Distributed Partial Clustering

SUDIPTO GUHA, University of Pennsylvania, United States
YI LI, Nanyang Technological University, Singapore
QIN ZHANG, Indiana University Bloomington, United States

Recent years have witnessed an increasing popularity of algorithm design for distributed data, largely due to the fact that massive datasets are often collected and stored in different locations. In the distributed setting, communication typically dominates the query processing time. Thus, it becomes crucial to design communication efficient algorithms for queries on distributed data. Simultaneously, it has been widely recognized that partial optimizations, where we are allowed to disregard a small part of the data, provide us significantly better solutions. The motivation for disregarded points often arise from noise and other phenomena that are pervasive in large data scenarios.

In this paper we focus on partial clustering problems, k-center, k-median and k-means objectives, in the distributed model, and provide algorithms with communication sublinear of the input size. As a consequence we develop the first algorithms for the partial k-median and means objectives that run in subquadratic running time. We also initiate the study of distributed algorithms for clustering uncertain data, where each data point can possibly fall into multiple locations under certain probability distribution.

CCS Concepts: • **Theory of computation → Facility location and clustering**; • **Computing methodologies → Distributed algorithms**.

Additional Key Words and Phrases: clustering, distributed computing, k-means, k-medians, k-centers

ACM Reference Format:
Sudipto Guha, Yi Li, and Qin Zhang. 2019. Distributed Partial Clustering. *ACM Trans. Parallel Comput.* 1, 1, Article 1 (January 2019), 20 pages. https://doi.org/10.1145/3322808

1 INTRODUCTION

The challenge of optimization over large quantities of data has brought communication efficient *distributed* algorithms to the fore. From the perspective of optimization, it has also become clear that *partial optimizations*, where we are allowed to disregard a small part of the input, enable us to provide significantly better optimization solutions compared with those which are forced to account for the whole input [4, 19]. While several algorithms for distributed clustering have been proposed, partial optimizations for clustering problems, introduced by Charikar et al. [4], have not received as much attention. While the results of Chen [6] improve the approximation ratios, the running time of the k-median and k-means versions have not been improved and the (at least) quadratic running times have remained as a barrier.

Sudipto Guha was supported in part by NSF award 1546151. Qin Zhang was supported in part by NSF CCF-1525024 and IIS-1633215.
Authors' addresses: Sudipto Guha, University of Pennsylvania, Philadelphia, PA, 19104, United States, sudipto@cis.upenn.edu; Yi Li, Nanyang Technological University, Singapore, 637371, yili@ntu.edu.sg; Qin Zhang, Indiana University Bloomington, Bloomington, IN, 47401, United States, qzhangcs@indiana.edu.

1539-9087/2019/1-ART1 $15.00
https://doi.org/10.1145/3322808

In this paper we study partial clustering under the standard (k, t)-median/means/center objective functions, where k is the number of centers we can use and t is the maximum number of points we can ignore. See Definition 1.1 below for a formal definition. In the distributed setting, let s denote the number of sites. The (k, t)-center problem has recently been studied by Malkomes et al. [21], who gave a 2-round $O(1)$-approximation algorithm with $\tilde{O}(sk + st)$ bits of communication[1], assuming that each point can be encoded in $O(1)$ bits. In fact, we observe that results from streaming algorithms [15] can in fact provide us 1-round $O(1)$-approximation algorithms with $\tilde{O}(sk + st)$ bits of communication for (k, t)-center, (k, t)-median, and (k, t)-means. However, in many scenarios of interest, we have $n > t \gg k$ and $t \gg s$. Thus the st term generates a significant communication burden. In this paper we reduce $\tilde{O}(st)$ to $\tilde{O}(t)$ for the (k, t)-center problem, as well as for (k, t)-median and (k, t)-means problems and unify their treatment. We also provide the first subquadratic algorithms for median and means version of this problem.

Large data sets often have erroneous values. Stochastic optimization has recently attracted a lot of attention in the field of databases, and has substantiated as a subfield called 'uncertain/probabilistic databases' (see, e.g., [22]). For the clustering problem, a method of choice is to first model the underlying uncertainty and then cluster the uncertain data. Clustering under uncertainty has been studied in centralized models [8, 16], but the algorithms proposed therein do not consider communication costs. Note that it typically requires significantly more communication to communicate a distribution (for an uncertain point) than a deterministic point, and thus black box adaptations of centralized algorithms do not work well in the distributed setting. In this paper we propose communication-efficient distributed algorithms for handling *both* data uncertainty and partial clustering. To the best of our knowledge neither distributed clustering of uncertain data nor partial clustering of uncertain data has been studied. We note that both problems are fairly natural, and likely to be increasingly useful as distributed cloud computing becomes commonplace.

Models and Problems. We study the clustering problems in the *coordinator* model, which is a popular model in the study of multiparty communication, see, e.g. [23]. In the coordinator model, there are s sites and one central coordinator, who are connected by a star communication network with the coordinator at the center. However, direct communication between sites can be simulated by routing via the coordinator, which at most doubles the communication. The computation is in terms of rounds. At each round, the coordinator sends a message (could be an empty message) to each site and every site sends a message (could be an empty message) back to the coordinator. The coordinator outputs the answer at the end[2]. The input $\mathbb{A}$ is partitioned into $(\mathbb{A}_1, \ldots, \mathbb{A}_s)$ among the s sites. Let $n_i = |\mathbb{A}_i|$, and $n = |\mathbb{A}| = \sum_{i\in[s]} n_i$ be the total input size.

We will consider clustering over a graph with n nodes and an oracle distance function $d(\cdot, \cdot)$. An easy example of such is points in Euclidean space. More complicated examples correspond to documents and images represented in a feature space and the distance function is computed via a kernel. We now give the definitions of (k, t)-center/median/means.

Definition 1.1 ((k, t)-center,median,means). Let $\mathbb{A}$ be a set of n points and k, t are integer parameters ($1 \le k \le n$, $0 \le t \le n$). In the (k, t)-median problem we want to compute

$$\min_{K,\mathbb{O}\subseteq\mathbb{A}} \sum_{p\in\mathbb{A}\setminus\mathbb{O}} d(p, K) \quad \text{subject to} \quad |K| \le k \ \text{ and } \ |\mathbb{O}| \le t,$$

[1] In this paper, we hide poly $\log n$ factors in the $\tilde{O}$ notation, even when the function in $O(\cdot)$ does not depend on n. Note that this is different from the typical usage that $\tilde{O}(f)$ hides the factors of poly$(\log f)$.

[2] We note that any algorithm in the coordinator model can also be implemented in parallel computation models such as MapReduce [10] – we can just pick an arbitrary machine as the coordinator.

where $d(p, K) = \min_{x \in K} d(p, x)$. We typically call K the *centers* and $\mathbb{O}$ the *outliers*. In the (k, t)-means and the (k, t)-center problem we replace the objective function $\sum_{p \in \mathbb{A} \setminus \mathbb{O}} d(p, K)$ with $\sum_{p \in \mathbb{A} \setminus \mathbb{O}} d^2(p, K)$ and $\max_{p \in \mathbb{A} \setminus \mathbb{O}} d(p, K)$ respectively.

In the definition above, we assume that centers are chosen from the input points. In the Euclidean space, compared with the setting of unconstrained centers, such restriction will only affect the approximation by a factor of 2.

For the uncertain data, we follow the assigned clustering introduced in [8]. Let $\mathcal{P}$ be a finite set of points in a metric space. There are n input nodes $\mathbb{A}$, where node j follows distribution $\mathcal{D}_j$ over $\mathcal{P}$. Each site i knows the distributions $\mathcal{D}_j$ associated with the nodes $j \in \mathbb{A}_i$.

Definition 1.2 (Clustering Uncertain Data). In clustering with uncertainty, the output is a subset $K \subseteq \mathcal{P}$ of size k (centers), a subset $\mathbb{O} \subseteq \mathcal{P}$ of size at most t (ignored points), as well as a mapping $\pi : \mathbb{A} \to K$. In every realization $\sigma : \mathbb{A} \to \mathcal{P}$ of the values of the input nodes, node $j \in \mathbb{A}$ (now realized as $\sigma(j) \in \mathcal{P}$) is assigned to the same center $\pi(j) \in K$. In uncertain (k, t)-median, the goal is to minimize the expected cost

$$\mathop{\mathbb{E}}_{\sigma \sim \prod_{j \in \mathbb{A}} \mathcal{D}_j} \left[\sum_{j \in \mathbb{A} \setminus \mathbb{O}} d(\sigma(j), \pi(j)) \right] = \sum_{j \in \mathbb{A} \setminus \mathbb{O}} \mathop{\mathbb{E}}_{\sigma \sim \mathcal{D}_j} \left[d(\sigma(j), \pi(j)) \right]. \tag{1}$$

The definition of uncertain (k, t)-means is basically the same as uncertain (k, t)-median, except that we replace the objective function (1) with $\sum_{j \in \mathbb{A} \setminus \mathbb{O}} \mathbb{E}_{\sigma \sim \mathcal{D}_j} \left[d^2(\sigma(j), \pi(j)) \right]$. For uncertain (k, t)-center, we have two objectives:

$$\max_{j \in \mathbb{A} \setminus \mathbb{O}} \left(\mathop{\mathbb{E}}_{\sigma \sim \mathcal{D}_j} \left[d(\sigma(j), \pi(j)) \right] \right) \tag{2}$$

$$\mathop{\mathbb{E}}_{\sigma \sim \prod_j \mathcal{D}_j} \left[\max_{j \in \mathbb{A} \setminus \mathbb{O}} d(\sigma(j), \pi(j)) \right] \tag{3}$$

Note that these two objectives are *not* equivalent, since $\mathbb{E}$ and max do not commute in Equation (3) and we cannot equate it to (2). Equation (2) is in the same spirit as Equation (1), and corresponds to a *per point* measurement. We term this problem as uncertain (k, t)-center-pp. Equation (3) corresponds to a more *global* measurement and we term this problem as uncertain (k, t)-center-g. This version was considered in [8, 16].

Our Results. We present all our results in Table 1, which include both one-round and two-round algorithms. In the column of 'Local Time', the first is the local computation time of all sites, and the second is the local computation time at the coordinator. Observe that the total running time is $\tilde{O}(\sum_i n_i^2)$, which becomes $\tilde{O}(n^2/s)$ if the partitions are balanced. This shows that we can reduce the running time by distributing the clustering across many sites.

In particular we have obtained the following algorithms which finish in two-rounds in the coordinator model. We say a solution is an (α, β)-approximation if it is a solution of cost αC while excluding βt points, where C is the optimum cost for excluding t points. In addition, we denote by B the number of bits required to encode a point.

(1) We give $(O(1), 1)$-approximation algorithms with $\tilde{O}((sk + t)B)$ communication for the (k, t)-median (Section 3) and the (k, t)-center (Theorem 4.3) problems. The lower bounds in [5] for the $t = 0$ case indicate that these communication costs are tight, if we want to output all the outliers (which our algorithms do), up to logarithmic factors. We also give an $(O(1 + 1/\epsilon), 1 + \epsilon)$-approximation algorithm with $\tilde{O}((sk + t)B)$ communication for the (k, t)-median (with better running time) and the (k, t)-means (Theorem 3.1) problems.

