How to Solve Fair k-Center in Massive Data Models: Appendix

1 Algorithms

The definition of clustering cost (Definition 1) immediately implies the following observations.

Observation 1. *Let $A \supseteq A'$ and $B \subseteq B'$ be sets of points in a metric space given by a distance function d. The clustering cost of A for B is at most the clustering cost of A' for B'.*

Observation 2. *Let A_1, A_2, B_1, B_2 be sets of points in a metric space given by a distance function d. Suppose the clustering cost of each A_i for B_i is at most τ. Then the clustering cost of $A_1 \cup A_2$ for $B_1, \cup B_2$ is at most τ.*

The following lemma follows easily from the triangle inequality.

Lemma 1 (Lemma 1 from the paper, restated)**.** *Let $A, B, C \subseteq X$. The clustering cost of A for C is at most the clustering cost of A for B plus the clustering cost of B for C.*

Proof. Let d be the metric and let r_{AB} and r_{BC} denote the clustering costs of A for B and of B for C respectively. For every $a \in A$, there exists $b \in B$ such that $d(a, b) \leq r_{AB}$. But for this b, there exists $c \in C$ such that $d(b, c) \leq r_{BC}$. Thus, for every $a \in A$, there exists a $c \in C$ such that $d(a, c) \leq r_{AB} + r_{BC}$, by the triangle inequality. This proves the claim. □

The pseudocodes of procedures `getPivots`(), `getReps`(), and `HittingSet`() are given by Algorithms 1, 2, and 3 respectively.

Observation 3. *The procedure `getPivots` performs a single pass over the input set T. The set P returned by `getPivots`(T, d, r) contains points separated pairwise by distance more than r. The clustering cost of P for T is at most r. Therefore, by Lemma 2 from the paper, if there is a set of k points whose clustering cost for T is at most $r/2$, then $|P| \leq k$ pivots.*

Observation 4. *The procedure `getRep` executes a single pass over the input set T. The points in each set I_p returned by `getRep`(T, d, g, P, r) belong to distinct groups and are all within distance r from p. For every point q within distance r from $p \in P$, I_p contains a point in the same group as q (possibly q itself).*

Algorithm 1 `getPivots`(T, d, r)

Input: Set T with metric d, radius r.
$P \leftarrow \{p\}$ where p is an arbitrary point in T.
for each $q \in T$ (in an arbitrary order) **do**
  if $\min_{p \in P} d(p, q) > r$ **then**
    $P \leftarrow P \cup \{q\}$.
  end if
end for
Return P.

Algorithm 2 `getReps`(T, d, g, P, r)

Input: Set T with metric d, group assignment function g, subset $P \subseteq T$, radius r.
for each $p \in P$ **do**
  $I_p \leftarrow \{p\}$.
end for
for each $q \in T$ (in an arbitrary order) **do**
  for each $p \in P$ **do**
    if $d(p, q) \leq r$ and I_p doesn't contain a point from q's group **then**
      $I_p \leftarrow I_p \cup \{q\}$.
    end if
  end for
end for
Return $\{I_p : p \in P\}$.

Algorithm 3 `HittingSet`$(\mathcal{N}, g, \overline{k})$

Input: Collection $\mathcal{N} = (N_1, \ldots, N_K)$ of pairwise disjoint sets of points, group assignment function g, vector $\overline{k} = (k_1, \ldots, k_m)$ of capacities.
Construct bipartite graph $G = (\mathcal{N}, V, E)$ as follows.
$V \leftarrow \biguplus_{j=1}^{m} V_i$, where V_j is a set of k_j vertices.
for each N_i and **each** group j **do**
  if $\exists\, p \in N_i$ such that $g(p) = j$ **then**
    Connect N_i to all vertices in V_j.
  end if
end for
Find the maximum cardinality matching H of G.
$C \leftarrow \emptyset$.
for each edge (N_i, v) of H **do**
  Let p be a point in N_i from group j, where $v \in V_j$.
  $C \leftarrow C \cup \{p\}$.
end for
Return C.

