

Improved Streaming Algorithm for Fair k -Center Clustering

Longkun Guo¹, Zeyu Lin¹, Chaoqi Jia², Chao Chen²

¹School of Mathematics and Statistics,
Fuzhou University, Fuzhou 350116, China

²School of Accounting, Information Systems and Supply Chain,
RMIT University, Melbourne, VIC 3000, Australia

Abstract

Many real-world applications pose challenges in incorporating fairness constraints into the k -center clustering problem, where the dataset consists of m demographic groups, each with a specified upper bound on the number of centers to ensure fairness. Focusing on big data scenarios, this paper addresses the problem in a streaming setting, where data points arrive one by one sequentially in a continuous stream. Leveraging a structure called the λ -independent center set, we propose a one-pass streaming algorithm that first computes a reserved set of points during the streaming process. Then, for the post-streaming process, we propose an approach for selecting centers from the reserved point set by analyzing all three possible cases, transforming the most complicated one into a specially constrained vertex cover problem in an auxiliary graph. Our algorithm achieves a tight approximation ratio of 5 while consuming $O(k \log n)$ memory. It can also be readily adapted to solve the offline fair k -center problem, achieving a 3-approximation ratio that matches the current state of the art. Furthermore, we extend our approach to a semi-structured data stream, where data points from each group arrive in batches. In this setting, we present a 3-approximation algorithm for $m = 2$ and a 4-approximation algorithm for general m . Lastly, we conduct extensive experiments to evaluate the performance of our approaches, demonstrating that they outperform existing baselines in both clustering cost and runtime efficiency.

1 Introduction

Fair k -center clustering is a popular problem in various fields including data summarization (Kleindessner, Awasthi, and Morgenstern 2019; Angelidakis et al. 2022) and machine learning (Chierichetti et al. 2017; Jones, Nguyen, and Nguyen 2020). The problems have been widely studied, with many definitions of fairness proposed and corresponding approximation algorithms discussed, such as group fairness (Chierichetti et al. 2017; Wu et al. 2024; Backurs et al. 2019), data summarization fairness (Kleindessner, Awasthi, and Morgenstern 2019; Jones, Nguyen, and Nguyen 2020; Wu et al. 2024; Lin, Guo, and Jia 2024), colorful fairness (Bandyapadhyay et al. 2019; Anegg, Vargas Koch, and Zenklusen 2022; Jia, Sheth, and Svensson 2022) and so on. In this paper, we focus on the data summarization fairness k -center problem. We consider a dataset S of size n divided

into m disjoint groups, as $S = \{S_1 \cup \dots \cup S_m\}$. Our goal is to select k clusters to minimize the maximum distance from each point to its nearest center, while the number of centers chosen from each group S_l is bound by k_l . This formulation can avoid the biased for some sensitive features by controlling the number of objects from each category in the output. For instance, it can dictate the number of movies from each genre shown to a user in a recommendation system or limit the number of old messages included in a summary of a user's feed (Mahabadi and Trajanovski 2024).

However, a traditional challenge in clustering is the need to process large-scale datasets across numerous applications. In such scenarios, storing the entire input in memory becomes impractical, giving rise to the streaming model. In this model, data points arrive sequentially in a stream, and only a limited portion can be retained in memory due to space constraints. The streaming algorithm must decide, upon the arrival of each data point, whether to store it or discard it. Notably, designing effective streaming algorithms is more challenging than developing offline algorithms, as decisions must be made based on partial information rather than having access to the entire dataset. Motivated by this challenge, we aim to propose the approximation algorithms for the fair k -center problem under the streaming setting.

1.1 Related Work

Data summarization fairness k -center. To the best of our knowledge, the fair k -center clustering problem was first formally addressed in the context of data summarization by Kleindessner, Awasthi, and Morgenstern (2019). They proposed a 5-approximation algorithm for the case of two groups and a $(3 \times 2^{m-1} - 1)$ -approximation for the general case with m groups. After them, the improvement was made by Jones, Nguyen, and Nguyen (2020), who reduced the approximation ratio to 3 for arbitrary m groups. Prior to these two relevant works, the matroid center problem which includes a generalization of the k -center problem that enforces a matroid constraint on the set of centers rather than a simple cardinality constraint, was studied by Chen et al. (2016). They developed a 3-approximation algorithm, although with considerably higher time complexity. More recently, Chen et al. (2024) introduced the fair k -supplier problem, which involves selecting k facilities from a dataset partitioned into m disjoint groups, subject to group-wise upper bounds on the number

of facilities selected. They presented a 5-approximation algorithm for this problem by maximum matching techniques.