Objective	Approx.	Centers	Ignored	Rounds	Total Comm.	Local Time
median	$O(1)$	k	t	1	$\tilde{O}((sk+st)B)$	$\tilde{O}(n_i^2)$, $\tilde{O}(k^2s^3t^5)$
				2	$\tilde{O}((sk+t)B)$	$\tilde{O}(n_i^2)$, $\tilde{O}(k^2t^2(sk+t)^3)$
			$(2+\delta)t$	2	$\tilde{O}(s/\delta + skB)$	$\tilde{O}(n_i^2)$, $\tilde{O}(s^2k^7)$
means/ median	$O(1+1/\epsilon)$	k, $(1+\epsilon)t$ or $(1+\epsilon)k$, t		1	$\tilde{O}((sk+st)B)$	$\tilde{O}(n_i^2)$, $\tilde{O}((sk+st)^2)$
				2	$\tilde{O}((sk+t)B)$	$\tilde{O}(n_i^2)$, $\tilde{O}((sk+t)^2)$
		k $(1+\epsilon)k$	$(2+\epsilon+\delta)t$ $(2+\delta)t$	2	$\tilde{O}(s/\delta + skB)$	$\tilde{O}(n_i^2)$, $\tilde{O}((sk)^2)$
center	$O(1)$	k	t	1	$\tilde{O}((sk+st)B)$	$\tilde{O}((k+t)n_i)$, $\tilde{O}((sk+st)^2)$
				2	$\tilde{O}((sk+t)B)$	$\tilde{O}((k+t)n_i)$, $\tilde{O}((sk+t)^2)$
			$(2+\delta)t$	2	$\tilde{O}(s/\delta + skB)$	$\tilde{O}(n_i^2)$, $\tilde{O}((sk)^2)$
uncertain median/ means/ center-pp	as in the regular case above					regular case runtime $+O(n_iT)$, unchanged
center-g	$O(1+1/\epsilon)$	k	$(1+\epsilon)t$	2	$\tilde{O}(skB + tI + s\log\Delta)$	$\tilde{O}(n_i^2\log\Delta)$, $\tilde{O}((sk+t)^2)$
	$O(1)$		t	1	$\tilde{O}(s(kB+tI)\log\Delta)$	$\tilde{O}((k+t)n_i\log\Delta)$, $\tilde{O}(s^2(k+t)^2)$

Table 1. Our results. T denotes the runtime to compute 1-median/mean of a node distribution, I is the information encoding a node in the uncertain data case, B the information encoding a point and Δ the ratio between the maximum pairwise distance and the minimum pairwise distance in the dataset. The algorithms for $k, (1+\epsilon)t$ bicriteria are randomized algorithms with success probability at least $1 - 1/\text{poly}(n)$, but when $\epsilon = 1$ the randomized algorithm can be derandomized. All other algorithms are deterministic.

(2) We show that for (k,t)-median/means and (k,t)-center-pp the above results are achievable even on uncertain data (Theorem 5.6). For uncertain (k,t)-center-g we obtain an $(O(1+1/\epsilon), 1+\epsilon)$-approximation algorithm with $\tilde{O}(skB + tI + s\log\Delta)$ communication, where I is the information to encode the distribution of an uncertain point, and Δ is the ratio between the maximum pairwise distance and the minimum pairwise distance in the dataset (Theorem 5.14).

Our results for the (k,t)-center problem improves that in [21]. And as far as we are aware, our results on distributed (k,t)-median/means and of uncertain input are the first of their kinds. Our results for distributed (k,t)-median or means also lead to *subquadratic* time constant factor approximation centralized algorithms, which have been left open for many years.

Our two-round algorithms can be easily modified to be one-round at the cost of bigger communication costs and longer runtime, which will be explained in Section 6.

Technical Overview. The high level idea of our algorithms is fairly natural: Each site first performs a *preclustering*, i.e., it computes some local solution on its own dataset. Then each site sends the centers of the local solution, number of attached points to each center and the ignored points to the coordinator, who will then solve the induced *weighted* clustering problem.

A major difficulty is to determine how many points to ignore in the local solution at each site. Certainly for the sake of safety each site can ignore t points and send all ignored t points to the coordinator for a final decision. This would however incur $\Theta(st)$ bits of communication. To reduce the communication of this part to $O(t)$, we hope to find $\{t_1, \ldots, t_s\}$ such that $\sum_i t_i = t$ and each site i sends a solution with just t_i ignored points. At the cost of an extra round of communication, we solve the minimization problem $\sum_i f_i(t_i)$ subject to $\sum_i t_i = t$ for convex functions $\{f_i\}$. It is tempting to take $f_i(t_i)$ to be the cost of local solution with t_i ignored points on site i, however, such f_i is not necessarily convex. The remedy is to take a lower convex hull of f_i instead, which can be shown to have only a mild effect on the solution cost. The convex hull of t points can be found in $O(t\log t)$ time, and we can further reduce the runtime without compromising approximation ratio by computing local solutions on each site for only $\log t$ geometrically increasing values of t_i.

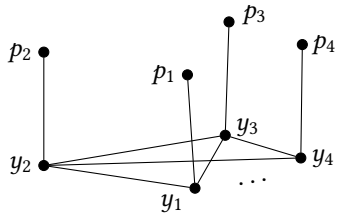

Fig. 1. An example of a compressed graph produced

For uncertain data, it is natural to reduce the clustering problems to the deterministic case. To this end, we 'collapse' each node j to its optimal center in $\mathcal{P}$. For instance, for the (k, t)-median problem, each node j is 'collapsed' to $y_j = \arg\min_{y \in \mathcal{P}} \mathbb{E}_\sigma[d(\sigma(j), y)]$, called the 1-median of node j. It may be tempting to consider the clustering problem on the set of 1-medians, but the 'collapse' cost is lost, hence we construct a *compressed graph* G that allows us to keep track of the collapse costs. The graph looks like a clique with tentacles, see Figure 1. The 1-medians form a clique in G with edge weight being the distance in the underlying metric space; for each 1-median y_j, we add a tentacle (an edge) from y_j to a new vertex p_j with edge weight being the collapse cost $\mathbb{E}_\sigma[d(\sigma(j), y_j)]$. We manage to show that the original clustering problem is equivalent, up to a constant factor in cost, to the clustering problem on the compressed graph where the facility vertices are 1-medians $\{y_j\}$ and the demand vertices are $\{p_j\}$. Our previous framework for deterministic data is then applied to the compressed graph.

Lastly, for the global center problem with uncertain data, we build upon the approach developed in [16], which uses a truncated distance function $\mathcal{L}_\tau(x, y) = \max\{d(x, y) - \tau, 0\}$ instead of the usual metric distance $d(\cdot, \cdot)$. Our algorithm performs a parametric search on τ, and applies our previous framework to solve the global problem using local solutions. Now in the analysis of the approximation ratio we need to relate the optimum solution to the solution with truncated distance function, which is a fairly nontrivial task.

Related Work. In the centralized model, Charikar et al. gives a 3-approximation algorithm for (k, t)-center, and an $(O(1), O(1))$ bicriteria algorithm for (k, t)-median [4]. This bicriteria was later removed by Chen [6], who designed an $O(1)$-approximation algorithm using $\tilde{O}(k^2(k + t)^2n^3)$ time. Feldman and Schulman studied the (k, t)-median problem with different loss functions using the *coreset* technique [13].

On uncertain data, Cormode and McGregor considered k-center/median/means where each $\mathcal{D}_i$ is a discrete distribution [8]. Guha and Munagala provided a technique to reduce the uncertain k-center to the deterministic k-median problem [16]. Wang and Zhang studied the special case of k-center on the line [24]. We refer the readers to the survey by Aggarwal [1].

Clustering on distributed data has been studied only recently. In the coordinator model, in the d-dimensional Euclidean space, Balcan et al. obtained $O(1)$-approximation algorithms with $\tilde{O}((kd + sk)B)$ bits of communication for both k-median and k-means [2]. Their results on k-means were further improved by Liang et al. [20] and Cohen et al. [7]. Chen et al. provided a set lower bounds for these problems [5]. In the MapReduce model, Ene et al. designed several $O(1)$-approximation $O(1)$-round algorithms for the k-center and the k-median problems [12]. Im and Moseley further studied the partial clustering variant [17], however their algorithms require communication polynomial in n. Cormode et al. studied the k-center maintenance problem in the

[2]For a general discrete distribution on m points in Euclidean space with $\mathcal{P}$ be the whole space, $T = O(m)$ [11]; for special distributions such as normal distribution, $T = O(1)$.

distributed data stream model where the coordinator can keep track of the cluster centers at any time step [9].

2 PRELIMINARIES

Notation. We use the following notations in this paper.

- $\mathrm{sol}(Z, k, t, d)$: A solution (computed by an algorithm) to the median/means/center problem on point set Z with at most k centers and at most t outliers, under the distance function d;
- $\mathrm{opt}(Z, k, t, d)$: An optimal solution to the median/means or center problem on point set Z with at most k centers and at most t outliers, under d;
- $C_{\mathrm{sol}}(Z, k, t, d)$: The cost of the solution $\mathrm{sol}(Z, k, t, d)$;
- $C_{\mathrm{opt}}(Z, k, t, d)$: The cost of the solution $\mathrm{opt}(Z, k, t, d)$;
- $\pi(j)$: The center to which point j is attached.

When Z lies in a metric space and d agrees with the distance function on the metric space, we omit the parameter d in the notations above.

Combining Preclustering Solutions. We review a theorem from [15], which concerns combining local solutions into a global solution. The problems considered in the theorem have *no* outliers ($t = 0$) and lie in a metric space, so we abbreviate the notation $\mathrm{sol}(Z, k, t, d)$ to $\mathrm{sol}(Z, k)$, etc.

Theorem 2.1 ([15]). *Suppose that $\mathbb{A} = \mathbb{A}_1 \uplus \cdots \uplus \mathbb{A}_s$ (disjoint union) and $\{\mathrm{sol}(\mathbb{A}_i, k)\}$ are the preclustering solutions at sites. Let $\mathbb{M} = \{\pi(j) : j \in \mathbb{A}\}$ and $L = \sum_{j \in \mathbb{A}} d(j, \pi(j))$, where $\pi(j)$ denotes the preclustering assignment. Consider the weighted k-median problem on $\mathbb{M}$ where the weight of $m \in \mathbb{M}$ is defined to be the number of points that are assigned to m in the preclustering, that is, $|\{j \mid j \in \mathbb{A}, \pi(j) = m\}|$. Then*

(i) There exists a weighted k-median solution $\mathrm{sol}(\mathbb{M}, k)$ such that $C_{\mathrm{sol}}(\mathbb{M}, k) \le 2(L + C_{\mathrm{opt}}(\mathbb{A}, k))$.
(ii) Given any weighted k-median solution $\mathrm{sol}(\mathbb{M}, k)$, there exists a k-median solution $\mathrm{sol}(\mathbb{A}, k)$ such that $C_{\mathrm{sol}}(\mathbb{A}, k) \le \mathrm{sol}(\mathbb{M}, k) + L$.