The procedure `HittingSet` constructs the following bipartite graph. The left side vertex set contains K vertices: one for each N_i. The right side vertex set is $V = \biguplus_{j=1}^{m} V_j$, where V_j contains k_j vertices for each group j. If N_i contains a point from group j, then its vertex is connected to the all of V_j. Each matching H in this bipartite graph encodes a feasible subset C of $\biguplus_{i=1}^{K} N_i$ as follows. For each edge $e = (N_i, v) \in H$ where $v \in V_j$, add to C the point from N_i belonging to group j. Observe that since $|V_j| = k_j$ and H is a matching, C contains at most k_j points from group j. Moreover, $|C| = |H|$, and hence, a maximum cardinality matching in the bipartite graph encodes a set C intersecting as many of the N_i's as possible.

In our implementation, we enhance the efficiency of `HittingSet` as follows. For each group, we introduce only one vertex in the right side vertex set and construct the bipartite graph like `HittingSet`, directing edges from left to right. We further connect a source to the left side vertices with unit capacity edges, and the right side vertices to a sink with edges of capacities k_j. We find the maximum (integral) source-to-sink flow using the Ford-Fulkerson algorithm. For each i and j, if the edge (N_i, j) exists and carries nonzero flow, then we include in C the point in N_i that belongs to group j. Our runtime is bounded as follows.

Lemma 2. *The runtime of* `HittingSet`() *is* $O(K^2 \cdot \max_i |N_i|)$.

Proof. The number of edges in the constructed bipartite graph is $O(K \cdot \max_i |N_i|)$ whereas the value of the max-flow is no more than K. The runtime of the Ford-Fulkerson algorithm is of the order of the size of the number of edges times the value of max-flow. Therefore, the runtime of `HittingSet()`, which is dominated by the runtime of the Ford-Fulkerson algorithm, turns out to be $O(K^2 \cdot \max_i |N_i|)$. $\square$

2 Distributed k-Center Lower Bound

In this section, we present the formal details of the lower bound discussed in Section 4 of the main paper. For a natural number n, $[n]$ denotes the set $\{1, 2, \ldots, n\}$.

The metric space $\mathcal{M}(n')$. The point set of this metric space on $n = 9n' + 7$ points is given by

$$S := \{a^*, b_1^*, b_2^*, c^*, a, b, c\} \cup S_1 \cup S_2 \cup S_3,$$

where $|S_1| = |S_2| = |S_3| = 3n'$. Note that S_1, S_2, S_3 are pairwise disjoint and are also disjoint from $\{a^*, b_1^*, b_2^*, c^*, a, b, c\}$. We will call the points $\{a^*, b_1^*, b_2^*, c^*, a, b, c\}$ *critical.* The metric $d : S \times S \longrightarrow \mathbb{R}$ is the shortest-path-length metric induced by the graph shown in Figure 1 (where x is not a point in S but is only used to define the pairwise distances). The pairwise distances are given in Table 1. Note that if the table entry i, j is indexed by sets, then the entry corresponds to the distance between distinct points in the sets. The following observation can be verified by a case-by-case analysis.

Observation 5. *The sets $\{a^*, b_1^*, c^*\}$ and $\{a^*, b_2^*, c^*\}$ are the only optimum solutions of the 3-center problem on $\mathcal{M}(n')$ and they have unit clustering cost. The clustering cost of any subset of S_1 is 4 due to point c. Similarly, the clustering cost of any subset of S_3 is 4 due to point a.*

