New Algorithms for Distributed Fair k-Center Clustering: Almost Accurate as Sequential Algorithms

Xiaoliang Wu
School of Computer Science and Engineering, Central South University
Xiangjiang Laboratory
Changsha, China
xiaoliangwu@csu.edu.cn

Qilong Feng*
School of Computer Science and Engineering, Central South University
Xiangjiang Laboratory
Changsha, China
csufeng@mail.csu.edu.cn

Ziyun Huang
Department of Computer Science and Software Engineering, Penn State Erie, The Behrend College
Erie, United States
zxh201@psu.edu

Jinhui Xu
Department of Computer Science and Engineering, State University of New York at Buffalo
Buffalo, United States
jinhui@cse.buffalo.edu

Jianxin Wang*
Hunan Provincial Key Lab on Bioinformatics, Central South University
Xiangjiang Laboratory
Changsha, China
jxwang@mail.csu.edu.cn

ABSTRACT

Fair clustering problems have been paid lots of attention recently. In this paper, we study the k-Center problem under the group fairness and data summarization fairness constraints, denoted as Group Fair k-Center (GFkC) and Data Summarization Fair k-Center (DSFkC), respectively, in the massively parallel computational (MPC) distributed model. The previous best results for the above two problems in the MPC model are a 9-approximation with violation 7 (WWW 2022) and a $(17+\epsilon)$-approximation without fairness violation (ICML 2020), respectively. In this paper, we obtain a $(3+\epsilon)$-approximation with violation 1 for the GFkC problem in the MPC model, which is almost as accurate as the best known approximation ratio 3 with violation 1 for the sequential algorithm of the GFkC problem. Moreover, for the DSFkC problem in the MPC model, we obtain a $(4+\epsilon)$-approximation without fairness violation, which is very close to the best known approximation ratio 3 for the sequential algorithm of the DSFkC problem. Empirical experiments show that our distributed algorithms perform better than existing state-of-the-art distributed methods for the above two problems.

KEYWORDS

Machine Learning; Fairness; Clustering

ACM Reference Format:
Xiaoliang Wu, Qilong Feng*, Ziyun Huang, Jinhui Xu, and Jianxin Wang*. 2024. New Algorithms for Distributed Fair k-Center Clustering: Almost Accurate as Sequential Algorithms. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

* Corresponding authors.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

Clustering is one of the most popular problems in machine learning, and has lots of applications in data mining, image classification, etc. Given a set of points, the goal of clustering is to partition the point set into several disjoint clusters such that the points in the same cluster are close to each other, and the points in different clusters are far away from each other. Several classic clustering models have been extensively studied, such as k-Center, k-Median, and k-Means. In this paper, we focus on the k-Center problem that is known to be NP-hard [20], and admits a 2-approximation algorithm [21, 23]. Many variations of the k-Center problem have been studied in the literature [1, 3, 4, 10–12, 17, 19, 26].

Recently, fair clustering has been extensively studied, and lots of definitions about fairness have been proposed, such as group fairness [2, 6, 8, 22], data summarization fairness [16, 24, 27], proportional fairness [14, 28, 30], individual fairness [29, 31], etc. In this paper, we focus on two fairness notions, i.e., group fairness and data summarization fairness. For an instance of the clustering problem under group fairness or data summarization fairness, the points in the instance are divided into several groups, and each point is assigned a color to denote which group it is in.

We take two examples to illustrate the application of group fairness and data summarization fairness, respectively. As pointed out in [15], if the given clustering instance contains some minority groups, and the properties of the minority groups are not considered in the clustering process, then in the clustering results, the proportion of some minority groups assigned to some clusters may be far from the true proportions in real life, which results in the unfair treatments to minority groups. For the data summarization problem, the objective is to find a small subset to summarize the whole dataset. Most algorithms for data summarization are usually biased with respect to some sensitive attributes, and this leads to the study of the data summarization problem under fairness constraints, for example, a Google Images search for the keyword "CEO" returns a much higher proportion of men than the real life proportion of male CEOs, leading to prejudice against women [25].

In this paper, we focus on the k-Center problem under group fairness and data summarization fairness constraints, denoted as Group Fair k-Center (GFkC) and Data Summarization Fair k-Center (DSFkC), respectively. Given a set C of n points in a metric space, where C comprises m disjoint groups $C_1, \ldots, C_m$, and the points in C_h are colored with color h ($h \in \{1, \ldots, m\}$). For the GFkC problem, two fair vectors $\alpha, \beta \in [0, 1]^m$ are given, and the goal is to partition C into k clusters such that the proportion of points with color h in each cluster is at least β_h and at most α_h, and the k-Center problem objective is minimized. For the DSFkC problem, a vector $\gamma = (k_1, \ldots, k_m)$ with $\sum_{h=1}^{m} k_h = k$ is given, and the goal is to partition C into k clusters such that the number of cluster centers with color h is equal to k_h, and the k-Center problem objective is minimized. In this paper, we study the GFkC and DSFkC problems in the massively parallel computational (MPC) distributed model, which is used to deal with massive data. In this model, the input data points are distributed arbitrarily across various machines. Suppose that there are l machines, and the points in the set C are partitioned across l machines, denoted as $\{M_1, \ldots, M_l\}$ where $\cup_{j=1}^{l} M_j = C$. The MPC model computes final solution in several rounds. In each round, each machine j ($j \in \{1, \ldots, l\}$) performs some computation on M_j, and communicates with a coordinator at the end of the round. The goal of the MPC model is to optimize the communication cost and the number of rounds.

Chierichetti *et al.* [15] introduced the definition of fairness with only two colors, requiring that the proportion of two colors has approximately equal representation in every cluster. Bercea *et al.* [8] proposed the notion of group fairness. They presented a 3-approximation with an additive 1 violation for the group fairness constraints using linear programming and min-cost flow network for the GFkC problem. The value of violation represents the extent to which the fairness constraints are violated (see [8] with details). Ahmadian *et al.* [2] studied the GFkC problem with only an upper bound constraint α. They presented a 3-approximation with an additive 2 violation (the definition of this violation is different from the one in [8]) using linear programming and min-cost flow network. For the GFkC problem under the condition that colors are allowed to overlap, a 4-approximation algorithm with $(4\Delta + 3)$ violation [6] and a 3-approximation algorithm with $(4\Delta + 3)$ violation [22] were presented, respectively, where Δ is the maximum number of colors a single point can belong to. Recently, Bera *et al.* [7] considered the GFkC problem in the MPC model. They gave a 2-round algorithm that communicates $O(mkl \log n)$ amount of data, and achieves a 9-approximation with an additive 7 violation.

Chen *et al.* [13] studied the matroid center problem that generalizes the DSFkC problem, and gave a 3-approximation with running time $\Omega(n^2 \log n)$. For the DSFkC problem, there was a $(3 \cdot 2^{m-1} - 1)$-approximation with running time $O(nkm^2 + km^4)$ based on a swap technique [27]. Jones, Lê Nguyên and Nguyen [24] improved the time complexity to $O(nk)$, and maintained the approximation factor 3 using the maximum matching method. Chiplunkar, Kale and Ramamoorthy [16] considered the DSFkC problem in the MPC model. They gave a 2-round algorithm that communicates $O(mkl \log n)$ amount of data, and achieves a $(17 + \epsilon)$-approximation.