Consequently, there exists a k-median solution $\mathrm{sol}(\mathbb{A}, k)$ such that $C_{\mathrm{sol}}(\mathbb{A}, k) \le 2\gamma(L + C_{\mathrm{opt}}(\mathbb{A}, k)) + L$ and centers are restricted to $\mathbb{M}$, where γ is the best approximation ratio for the k-median problem.

Corollary 2.2. *The result in Theorem 2.1 extends to*

(i) the k-center problem;
(ii) the k-means problem with weaker constants, using a relaxed triangle inequality;
(iii) the (k, t)-median/means/center approximation on the weighted point set $\mathbb{M}$ (with γ being the corresponding bicriteria approximation ratio), provided *the preclustering does not ignore any points. Otherwise the total number of ignored points is the sum of the ignored points in the clustering and preclustering phases.*

3 (k, t)-MEDIAN AND (k, t)-MEANS

In this section, we first present a two-round algorithm in Section 3.1 and then analyse it in Section 3.2. We improve the algorithm for a better communication complexity in Section 3.3 in the case where we are only interested in the clustering and do not output the list of outliers. Finally in Section 3.4 we show how to obtain an subquadratic-time centralized algorithm by simulating a distributed algorithm sequentially.

3.1 Algorithm

Our algorithm for distributed (k, t)-median clustering is provided in Algorithm 1. For integer pairs (i, q), we consider the lexicographical order as partial order, that is,

$$(i_1, q_1) \prec (i_2, q_2) \quad \text{if} \quad \begin{cases} i_1 < i_2; \text{ or} \\ i_1 = i_2 \text{ and } q_1 < q_2. \end{cases} \tag{4}$$

Algorithm 1 Distributed $(k, (1+\epsilon)t)$-median clustering

Input: $\mathbb{A} = \mathbb{A}_1 \uplus \cdots \uplus \mathbb{A}_s$, $k \geq 1$, $t \geq 0$ and $\rho > 1$
Output: $\mathrm{sol}(\mathbb{A}, k, (1+\epsilon)t)$ such that $C_{\mathrm{sol}}(\mathbb{A}, k, (1+\epsilon)t) = O(1 + 1/\epsilon) \cdot C_{\mathrm{opt}}(\mathbb{A}, k, t)$

1: **for** each site i **do**
2: $\quad \mathbb{I} \leftarrow \{\lfloor \rho^r \rfloor : 1 \leq r \leq \lfloor \log_\rho t \rfloor,\ r \in \mathbb{Z}\} \cup \{0, t\}$
3: $\quad$ Compute $\mathrm{sol}(\mathbb{A}_i, 2k, q)$ for each $q \in \mathbb{I}$ ▷ Use the algorithm in Theorem 3.2
4: $\quad$ Compute the (lower) convex hull of the point set $\{(q, C_{\mathrm{sol}}(\mathbb{A}_i, 2k, q))\}_{q \in \mathbb{I}}$, which induces a function $f_i(\cdot)$ defined on $\{0, \ldots, t\}$
5: $\quad$ Send the function $f_i(\cdot)$ to the coordinator
6: **end for**
7: Coordinator computes $\ell(i, q) = f_i(q-1) - f_i(q)$ for each $1 \leq i \leq s$ and each $1 \leq q \leq t$
8: Coordinator *stably* sorts all $\{\ell(i, q)\}$ in decreasing order[3]
9: Coordinator finds $\ell(i_0, q_0)$ of rank[4] ρt and sends $\ell(i_0, q_0)$, i_0 and q_0 to all sites
10: **for** each site i **do**
11: $\quad t_i \leftarrow \max\{q : \ell(i, q) \geq \ell(i_0, q_0)\}$ ▷ define $\max \emptyset = 0$
12: $\quad$ **if** $i = i_0$ **then**
13: $\quad\quad t_i \leftarrow \min\{q \in \mathbb{I} : q \geq q_0 \text{ and } C_{\mathrm{sol}}(\mathbb{A}_i, 2k, q_0) = f_{i_0}(q_0)\}$
14: $\quad$ **end if**
15: $\quad$ Send the coordinator the $2k$ centers built in $\mathrm{sol}(\mathbb{A}_i, 2k, t_i)$, the number of points attached to each center, and the t_i unassigned points
16: **end for**
17: Coordinator considers the union of the centers obtained from each site and the unassigned points, and outputs $\mathrm{sol}(\mathbb{A}, k, (1+\epsilon)t)$. ▷ Use Theorem 3.2 and Remark 2

On a high level, the algorithm consists of two rounds. In the first round, each site guesses the number of points to exclude in its local input by computing the solution $\mathrm{sol}(\mathbb{A}_i, 2k, q)$ for logarithmically many values of q, then computes a convex hull of the costs for these values of q and sends the convex hull to the coordinator. The coordinator then determines a cost threshold, which will in turn determine the number of points to exclude on each individual site, and guarantees that the overall solution obtained this way is a constant-factor approximation. The coordinator then sends the threshold back to the sites. In the second round, each site determines the number of points to exclude from its local input, and sends the preclustering results to the coordinator and the coordinator then computes an overall solution.

We present the main theorem regarding the algorithm and defer the proof to the next subsection.

Theorem 3.1. *For the distributed (k, t)-median problem, with probability at least $1 - 1/\mathrm{poly}(n)$, Algorithm 1 with $\rho = 2$ outputs $\mathrm{sol}(\mathbb{A}, k, (1+\epsilon)t)$ satisfying $C_{\mathrm{sol}}(\mathbb{A}, k, (1+\epsilon)t)) \leq O(1 + 1/\epsilon) \cdot C_{\mathrm{opt}}(\mathbb{A}, k, t)$. The sites communicate a total of $\tilde{O}((sk + t)B)$ bits of information with the coordinator over 2 rounds. The runtime at each site is $\tilde{O}(n_i^2)$ and the runtime at the coordinator is $\tilde{O}((sk + t)^2)$. The same result holds for (k, t)-means with larger constants in the approximation ratio and the runtime. The algorithm can be derandomized when $\epsilon = 1$.*

Remark 1. *In Line 17 of Algorithm 1, (i) no input point is ignored in the preclustering; (ii) if the preclustering aggregated q points but the coordinator's algorithm chooses less than q copies (to exclude exactly t) then the proofs are not affected in any way.*

3.2 Analysis of Algorithm

To prove the correctness of Algorithm 1, we begin with a theorem about approximating (k,t)-median or means with a different trade-off between the approximation ratio and the runtime from that in [4]. The result in [4] is built upon the algorithm in [3], which increases the running time to cubic time for a better approximation factor. In the theorem below we aim at quadratic runtime and build our algorithm upon [18].

Theorem 3.2. *Let $\epsilon \in (0,1]$. We can compute $\mathrm{sol}(Z,k,(1+\epsilon)t)$ and $\mathrm{sol}(Z,(1+\epsilon)k,t)$ for the (k,t)-median problem using an algorithm in $\tilde{O}(|Z|^2)$ time such that*

(a) $C_{\mathrm{sol}}(Z,k,(1+\epsilon)t) \le O(1+1/\epsilon)\cdot C_{\mathrm{opt}}(Z,k,t)$, and
(b) $C_{\mathrm{sol}}(Z,(1+\epsilon)k,t) \le O(1+1/\epsilon)\cdot C_{\mathrm{opt}}(Z,k,t)$.

The algorithm for approximation guarantee (a) is randomized and succeeds with probability at least $1-1/\mathrm{poly}(\log|Z|)$, while the algorithm for approximation guarantee (b) is deterministic.

The result extends to the (k,t)-means problem with slightly larger constants.

Proof. Using [18], we can obtain in $\tilde{O}(|Z|^2)$ time two solutions with k_1, k_2 centers and each solution ignores exactly t outliers, where $k_1 < k < k_2$. Although not explicitly stated in [18], but as observed in [4], the algorithm is applicable to the outlier case as we can simply stop the algorithm when there are t points unprocessed.

Next we shall show approximation guarantee (a) by constructing a solution of exactly k centers with a similar procedure as the randomized rounding in [18]. Set $a = (k_2-k)/(k_2-k_1)$. First, we iteratively pair off every center in the small solution with its nearest (remaining) center in the large solution. Then with probability a we choose all the centers in the small solution and with probability $1-a$ we choose the paired centers in the large solution. At last, we choose $k-k_1$ centers at random from the remaining centers in the large solution.

In the current case we also have two solutions and each solution ignores exactly t outliers. Notice that if a point is labeled outlier in one solution and not in the other, it must be directly connected to a center (in the language of [18]). If we choose all the centers in the small solution then we cannot have more than t outliers. If we choose the large solution then we may exceed t outliers if all the points labeled outliers in the large solution were excluded and some of the points clustered in the small solution (but not in the large) cannot be accommodated because the corresponding center was not chosen. In expectation we have at most $a\cdot t$ extra outliers.

Let S_1 be the cost of the solution if the first step chooses the small solution and S_2 be the cost of the solution if the first step chooses the paired centers in the large solution. The same argument in [18] shows that

$$a\,\mathbb{E}(S_1) + (1-a)\,\mathbb{E}(S_2) \le 6C_{\mathrm{opt}}(Z,k,t).$$

Hence $\mathbb{E}(S_1) \le (6/a)C_{\mathrm{opt}}(Z,k,t)$.

- If $a \ge \epsilon/2$, we choose the small solution and run the derandomized version of the rounding part, which will give a solution of cost at most $\mathbb{E}(S_1) \le (12/\epsilon)C_{\mathrm{opt}}(Z,k,t)$. The derandomized rounding runs in time $\tilde{O}(|Z|^2)$ [18] and the claim of the overall runtime follows.
- Otherwise, $\mathbb{E}(S_2) \le \frac{6}{1-a}C_{\mathrm{opt}}(Z,k,t) \le \frac{6}{1-\epsilon/2}C_{\mathrm{opt}}(Z,k,t) \le 12C_{\mathrm{opt}}(Z,k,t)$. By Markov's inequality and a union bound, each run of (randomized) rounding fails with probability $\le 1/2+1/3 = 5/6$, producing a solution with more than $t+\epsilon t$ outliers or with cost more than $36C_{\mathrm{opt}}(Z,k,t)$. If we run the rounding $\Theta(\log|Z|)$ times we can, with probability at least

$1/\text{poly}(|Z|)$, find a solution with at most $t + \epsilon t$ outliers and cost at most $36C_{\text{opt}}(Z, k, t)$ (by picking the solution with minimum cost among all solutions with at most $t + \epsilon t$ outliers). The randomized rounding runs in time $\tilde{O}(|Z|)$ [18] and the claim of the overall runtime follows.

For approximation guarantee (b), when $k_2 \le (1+\epsilon)k$ we use the large solution, and the cost is at most $3C_{\text{opt}}(Z, k_2, t) \le 3C_{\text{opt}}(Z, k, t)$. When $k_2 \ge (1+\epsilon)k$, it holds that $a \ge 1 - 1/(1+\epsilon) \ge \epsilon/2$, a case that was discussed above. Note that the algorithm can be derandomized in this case. The solution is a $(12/\epsilon)$-approximation and the theorem follows.