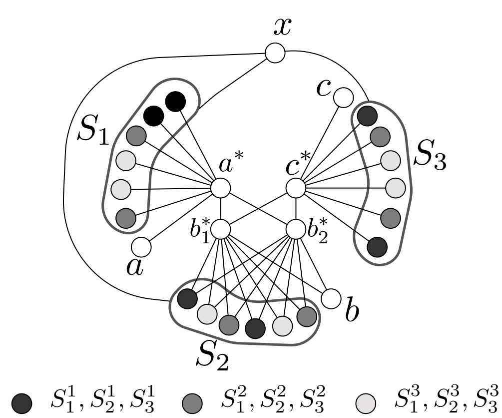

Figure 1: The underlying metric for $n' = 2$

Input Distribution $\mathcal{D}$ on the Processors' Inputs. For $i \in [3]$, let S_i^1, S_i^2, S_i^3 be an arbitrary equipartition of S_i (and therefore, $|S_i^j| = n'$ for all i, j). Define the sets $Y_1^j = \{b_1^*, b_2^*, a\} \cup S_1^j$, $Y_2^j = \{a^*, c^*, b\} \cup S_2^j$ and $Y_3^j = \{b_1^*, b_2^*, c\} \cup S_3^j$, for $j \in [3]$. Observe that each Y_i^j contains exactly $n' + 3$ points separated pairwise by distance 2, and moreover, three of the $n' + 3$ points are critical. We assign the sets Y_i^j randomly to the nine processors after a random relabeling. Formally, we pick a uniformly random bijection $\pi : S \longrightarrow [n]$ as the relabeling and another uniformly random bijection $\Gamma : [3] \times [3] \longrightarrow [9]$, independent of π, as the assignment. We assign the set $\pi(Y_i^j)$ to processor $\Gamma(i, j)$ for every i, j. When a processor or the coordinator

	a^*	b_1^*	b_2^*	c^*	a	b	c	S_1	S_2	S_3
a^*	0	1	1	2	1	2	3	1	2	3
b_1^*	1	0	2	1	2	1	2	2	1	2
b_2^*	1	2	0	1	2	1	2	2	1	2
c^*	2	1	1	0	3	2	1	3	2	1
a	1	2	2	3	0	3	4	2	3	4
b	2	1	1	2	3	0	3	3	2	3
c	3	2	2	1	4	3	0	4	3	2
S_1	1	2	2	3	2	3	4	2	2	2
S_2	2	1	1	2	3	2	3	2	2	2
S_3	3	2	2	1	4	3	2	2	2	2

Table 1: Pairwise Distances

queries the distance between p and q where $p, q \in [n]$, it gets $d(\pi^{-1}(p), \pi^{-1}(q))$ as an answer. Note that neither the processors nor the coordinator knows π or Γ. Let the random variable $\mathcal{P} = (\mathcal{P}_1, \ldots, \mathcal{P}_9)$ denote the partition of the set of labels into a sequence of nine subsets induced by π and Γ, where $\mathcal{P}_r$ is the set of labels of points assigned to processor r, that is, $\mathcal{P}_{\Gamma(i,j)} = \pi(Y_i^j)$.

Lemma 3. *Consider any deterministic distributed algorithm for the* 9 *processor* 3*-center problem on* $\mathcal{M}(n')$ *and input distribution* $\mathcal{D}$*, in which each processor communicates an* ℓ*-sized subset of its input points, and the coordinator outputs* 3 *of the received points. If* $\ell \leq (n'+3)/54$*, then with probability at least* $1/84$*, the output is no better than a* 4*-approximation.*

Here, although the probability with which the coordinator fails to outputs a better-than-4-approximation is only $1/84$, it can be *amplified* to $1-\varepsilon$, for any $\varepsilon > 0$. We discuss the amplification result before presenting the proof of the above lemma.