For the GFkC and DSFkC problems in the MPC model, there exist some obstacles to obtain better approximation algorithms. It is known that for the GFkC and DSFkC problems in the single machine, the best known results have approximation ratio 3, respectively. However, for the above two problems in the MPC model, the best known results have approximation ratios 9 and $17 + \epsilon$, respectively. Therefore, the gaps between the ratios of the distributed algorithms in the MPC model and the ratios of the sequential algorithms are still large. The key idea of the previous distributed algorithms for the two problems in the MPC model is first to obtain a set of k centers on each machine. Then, each machine sends the k centers to the coordinator to obtain the final solution. The quality of the set of k centers obtained on each machine greatly impacts on the finding of the final approximate solution in the coordinator. However, the above process of obtaining k centers has an approximation loss of 2 on each machine, which is hard to find an approximation algorithm with same ratio of the sequential algorithm. Moreover, due to the existence of fairness constraints, it remains troublesome to obtain a solution satisfying fairness constraints with a small loss in approximation guarantee in the coordinator. Therefore, how to find a set of high-quality centers on each machine (the gap between the approximate solution obtained by the set of centers on each machine and the optimal solution is small), and how to obtain a desired feasible solution satisfying fairness constraints with a small approximation loss in the coordinator are still challenging for the GFkC and DSFkC problems in the MPC model. The previous distributed algorithm for the GFkC problem in the MPC model communicates with the coordinator by a factor of $O(\log n)$ due to the guess of optimal cost of given instance. Therefore, how to reduce the communication cost of the distributed algorithm for the GFkC problem in the MPC model is still challenging.

Our Contributions: In this paper, to overcome the above obstacles, we propose some new approximation results for the GFkC and DSFkC problems in the MPC model. The main contributions of our paper are summarized as follows.

- For the GFkC problem in the MPC model, we obtain a $(3+\epsilon)$-approximation with violation 1, which is almost as accurate as the best approximation ratio of the sequential algorithm for the GFkC problem. For the DSFkC problem in the MPC model, we obtain a $(4 + \epsilon)$-approximation without fairness violation, which is very close to the ratio 3 of the sequential algorithm for the DSFkC problem.
- Compared with the distributed algorithms in [7] for the GFkC problem, the optimal cost is guessed by considering all possible $O(n^2)$ distances for given instance, resulting in a factor $O(\log n)$ in communication cost. By guessing a lower bound of the optimal cost on each machine, and then applying a geometric guessing method, we present a process to find the final solution in the coordinator, and achieve better communication cost without factor $O(\log n)$.
- We test our distributed algorithms on real datasets, and the experiment results show that our algorithms perform better compared with the state-of-the-art algorithms.

We summarize the results in the literature and ours in Table 1. Formally, we have the following results.

Theorem 1.1. *Assume that the input data points are already distributed across the machines. There exists a 2-round MPC algorithm*

Table 1: Approximation results for the GFkC and DSFkC problems in the MPC model.

Problem	Approximation	Communication	Violation	Reference
GFkC	9	$O(kml \cdot \log n)$	7	[7]
	$3+\epsilon$	$O(kml \cdot (4/\epsilon)^D)$	1	Theorem 1.1
DSFkC	$17+\epsilon$	$O(kml)$	0	[16]
	$4+\epsilon$	$O(kml \cdot (4/\epsilon)^D)$	0	Theorem 1.2

that achieves a $(3+\epsilon)$-approximation with violation 1, and communicates $O(kml \cdot (4/\epsilon)^D)$ amount of data for the GFkC problem, where $\epsilon > 0$ is a parameter, and D is the doubling dimension of the input data points, respectively.

Theorem 1.2. *Assume that the input data points are already distributed across the machines. There exists a 2-round MPC algorithm that achieves a $(4+\epsilon)$-approximation, and communicates $O(kml \cdot (4/\epsilon)^D)$ amount of data for the DSFkC problem, where $\epsilon > 0$ is a parameter, and D is the doubling dimension of the input data points, respectively.*

2 PRELIMINARIES

Given a set C of points in a metric space $(\mathcal{X}, d)$, for a point $v \in C$ and a set $S \subseteq C$, let $d(v,S) = \min_{s\in S} d(v,s)$. For any $m \in \mathbb{N}^{\geq 1}$, let $[m]$ denote $\{1,\ldots,m\}$. For any nonempty subset $S \subseteq C$ of centers and any $s \in S$, a ball $B(s,r)$ is the set of points that are within a distance r from s, i.e., $B(s,r) = \{v \in C \mid d(s,v) \leq r\}$. For any nonempty subset $S, C' \subseteq C$, let $cost(S,C') = \max_{v\in C'} d(v,S)$ be the clustering cost of S for C'.

Definition 2.1 (the k-Center problem). Given a set C of points in a metric space $(\mathcal{X}, d)$ and an integer k, the goal is to find a subset $S \subseteq C$ of k centers such that $cost(S,C)$ is minimized.

Given an instance (C,d,k) of the k-Center problem, S is called a feasible solution if $S \subseteq C$ is a set of size k.

Definition 2.2 (the GFkC problem). Given a set C of points in a metric space $(\mathcal{X}, d)$, an integer k, a set of colors $H = \{1,\ldots,m\}$, m disjoint groups $\mathcal{G} = \{C_1,\ldots,C_m\}$ with $\bigcup_{h=1}^m C_h = C$, and two vectors $\alpha = (\alpha_1,\ldots,\alpha_m)$, $\beta = (\beta_1,\ldots,\beta_m)$, where the points in C_h are colored with color $h \in H$, and β_h, α_h are the lower and upper bounds on the proportions of group C_h points in each cluster, respectively, the goal is to find a set $S \subseteq C$ of k centers, and a mapping $\phi : C \to S$ such that the cost $\max_{v\in C} d(v,\phi(v))$ is minimized, and ϕ satisfies the following group fairness constraints.

$$\beta_h |O_i| \leq |O_i^h| \leq \alpha_h |O_i|, \forall i \in S, \forall h \in H \tag{1}$$

where $O_i = \{v \in C \mid \phi(v) = i\}$ is the set of points in cluster i, and $O_i^h = \{v \in C_h \mid \phi(v) = i\}$ is the set of points in cluster i with color h, respectively.

Given an instance $(C,d,k,\mathcal{G},H,\alpha,\beta)$ of the GFkC problem, a pair (S,ϕ) is called a feasible solution if $S \subseteq C$ is a set with size k, and $\phi : C \to S$ is a mapping satisfying constraint (1).

Definition 2.3 (the DSFkC problem). Given a set C of points in a metric space $(\mathcal{X}, d)$, an integer k, a set of colors $H = \{1,\ldots,m\}$, m disjoint groups $\mathcal{G} = \{C_1,\ldots,C_m\}$ with $\bigcup_{h=1}^m C_h = C$, and a vector $\gamma = (k_1,\ldots,k_m)$ with $\sum_{h=1}^m k_h = k$, where the points in C_h are colored with color $h \in H$, the goal is to find a subset $S \subseteq C$ of k centers such that the cost $\max_{v\in C} d(v,S)$ is minimized, and S satisfies the following data summarization fairness constraints.

$$|\{v \mid v \in S, v \in C_h\}| = k_h, \forall h \in H \tag{2}$$

Given an instance $(C,d,k,\mathcal{G},H,\gamma)$ of the DSFkC problem, S is called a feasible solution if $S \subseteq C$ with size k satisfies constraint (2).

Definition 2.4 (doubling dimension). Given a set C of points in a metric space $(\mathcal{X}, d)$, the doubling dimension of C is the smallest number D such that for any radius r and a point $v \in C$, all points in the ball $B(v,r)$ are always covered by the union of at most 2^D balls with radius $r/2$.