Note that the above rounding argument uses the triangle inequality. While the triangle inequality does not hold for squares of distances (as in the k-means objective function), we instead use $2(x^2 + y^2) \ge (x + y)^2$. □

Remark 2. *The result generalizes to the weighted k-median/mean problem, since [18] also works for the weighted variant.*

Remark 3. *When $\epsilon = 1$, the algorithm for approximation guarantee (a) can be derandomized because the total number of outliers cannot exceed $2t$ in rounding, and the derandomization in [18] applies. This observation will be used in Section 3.4.*

Throughout the rest of the section, we denote by t_i^* the number of ignored points from $\mathbb{A}_i$ in the global optimum solution $\text{opt}(\mathbb{A}, k, t)$. We need the following lemmas.

Lemma 3.3. *It holds that $\sum_{i=1}^{s} C_{\text{opt}}(\mathbb{A}_i, k, t_i^*) \le 2C_{\text{opt}}(\mathbb{A}, k, t)$. For (k, t)-means the constant changes from 2 to 4.*

Proof. We shall use an argument used in [15]. Let π_{opt} be the center projection function and K be the set of optimum centers in the optimal solution $\text{opt}(\mathbb{A}, k, t)$. For each $\mathbb{A}_i$, we construct a solution $\text{sol}(\mathbb{A}_i, k, t_i^*)$ by excluding the points excluded in $\text{opt}(\mathbb{A}, k, t)$ and choosing $\left\{\arg\min_{u \in \mathbb{A}_i} d(u, k) : k \in K\right\}$ to be the centers. Then by the triangle inequality,

$$C_{\text{sol}}(\mathbb{A}_i, k, t_i^*) \le 2 \sum_{x \in \mathbb{A}_i} d(x, \pi_{\text{opt}}(x)).$$

Summing over $i = 1, \ldots, s$ yields $\sum_{i=1}^{s} C_{\text{sol}}(\mathbb{A}_i, k, t_i^*) \le 2C_{\text{opt}}(\mathbb{A}, k, t)$. The result for k-means follows from applying the triangle inequality with $(a+b)^2 \le 2(a^2 + b^2)$. □

Lemma 3.4. *The $t_1, \ldots, t_s$ computed in Step 11 of Algorithm 1 minimizes $\sum_i f_i(t_i)$ subject to $\sum_i t_i \le \rho t$ and $0 \le t_i \le t$.*

Proof. Suppose that $t_1', \ldots, t_s'$ is a minimizer. Since $f_i(\cdot)$ is non-increasing for all i, it must hold that $\sum_i t_i' = \rho t$. By the definition of t_i, it also holds that $\sum_i t_i = \rho t$. If $(t_1', \ldots, t_s') \ne (t_1, \ldots, t_s)$, there must exist i, j such that $t_i' > t_i$ and $t_j' < t_j$. By the definition of t_i and the sorting of $\{\ell(i, q)\}$, we know that

$$\ell(i, t_i + 1) \le \ell(i_0, q_0), \quad \ell(j, t_j) \ge \ell(i_0, q_0).$$

From convexity of f_i and that $t_i' \ge t_i + 1$ and $t_j' + 1 \le t_j$, it follows that

$$f_i(t_i' - 1) - f_i(t_i') \le \ell(i_0, q_0) \le f_j(t_j') - f_j(t_j' + 1)$$

which means that increasing t_j' by 1 and decreasing t_i' by 1 will not decrease the sum

$$G(q_1', \ldots, q_s') := \sum_i (f_i(0) - f_i(t_i')).$$

[4] *Stably* means that when $\ell(i_1, q_1) = \ell(i_2, q_2)$, the sorting algorithm puts $\ell(i_1, q_1)$ before $\ell(i_2, q_2)$ if $(i_1, q_1) < (i_2, q_2)$ as defined in (4).

[4] Element of *rank* r means the r-th element in a sorted list

Therefore $\sum_i f_i(t_i') = \sum_i f_i(0) - G(t_1', \ldots, t_s')$ will not increase. We can continue this procedure until $(t_1', \ldots, t_s') = (t_1, \ldots, t_s)$. □

Recall that $\mathbb{I} = \{\lfloor \rho^r \rfloor : 1 \le r \le \lfloor \log_\rho t \rfloor,\ r \in \mathbb{Z}\} \cup \{0, t\}$, defined in Line 2 of Algorithm 1.

Lemma 3.5. *It holds for all $i \neq i_0$ that $t_i \in \mathbb{I}$ and $C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i) = f_i(t_i)$, where i_0 is computed in Step 9 and t_i's in Step 11 of Algorithm 1.*

Proof. Since $0 \in \mathbb{I}$, we need only to consider the i's with $t_i \neq 0$. By the selection of (i_0, q_0) in the sorted list, it must hold that

$$\begin{aligned} \ell(i, t_i) &\ge \ell(i_0, q_0) > \ell(i, t_i + 1) \quad \text{for} \quad i < i_0 \\ \ell(i, t_i) &> \ell(i_0, q_0) \ge \ell(i, t_i + 1) \quad \text{for} \quad i > i_0, \end{aligned}$$

which implies that $\ell(i, t_i) > \ell(i, t_i + 1)$ whenever $i \neq i_0$, i.e.,

$$f_i(t_i - 1) - f_i(t_i) > f_i(t_i) - f_i(t_i + 1), \quad i \neq i_0.$$

Hence $(i, f_i(t_i))$ is a vertex of the convex hull for all $i \neq i_0$, that is, $t_i \in \mathbb{I}$ and $f_i(t_i) = C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i)$. □

Now we are ready to bound the 'goodness' of local solutions.

Lemma 3.6. *Let $\rho = 2$. It holds that $\sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i) \le C \cdot C_{\mathrm{opt}}(\mathbb{A}, 2k, t)$ for some absolute constant C and $\sum_i t_i \le 3t$, where $t_1, \ldots, t_s$ are computed in Step 11 and may be updated in Step 13 of Algorithm 1.*

Proof. Let $\hat{t}_i = \min\{q \in \mathbb{I} : q \ge t_i^*\}$. It follows from Lemma 3.3 with $\sum_i t_i^* \le t$ that

$$2C_{\mathrm{opt}}(\mathbb{A}, k, t) \ge \sum_i C_{\mathrm{opt}}(\mathbb{A}_i, k, t_i^*) \ge \sum_i C_{\mathrm{opt}}(\mathbb{A}_i, k, \hat{t}_i) \ge \frac{1}{C_0} \sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, \hat{t}_i),$$

where the last inequality follows from Theorem 3.2 (applied with $\epsilon = \rho - 1 = 1$) and C_0 is the approximation factor therein. Observe that $\hat{t}_i \le 2t_i^*$ and thus $\sum_i \hat{t}_i \le 2\sum_i t_i^* \le 2t$, and

$$\sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, \hat{t}_i) \ge \sum_i f_i(\hat{t}_i) \ge \sum_i f_i(t_i),$$

where the last equality follows from Lemma 3.4, and t_i's are computed in Step 11.

Now, by Lemma 3.5, $f_i(t_i) = C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i)$ for all except one i. The exceptional t_i will be replaced by a bigger value, which will not increase $f_i(t_i)$ by the monotonicity of f_i, and the first part follows. This update will increase $\sum_i t_i$ by at most t and thus $\sum_i t_i \le 3t$. □

Now, Theorem 3.1 follows straightforwardly from Lemma 3.6 and Theorem 3.2. Note that $|\mathbb{I}| = O(\log t)$.

Proof of Theorem 3.1. The communication cost is straightforward. By Lemma 3.6, the coordinator will solve the problem of at most $2sk + 3t$ points. The claims on approximation ratio and the runtime then follow from Theorem 3.2, noting that it takes time $O(|\mathbb{I}| \log |\mathbb{I}|) = \tilde{O}(1)$ to find the convex hull. All s sites run deterministic algorithms and only the coordinator runs a randomized algorithm, and the failure probability follows. □

3.3 Improvement When Not Outputting Outliers

In this subsection we discuss the scenario where we are only interested in the clustering and not the list of ignored points. We show that the communication complexity can be improved from $\tilde{O}(s(kB+t))$ to $\tilde{O}(skB)$ at the cost of slightly more ignored points in the bicriteria approximation. See Theorem 3.8 for a precise statement.

We set $\rho = 1+\delta$ and change line 12 to line 15 of Algorithm 1 to the following. The sites do not send the ignored nodes but just the number of them, and the exceptional site runs a slightly more convoluted algorithm.

12: **if** $i \neq i_0$ **then**
13: Send the coordinator t_i, the $2k$ centers built in $\mathrm{sol}(\mathbb{A}_i, 2k, t_i)$ and the number of points attached to each center
14: **else**
15: $t_{i,1} = \max\{q \in \mathbb{I} : q \leq t_i \text{ and } C_{\mathrm{sol}}(\mathbb{A}_i, 2k, q) = f_i(q)\}$
16: $t_{i,2} = \min\{q \in \mathbb{I} : q \geq t_i \text{ and } C_{\mathrm{sol}}(\mathbb{A}_i, 2k, q) = f_i(q)\}$
17: Combine $\mathrm{sol}(\mathbb{A}_i, 2k, t_{i,1})$ and $\mathrm{sol}(\mathbb{A}_i, 2k, t_{i,2})$ to form a solution $\mathrm{sol}(\mathbb{A}_i, 4k, t_i)$ by taking the union of the medians, attaching each point to the closest center among the combined centers, and ignoring the points with largest t_i distances.
18: Send to the coordinator t_i, the combined centers and the number of points attached to each center.
19: **end if**

Observe that Lemma 3.6 still holds with $\sum_i t_i \leq (1+\delta)t$, since we are not changing the exceptional t_i. For the exceptional site i, suppose that $t_i = (1-\theta)t_{i,1} + \theta t_{i,2}$ for some $\theta \in (0,1)$, we have $(1-\theta)f_i(t_{i,1}) + \theta f_i(t_{i,2}) \leq f_i(t_i)$. We now argue the next critical lemma.

Lemma 3.7. $C_{\mathrm{sol}}(\mathbb{A}_i, 4k, t_i) \leq (1-\theta)f_i(t_{i,1}) + \theta f_i(t_{i,2})$.

Proof. We will prove the lemma by carefully designing an assignment of $n - t_i$ points to the $4k$ centers which is bounded above by the right hand side. Since choosing the minimum $n - t_i$ distances will only result in a smaller value, the lemma would follow.