Computation of k -center for bigdata. For large-scale datasets in real-time streaming scenarios, streaming k -center clustering was first studied by Matthew McCutchen and Khuller (2008), which provided a streaming $(4 + \epsilon)$ -approximation algorithm with outliers, where up to z input points can be dropped and $O(kz/\epsilon)$ memory was consumed. Then, a deterministic one-pass streaming algorithm was developed for the same problem by Ceccarello, Pietracaprina, and Pucci (2019), achieving an improved approximation ratio $(3 + \epsilon)$ with $O((k + z)(96/\epsilon)^d)$ memory with d -dimension. For the matroid center problem, a $(17 + \epsilon)$ -approximation one-pass algorithm with a running time $O_\epsilon((nk + k^{3.5}) + k^2 \log(\Lambda))$ has been developed by Kale (2019), where k is the rank of the matroid, Λ is the aspect ratio of the metric, and ϵ terms are hidden by the O_ϵ notation. For the fair k -center problem, Chiplunkar, Kale, and Ramamoorthy (2020) provided a distributed algorithm achieving $(17 + \epsilon)$ approximation ratio with $O(kn/l + mk^2l)$ running time for l processors. They also developed a two-pass streaming algorithm with approximation ratio 3 for the fair k -center problem. Recently, the fair range k -center was addressed by Nguyen, Nguyen, and Jones (2022) with an achieved approximation ratio 13. In addition, a one-pass streaming algorithm for fair k -center clustering proposed by Lin, Guo, and Jia (2024) achieved the state-of-the-art approximation ratio of 7. In this paper, we improve the approximation ratio to 5 by utilizing the independent centers and modifying the post-streaming algorithms.

1.2 Our Contributions

The main contribution of this paper is summarized as follows:

- Devise a one-pass streaming algorithm for fair k -center, achieving an improved approximation ratio of 5 while consuming $O(k \log n)$ memory. This significantly improves the previous SOTA ratio 7 due to Lin, Guo, and Jia (2024).
- For semi-structured data streams where data points of each group S_l are streamed as a batch, we achieve ratios 3 and 4 for $m = 2$ and general m , respectively.
- Construct extensive experiments on real-world datasets to demonstrate the practical performance gains of our algorithms in both clustering accuracy and runtime efficiency.

Notably, we also provide an example to show that the achieved ratio 5 is tight under the $o(n)$ memory constraint. Moreover, the streaming algorithm with ratio 5 can be slightly tuned to achieve the SOTA ratio 3 for offline fair k -center previously due to Jones, Nguyen, and Nguyen (2020).

2 Preliminary and Algorithmic Framework

Let S be a finite set of data points with size n distributed in the metric space where $d : S \times S \rightarrow \mathbb{R}_{\geq 0}$ is the distance on S that satisfies the triangle inequality. For a given parameter $k \in \mathbb{N}$, the traditional k -center clustering problem is to select a set of k points $C \subseteq S$ to serve all points in S such that $\max_{s \in S} d(s, C)$ is minimized, where $d(s, C) = \min_{c \in C} d(s, c)$ is the distance between a point

s and the center set C . Let the data set be divided into m disjoint groups $S = \{S_1 \cup \dots \cup S_m\}$, where there exists a fairness constraint that the number of chosen centers from group S_l is constrained by an upper bound k_l , where k_l is the fair constraint for each group l for $l \in [m]$ and $\sum_{l=1}^m k_l = k$, where we use $[m] = \{1, 2, \dots, m\}$ for brevity. Then, the fair k -center clustering problem is to find a center set C satisfying the formulation as follows:

$$\begin{aligned} \min_{C \subseteq S} \quad & \max_{s \in S} d(s, C) \\ \text{s.t.} \quad & |C \cap S_l| \leq k_l, \forall l \\ & |C \cap S| = \sum_{l=1}^m |C \cap S_l| \leq k. \end{aligned}$$

2.1 Algorithmic Framework

In this subsection, we introduce the general framework of our algorithms. Throughout this paper, we assume that the optimal radius of the fair k -center problem is known as r^* (i.e., $r^* = \max_{s \in S} d(s, C^*)$ for the optimum center set C^*). Although the exact value of r^* is actually unknown, we can employ the previous approach for finding a suitable replacement of r^* due to Guo et al. (2025). In general, all of our algorithms mainly proceed in two stages.