3 OBTAINING CANDIDATE SET ON EACH MACHINE

In this section, we show how to construct a candidate set, which is a set of centers on each machine. Since the quality of a set of centers obtained on each machine and the existence of fairness constraints greatly impact the finding of the approximate solutions in the coordinator, we first construct a candidate set on each machine. Using the method in [9], we can achieve a small gap between the approximation solution obtained by the candidate set and the optimal solution.

Theorem 3.1 ([9]). *Given an instance $\mathcal{I} = (C,d,k)$ of the k-Center problem and a parameter $\epsilon > 0$, assume that D is the doubling dimension of C, and τ is the cost of optimal solution of $\mathcal{I}$, respectively. Then, there is an algorithm that returns a subset $E \subseteq C$ with size $k \cdot (4/\epsilon)^D$ such that for any $v \in C$, $d(v,E) \leq \epsilon\tau$.*

For completeness, we refer to the algorithm in Theorem 3.1 by ϵ-CSC (see Algorithm 1). Given an instance $\mathcal{I} = (C,d,k)$ of the k-Center problem and a parameter $\epsilon > 0$, let E be the set of centers returned by algorithm ϵ-CSC. Formally, we call E an ϵ-candidate set of $\mathcal{I}$. In algorithm ϵ-CSC, it involves an important subroutine, which is a classic greedy algorithm (denoted as Greedy-kC) in [21] for solving the k-Center problem. Here we briefly review how Greedy-kC works. Given an instance (C,d,k) of the k-Center problem, Greedy-kC first selects an arbitrary point from C as center. Then, it iteratively selects the next center that is the farthest point from all chosen centers until k centers are chosen. Moreover, we have the following theorem given in [21].

Theorem 3.2 ([21]). *Greedy-kC is a 2-approximation algorithm for the k-Center problem.*

Here, we give a brief introduction of algorithm ϵ-CSC in [9]. Theorem 3.2 implies that Greedy-kC returns a 2-approximate solution

Algorithm 1: ϵ-CSC

Input: A set C of points, a metric d, a positive integer k, and a parameter $\epsilon > 0$
Output: A subset E of C

$E \leftarrow \emptyset$;
$S \leftarrow$ Greedy-kC(C, d, k);
$E \leftarrow S$;
while $cost(E, C) > (\epsilon/2) \cdot cost(S, C)$ **do**
  $v \leftarrow \arg\max_{v \in C} d(v, E)$;
  $E \leftarrow E \cup \{v\}$;
return E.

of the k-Center problem instance with only k centers. Intuitively, if we continue to pick more and more centers from C, then a better approximate solution can be obtained. Therefore, the general idea of algorithm ϵ-CSC is to use the greedy strategy of Greedy-kC on C to iteratively select points as centers such that the cost of the chosen centers is small. More precisely, for a given instance (C, d, k) of the k-Center problem and a parameter $\epsilon > 0$, algorithm ϵ-CSC starts with an empty-set E. Then, it runs Greedy-kC(C, d, k) to obtain a set S of k centers, and adds the centers in S to E. Finally, it continues to use the greedy strategy to select some centers from C, and adds them to E until $cost(E, C) \leq (\epsilon/2) \cdot cost(S, C)$ holds.

Let τ and τ_f denote the costs of optimal solutions of the k-Center problem instance and the GFkC (or DSFkC) problem instance, respectively. Then, we have $\tau \leq \tau_f$ since the feasible solution of the GFkC (or DSFkC) problem instance is also a feasible solution of the k-Center problem instance. Therefore, an ϵ-candidate set to the k-Center problem instance is also the one to the GFkC (or DSFkC) problem instance due to $\tau \leq \tau_f$. We now use the GFkC problem as an example to show how to construct an ϵ-candidate set in the MPC model. The analysis can be easily adapted to the DSFkC problem. For a given instance $\mathcal{I} = (C, d, k, \mathcal{G}, H, \alpha, \beta)$ of the GFkC problem and a parameter $\epsilon > 0$, assume that the points in C are partitioned across l machines. Let M_j be the set of points distributed to machine $j \in [l]$, and E_j be the output of algorithm ϵ-CSC(M_j, d, k, ϵ), respectively. Let $E = \cup_{j=1}^{l} E_j$.

Lemma 3.3. *Given an instance $\mathcal{I} = (C, d, k, \mathcal{G}, H, \alpha, \beta)$ of the GFkC problem and a parameter $\epsilon > 0$, assume that D is the doubling dimension of C, and τ_f is the cost of optimal solution of $\mathcal{I}$, respectively. Then, for any $v \in C$, $d(v, E) \leq \epsilon\tau_f$. Moreover, $|E| = kl \cdot (4/\epsilon)^D$.*

Proof. For each machine $j \in [l]$, let S_j be the output of Greedy-kC(M_j, d, k) when step 2 of ϵ-CSC(M_j, d, k, ϵ) is executed. Note that S_j is the set of k centers selected by ϵ-CSC. Since M_j is a subset of C, by Theorem 3.2, we have $cost(S_j, M_j) \leq 2\tau \leq 2\tau_f$. After calling the algorithm ϵ-CSC, we have $cost(E_j, M_j) \leq (\epsilon/2) \cdot cost(S_j, M_j)$. Thus, for any $v \in M_j$, $d(v, E_j) \leq cost(E_j, M_j) \leq (\epsilon/2) \cdot cost(S_j, M_j) \leq \epsilon\tau_f$. Combining all l machines, for any $v \in C$, we have $d(v, E) \leq \epsilon\tau_f$. Since there are l machines, by Theorem 3.1, $|E| = kl \cdot (4/\epsilon)^D$. □

For simplicity, we define a mapping $\pi : C \rightarrow E$ that maps each point $v \in M_j$ ($j \in [l]$) to its closest center in E_j. Therefore, by Lemma 3.3, we get that for any $v \in C$, $d(v, \pi(v)) \leq \epsilon\tau_f$. Note that many real-world datesets often have lower intrinsic dimensions [5], i.e., the set C has a low doubling dimension D. Moreover, once the dimension of the dataset is given, the doubling dimension of the dataset is fixed. Since the exact value of the doubling dimension of the testing dataset is often difficult to compute, it is usually assumed to be a constant [18]. Therefore, the construction of the set E does not cause large communication cost in the coordinator.

4 DISTRIBUTED ALGORITHMS

In this section, we show how to obtain the desired distributed algorithms for the GFkC and DSFkC problems in the MPC model. For the distributed algorithms in [7, 16], the final solutions are obtained based on the centers returned by each machine. However, it is hard to obtain feasible solutions with a small approximation loss in the coordinator due to the existence of fairness constraints. We overcome the obstacle caused by the fairness constraints, and prove theoretically that for the GFkC and DSFkC problems in the MPC model, there must exist feasible solutions with small factors $(3 + \epsilon)$ and $(4 + \epsilon)$ of the optimal solution in the coordinator, respectively. Given an instance $\mathcal{I} = (C, d, k, \mathcal{G}, H, \alpha, \beta)$ (or $\mathcal{I} = (C, d, k, \mathcal{G}, H, \gamma)$) of the GF$k$C (or DSF$k$C) problem, assume that D is the doubling dimension of C, and the points in C are distributed among l machines. Let M_j be the set of points distributed to machine $j \in [l]$. Moreover, we have $\cup_{j=1}^{l} M_j = C$. Assume that τ_f is the cost of optimal solution for given instance. In Subsection 4.3, we show how to obtain τ_f by a geometric guessing method in the coordinator with an approximation loss $(1 + \delta)$, where $\delta > 0$ is a parameter.