For $j = 1, 2$, let π_j be the center projection function in $\mathrm{sol}(\mathbb{A}_i, 2k, t_{i,j})$ and P_i the set of clustered points in $\mathrm{sol}(\mathbb{A}_i, 2k, t_{i,j})$. For $x \in P_1 \cap P_2$, we attach x to the nearer one between the two centers $\pi_1(x)$ and $\pi_2(x)$, and the incurred cost is

$$\min\{d(x,\pi_1(x)), d(x,\pi_2(x))\} \leq (1-\theta)d(x,\pi_1(x)) + \theta d(x,\pi_2(x)). \tag{5}$$

For $x \in P_1 \triangle P_2$, since only one of $\pi_1(x)$ and $\pi_2(x)$ exist, we abbreviate it as $\pi(x)$ for simplicity. Define $h(x)$ for each $x \in P_1 \triangle P_2$ as

$$h(x) = \begin{cases} (1-\theta)\cdot d(x,\pi(x)), & x \in P_1 \setminus P_2; \\ \theta \cdot d(x,\pi(x)), & x \in P_2 \setminus P_1. \end{cases}$$

Let $r = |P_1 \cap P_2|$, $r_1 = |P_1 \setminus P_2|$ and $r_2 = |P_2 \setminus P_1|$. It holds that $r + r_1 = n - t_{i,1}$ and $r + r_2 = n - t_{i,2}$, thus $r_1 > r_2$ and

$$(1-\theta)r_1 + \theta r_2 = n - t_i - r.$$

Define $Q_1 = P_1 \setminus P_2$ and $Q_2 = P_2 \setminus P_1$. Pick $x = \arg\min_{z \in Q_1 \cup Q_2} h(z)$. If $x \in Q_1$, pick an arbitrary $u \in Q_2$, otherwise pick $u \in Q_1$. Attach x to $\pi(x)$ in the $4k$-center solution we are constructing and mark u as outlier. Note that this incurs a cost of

$$d(x,\pi(x)) \leq \begin{cases} (1-\theta)d(x,\pi(x)) + \theta d(u,\pi(u)), & x \in Q_1; \\ (1-\theta)d(u,\pi(u)) + \theta d(x,\pi(x)), & x \in Q_2, \end{cases} \tag{6}$$

by our choice of x, because one of the combination terms is exactly $h(x)$ and it is smaller than $h(u)$, which is exactly the other term. Then we remove x and u from Q_1 or Q_2 depending on the case. Now, $|Q_1| = r_1 - 1$ and $|Q_2| = r_2 - 1$, and note that

$$(1-\theta)(r_1-1) + \theta(r_2-1) = n - t_i - r - 1.$$

Since $r_1 > r_2$, we can continue this process until $Q_2 = \emptyset$. At this point we have run the procedure above r_2 times, and it holds that

$$(1-\theta)r_1 = n - t_i - r - r_2.$$

Note that $r_1 \geq n - t_i - r - r_2$, so we can choose $E \subseteq Q_1$ to be the points with smallest $n - t_i - r - r_2$ values of h. Attach points in E to their respective centers and mark the remaining points in Q_1 as outliers. This incurs a cost of

$$\sum_{x\in E} d(x,\pi(x)) \leq \frac{n-t_i-r-r_2}{r_1} \sum_{x\in Q_1} d(x,\pi(x)) = (1-\theta)\sum_{x\in Q_1} d(x,\pi(x)) \tag{7}$$

In total we have assigned $r + r_2 + (n - t_i - r - r_2) = n - t_i$ points as desired. The desired upper bound on cost follows from (i) summing both sides of (5) over $P_1 \cap P_2$; (ii) summing both sides of (6) over x and the corresponding u during the pairing procedure; and (iii) Equation (7). Note that (ii) covers $(P_1 \triangle P_2) \setminus Q_1$, where Q_1 is the post-pairing set. □

As a consequence of Lemma 3.7, $C_{\text{sol}}(\mathbb{A}_i, 4k, t_i) \leq f_i(t_i)$. Thus the upper bound on the approximation ratio still holds. Finally, note that $|\mathbb{I}| = \tilde{O}(1/\delta)$ and we conclude that

Theorem 3.8. *For the distributed (k,t)-median problem, with probability at least $1 - 1/\text{poly}(n)$, the modified Algorithm 1 with $\rho = 1+\delta$ outputs $\text{sol}(\mathbb{A}, k, (2+\epsilon+\delta)t)$ satisfying $C_{\text{sol}}(\mathbb{A}, k, (2+\epsilon+\delta)t) \leq O(1+1/\epsilon)\cdot C_{\text{opt}}(\mathbb{A}, k, t)$. The sites communicate a total of $\tilde{O}(s\delta^{-1} + skB)$ bits of information with the coordinator over 2 rounds. The runtime on site i is $\tilde{O}(n_i^2/\delta)$ and the runtime on the coordinator is $\tilde{O}((sk)^2)$. The same result holds for (k,t)-means with a larger constant in the approximation ratio.*

3.4 Subquadratic-time Centralized Algorithm

We now show an unusual application of Theorem 3.1 in speeding up existing constant-factor approximation algorithms for (k,t)-median (or means). Note that the centralized bicriteria approximation algorithms in [4] have a runtime of $\tilde{O}(n^3)$ on n points, while the modifications in Theorem 3.2 improve the running time to $\tilde{O}(n^2)$, this leaves open the important question: *Are there algorithms with provable constant factor approximation guarantees which are subquadratic?* Observe that the question is even more pertinent in the context of unicriterion approximation, for which the only known result is a $\tilde{O}(n^3k^2t^2)$-time constant-factor approximation of (k,t)-median [6]. In the sequel we show that the running time can be brought to almost linear time. The improvement arises from the fact that we can simulate a distributed algorithm sequentially.

Lemma 3.9. *Suppose that we are given a $\tilde{O}(n^{1+\alpha_0}k^2)$ time algorithm for bicriteria approximation which produces $2k$ centers or $2t$ outliers with approximation factor γ, where $\alpha_0 \leq 1$. Then we can produce a similar algorithm with running time $\tilde{O}(t^2) + \tilde{O}\left(n^{\frac{2+2\alpha_0}{2+\alpha_0}}k^2\right)$ and approximation $c_0\gamma$ for some absolute constant $c_0 > 0$.*

Proof. We will apply Theorem 3.1 after dividing the data arbitrarily in s pieces of size n/s. The sequential simulation of the s sites will take time $\tilde{O}(s\,(n/s)^{1+\alpha_0}\,k^2)$ based on the statement of the lemma. The coordinator will require time $\tilde{O}((sk+t)^2) = \tilde{O}(s^2k^2) + \tilde{O}(t^2)$. Observe that we can now

balance $n^{1+\alpha_0} = s^{2+\alpha_0}$, which provides us the optimum s to use and achieve a running time of

$$\tilde{O}(t^2) + \tilde{O}(s^2k^2) = \tilde{O}(t^2) + \tilde{O}\left(n^{\frac{2+2\alpha_0}{2+\alpha_0}}k^2\right). \qquad \square$$

Theorem 3.10. *Let $\alpha > 0$ and suppose that $t \le \sqrt{n}$. There exists a centralized algorithm for the (k,t)-median problem that runs in $\tilde{O}(n^{1+\alpha}k^2)$ time and outputs a solution $\mathrm{sol}(\mathbb{A}, k, 2t)$ satisfying $C_{\mathrm{sol}}(\mathbb{A}, k, 2t) \le (1 + 1/\alpha)^{O(1)}C_{\mathrm{opt}}(\mathbb{A}, k, t)$.*

Proof. Note that the algorithm in Theorem 3.2 has runtime $\tilde{O}(n^2)$, so we can take $\alpha_0 = 1$ in Lemma 3.9 to obtain an algorithm of approximation ratio $\gamma = 6$ and runtime $\tilde{O}(t^2 + n^{4/3}k^2)$, which is $\tilde{O}(n^{4/3}k^2)$ by our assumption that $t \le \sqrt{n}$. Repeatedly applying Lemma 3.9 for j times gives an algorithm of runtime $\tilde{O}(n^{1+1/(2^j-1)}k^2)$ and approximation ratio $(c_0\gamma)^j$. Let $j = \log(1 + 1/\alpha)$, the runtime becomes $O(n^{1+\alpha}k^2)$ and the approximation ratio $(1 + 1/\alpha)^{\log(c_0\gamma)} = (1 + 1/\alpha)^{O(1)}$. $\square$

Remark 4. *We remark that*

(i) the theorem above also holds for $\mathrm{sol}(\mathbb{A}, 2k, t)$, where the number of centers, instead of the outliers, is relaxed;

(ii) for the unicriterion approximation, if we use the algorithm of runtime $\tilde{O}(n^3t^2k^2)$ from [6] instead of the result of Theorem 3.2, we need to balance s^3 and $s(n/s)^{1+\alpha_0}$ for an analogy of Lemma 3.9, which will eventually lead to an algorithm of runtime $O(n^{1+\alpha}t^2k^2)$, provided that $t \le n^{1/5}$.

4 (k,t)-CENTER CLUSTERING

Our algorithm for (k,t)-center clustering is presented in Algorithm 2. It is similar to Algorithm 1 but only simpler, because the preclustering stage admits a simpler algorithm due to Gonzalez [14]. For the k-center problem on a point set Z of n points, Gonzalez's algorithm outputs a re-ordering of points in Z, say, $p_1, \ldots, p_n$, such that for each $1 \le r \le n$, the solution $\mathrm{sol}(Z, r)$ of choosing $\{p_1, \ldots, p_r\}$ as the r centers is a 2-approximation for the r-center problem on Z, i.e., $C_{\mathrm{sol}}(Z, r) \le 2C_{\mathrm{opt}}(Z, r)$.

Algorithm 2 Distributed (k,t)-center clustering

1: **for** each site i **do**
2: Run Gonzalez's algorithm and obtain a re-ordering $\{a_1, \ldots, a_{n_i}\}$ of the points in $\mathbb{A}_i$
3: **for** each $1 \le q \le t$ **do**
4: Compute $\ell(i,q) \leftarrow \min\{d(a_j, a_{k+q}) : j < k + q\}$
5: **end for**
6: **end for**
7: Sites and coordinator sort $\{\ell(i,q)\}$, and follow the subsequent steps as in Algorithm 1, where the coordinator in the last step runs the algorithm in [4] for the k-center problem with exactly t outliers.

The core argument is that the k-center algorithm of Gonzalez can be used to simultaneously (a) precluster the local data into local solutions and (b) provide a witness that can be compared globally.

Remark 5. *In Algorithm 2, (i) none of the original points is ignored in the preclustering, and (ii) it is possible that the preclustering aggregated q points but the coordinator's algorithm chooses less than q copies to exclude exactly t points. This does not affect the proofs of (k,t)-center clustering.*

We now analyze the performance of Algorithm 2. Denote by t_i^* the number of points ignored from $\mathbb{A}_i$ in the global optimum solution $\mathrm{opt}(\mathbb{A}, k, t)$. First we show two structural lemmas.

Lemma 4.1. $2C_{\text{opt}}(\mathbb{A}_i, k, t) \geq \max_i C_{\text{opt}}(\mathbb{A}_i, k, t_i^*)$.

Proof. Use the same argument in the proof of Lemma 3.3. □

Lemma 4.2. $\max_i C_{\text{opt}}(\mathbb{A}_i, k, t_i^*) \geq \min_{\sum_i t_i \leq t} \left(\max_i C_{\text{opt}}(\mathbb{A}_i, k, t_i)\right)$.

Proof. It follows from the fact that $\sum_i t_i^* = t$. □

Theorem 4.3. *For the distributed (k,t)-center problem, Algorithm 2 outputs* $\text{sol}(\mathbb{A}, k, t)$ *satisfying* $C_{\text{sol}}(\mathbb{A}, k, t) \leq O(1) \cdot C_{\text{opt}}(\mathbb{A}, k, t)$. *The sites communicate a total of* $\tilde{O}((sk+t)B)$ *bits of information to the coordinator over* 2 *rounds. The runtime on site* i *is* $\tilde{O}((k+t)n_i)$ *and the runtime on the coordinator is* $\tilde{O}((sk+t)^2)$.