Lemma 4. *Let* $\varepsilon > 0$ *and* $0 < c < 1/486$ *be arbitrary constants, and let*

$$\alpha = \left\lceil \frac{84 \ln(1/\varepsilon)}{1 - 486c} \right\rceil.$$

Then there exists an instance of the (3α)*-center problem such that, in the distributed setting with* 9 *processors, each communicating at most a* c *fraction of its input points to the coordinator, the coordinator fails to output a better than* 4*-approximation with probability at least* $1 - \varepsilon$*.*

Proof. The underlying metric space consists of α disjoint copies of $\mathcal{M}(n')$ separated by an arbitrarily large distance from one another. The point set of each copy is distributed to the nine processors as described earlier, and these distribtions are independent. Thus, each processor receives $\alpha \cdot (n'+3)$ points. Observation 5 implies that in this instance, the optimum set of 3α centers (the union of optimum sets of 3 centers in each copy) has unit cost. Also, in order to get a better than 4-approximation, the coordinator must output a better than 4-approximate solution from every copy. We prove that this is unlikely.

By our assumption, each processor sends at most $c\alpha \cdot (n'+3)$ points to the coordinator, where $c < 1/486$. Therefore, for each processor, there exist at most $54c\alpha$ copies from which it sends more than $(n'+3)/54$

points to the coordinator. Since we have 9 processors, there exist at most $9 \times 54c\alpha = 486c\alpha$ copies from which more than $(n'+3)/54$ points are sent by some processor. From each of the remaining $(1-486c)\alpha$ copies, no processor sends more than $(n'+3)/54$ points. By Lemma 3, the coordinator succeeds on each of these copies independently with probability at most $1-1/84$, in producing a better than 4 approximation. Therefore, the probability that the coordinator succeeds in all the $(1-486c)\alpha$ copies is bounded as

$$\left(1-\frac{1}{84}\right)^{(1-486c)\alpha} \leq \exp\left(-\frac{1-486c}{84}\cdot\alpha\right) \leq \varepsilon,$$

where the last inequality follows by substituting the value of α. Thus, the coordinator fails to produce a better than 4-approximation with probability at least $1-\varepsilon$. □

Proof of Lemma 3. Consider any one of the nine processors. It gets the set $\pi(Y_i^j)$ for a uniformly random $(i,j) \in [3]\times[3]$. Since π is a uniformly random labeling and points in Y_i^j are pairwise equidistant, the processor is not able to identify the three critical points in its input. This happens even if we condition on the values of Γ. Formally, conditioned on Γ and $\mathcal{P}$, all subsets of $\mathcal{P}_r$ of size 3 are equally likely to be the set of labels of the three critical points in processor r's input, i.e., Y_i^j where $(i,j)=\Gamma^{-1}(r)$. As a consequence, the probability that at least one of the three critical points appears in the set of at most ℓ points the processor communicates is at most $3\ell/|Y_i^j| = 3\ell/(n'+3)$, even when we condition on Γ. For a given processor $r\in[9]$, let O_r be the set of labels it sends to the coordinator, and define B_r to be the event that O_r contains the label of a critical point. Then $\Pr[B_r \mid \Gamma,\mathcal{P}] \leq 3\ell/(n'+3)$. Next, define G to be the event that no processor sends the label of any critical point to the coordinator, that is, $G=\cap_{r=1}^{9} B_r^c$, where B_r^c is the complement of B_r. Then by the union bound and the fact that $\ell \leq (n'+3)/54$, we have for every partition P of the label set and every bijection $\gamma : [3]\times[3] \longrightarrow [9]$,

$$\Pr[G \mid \Gamma=\gamma, \mathcal{P}=P] \geq 1-9\cdot\frac{3\ell}{n'+3} \geq \frac{1}{2}. \tag{1}$$

Suppose the coordinator outputs O, a set of three labels, on receiving $O_1,\ldots,O_9$. Then $O \subseteq O_{r_1}\cup O_{r_2}\cup O_{r_3}$ for some $r_1,r_2,r_3\in[9]$. Observe that $O_1,\ldots,O_9$, O, and $\{r_1,r_2,r_3\}$ are all completely determined[1] by $\mathcal{P}$. In contrast, due to the random labeling π, the mapping Γ is independent of $\mathcal{P}$. Therefore,