4.1 The GFkC Problem in the MPC Model

In this section, we consider the GFkC problem in the MPC model, and present a distributed algorithm, called Group-Fair-kC (see Algorithm 2), which can achieve a $(3 + \epsilon)$-approximation with an additive 1 violation for the group fairness constraints. The general idea of Group-Fair-kC is as follows. Given an instance $\mathcal{I} = (C, d, k, \mathcal{G}, H, \alpha, \beta)$ of the GFkC problem and a parameter $\epsilon > 0$, our algorithm has two rounds. In the first round, we run the algorithm ϵ-CSC(M_j, d, k, ϵ) to obtain a set $E_j = \{s_j^1, \ldots, s_j^t\} \subseteq M_j$ of t centers on each machine $j \in [l]$, where $t = k \cdot (4/\epsilon)^D$. Based on E_j, we construct a weighted set U_j of points with size mt as follows, where each point in U_j is associated with an integer weight. For each center $s_j^i \in E_j$ ($i \in [t]$) and each color $h \in H$, we add a point v_j^{ih} with the same position as s_j^i with weight w_j^{ih} to U_j, where w_j^{ih} is equal to the total number of points with color h such that s_j^i is the closest center in E_j (i.e., $w_j^{ih} = |\{v \in M_j \cap C_h \mid \pi(v) = s_j^i\}|$). Then, we send (E_j, U_j) to the coordinator for each machine $j \in [l]$ at the end of the round. Let $E = \cup_{j=1}^{l} E_j$ and $U = \cup_{j=1}^{l} U_j$. In the second round, we first run algorithm Greedy-kC(E, d, k) to obtain a set $S = \{s_1, \ldots, s_k\}$ of k centers. Then, we obtain the final solution by solving the Weighted Fair Assignment problem (see Definition 4.2) that assigns the weighted points in U to the centers in S.

Let $S = \{s_1, \ldots, s_k\} \subseteq E$ be the set of k centers returned by Greedy-kC(E, k, d) in the second round. The following lemma easily follows from Theorem 3.2 and Lemma 3.3.

Lemma 4.1. *For any $v \in C$, we have $d(v, S) \leq (2 + \epsilon)\tau_f$.*

We consider the following Weighted Fair Assignment problem.

Algorithm 2: Group-Fair-kC

Input: An instance $\mathcal{I} = (C, d, k, \mathcal{G}, H, \alpha, \beta)$ of the GFkC problem, the points in M_j distributed across the j-th machine ($j \in [l]$), and a parameter $\epsilon > 0$

Output: A feasible solution of $\mathcal{I}$

for $j = 1$ **to** l **do**
  $E_j \leftarrow \epsilon$-CSC(M_j, d, k, ϵ);
  $U_j \leftarrow \emptyset$;
  for $i = 1$ **to** $|E_j|$ **do**
    for $h = 1$ **to** m **do**
      $v_j^{ih} \leftarrow$ construct a point with the same position as s_j^i with color h;
      $w_j^{ih} \leftarrow |\{v \in M_j \cap C_h \mid \pi(v) = s_j^i\}|$;
      $U_j \leftarrow U_j \cup \{v_j^{ih}\}$;
Send (E_j, U_j) to the coordinator for each machine $j \in [l]$;
$S \leftarrow$ Greedy-kC$(\cup_{j=1}^{l} E_j, d, k)$;
$\phi \leftarrow$ solve the Weighted Fair Assignment problem on U and S;
return (S, ϕ).

Definition 4.2 (the Weighted Fair Assignment problem [7]). Given a weighted set U of points in a metric space $(\mathcal{X}, d)$, where each point $u \in U$ is associated with an integer weight w_u, an integer k, a set of colors $H = \{1, \ldots, m\}$, m disjoint groups $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_m\}$ with $\cup_{h=1}^{m} \mathcal{U}_h = U$, a set $S = \{s_1, \ldots, s_k\}$ of k centers, and two vectors $\alpha = \{\alpha_1, \ldots, \alpha_m\}$, $\beta = \{\beta_1, \ldots, \beta_m\}$, where the points in $\mathcal{U}_h$ are colored with color $h \in H$, the goal is to find a mapping $\psi : (U \times S) \to \mathbb{N} \cup \{0\}$ satisfying $\sum_{s \in S} \psi(u, s) = w_u$ for any $u \in U$, and the following weighted fairness constraints, i.e.,

$$\beta_h \le \frac{\sum_{u \in \mathcal{U}_h} \psi(u, s)}{\sum_{u \in U} \psi(u, s)} \le \alpha_h, \forall s \in S, \forall h \in H, \tag{3}$$

and such that the cost $\max_{u \in U} \max_{s \in S, \psi(u,s) > 0} d(u, s)$ is minimized.

Given an instance $(U, S, d, k, \mathcal{U}, H, \alpha, \beta)$ of the Weighted Fair Assignment problem, we call $\psi : (U \times S) \to \mathbb{N} \cup \{0\}$ a feasible solution if ψ satisfies constraint (3), and $\sum_{s \in S} \psi(u, s) = w_u$ for any $u \in U$. We define the cost of ψ as $cost(\psi) = \max_{u \in U} \max_{s \in S, \psi(u,s) > 0} d(u, s)$, which is the maximum distance between a point $u \in U$ and a center $s \in S$ with $\psi(u, s) > 0$. Note that a point $u \in U$ is assigned to a center $s \in S$ if and only if $\psi(u, s) > 0$. It is easy to get that the cost of optimal solution of the Weighted Fair Assignment problem instance is the same as the GFkC problem instance.

Lemma 4.3. *For the Weighted Fair Assignment problem instance $\mathcal{J} = (U, S, d, k, \mathcal{U}, H, \alpha, \beta)$, there must exist a solution ψ satisfying constraint (3) with $cost(\psi) \le (3+\epsilon)\tau_f$, where τ_f is the cost of optimal solution of $\mathcal{J}$.*

Proof. Let (S^*, ϕ^*) be an optimal solution of the GFkC problem instance $\mathcal{I}$ with cost τ_f, where $S^* = \{s_1^*, \ldots, s_k^*\}$ is the set of k optimal centers. Let $O^* = \{O_1^*, \ldots, O_k^*\}$ be the corresponding k optimal clusters under mapping ϕ^*. For any $i \in [k]$ and $h \in H$, let $O_i^*(h)$ be the set of points in O_i^* with color h. Obviously, we have $C = \cup_{i \in [k]} O_i^*$. For any $i \in [k]$, we have $O_i^* = \cup_{h \in H} O_i^*(h)$. For any $i \in [k]$, let $\sigma(s_i^*) = \arg\min_{s \in S} d(s, s_i^*)$ denote the closest center in S to s_i^*. By Lemma 4.1, we get that there exists a center in S with distance at most $(2+\epsilon)\tau_f$ to s_i^*. Thus, we have $d(s_i^*, \sigma(s_i^*)) \le (2+\epsilon)\tau_f$ since $\sigma(s_i^*)$ is the closest center in S to s_i^*. For any $v \in C$, let $\phi(v) = \sigma(\phi^*(v))$. We now prove that (S, ϕ) is a $(3+\epsilon)$-approximate solution of $\mathcal{I}$, i.e., the cost of (S, ϕ) is at most $(3+\epsilon)\tau_f$, and $\phi : C \to S$ satisfies the group fairness constraints.