- **Streaming stage.** Select a set of representative data points along the stream according to the optimum radius r^* , where the selected set might be with a size larger than k ;
- **Post-streaming Stage.** Compute the set of actually centers from the set of streamed and stored representative data points according to r^* .

2.2 The streaming stage

For the first stage, we employ the λ -independent center set as defined below, but simply tuned the streaming stage for different ratios:

Definition 1. (λ -independent center set) $\Gamma \subseteq S$ is a λ -independent center set of S , if and only if it satisfies the following two conditions:

- 1) For any two points $p, q \in \Gamma$, the distance between them is larger than λ , i.e. $d(p, q) > \lambda$.
- 2) For any point $p \in S$, there exists a point $q \in \Gamma$, such that $d(p, q) \leq \lambda$.

We say a λ -independent center set $\Gamma \subseteq S$ is *minimal* iff removal of any point from Γ makes Γ no longer a λ -independent center set. Assume C^* is the center set of an optimal solution and recalled that r^* is the optimal radius. Then we have:

Lemma 1. For a minimal λ -independent center set $\Gamma \subseteq S$, if $\lambda \geq 2r^*$, then $|\Gamma| \leq k$.

Proof. We need only to show the case for $\lambda = 2r^*$. Suppose $|\Gamma| > k$. Then, by the Pigeonhole principle, there must exist at least a pair of points $i, j \in \Gamma$, which are covered by the same center in the optimal solution, say c^* in C^* . Then, both $d(i, c^*) \leq r^*$ and $d(c^*, j) \leq r^*$ hold. By the triangle inequality, we then have $d(i, j) \leq d(i, c^*) + d(c^*, j) \leq$

$r^* + r^* = 2r^*$. On the other hand, we have $d(i, j) > 2r^*$ according to Cond. (1) in Def. 1, resulting in a contradiction and hence completing the proof. $\square$

3 Fair k-center Clustering in Data Streams

In this section, we present our two-stage streaming algorithm for the case $m = 2$, and then extend to general m .

3.1 Overview of the algorithm

For Stage 1, we need only to simply construct a minimal λ -independent center set Γ_l for each $l \in \{1, 2\}$ regarding $\lambda = 2r^*$ along the stream:

Upon each arriving point i , if $i \in S_l$ and $d(i, \Gamma_l) > \lambda$ both hold, grow Γ_l (initially empty) by adding i .

Next, we present details of Stage 2 with a ratio proof based on analyzing the sizes of Γ_1 and Γ_2 . There are exactly three possibilities regarding the sizes of Γ_1 and Γ_2 :

- (1) $|\Gamma_1| \leq k_1, |\Gamma_2| \leq k_2$;
- (2) $|\Gamma_1| \leq k_1, |\Gamma_2| > k_2$ or $|\Gamma_1| > k_1, |\Gamma_2| \leq k_2$;
- (3) $|\Gamma_1| > k_1, |\Gamma_2| > k_2$.

For Case (1), we can directly use $C = \Gamma_1 \cup \Gamma_2$ as the desired center set, because we have: (1) $|C| = |\Gamma_1 \cup \Gamma_2| \leq k_1 + k_2 = k$; (2) for any point $s \in S$, $d(s, C) \leq 2r^*$ holds due to the definition of λ -independent center set and $\lambda = 2r^*$.

Then, we will give a method to solve Case (2), and show that Case (3) can be reduced to Case (2).