Proof. The approximation ratio follows from a similar argument to that of Theorem 3.1, using Lemma 4.1 and 4.2. The coordinator runtime follows from [4, Theorem 3.1] and the site runtime from [14], noting that we need only the first $k+t$ points of the reordering of each $\mathbb{A}_i$. The communication cost is clear from Algorithm 2. □

5 CLUSTERING UNCERTAIN INPUT

Recall that in the setting of clustering with uncertainty there is an underlying metric space $(\mathcal{P}, d)$. We are given a set of input nodes $j \in \mathbb{A}$ which correspond to distributions $\mathcal{D}_j$ on $\mathcal{P}$. In this section we shall use nodes to indicate the input and points to indicate deterministic objects in the metric space $\mathcal{P}$. We shall denote by $\sigma(j)$ a realization of node j and by $\pi(j)$ the center node to which j is attached. Note that $\pi(j)$ is a fixed point that is independent of the realization of the nodes. Our goal in the (k,t)-median problem in this context is to compute

$$\min_{\substack{K\subseteq\mathcal{P},\mathbb{O}\subseteq\mathbb{A}\\ |K|\leq k,|\mathbb{O}|\leq t}} \left[\sum_{j\in\mathbb{A}\setminus\mathbb{O}} \left(\min_{\pi(j)} \mathop{\mathbb{E}}_{\sigma} [d(\sigma(j),\pi(j))]\right)\right]. \tag{8}$$

For (k,t)-means we use $d^2(\cdot,\cdot)$ and for (k,t)-center-pp we use $\max_j$ instead of $\sum_j$.

Define $\widehat{d} : \mathbb{A} \times \mathcal{P} \to \mathbb{R}$ as $\widehat{d}(j,p) = \mathbb{E}_\sigma[d(\sigma(j),p)]$, the objective function (8) is then reduced to the usual (k,t)-median problem with the new distance function $\widehat{d}$. However, this definition only allows the computation of distance between an input node and a point in $\mathcal{P}$. To extend $\widehat{d}$ to a pair of input nodes, the site holding $\mathbb{A}_i$ will need to know the point set $\bigcup_{j\in\mathbb{A}_{i'}} \text{supp}(\mathcal{D}_j)$ from some other site i'. This will blow up the communication cost, and thus naively using this distance function in combination with the algorithms developed previously will not work well. To circumvent this issue we combine the notion of 1-median introduced in [8] along with the framework in Theorem 2.1, and introduce a compression scheme to evaluate distances.

Definition 5.1. For each node j, define its 1-median and 1-mean to be

$$y_j = \arg\min_{y\in\mathcal{P}} \mathop{\mathbb{E}}_{\sigma}[d(\sigma(j),y)], \quad y_j' = \arg\min_{y\in\mathcal{P}} \mathop{\mathbb{E}}_{\sigma}[d^2(\sigma(j),y)],$$

respectively.

Definition 5.2 (Compressed graph). The compressed graph $G(\mathbb{A})$ is a weighted graph on vertices $\mathcal{P} \cup \{p_j\}_{j\in\mathbb{A}}$, where the edges are as follows: (1) each pair $(u,v) \in \mathcal{P}$ is an edge with weight $d(u,v)$, and (2) for each $j \in \mathbb{A}$, the vertex p_j is connected only to y_j with weight $\ell_j = \mathbb{E}_\sigma[d(\sigma(j),y_j)]$. Define the distance $d_G(u,v)$ between two vertices u, v in G to be the length of the shortest path between u and v in G.

For the compressed graph G, we can also consider the following (k,t)-median problem, where we restrict the demand points to $\{p_j\}$ and the possible centers to $\{y_j\}$, and the distance function is the length of shortest path on G. We continue to use the notations $\mathrm{sol}(G,k,t)$, $C_{\mathrm{sol}}(G,k,t)$, etc., to denote the solution and the corresponding cost of (k,t)-median problem on G. The following two lemmas show that (k,t)-median problem in Eqn (8) is, up to some constant factor in the approximation ratio, equivalent to the (k,t)-median problem on the compressed graph.

Lemma 5.3. *If there exists a solution* $\mathrm{sol}(\mathbb{A},k,t)$ *of cost* $C_{\mathrm{sol}}(\mathbb{A},k,t)$ *to the objective in Equation* (8)*, then there exists a solution* $\mathrm{sol}(G(\mathbb{A}),k,t)$ *on the compressed graph such that* $C_{\mathrm{sol}}(G(\mathbb{A}),k,t) \le 5C_{\mathrm{sol}}(\mathbb{A},k,t)$.

Proof. Let $\mathbb{A}'$ be the set of clustered nodes in the feasible (k,t)-median solution of the original problem with the objective in (8). Define the set of center points $M = \{y_j : j \in \mathbb{A}'\}$. For each $j \in \mathbb{A}'$, let $y_{\pi(j)} = \arg\min_{y\in M} d(\pi(j),y)$. Let $\mathrm{sol}(G(\mathbb{A}),k,t)$ be the solution of connecting each point p_j ($j \in \mathbb{A}'$) to $y_{\pi(j)}$ in the compressed graph G. We try to upper bound the cost $C_{\mathrm{sol}}(G(\mathbb{A}),k,t)$:

$$\begin{aligned}
C_{\mathrm{sol}}(G(\mathbb{A}),k,t) &= \sum_{j\in\mathbb{A}'} d_G(y_{\pi(j)},p_j) && \text{(definition of } C_{\mathrm{sol}}\text{)}\\
&= \sum_{j\in\mathbb{A}'} \big(d(y_{\pi(j)},y_j) + d_G(y_j,p_j)\big) && \text{(definition of } d_G\text{)}\\
&\le \sum_{j\in\mathbb{A}'} d(y_{\pi(j)},\pi(j)) + \sum_{j\in\mathbb{A}'} d(\pi(j),y_j) + \sum_{j\in\mathbb{A}'} d_G(y_j,p_j) && \text{(triangle inequality)}\\
&\le 2\sum_{j\in\mathbb{A}'} d(\pi(j),y_j) + \sum_{j\in\mathbb{A}'} \ell_j,
\end{aligned}$$

where the last line follows from $d(y_{\pi(j)},\pi(j)) \le d(\pi(j),y_j)$ by the definition (optimality) of $y_{\pi(j)}$.

Observe that for any realization $\sigma(j)$, it holds that

$$d(y_j,\pi(j)) \le d(y_j,\sigma(j)) + d(\sigma(j),\pi(j)).$$

Taking expectation over σ,

$$d(y_j,\pi(j)) \le \mathop{\mathbb{E}}_{\sigma} d(y_j,\sigma(j)) + \mathop{\mathbb{E}}_{\sigma} d(\sigma(j),\pi(j)) = \ell_j + \mathop{\mathbb{E}}_{\sigma} d(\sigma(j),\pi(j)).$$

Summing over $j \in \mathbb{A}'$,

$$\sum_{j\in A'} d(y_j,\pi(j)) \le \sum_{j\in\mathbb{A}'} \ell_j + \sum_{j\in\mathbb{A}'} \mathop{\mathbb{E}}_{\sigma} d(\sigma(j),\pi(j)) \le \sum_{j\in\mathbb{A}'} \ell_j + C_{\mathrm{opt}}(\mathbb{A},k,t). \tag{9}$$

We next bound $\sum_{j\in\mathbb{A}'} \ell_j$. This is exactly the cost of connecting each $j \in \mathbb{A}'$ to its 1-median, which is the optimal solution of at most $n-t$ centers for $\mathbb{A}'$. The optimal cost for $n-t$ centers is clearly less than that for k centers and hence $\sum_{j\in\mathbb{A}'} \ell_j \le C_{\mathrm{opt}}(\mathbb{A},k,t)$.

Therefore $C_{\mathrm{sol}}(G(\mathbb{A}),k,t) \le 2\cdot 2C_{\mathrm{opt}}(\mathbb{A},k,t) + C_{\mathrm{opt}}(\mathbb{A},k,t) = 5C_{\mathrm{opt}}(\mathbb{A},k,t)$ as claimed. □

Lemma 5.4. *If there exists a solution* $\mathrm{sol}(G(\mathbb{A}),k,t)$ *of cost* $C_{\mathrm{sol}}(G(\mathbb{A}),k,t)$ *on the compressed graph, then there exists a solution* $\mathrm{sol}(\mathbb{A},k,t)$ *for the problem formulated in* (8) *such that* $C_{\mathrm{sol}}(\mathbb{A},k,t) \le 2C_{\mathrm{sol}}(G(\mathbb{A}),k,t)$.

Proof. Let $\mathbb{A}''$ be the set of clustered nodes in $\mathrm{sol}(G(\mathbb{A}),k,t)$. A similar argument of increasing the number of centers as in Lemma 5.3 yields that $\sum_{j\in\mathbb{A}''} \ell_j \le C_{\mathrm{sol}}(G(\mathbb{A}),k,t)$. Suppose that p_j is assigned to $\pi(j)$ in $\mathrm{sol}(G(\mathbb{A}),k,t)$ in the compressed graph. Note that $\pi(j) \in \mathcal{P}$. Let $\mathrm{sol}(\mathbb{A},k,t)$ be

Algorithm 3 A Compression Scheme for Distributed Partial Clustering of Uncertain Data

1: **for** each site i **do**
2: Compute $\ell_j = \mathbb{E}_\sigma[d(\sigma(j), y_j)]$ for all $j \in \mathbb{A}_i$
3: Construct the compressed graph of $\mathbb{A}_i$ as described in Definition 5.2
4: Run any algorithm corresponding to Section 3 and Section 4 on the compressed graph, with the following change: whenever the site has to communicate p_j, it also sends y_j (or y'_j) and the values of $\mathbb{E}_\sigma[d(\sigma(j), y_j)]$ (or $\mathbb{E}_\sigma[d^2(\sigma(j), y'_j)]$).
5: **end for**

the solution of attaching j to $\pi(j)$ in $\mathcal{P}$, and the cost can be bounded as

$$\begin{aligned} C_{\text{sol}}(\mathbb{A}, k, t) &= \sum_{j \in \mathbb{A}''} \mathop{\mathbb{E}}_{\sigma} \left(d(\sigma(j), \pi(j))\right) && \text{(definition of } C_{\text{sol}}\text{)} \\ &\le \sum_{j \in \mathbb{A}''} \mathop{\mathbb{E}}_{\sigma} \left(d(\sigma(j), y_j)\right) + \sum_{j \in \mathbb{A}''} d(y_j, \pi(j)) && \text{(triangle inequality)} \\ &\le \sum_{j \in \mathbb{A}''} \ell_j + \sum_{j \in \mathbb{A}''} d_G(p_j, \pi(j)) && \text{(definition of } d_G\text{, see below)} \\ &\le 2C_{\text{sol}}(G(\mathbb{A}), k, t), && \text{(definition of } C_{\text{sol}}\text{)} \end{aligned}$$

where the third line follows from $d_G(p_j, \pi(j)) = d(p_j, y_j) + d(y_j, \pi(j)) \ge d(y_j, \pi(j))$. □

The equivalence between the original problem and the one on the compressed graph also holds for the (k, t)-center-pp and the (k, t)-means problems.