For any $v \in O_i^*$ ($i \in [k]$), by the triangle inequality, we have

$$\begin{aligned} d(v, \phi(v)) = d(v, \sigma(\phi^*(v))) &\le d(v, \phi^*(v)) + d(\phi^*(v), \sigma(\phi^*(v))) \\ &\le \tau_f + (2+\epsilon)\tau_f \le (3+\epsilon)\tau_f. \end{aligned}$$

Hence, the cost of (S, ϕ) is at most $(3+\epsilon)\tau_f$. Since (S^*, ϕ^*) is a feasible solution of $\mathcal{I}$, for any $i \in [k]$ and $h \in H$, we have

$$\beta_h \le \frac{|O_i^*(h)|}{|O_i^*|} \le \alpha_h.$$

For any $s \in S$, let $N(s) = \{s_i^* \in S^* \mid \sigma(s_i^*) = s\}$ denote all centers in S^* such that s is the closest center. Note that $\{v \in C \mid \phi(v) = s\} = \cup_{s_i^* \in N(s)} O_i^*$. Similarly, for any $h \in H$, we have $\{v \in C_h \mid \phi(v) = s\} = \cup_{s_i^* \in N(s)} O_i^*(h)$. Consequently, for any $s \in S$ and $h \in H$, we have

$$\frac{|\{v \in C_h \mid \phi(v) = s\}|}{|\{v \in C \mid \phi(v) = s\}|} = \frac{\sum_{s_i^* \in N(s)} |O_i^*(h)|}{\sum_{s_i^* \in N(s)} |O_i^*|}.$$

By using the scaling technique, we have

$$\min_{s_i^* \in N(s)} \frac{|O_i^*(h)|}{|O_i^*|} \le \frac{\sum_{s_i^* \in N(s)} |O_i^*(h)|}{\sum_{s_i^* \in N(s)} |O_i^*|} \le \max_{s_i^* \in N(s)} \frac{|O_i^*(h)|}{|O_i^*|}.$$

Then, we have

$$\beta_h \le \frac{\sum_{s_i^* \in N(s)} |O_i^*(h)|}{\sum_{s_i^* \in N(s)} |O_i^*|} \le \alpha_h.$$

Thus, ϕ satisfies the group fairness constraints.

We now prove that for the Weighted Fair Assignment problem instance $\mathcal{J} = (U = \cup_{j=1}^{l} U_j, S, d, k, \mathcal{U}, H, \alpha, \beta)$, a mapping ψ satisfying constraint (3) with $cost(\psi) \le (3+\epsilon)\tau_f$ based on the solution (S, ϕ) of $\mathcal{I}$ can be constructed. For any point $v \in C$, assume that v is in machine $j \in [l]$ with color $h \in H$, i.e., $v \in M_j \cap C_h$. By the definition of the mapping π, we get that $\pi(v)$ is the closest center in E_j to v with $d(\pi(v), v) \le \epsilon\tau_f$. Suppose that $i \in [t]$ is the index of the center $\pi(v)$ in E_j. Recall that for each color $h \in H$, the weighted set U_j contains a point v_j^{ih} with the same position as $\pi(v) \in E_j$. Thus, we have $d(v_j^{ih}, v) \le \epsilon\tau_f$ since v_j^{ih} and $\pi(v)$ have the same positions. In the solution (S, ϕ), point v is assigned to the center $\phi(v) \in S$. Then, we assign 1 unit of weight of the point v_j^{ih} to $\phi(v)$, i.e., set $\psi(v_j^{ih}, \phi(v)) = 1$. By the triangle inequality, we have

$$d(v_j^{ih}, \phi(v)) \le d(v_j^{ih}, v) + d(v, \phi(v)) \le (3+\epsilon)\tau_f.$$

Thus, we have $cost(\psi) \le (3+\epsilon)\tau_f$.

We now prove that the mapping ψ satisfies constraint (3). By the above process, we get that the total weight of points with color $h \in H$ in U assigned to a center $s \in S$ is exactly equal to the number of points of this color assigned to s in the solution (S, ϕ), i.e., for any $s \in S$ and $h \in H$, we have $\sum_{u \in \mathcal{U}_h} \psi(u, s) = |\{v \in C_h \mid \phi(v) = s\}|$.

Then, for any $s \in S$, we have $\sum_{u\in U}\psi(u,s) = |\{v \in C \mid \phi(v) = s\}|$. Thus, for any $s \in S$ and $h \in H$, we have

$$\frac{\sum_{u\in\mathcal{U}_h}\psi(u,s)}{\sum_{u\in U}\psi(u,s)} = \frac{|\{v \in C_h \mid \phi(v) = s\}|}{|\{v \in C \mid \phi(v) = s\}|}.$$

Since the solution (S,ϕ) satisfies the group fairness constraints, the mapping ψ satisfies constraint (3). □

Lemma 4.3 shows the existence of a solution ψ of $\mathcal{J}$ satisfying constraint (3) with cost at most $(3+\epsilon)\tau_f$. Note that a solution ψ of $\mathcal{J}$ induces an assignment from the set U to the centers in S. The natural idea to obtain such an assignment is firstly to use the linear programming method, which results in a fractional assignment satisfying the group fairness constraints with cost at most $(3+\epsilon)\tau_f$. Then, we can obtain an integral assignment by using a rounding method that rounds the feasible fractional assignment obtained to an integral assignment. Therefore, we model the Weighted Fair Assignment problem instance $\mathcal{J} = (U, S, d, k, \mathcal{U}, H, \alpha, \beta)$ as a linear program, which can be solved by the method in [7].

Theorem 4.4 ([7]). *Given an instance $\mathcal{J}$ of the Weighted Fair Assignment problem, assume that $\phi' : U \to S$ is a fractional assignment obtained by the linear programming with cost at most $(3+\epsilon)\tau_f$. Then, there is a rounding algorithm that returns an integral assignment $\overline{\phi}$ with cost at most $(3+\epsilon)\tau_f$ with an additive 7 violation for group fairness constraints.*

By Theorem 4.4, we can obtain an integral assignment $\overline{\phi} : U \to S$ with an additive 7 violation that assigns each point $u \in U$ to a center $s \in S$ with $d(u,s) \le (3+\epsilon)\tau_f$. However, the above solution is for the weighted points in U, we need to convert it to obtain a solution for the original points in C. Based on the integral assignment $\overline{\phi}$, we can obtain a solution (S,ϕ) for the GFkC problem instance, where $\phi : C \to S$ is a mapping that assigns each point in C to a center in S. For any $v \in C$, assume that point v has color $h \in H$, and we get that $\pi(v)$ is the closest center in E to v. Recall that the set U contains a point with color h, denoted by u^h, with the same position as $\pi(v)$. Assume that in the above integral assignment $\overline{\phi}$, the weight point u^h is assigned to a center $s \in S$. Then, we assign v to s, i.e., set $\phi(v) = s$. Obviously, by Theorem 3.1, the above process incurs an additional cost of $\epsilon\tau_f$. Therefore, the cost of (S,ϕ) is still at most $(3+\epsilon)\tau_f$.

In fact, the violation can be improved to 1 instead of 7 by using the rounding method in [8]. However, compared with the rounding method in [8], the method in [7] is fast and scalable in practice . Since the sizes of E_j and U_j are $O(k\cdot(4/\epsilon)^D))$ and $O(km\cdot(4/\epsilon)^D))$ for each machine $j \in [l]$, respectively, the algorithm Group-Fair-kC communicates $O(kml\cdot(4/\epsilon)^D)$ amount of data. By the above discussion, Theorem 1.1 can be proved.