3.2 Proccession of Case (2)

Without loss of generality, we assume that $|\Gamma_l| > k_l$ and $|\Gamma_{3-l}| \leq k_{3-l}$ for $l \in \{1, 2\}$. Our algorithm constructs a new set $\Gamma'_l = \{i | i \in \Gamma_l, d(i, \Gamma_{3-l}) > 3r^*\}$, and simply uses $C = \Gamma_{3-l} \cup \Gamma'_l$ as the center set. The correctness of the such C can be derived from the following theorem:

Theorem 1. *For $C = \Gamma'_l \cup \Gamma_{3-l}$, we have: (1) $|\Gamma'_l| \leq k_l$, $|\Gamma_{3-l}| \leq k_{3-l}$, $|C| \leq k$; (2) for $\forall s \in S$, $d(s, C) \leq 5r^*$.*

Proof. To prove $|\Gamma'_l| \leq k_l$ of Cond. (1), we first show that each $i \in \Gamma'_l$ can not be covered by any point of S_{3-l} , i.e., $d(i, j) > r^*$ holds for any point $j \in S_{3-l}$. Suppose this is not true, i.e., for a point $i \in \Gamma'_l$, there exists a point j in S_{3-l} with $d(i, j) \leq r^*$. According to the definition of the $2r^*$ -independent center set, there must exist a point $p \in \Gamma_{3-l}$ with $d(j, p) \leq 2r^*$. So we get $d(i, p) \leq d(i, j) + d(j, p) \leq r^* + 2r^* = 3r^*$, where the second inequality is by the triangle inequality. This contradicts with the definition of Γ'_l that $d(i, \Gamma_{3-l}) > 3r^*$ holds for each $i \in \Gamma'_l$. Then, $|\Gamma'_l| \leq k_l$ must hold, because otherwise there must exist at least two points $p, q \in \Gamma'_l$ belonging to an identical cluster in the optimal solution. That means $d(p, q) \leq 2r^*$, contradicting the fact $d(p, q) > 2r^*$ as p, q are two points of the $2r^*$ -independent center set Γ_l .

Next, we show $d(s, C) \leq 5r^*$. Firstly, for each point $s \in S_l$, following the algorithm, there exists a point $p \in \Gamma_l$ such that $d(s, p) \leq 2r^*$. Then if point $p \in \Gamma'_l$, $d(s, p) \leq 2r^*$ is true; otherwise, i.e. $p \notin \Gamma'_l$, then there exists a point $q \in \Gamma_{3-l}$ with $d(p, q) \leq 3r^*$, indicating $d(s, q) \leq d(s, p) + d(p, q) \leq$

$2r^* + 3r^* = 5r^*$. Secondly, for each point $s \in S_{3-l}$, there exists a point $p \in \Gamma_{3-l}$ such that $d(s, p) \leq 2r^*$. Moreover, Γ_{3-l} remains unchanged during the algorithm, so $d(s, p) \leq 2r^*$ remains true for $p \in \Gamma_{3-l}$. Therefore, for each $s \in S = S_1 \cup S_2$, $d(s, C) \leq 5r^*$ holds. $\square$

3.3 Proccession of Case (3)

We propose a more sophisticated algorithm to ensure the approximation ratio 5 for Case (3). The key idea of our algorithm is to construct an auxiliary bipartite graph $G(\Gamma_1 \cup \Gamma_2)$, such that the center selection problem regarding $\Gamma_1 \cup \Gamma_2$ is transformed into a special constrained vertex cover problem therein. The construction of G simply proceeds as: (1) Set $V(G) = \Gamma_1 \cup \Gamma_2$; (2) For a pair of $p \in \Gamma_1$ and $q \in \Gamma_2$, add an edge $e(p, q) \in E(G)$ if and only if $d(p, q) \leq 3r^*$ holds.

Then, the aim is to find a vertex cover in G with a size bounded by k that also satisfies the fairness constraints k_1 and k_2 . Notably, although finding a vertex cover is NP -hard in general, we manage to devise a polynomial-time exact algorithm for the problem based on certain special properties of the constructed auxiliary graph.

The key idea of our algorithm is to repeatedly eliminate degree 0 and degree 1 points in G when there exist any, by selecting the point covering such points from $V(G)$ as a new center of the center set C . While G contains no degree-0 or degree-1 points, our algorithm arbitrarily chooses an edge in G , and selects one of its endpoints as a new center by adding it to C . The procedure repeats until the center set C covers all points of G . The detailed algorithm is illustrated in Alg. 1.