Lemma 5.5. *Lemma 5.3 and Lemma 5.4 both hold*
(a) for (k, t)-center-pp with the same constants; and
(b) for (k, t)-means with slightly larger constants.

Proof. (a) Observe that $\sum_j$ is replaced with $\max_j$ and Equation (9) rewrites to

$$\max_{j \in \mathbb{A}'} d(y_j, \pi(j)) \le \max_{j \in \mathbb{A}'} \ell_j + C_{\text{opt}}(\mathbb{A}, k, t).$$

The remainder of the equations hold with this transformation.
(b) Note that we used triangle inequality in the proof above. Although the square of the distance does not obey the triangle inequality, we can nevertheless apply $(a + b)^2 \le 2a^2 + 2b^2$ after the triangle inequality. The derivations above will go through and the results hold with slightly larger constants. □

The overall algorithm is summarized in Algorithm 3. Note that we cannot just cluster the $\{y_j\}$; the graph is necessary. To implement the algorithm, we need to show that each site is able to compute the distance function individually. Indeed, note that any site that contains p_j will also contain the corresponding y_j or y'_j and the value $\mathbb{E}_\sigma[d(\sigma(j), y_j)]$ or $\mathbb{E}_\sigma[d^2(\sigma(j), y'_j)]$ respectively. Therefore the distance oracle on the graph can be implemented by the site in constant time.

Theorem 5.6. *For the distributed (k, t)-median problem, with probability at least $1 - 1/\text{poly}(n)$, Algorithm 3 outputs $\text{sol}(\mathbb{A}, k, (1 + \epsilon)t)$ such that $C_{\text{sol}}(\mathbb{A}, k, (1 + \epsilon)t) = O(1 + 1/\epsilon) \cdot C_{\text{opt}}(\mathbb{A}, k, t)$. The sites communicate a total of $\tilde{O}((sk + t)B)$ bits of information to the coordinator over 2 rounds. The runtime on site i is $\tilde{O}(n_i^2 + n_i T)$, where T is the runtime to compute 1-median, and the runtime on the coordinator is $\tilde{O}((sk + t)^2)$. The same result holds for the (k, t)-median and center-pp problems with larger constants.*

Proof. By Lemma 5.4 for the median problem and Lemma 5.5 for the means and center-pp problems, it suffices to show that we can solve the (k,t)-median problem on the compressed graph. The result then follows from Theorem 3.1 and Theorem 3.8 with the following amendments: When a site sends the t or t_i potential outliers, it needs to send the y_j and the corresponding values $\mathbb{E}_\sigma[d(\sigma(j),y_j)]$ or $\mathbb{E}_\sigma[d^2(\sigma(j),y'_j)]$, which at most doubles the communication cost. The runtime is increased by $O(n_iT)$ due to Step 2 since computing ℓ_j on the compressed graph takes $O(T)$ time. □

Other results claimed in Table 1 follow from analogous amendments to Theorem 3.8.

The global k-Center case. We now focus on (k,t)-center-g. In this setting we optimize

$$\min_{\substack{K\subseteq\mathcal{P},\mathbb{O}\subseteq\mathbb{A}\\ |K|\le k,|\mathbb{O}|\le t}}\left(\mathop{\mathbb{E}}_{\sigma\sim\prod_j\mathcal{D}_j}\left[\max_{j\in\mathbb{A}\setminus\mathbb{O}} d(\sigma(j),\pi(j))\right]\right).$$

Definition 5.7 (Truncated distance [16]). For $\tau\ge 0$, define $\mathcal{L}_\tau:\mathcal{P}\times\mathcal{P}\to\mathbb{R}$ as $\mathcal{L}_\tau(u,v)=\max\{d(u,v)-\tau,0\}$ and $\rho_\tau:\mathbb{A}\times\mathcal{P}\to\mathbb{R}$ as $\rho_\tau(j,u)=\mathbb{E}_\sigma[\mathcal{L}_\tau(\sigma(j),u)]$. Note that $\mathcal{L}_\tau(\cdot,\cdot)$ is not a metric for $\tau>0$.

Definition 5.8. Given a node set $Z\subseteq\mathbb{A}$, let $\mathcal{P}(Z)\subseteq\mathcal{P}$ be the associated point set corresponding to possible realizations of nodes in Z. Let $\mathrm{sol}(Z,k,t,\rho_\tau)$ and $\mathrm{opt}(Z,k,t,\rho_\tau)$ be a solution by algorithm and the global optimum solution respectively to the (k,t)-median problem on node set Z where the centers are restricted to $\mathcal{P}(Z)$ and the weighted assignment cost of assigning node $j\in Z$ to center $m\in\mathcal{P}(Z)$ is $\rho_\tau(j,m)$. The costs $C_{\mathrm{sol}}(Z,k,t,\rho_\tau)$ and $C_{\mathrm{opt}}(Z,k,t,\rho_\tau)$ are defined analogously.

Let $d_{\min}$ and $d_{\max}$ denote the minimum and the maximum distance, respectively, between two distinct points in $\mathcal{P}$ and let $\Delta=d_{\max}/d_{\min}$. The algorithm is presented in Algorithm 4.

Algorithm 4 Algorithm for (k,t)-center-g

1: All parties compute $d_{\min}$ and $d_{\max}$
2: Each party creates $\mathbb{T}=\{2^i d_{\min}/18 : 0\le i\le\lceil\log_2\Delta\rceil+2\}$
3: **for** each $\tau\in\mathbb{T}$ **do**
4: All parties run Algorithm 2 with the following changes: when it calls Algorithm 1 as a subroutine, $\mathrm{sol}(\mathbb{A}_i,2k,q)$ in Algorithm 1 is replaced with $\mathrm{sol}(\mathbb{A}_i,2k,q,\rho_{6\tau})$ and the sites obtain the numbers of local outliers $\{t_i(\tau)\}$
5: **end for**
6: Coordinator finds $\hat\tau=\max\{\tau\in\mathbb{T}:\sum_i C_{\mathrm{sol}}(\mathbb{A}_i,2k,t_i(\tau),\rho_{6\tau})\ge 2C_0\tau\}$, where C_0 is the approximation factor for (b) in Lemma 5.9.
7: Coordinator solves (k,t)-center-g on the preclustering solutions $\mathrm{sol}(\mathbb{A}_i,2k,t_i(\hat\tau),\rho_{6\tau})$ and outputs $\mathrm{sol}(\mathbb{A},k,(1+\epsilon)t)$.

Now we try to analyze the performance of Algorithm 4. We first show an analogy of Theorem 3.2 that we can compute a constant approximation to $C_{\mathrm{opt}}(Z,k,t,\rho_\tau)$.

Lemma 5.9. *Let $\tau\ge 0$. For the (k,t)-center problem on Z, we can compute in $\tilde O((k+t)|Z|)$ time* $\mathrm{sol}(Z,k,(1+\epsilon)t,\rho_{9\tau})$ *or* $\mathrm{sol}(Z,(1+\epsilon)k,t,\rho_{3\tau})$ *such that*

(a) $C_{\mathrm{sol}}(Z,k,(1+\epsilon)t,\rho_{9\tau})\le O(1+1/\epsilon)\cdot C_{\mathrm{opt}}(Z,k,t,\rho_\tau)$*;*
(b) $C_{\mathrm{sol}}(Z,(1+\epsilon)k,t,\rho_{3\tau})\le O(1+1/\epsilon)\cdot C_{\mathrm{opt}}(Z,k,t,\rho_\tau)$.

The algorithm for approximation guarantee (a) is randomized and succeeds with probability at least $1-1/\mathrm{poly}(|Z|)$ *while the algorithm for (b) is deterministic.*

Proof. The proof is similar to that of Theorem 3.2. The only different part is the accounting for the truncation. For the $(1+\epsilon)k$ result we note a pseudo-triangle inequality (see [16, Lemma 4.1]) $\rho_{3\tau}(j,m) \le \rho_\tau(j,m') + \rho_\tau(i,m') + \rho_\tau(i,m)$ for any m', since in this case we assign points within three hops. For the $(1+\epsilon)t$ result we assign within 9 hops—each point has a center in the large and small solutions within 3 hops. The pairing of the centers in the two solutions show that the pair of a center in the small solution exists within 6 hops. The whole argument for Theorem 3.2 then goes through. □

Remark 6. *Similar to Remark 3, the algorithm for approximation guarantee (a) in the preceding lemma (Lemma 5.9) can be made deterministic when* $\epsilon = 1$.

We next show that the $\hat{\tau}$ computed in Step 6 is a good choice of τ and will ensure that the preclustering solutions $\mathrm{sol}(\mathbb{A}_i, 2k, t_i(\hat{\tau}), \rho_{2\hat{\tau}})$ can be combined to yield a good global solution. Specifically we have the following two lemmas.

Lemma 5.10. *The* $\hat{\tau}$ *computed in Step 6 satisfies the following two conditions.*

(i) $\sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i(2\hat{\tau}), \rho_{12\hat{\tau}}) \le 4C_0\hat{\tau}$;
(ii) $\sum_i C_{\mathrm{opt}}(\mathbb{A}_i, k, t_i', \rho_{2\hat{\tau}}) \ge 2\hat{\tau}$ *for all* $\{t_i'\}$ *s.t.* $\sum_i t_i' \le t$,

Proof. Note that $\tau_{\max} = \max \mathbb{T} > d_{\max}/6$, it always holds that $\rho_{6\tau_{\max}} = 0$. Thus the condition $\sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i(\tau_{\max}), \rho_{6\tau_{\max}}) < 2C_0\tau_{\max}$ always holds, and $\hat{\tau}$ exists. Condition (i) follows from the maximality of $\hat{\tau}$, which means that $2\hat{\tau}$ will not satisfy the constraint.

Next we show that condition (ii) holds. Let $\{t_i'\}$ be an arbitrary sequence satisfying that $\sum_i t_i' \le t$. Similarly to the proof of Lemma 3.4, one can show that $\sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i', \rho_{6\hat{\tau}}) \ge \sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i(\hat{\tau}), \rho_{6\hat{\tau}})$, using the fact that $\sum_i t_i' \le t < \rho t = \sum_i t_i$. Combining with Lemma 5.9 with $\epsilon = 1$, we have that

$$C_0 \sum_i C_{\mathrm{opt}}(\mathbb{A}_i, k, t_i', \rho_{2\hat{\tau}}) \ge \sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i', \rho_{6\hat{\tau}}) \ge \sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i(\hat{\tau}), \rho_{6\hat{\tau}}) \ge 2C_0\hat{\tau},$$

whence condition (ii) follows. □

Lemma 5.11. *Suppose that* $\hat{\tau}$ *satisfies the condition (i) and (ii) of Lemma 5.10, a* γ*-approximation of the weighted center-g problem induced by preclustering* $\mathrm{sol}(\mathbb{A}_i, 2k, t_i(\hat{\tau}), \rho_{6\hat{\tau}})$ *is an* $O(\gamma)$ *approximation of* $C_{\mathrm{opt}}(\mathbb{A}, k, t)$.