4.2 The DSFkC Problem in the MPC Model

In this section, we consider the DSFkC problem in the MPC model, and present a distributed algorithm, called Data-Summarization-Fair-kC (see Algorithm 3), which achieves a $(4+\epsilon)$-approximation without fairness violation. The general idea of Data-Summarization-Fair-kC is as follows. Given an instance $\mathcal{I} = (C, d, k, \mathcal{G}, H, \gamma)$ of the DSFkC problem and a parameter $\epsilon > 0$, our algorithm has two rounds. In the first round, we run the algorithm ϵ-CSC(M_j, d, k, ϵ)

Algorithm 3: Data-Summarization-Fair-kC

Input: An instance $\mathcal{I} = (C, d, k, \mathcal{G}, H, \gamma)$ of the DSFkC problem, the points in M_j distributed across the j-th machine ($j \in [l]$), and a parameter $\epsilon > 0$

Output: A feasible solution of $\mathcal{I}$

```
for j = 1 to l do
    E_j ← ε-CSC(M_j, d, k, ε);
    U_j ← E_j;
    for i = 1 to |E_j| do
        for h = 1 to m do
            v_j^{ih} ← a point in M_j with color h such that s_j^i is
              the closest center in E_j;
            U_j ← U_j ∪ {v_j^{ih}};
Send (E_j, U_j) to the coordinator for each machine j ∈ [l];
S ← Greedy-kC(∪_{j=1}^{l} E_j, d, k);
Ŝ ← Get-Solution(C, d, k, G, H, γ, S, ∪_{j=1}^{l} U_j, τ_f, ε);
return Ŝ.
```

to obtain a set $E_j = \{s_j^1, \ldots, s_j^t\} \subseteq M_j$ of t centers on each machine $j \in [l]$, where $t = k\cdot(4/\epsilon)^D$. Based on E_j, we construct a representative set U_j of points with size mt as follows. We start with $U_j = E_j$. Then, for each center $s_j^i \in E_j$ ($i \in [t]$) and each color $h \in H$, we add a point $v_j^{ih} \in M_j$ with color h to U_j (if it exists) such that $\pi(v_j^{ih}) = s_j^i$ (note that there may be multiple points in M_j with color h such that the closest center is s_j^i, and we select any point v_j^{ih} from these points). Next, we send (E_j, U_j) to the coordinator for each machine $j \in [l]$ at the end of the round. Let $E = \cup_{j=1}^{l} E_j$ and $U = \cup_{j=1}^{l} U_j$. In the second round, we first run algorithm Greedy-kC(E, d, k) to obtain a set $S = \{s_1, \ldots, s_k\}$ of k centers. Then, we get the final solution by calling the algorithm Get-Solution (see Algorithm 4), which finds a maximum matching between a bipartite graph based on S and U. The matching method in Get-Solution is similar to that of [16, 24], which is a commonly technique used for solving the DSFkC problem to get k centers satisfying the data summarization fairness constraints.

Lemma 4.5. *Given an instance $\mathcal{I} = (C, d, k, \mathcal{G}, H, \gamma)$ of the DSFkC problem and a parameter $\epsilon > 0$, Get-Solution returns a set $\hat{S}$ satisfying the data summarization fairness constraints such that for any $v \in C$, $d(v, \hat{S}) \le (4+\epsilon)\tau_f$.*

Proof. Let $E = \cup_{j=1}^{l} E_j$ and $U = \cup_{j=1}^{l} U_j$. Let $S = \{s_1, \ldots, s_k\}$ be the set of k centers returned by Greedy-kC(E, d, k). Let $S^* = \{s_1^*, \ldots, s_k^*\}$ be an optimal solution of $\mathcal{I}$ with cost τ_f. We first prove the existence of $\hat{S}$. For each $s_i^* \in S^*$ ($i \in [k]$), by Lemma 4.1, we get that there exists a center in S with distance at most $(2+\epsilon)\tau_f$ to s_i^*. Assume that s_i^* is in machine $j \in [l]$, i.e., $s_i^* \in M_j$. By the definition of the mapping π, $\pi(s_i^*)$ is the closest center in E_j to s_i^* with $d(s_i^*, \pi(s_i^*)) \le \epsilon\tau_f$. Recall that for each color $h \in H$, the set U_j contains a point with color h such that $\pi(s_i^*)$ is the closest center in E_j. Therefore, the set U_j must contain a point, denoted by $f(\pi(s_i^*))$, with the same color as s_i^* (possibly s_i^* itself) with

Algorithm 4: Get-Solution

Input: An instance $\mathcal{I} = (C, d, k, \mathcal{G}, H, \gamma)$ of the DSFkC problem, a set S of k centers, a set U of points, and parameters $\tau_f, \epsilon > 0$

Output: A feasible solution $\hat{S}$ of $\mathcal{I}$

Let $V_1 = V_2 = \emptyset, A = \emptyset$;
for $i = 1$ **to** k **do**
  Construct a vertex u_i, and add it to V_1;
for $h = 1$ **to** m **do**
  Construct a set V_h of k_h identical vertices, and add the vertices in V_h to V_2;
Let $G = (V_1 \cup V_2, A)$;
for $i = 1$ **to** k **do**
  for $h = 1$ **to** m **do**
    if $\exists\, v \in U \cap C_h$ **and** $d(s_i, v) \le (2+\epsilon)\tau_f$ **then**
      For each vertex $w \in V_h$, add edge (u_i, w) to A;
Find a maximum matching M of G;
$\hat{S} \leftarrow \emptyset$;
for *each edge* $(a, b) \in M$ **do**
  Let p be a point in U with color h such that $d(s_i, p) \le (2+\epsilon)\tau_f$, where a is the corresponding vertex of center $s_i \in S$, and b is in V_h, respectively;
  $\hat{S} \leftarrow \hat{S} \cup \{p\}$;
return $\hat{S}$.

$d(\pi(s_i^*), f(\pi(s_i^*))) \le \epsilon\tau_f$. Since E is a subset of C, by Lemma 3.2, there exists a center $s \in S$ such that $d(\pi(s_i^*), s) \le 2\tau_f$. By the triangle inequality, we have

$$d(f(\pi(s_i^*)), s) \le d(f(\pi(s_i^*)), \pi(s_i^*)) + d(\pi(s_i^*), s) \le (2+\epsilon)\tau_f.$$

Let $\hat{S} = \{f(\pi(s_1^*)), \ldots, f(\pi(s_k^*))\}$. Hence, there must exist a set $\hat{S} \subseteq U$ satisfying the data summarization fairness constraints such that for any $f(\pi(s_i^*)) \in \hat{S}$ $(i \in [k])$, there is a center $s \in S$ with $d(f(\pi(s_i^*)), s) \le (2+\epsilon)\tau_f$. Recall that Lemma 4.1 shows that for any $v \in C$, there exists a center in S with distance at most $(2+\epsilon)\tau_f$ to v. Therefore, by the triangle inequality, we get that for any $v \in C$, there must exist a center $\hat{s} \in \hat{S}$ with $d(v, \hat{s}) \le (4+\epsilon)\tau_f$.

We now prove that algorithm Get-Solution can give such a solution $\hat{S}$ of $\mathcal{I}$. Get-Solution starts with a bipartite graph $G = (V_1 \cup V_2, E)$. The left vertex set V_1 contains k vertices in total, where for each center $s_i \in S$ $(i \in [k])$, it contains one vertex. The right vertex set $V_2 = \cup_{h=1}^{m} V_h$ contains k vertices in total, where for each color $h \in H$, the set V_h contains k_h identical vertices. For each vertices $a \in V_1$ and $b \in V_2$, let (a, b) denote an edge between a and b For each $i \in [k]$ and $h \in H$, if there is a point $v \in U$ with color h such that $d(s_i, v) \le (2+\epsilon)\tau_f$, then the corresponding vertex u_i is connected to all vertices in V_h.