Lemma 2. *Each pair of points of Γ_l belongs to different clusters of the optimum solution when $\lambda = 2r^*$ before commencing of Alg. 1.*

Proof. When $\lambda = 2r^*$, according to Def. 1, the distance between each pair of points in Γ_l is larger than $2r^*$. Suppose that there exists a pair of points i, j in Γ_l such that they are in the same cluster of the optimum solution, that is, there exists a center c^* in the center set of the optimum solution such that $d(i, c^*) \leq r^*$ and $d(j, c^*) \leq r^*$ both hold, then $d(i, j) \leq d(i, c^*) + d(j, c^*) \leq r^* + r^* = 2r^*$, contradicting to $d(i, j) > 2r^*$. $\square$

Lemma 3. *For the set C , $|C \cap S_l| \leq k_l$ holds when Phase 1 of Alg. 1 completes.*

Proof. Following Steps 3-6 in Alg. 1, we add $i \in G$ with degree 0 to C . That is, each point $i \in \Gamma_l$ added in C satisfies $d(i, \Gamma_{3-l}) > 3r^*$. Then according to Cond. (1) in Thm. 1, $d(i, j) > r^*$ holds for any point $j \in S_{3-l}$. That is, each point $i \in C \cap \Gamma_l$ can not belong to a cluster centered at a point of S_{3-l} in the optimal solution. So similar to the proof of Cond. (1) of Thm. 1, we have $|C \cap S_l| \leq k_l$. $\square$

For briefness, we use Γ_l^j and C^j to respectively denote Γ_l and C in the j th iteration of the while-loop of Alg. 1.

Lemma 4. *In the j th iteration of Alg. 1's while-loop, we have: (1) $|\Gamma_l^j| \leq k - |C^j|$; (2) $|C^j| \leq k$; (3) $|C^j \cap S_l| \leq k_l$.*

Algorithm 1: Procession of Case (3)

Input: Γ_1 and Γ_2 with $|\Gamma_1| > k_1$, $|\Gamma_2| > k_2$ **Output:** Center set C .

// Phase 1: Construction of the auxiliary graph.

- 1 Set $C \leftarrow \emptyset$;
 - 2 Construct the auxiliary graph $G(\Gamma_1 \cup \Gamma_2, E)$, where an edge $e(p, q)$ exists in E if and only if $p \in \Gamma_1$, $q \in \Gamma_2$ and $d(p, q) \leq 3r^*$;
 - 3 **for** each point $i \in G$ with degree 0 **do**
 - 4 **if** $d(C, i) > 2r^*$ **then**
 - 5 Set $C \leftarrow C \cup \{i\}$;
 - 6 Remove each point $i \in G$ with degree 0;
 - 7 // Phase 2: Computation of C .
 - 7 **while** $|C| \leq k = k_1 + k_2$ and $V(G) \neq \emptyset$ **do**
 - 8 **if** there exists no degree-1 point in G **then**
 - 9 Arbitrarily select an edge $e(p, q)$ and remove it from G ;
 - 10 Set $C \leftarrow C \cup \{p\}$ and $G \leftarrow G \setminus \{p, q\}$;
 - 11 Set $\Gamma_l \leftarrow \Gamma_l \setminus \{p\}$ and $\Gamma_{3-l} \leftarrow \Gamma_{3-l} \setminus \{q\}$ for Γ_l containing p ;
 - 12 **else**
 - 13 Find $i \in G$ with largest $N_1(i)$, where $N_1(i)$ is the set of degree-1 neighbours of i in G ;
 - 14 Set $C \leftarrow C \cup \{i\}$ and $G \leftarrow G \setminus \{i\} \setminus N_1(i)$;
 - 15 Set $\Gamma_l \leftarrow \Gamma_l \setminus \{i\}$ and $\Gamma_{3-l} \leftarrow \Gamma_{3-l} \setminus N_1(i)$ for Γ_l containing i ;
 - 16 **if** there exists l with $|C \cap S_l| + |\Gamma_l| \leq k_l$ **then**
 - 17 Set $C \leftarrow C \cup \Gamma_l$ and compute $\Gamma'_{3-l} = \{p \in \Gamma_{3-l} \text{ and } d(p, C) > 3r^*\}$
 - 18 Return $C \leftarrow C \cup \Gamma'_{3-l}$.
-