To prove this lemma, we need the following two auxiliary lemmas.

Lemma 5.12. $2C_{\mathrm{opt}}(\mathbb{A}, k, t, \rho_\tau) \ge \sum_i C_{\mathrm{opt}}(\mathbb{A}_i, k, t_i^*, \rho_{2\tau})$, *where* t_i^* *is the number of ignored nodes from* $\mathbb{A}_i$ *in the global optimum solution* $\mathrm{opt}(\mathbb{A}, k, t, \rho_\tau)$.

Proof. Fix a realization of the nodes. The proof mimics Lemma 3.3 for each realization. It then uses the observation that $\mathcal{L}_\tau(u_1,u_2) + \mathcal{L}_\tau(u_2,u_3) \ge \mathcal{L}_{2\tau}(u_1,u_3)$ and takes the expectation. □

Lemma 5.13. *If* $C_{\mathrm{opt}}(Z,k,t,\rho_\tau) \ge \tau$ *then* $C_{\mathrm{opt}}(Z,k,t) \ge \tau/3$.

Proof. The case of $t = 0$ (no outliers) is proved in [16, Lemma 4.4]. For a general $t > 0$, let $Z' \subseteq Z$ be the set of clustered point in $\mathrm{opt}(Z,k,t)$, then $C_{\mathrm{opt}}(Z',k,0,\rho_\tau) = C_{\mathrm{opt}}(Z,k,t,\rho_\tau) \ge \tau$, thus $C_{\mathrm{opt}}(Z,k,t) = C_{\mathrm{opt}}(Z',k,0) \ge \tau/3$. □

Proof of Lemma 5.11. It follows from Lemma 5.12 and condition (ii) of Lemma 5.10 that

$$2C_{\mathrm{opt}}(\mathbb{A},k,t,\rho_{\hat{\tau}}) \ge \sum_i C_{\mathrm{opt}}(\mathbb{A}_i,k,t_i^*,\rho_{2\hat{\tau}}) \ge 2\hat{\tau},$$

where t_i^* is the number of ignored nodes from $\mathbb{A}_i$ in the global optimum solution $\mathrm{opt}(\mathbb{A},k,t,\rho_{\hat{\tau}})$. It then follows from Lemma 5.13 that $C_{\mathrm{opt}}(\mathbb{A},k,t) \ge \hat{\tau}/3$,

To simplify the notation, in the rest of the proof we shorthand $t_i(2\hat{\tau})$ as t_i. Let $\mathbb{A}_i^* \subseteq \mathbb{A}_i$ be the set of nodes clustered in the global optimum solution $\mathrm{opt}(\mathbb{A}, k, t)$. Consider "collapsing" the nodes in $\mathbb{A}_i^*$ to their corresponding centers in $\mathrm{sol}(\mathbb{A}_i, 2k, t_i, \rho_{12\hat{\tau}})$ while keeping the same centers in $\mathrm{sol}(\mathbb{A}, k, t)$. If a node in $\mathbb{A}_i^*$ is marked as an outlier in $\mathrm{sol}(\mathbb{A}_i, 2k, t_i, \rho_{12\hat{\tau}})$ then it is not moved, and it continues to be excluded from the calculation. This movement increases the expectation of the maximum assignment by $12\hat{\tau} + C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i, \rho_{12\hat{\tau}})$. Now consider the same process where we collapse $\mathbb{A}_i^*$ for all i. The total increase across the different i is $12\hat{\tau} + \sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i, \rho_{12\hat{\tau}})$ because the increase in $12\hat{\tau}$ arises from distance truncation and is common. Thus we achieve a solution of cost at most

$$\gamma \left(C_{\mathrm{opt}}(\mathbb{A}, k, t) + 12\hat{\tau} + \sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i, \rho_{12\hat{\tau}}) \right).$$

Now consider "expanding" the nodes of $\mathbb{A}_i$ from the preclustering to the distribution $\mathcal{D}_j$. By that logic the expected maximum can increase by at most $12\hat{\tau} + \sum_i C_{\mathrm{sol}}(\mathbb{A}_i, 2k, t_i, \rho_{12\hat{\tau}})$, which by condition (i) of Lemma 5.10 totals to $C'\gamma\hat{\tau} \le 3C'\gamma C_{\mathrm{opt}}(\mathbb{A}, k, t)$ for some constant C' that depends only on C_0. The lemma follows. □

We state the main theorem for the (k, t)-center-g problem to conclude this section.

Theorem 5.14. *For the distributed (k, t)-center-g problem, with probability at least $1 - 1/\mathrm{poly}(n)$, Algorithm 4 outputs $\mathrm{sol}(\mathbb{A}, k, (1+\epsilon)t)$ satisfying $C_{\mathrm{sol}}(\mathbb{A}, k, (1+\epsilon)t) = O(1 + 1/\epsilon) \cdot C_{\mathrm{opt}}(\mathbb{A}, k, t)$. The sites communicate a total of $\tilde{O}(skB + s\log\Delta + tI)$ bits of information to the coordinator over 2 rounds, where I is the bit complexity to encode a node. The runtime at site i is $\tilde{O}((k+t)n_i \log\Delta)$ and the runtime at the coordinator is $\tilde{O}((sk+t)^2)$. The algorithm can be derandomized when $\epsilon = 1$.*

Proof. The claim on approximation ratio follows from Lemma 5.11. To determine $\hat{\tau}$, the communication cost increases by a factor of $\log\Delta$; to send the preclustering solutions, the communication cost for sending the outliers increases by a factor of I. The runtime follows from Lemma 5.9 with an increase of a factor of $\log\Delta$. □

We remark that the dependence on $\log\Delta$ can be removed with another pass where each site computes a τ_i using binary search. The discussion is omitted in the interest of simplicity.

Other results claimed in Table 1 follow from analogous amendments to Theorem 3.8.

6 ONE-ROUND ALGORITHMS

We have shown two-rounds results in Table 1. The algorithms can be adapted to be one-round in a straightforward manner, by setting $t_i = t$ for all sites i. The results for (k, t)-median/means that ignores $(2+\delta)t$ or $(2+\epsilon+\delta)t$ points basically follow from Theorem 3.8, where for (k, t)-median with k centers (unicriterion) we need to apply again the 1-round result, and for (k, t)-median/means with $(1+\epsilon)k$ centers we simply use the second inequality of Theorem 3.2 instead of the first one at the final clustering step at the coordinator. The result for (k, t)-center that ignore $(2+\delta)t$ points is due to the following modifications on Algorithm 4: sites do not send the total $(1+\delta)t$ local outliers to the coordinator, and thereafter the coordinator performs the second level clustering with (another) t outliers, we have $(2+\delta)t$ outliers in total.

REFERENCES

[1] Charu C. Aggarwal. 2013. A Survey of Uncertain Data Clustering Algorithms. In *Data Clustering: Algorithms and Applications*. 457–482.

[2] Maria-Florina Balcan, Steven Ehrlich, and Yingyu Liang. 2013. Distributed k-means and k-median clustering on general communication topologies. In *Proceedings of NIPS*. 1995–2003.

[3] Moses Charikar and Sudipto Guha. 1999. Improved Combinatorial Algorithms for the Facility Location and k-Median Problems. In *Proceedings of FOCS*. 378–388.

[4] Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. 2001. Algorithms for Facility Location Problems with Outliers. In *Proceedings of SODA*. 642–651.
[5] Jiecao Chen, He Sun, D. Woodruff, and Qin Zhang. 2016. Communication-Optimal Distributed Clustering. In *Proceedings of NIPS*.
[6] Ke Chen. 2008. A constant factor approximation algorithm for k-median clustering with outliers. In *Proceedings of SODA*. 826–835.
[7] Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. 2015. Dimensionality Reduction for k-Means Clustering and Low Rank Approximation. In *Proceedings of STOC*. 163–172.
[8] Graham Cormode and Andrew McGregor. 2008. Approximation algorithms for clustering uncertain data. In *Proceedings of PODS*. 191–200.
[9] Graham Cormode, S Muthukrishnan, and Wei Zhuang. 2007. Conquering the divide: Continuous clustering of distributed data streams. In *Proceedings of ICDE*. IEEE, 1036–1045.
[10] Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: Simplified Data Processing on Large Clusters. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6 (OSDI'04)*. USENIX Association, Berkeley, CA, USA, 10–10.
[11] M. E. Dyer. 1986. On a Multidimensional Search Technique and Its Application to the Euclidean One Centre Problem. *SIAM J. Comput.* 15, 3 (1986), 725–738.
[12] Alina Ene, Sungjin Im, and Benjamin Moseley. 2011. Fast clustering using MapReduce. In *Proceedings of SIGKDD*. 681–689.
[13] Dan Feldman and Leonard J. Schulman. 2012. Data reduction for weighted and outlier-resistant clustering. In *Proceedings of SODA*. 1343–1354.
[14] Teofilo F. Gonzalez. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theor. Comput. Sci.* 38 (1985), 293–306. https://doi.org/10.1016/0304-3975(85)90224-5
[15] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. 2003. Clustering Data Streams: Theory and Practice. *IEEE Trans. on Knowl. and Data Eng.* 15, 3 (2003), 515–528.
[16] Sudipto Guha and Kamesh Munagala. 2009. Exceeding expectations and clustering uncertain data. In *Proceedings of PODS*. 269–278.
[17] Sungjin Im and Benjamin Moseley. 2015. Brief Announcement: Fast and Better Distributed MapReduce Algorithms for k-Center Clustering. In *Proceedings of SPAA*. 65–67.
[18] Kamal Jain and Vijay V. Vazirani. 2001. Approximation algorithms for metric facility location and k-Median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM* 48, 2 (2001), 274–296.
[19] Samir Khuller, Manish Purohit, and Kanthi K Sarpatwar. 2014. Analyzing the optimal neighborhood: Algorithms for budgeted and partial connected dominating set problems. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. 1702–1713.
[20] Yingyu Liang, Maria-Florina Balcan, Vandana Kanchanapally, and David P. Woodruff. 2014. Improved Distributed Principal Component Analysis. In *Proceedings of NIPS*. 3113–3121.
[21] Gustavo Malkomes, Matt J. Kusner, Wenlin Chen, Kilian Q. Weinberger, and Benjamin Moseley. 2015. Fast Distributed k-Center Clustering with Outliers on Massive Data. In *Proceedings of NIPS*. 1063–1071.
[22] Dan Suciu, Dan Olteanu, R. Christopher, and Christoph Koch. 2011. *Probabilistic Databases* (1st ed.). Morgan & Claypool Publishers.
[23] Pavol Ďuriš and José D.P. Rolim. 1998. Lower Bounds on the Multiparty Communication Complexity. *J. Comput. System Sci.* 56, 1 (1998), 90 – 95. https://doi.org/10.1006/jcss.1997.1547
[24] Haitao Wang and Jingru Zhang. 2015. One-dimensional k-center on uncertain data. *Theoretical Computer Science* 602 (2015), 114 – 124.