Let M be a maximum matching returned by the Ford-Fulkerson algorithm based on G that runs in polynomial time. Then, M immediately induces a solution $\hat{S}$ as follows. For each edge (a, b) in M, assume that vertex a corresponds to center $s_i \in S$, and vertex b is in V_h. We add a point $p \in U$ with color h to $\hat{S}$ such that $d(p, s_i) \le (2+\epsilon)\tau_f$. Since $|V_h| = k_h$, $\hat{S}$ contains k_h points with color h. Therefore, $\hat{S}$ is a set of k centers satisfying the data summarization fairness constraints. □

Since the sizes of E_j and U_j are $O(k \cdot (4/\epsilon)^D))$ and $O(km \cdot (4/\epsilon)^D))$ for each machine $j \in [l]$, respectively, the algorithm Data-Summarization-Fair-kC communicates $O(kml \cdot (4/\epsilon)^D)$ amount of data. By the above discussion, Theorem 1.2 can be proved.

4.3 Obtaining the Optimal Cost

In this section, we show how to obtain the optimal cost for given instance by a geometric guessing method. For the completeness of our algorithms, we do not provide the specific process of guessing the optimal cost in Group-Fair-kC and Data-Summarization-Fair-kC. Given an instance $\mathcal{I}$ of the DSFkC (or DSFkC) problem, assume that τ_f is the cost of the optimal solution of $\mathcal{I}$. Let Low and Upp be a lower bound and an upper bound of τ_f, respectively. Thus, we have $Low \le \tau_f \le Upp$. Then, for an arbitrarily small parameter δ, we can guess the optimal cost τ_f from $\{Low, Low(1+\delta), Low(1+\delta)^2, \ldots, Upp\}$, which has at most $\log_{1+\delta}(Upp/Low)$ choices. More precisely, in our distributed algorithms, for each machine $j \in [l]$, we first compute $r_j = cost(S_j, M_j)/2$, and send it to the coordinator, where S_j is the set of the first k centers in E_j. Note that the process of sending r_j to the coordinator will not increase the communication cost. It is easy to get that $r_j \le \tau_f$. Thus, we have $\max_{j\in[l]} r_j \le \tau_f$, i.e., $\max_{j\in[l]} r_j$ is a lower bound of τ_f. Then, we can run the algorithm Group-Fair-kC (or Data-Summarization-Fair-kC) in the coordinator with a parameter starting at $\max_{j\in[l]} r_j$ until it successfully finds a feasible solution of given instance. The role of δ is to guess the optimal cost τ_f of given instance. When the algorithm returns a feasible solution, the guessing value is at least τ_f and at most $(1+\delta)\tau_f$. For such case, the clustering cost is at most $(3+\epsilon)(1+\delta)\tau_f$. Therefore, δ does not influence the analysis of the approximation ratio. Moreover, the geometric guessing process is executed in the coordinator without additional communication cost.

5 EXPERIMENTS

In this section, we compare our proposed distributed algorithms Group-Fair-kC and Data-Summarization-Fair-kC with the state-of-the-art algorithms.

Datasets. We conduct experiments on 6 real datasets frequently used in fair clustering. For the GFkC problem, we use Reuters, Bank, and Creditcard. Reuters [2] dataset contains 50 English language texts from each of 50 authors, where author is considered as the sensitive attribute to generate 50 groups. Bank [6] dataset contains one record for each phone call in a marketing campaign by a Portuguese banking institution, where marital is used as the sensitive attribute to generate 3 groups. Creditcard [6] dataset contains a information for each credit card holders from Taiwan, where marital is considered as the sensitive attribute to generate 4 groups. For the DSFkC problem, we use SushiA, Adult, and Celeb-A from [16], and follow the settings in [16] to obtain the sensitive attributes for the three datasets. The datasets used in our experiments are summarized in Table 2.

Table 2: Datasets.

Problem	Dataset	Size	Dimension	Number of Groups
GFkC	Reuters	2,500	10	50
	Bank	4,000	3	3
	Creditcard	30,000	13	4
DSFkC	SushiA	5,000	11	2, 6, 12
	Adult	30,000	6	2, 5, 10
	Celeb-A	200,000	15360	2

Baseline Algorithms. We use three baseline algorithms in our experiments. The first one is the algorithm Greedy-kC that is a 2-approximation for the k-Center problem. We compare the cost returned by Greedy-kC in our experiments that is the maximum distance from a point to its closest center, and is also a lower bound of the optimal cost for the GFkC (or DSFkC) problem instance. Note that Greedy-kC is not strictly a baseline algorithm, and is used to provide a lower bound for other algorithms in our experiments. The second one is the distributed algorithm given in [7] that is a 9-approximation for the GFkC problem. The last one is the distributed algorithm given in [16] that is a $(17+\epsilon)$-approximation for the DSFkC problem. We denote the latter two distributed algorithms as Bera and Chiplunkar, respectively. Instead of comparing the costs returned by Bera and Chiplunkar, we follow the settings in [16], and use the ratios between its costs and the lower bounds returned by Greedy-kC, respectively. Note that the ratios are not the approximation ratios of the GFkC and DSFkC problems. Similarly, we also compare the ratios between the costs returned by our distributed algorithms and the lower bounds in experiments.

Results. Both our distributed algorithms first run algorithm ϵ-CSC(M_j, d, k, ϵ) to obtain a set E_j of $k \cdot (4/\epsilon)^D$ centers on each machine $j \in [l]$. However, the doubling dimension D is hard to compute in practice, and is not used as a parameter in our experiments. We follow the settings in [9], which varies the size of the set obtained on each machine in experiments. In our experiments, for each machine $j \in [l]$, we set $|E_j| = 10k$. Tables 3 and 4 compare the ratios between the costs of distributed algorithms for the GFkC and DSFkC problems and the costs returned by Greedy-kC, respectively. Note that in these tables, we abbreviate Greedy-kC as G, Bera as B, and Chiplunkar as C, respectively. Moreover, following the setting in [22], for the GFkC problem in Table 3, the parameters are fixed as $k = 4, \beta = 0$, and α is selected with feasible value. For the DSFkC problem in Table 4, the parameters of fairness constraints are selected from [16]. As shown in Table 3 and 4, our distributed algorithms achieve a lower cost to find fair clusters with various datasets and parameters.

Setups. We follow the settings in [7] with $l = 10$, i.e., all the algorithms in our experiments are tested on 10 machines. We follow the settings in [16] with $\delta = 0.1$. We use CPLEX to solve the linear program. For hardware, all the experiments are conducted on 72 Intel Xeon Gold 6230 CPUs and 500GB memory.

Table 3: Comparison of ratios between the costs of distributed algorithms for the GFkC problem and the costs returned by Greedy-kC on real datasets.

Dataset	α	Greedy-kC	Bera	Our
Reuters	0.05	3.01	3.57	**2.49**
	0.2	3.01	3.52	**2.49**
	0.4	3.01	3.52	**2.49**
Bank	0.8	2.85×10^3	31.27	**13.9**
	0.9	2.85×10^3	16.05	**13.9**
	1.0	2.85×10^3	9.20	**7.13**
Creditcard	0.6	1.65×10^6	3.49	**2.22**
	0.7	1.65×10^6	3.45	**2.18**
	0.8	1.65×10^6	3.45	**2.18**

Table 4: Comparison of ratios between the costs of distributed algorithms for the DSFkC problem and the costs returned by Greedy-kC on real datasets.