Observation 6. *Conditioned on $\mathcal{P}$, the bijection Γ is equally likely to be any of the $9!$ bijections from $[3]\times[3]$ to $[9]$.*

Next, define G' to be the event that $\{r_1,r_2,r_3\}$ is either $\Gamma(\{(1,1),(1,2),(1,3)\})$ or $\Gamma(\{(3,1),(3,2),(3,3)\})$. In words, G' is the event that the coordinator outputs labels of three points, all of which are contained in $Y_1^1\cup Y_1^2\cup Y_1^3$ or in $Y_3^1\cup Y_3^2\cup Y_3^3$. Note that the event $G'\cap G$ implies that the coordinator's output is contained in $S_1^1\cup S_1^2\cup S_1^3=S_1$ or in $S_3^1\cup S_3^2\cup S_3^3=S_3$. Therefore, by Observation 5, event $G'\cap G$ implies that the coordinator fails to output a better than 4-approximation. We are now left to bound $\Pr[G'\cap G]$ from below.

Since the set $\{r_1,r_2,r_3\}$ is completely determined by $\mathcal{P}$, the event G' is completely determined by $\mathcal{P}$ and Γ: for any $\mathcal{P}$, there exist exactly $2\cdot 3!\cdot 6!$ values of Γ which cause G' to happen. Formally,

Observation 7. *For every partition P of the label set, there exist exactly $2\cdot3!\cdot6!$ bijections $\gamma:[3]\times[3]\longrightarrow[9]$ such that $\Pr[G' \mid \mathcal{P}=P,\Gamma=\gamma]=1$, whereas $\Pr[G' \mid \mathcal{P}=P,\Gamma=\gamma']=0$ for all the other bijections $\gamma':[3]\times[3]\longrightarrow[9]$.*

[1] If O intersects less than three of the O_r's, then we define $\{r_1,r_2,r_3\}$ to be the lexicographically smallest set such that $O\subseteq O_{r_1}\cup O_{r_2}\cup O_{r_3}$.

Therefore, we have,

$$
\begin{aligned}
\Pr[G \cap G'] &= \sum_{P,\gamma} \Pr[G \cap G' \mid \mathcal{P} = P, \Gamma = \gamma] \cdot \Pr[\mathcal{P} = P, \Gamma = \gamma] \\
&= \sum_{(P,\gamma):\Pr[G' \mid \mathcal{P}=P,\Gamma=\gamma]=1} \Pr[G \mid \mathcal{P} = P, \Gamma = \gamma] \cdot \Pr[\Gamma = \gamma \mid \mathcal{P} = P] \cdot \Pr[\mathcal{P} = P] \\
&\geq \sum_{P} \sum_{\gamma:\Pr[G' \mid \mathcal{P}=P,\Gamma=\gamma]=1} \frac{1}{2} \cdot \frac{1}{9!} \cdot \Pr[\mathcal{P} = P] \\
&= \frac{1}{2} \cdot \frac{1}{9!} \cdot \sum_{P} |\{\gamma : \Pr[G' \mid \mathcal{P} = P, \Gamma = \gamma] = 1\}| \cdot \Pr[\mathcal{P} = P] \\
&= \frac{2 \cdot 3! \cdot 6!}{2 \cdot 9!} \cdot \sum_{P} \Pr[\mathcal{P} = P] \\
&= \frac{1}{84}.
\end{aligned}
$$

Here, we used Observation 7 for the second and fourth equality, and Equation (1) and Observation 6 for the inequality. Thus, the coordinator fails to output a better than 4-approximation with probability at least $1/84$, as required. □

Using Lemma 4 along with Yao's lemma, we get our main lower-bound theorem.

Theorem 1. *There exists $c > 0$ such that for any $\varepsilon > 0$, with $k = \Theta(\log(1/\varepsilon))$, any randomized distributed algorithm for k-center where each processor communicates at most cn points to the coordinator, who outputs a subset of those points as the solution, is no better than 4-approximation with probability at least $1 - \varepsilon$.*