comput.*, page to appear, 2013.

[14] C. Swamy and D. B. Shmoys. Fault-tolerant facility location. In *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, pages 735–736, 2003.