Dataset	Constraint	Greedy-kC	Chiplunkar	Our
Adult	[2, 2]	6.87	2.06	**1.99**
	[2]*5	5.21	2.32	**2.02**
	[2]*10	4.09	2.64	**2.02**
CelebA	[2, 2]	4.53×10^4	1.96	**1.83**
	[1, 3]	4.53×10^4	1.96	**1.91**
	[3, 1]	4.53×10^4	1.97	**1.93**
SushiA	[2, 2]	11.5	2.43	**2.17**
	[2]*6	9	2.67	**2.11**
	[2]*12	8	2.25	**2.12**

6 CONCLUSIONS

In this paper, we consider the GFkC and DSFkC problems in the MPC distributed model. Due to the fairness constraint, the problem is considerably more challenging than its non-fair counterpart. It is thus a non-trivial task to obtain a solution that satisfies the fairness constraints and meanwhile achieves a small approximation ratio in the coordinator model. The main contributions of this paper are: we show that there exist a $(3+\epsilon)$-approximate solution and a $(4+\epsilon)$-approximate solution for the GFkC and DSFkC problems, respectively, in the MPC distributed model, which are almost as accurate as the sequential algorithms. We show experimentally that our proposed distributed algorithms achieve better performance compared with the state-of-the-art algorithms. Moreover, we believe that our proposed distributed algorithms are useful in the k-Center problem with other fairness constraints in the MPC distributed model.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (62172446, 62350004, 62332020), Open Project of Xiangjiang Laboratory (22XJ02002), and Central South University Research Programme of Advanced Interdisciplinary Studies (2023QYJC023). This work was also carried out in part using computing resources at the High Performance Computing Center of Central South University.

REFERENCES

[1] Gagan Aggarwal, Rina Panigrahy, Tomás Feder, Dilys Thomas, Krishnaram Kenthapadi, Samir Khuller, and An Zhu. 2010. Achieving Anonymity via Clustering. *ACM Transactions on Algorithms* 6, 3 (2010), 49:1–49:19.

[2] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. 2019. Clustering without Over-Representation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 267–275.

[3] Sara Ahmadian and Chaitanya Swamy. 2016. Approximation Algorithms for Clustering Problems with Lower Bounds and Outliers. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming*. 69:1–69:15.

[4] Hyung-Chan An, Aditya Bhaskara, Chandra Chekuri, Shalmoli Gupta, Vivek Madan, and Ola Svensson. 2015. Centrality of Trees for Capacitated k-Center. *Mathematical Programming* 154, 1-2 (2015), 29–53.

[5] Mikhail Belkin. 2004. Problems of Learning on Manifolds. *The University of Chicago* (2004).

[6] Suman Kalyan Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. 2019. Fair Algorithms for Clustering. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 4955–4966.

[7] Suman K Bera, Syamantak Das, Sainyam Galhotra, and Sagar Sudhir Kale. 2022. Fair k-Center Clustering in MapReduce and Streaming Settings. In *Proceedings of the ACM Web Conference 2022*. 1414–1422.

[8] Ioana Oriana Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R. Schmidt, and Melanie Schmidt. 2019. On the Cost of Essentially Fair Clusterings. In *Proceedings of the 22nd International Conference on Approximation Algorithms for Combinatorial Optimization Problems and 23rd International Conference on Randomization and Computation*. 18:1–18:22.

[9] Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. 2019. Solving k-Center Clustering (with Outliers) in MapReduce and Streaming, almost as Accurately as Sequentially. *Proceedings of the VLDB Endowment* 12, 7 (2019), 766–778.

[10] Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. 2016. The Non-Uniform k-Center Problem. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming*. 67:1–67:15.

[11] Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. 2001. Algorithms for Facility Location Problems with Outliers. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*. 642–651.

[12] Shiri Chechik and David Peleg. 2015. The Fault-Tolerant Capacitated K-center Problem. *Theoretical Computer Science* 566 (2015), 12–25.

[13] Danny Z. Chen, Jian Li, Hongyu Liang, and Haitao Wang. 2016. Matroid and Knapsack Center Problems. *Algorithmica* 75, 1 (2016), 27–52.

[14] Xingyu Chen, Brandon Fain, Liang Lyu, and Kamesh Munagala. 2019. Proportionally Fair Clustering. In *Proceedings of the 36th International Conference on Machine Learning*. 1032–1041.

[15] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair Clustering Through Fairlets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 5029–5037.

[16] Ashish Chiplunkar, Sagar Kale, and Sivaramakrishnan Natarajan Ramamoorthy. 2020. How to Solve Fair k-center in Massive Data Models. In *Proceedings of the 37th International Conference on Machine Learning*. 1877–1886.

[17] Marek Cygan, MohammadTaghi Hajiaghayi, and Samir Khuller. 2012. LP Rounding for k-Centers with Non-uniform Hard Capacities. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science*. 273–282.

[18] Hu Ding, Haikuo Yu, and Zixiu Wang. 2019. Greedy Strategy Works for k-Center Clustering with Outliers and Coreset Construction. In *Proceedings of the 27th Annual European Symposium on Algorithms*. 40:1–40:16.

[19] Cristina G. Fernandes, Samuel P. de Paula, and Lehilton L. C. Pedrosa. 2018. Improved Approximation Algorithms for Capacitated Fault-Tolerant k-Center. *Algorithmica* 80, 3 (2018), 1041–1072.

[20] M. R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. WH Freeman.

[21] Teofilo F. Gonzalez. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science* 38 (1985), 293–306.

[22] Elfarouk Harb and Ho Shan Lam. 2020. KFC: A Scalable Approximation Algorithm for k-center Fair Clustering. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. 14509–14519.

[23] Dorit S. Hochbaum and David B. Shmoys. 1985. A Best Possible Heuristic for the k-Center Problem. *Mathematics of Operations Research* 10, 2 (1985), 180–184.

[24] Matthew Jones, Huy Lê Nguyên, and Thy Nguyen. 2020. Fair k-centers via Maximum Matching. In *Proceedings of the 37th International Conference on Machine Learning*. 4940–4949.

[25] Matthew Kay, Cynthia Matuszek, and Sean A Munson. 2015. Unequal Representation and Gender Stereotypes in Image Search Results for Occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3819–3828.

[26] Samir Khuller and Yoram J. Sussmann. 2000. The Capacitated K-Center Problem. *SIAM Journal on Discrete Mathematics* 13, 3 (2000), 403–418.

[27] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. 2019. Fair k-Center Clustering for Data Summarization. In *Proceeding of the 36th International Conference on Machine Learning*. 3448–3457.

[28] Bo Li, Lijun Li, Ankang Sun, Chenhao Wang, and Yingfan Wang. 2021. Approximate Group Fairness for Clustering. In *Proceedings of the 38th International Conference on Machine Learning*. 6381–6391.

[29] Sepideh Mahabadi and Ali Vakilian. 2020. Individual Fairness for k-Clustering. In *Proceedings of the 37th International Conference on Machine Learning*. 6586–6596.

[30] Evi Micha and Nisarg Shah. 2020. Proportionally Fair Clustering Revisited. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming*. 85:1–85:16.

[31] Maryam Negahbani and Deeparnab Chakrabarty. 2021. Better Algorithms for Individually Fair k-Clustering. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*. 13340–13351.