Proof. For Cond. (1), we first show it is true when $j = 1$. Before the first while loop commences, $|\Gamma_l^1| \leq k - |C^1|$ is satisfied. Suppose otherwise, there would exist a contradiction as analyzed below. We will show the distance between every two points belonging to $C^1 \cup \Gamma_l^1$ is larger than $2r^*$, so every two points therein can not belong to the same cluster in an optimal solution, and hence there are at least $|\Gamma_l^1| + |C^1| > k$ centers in an optimal solution, arising a contradiction. This fact is deduced following Steps 3-6 in Alg. 1, where points with a zero degree are added to C^1 for the first iteration. For each two points $i \in C^1 \cup \Gamma_l^1 \cap S_l$ and $j \in C^1 \cup \Gamma_l^1 \cap S_{3-l}$, the absence of edge $e(i, j)$ in G indicates $d(i, j) > 3r^*$ according to the construction of the auxiliary graph G . Moreover, for each two points $i, j \in C^1 \cup \Gamma_l^1 \cap S_l$, $d(i, j) > 2r^*$ holds because Γ_l is a $2r^*$ -independent center set for S_l ($l \in \{1, 2\}$), and i, j are points in initial Γ_l . Therefore, $d(i, j) > 2r^*$ holds for each two points $i, j \in C^1 \cup \Gamma_l^1$.

We demonstrate that Cond. (1) holds for the $(j + 1)$ th iteration by induction. Assuming Cond. (1) is valid for the j th iteration concerning Γ^j , we either remove two points or remove a point i along with its degree-1 neighbors, each belonging to Γ_1^j and Γ_2^j respectively. Consequently, after this

removal, it follows that:

$$|\Gamma_l^{j+1}| \leq |\Gamma_l^j| - 1 \leq k - |C^j| - 1 \leq k - |C^{j+1}|,$$

where the first inequality is due to the removal of at least one point from both Γ_1^j and Γ_2^j in every iteration. The second inequality is derived from our inductive assumption. The third inequality arises because, on one hand, we add one point to C^j to form C^{j+1} , and on the other hand, according to the algorithm, the removal of points from G does not result in new points of degree 0 in the auxiliary graph G . Hence, Cond. (1) is also true for the $(j + 1)$ th iteration.

Cond. (2) can be immediately derived from Cond. (1) because $|\Gamma_l^j| \geq 0$ holds.

Moving on to Cond. (3), Lem. 3 establishes that in Steps 3-6 of Alg. 1, the inequality $|C \cap \Gamma_l| = \gamma_l \leq k_l$ is valid. That is, Cond. (3) holds for $j = 1$. Our objective now is to verify that this inequality persists throughout the $(j + 1)$ th iteration of Alg. 1. Assume that Cond. (3) holds for j th iteration. According to Steps 8-15, Alg. 1 involves adding a single point to C in each iteration. If the algorithm does not terminate in the $(j + 1)$ th iteration, the inequality $|C^j \cap S_l| \leq k_l$ holds for each $l \in \{1, 2\}$ clearly. Otherwise, the algorithm terminates in the $(j + 1)$ th iteration. Upon the termination of Alg. 1, we encounter two cases: (1) $|C^j \cap S_l| + |\Gamma_l| \leq k_l$ and $|C^j \cap S_{3-l}| < k_{3-l}$; (2) $|C^j \cap S_{3-l}| = k_{3-l}$. Clearly, the inequality $|C^j \cap S_l| \leq k_l$ holds for each $l \in \{1, 2\}$ when Case (1) occurs. Regarding the latter Case (2), when a point from Γ_{3-l} is added to C^j in the $(j + 1)$ th iteration, we achieve $|C^j \cap S_{3-l}| = k_{3-l}$. In accordance with Cond. (1), this leads to $|C^j \cap S_l| + |\Gamma_l| \leq k - |C^j \cap S_{3-l}| = k - k_{3-l} = k_l$. Therefore, the inequality $|C^j \cap S_l| \leq k_l$ is consistently upheld for each $l \in \{1, 2\}$ during the execution of the entire algorithm. This completes the proof. $\square$

Theorem 2. Alg. 1 terminates in runtime $O(k^2)$ and outputs a feasible solution C to fair k -center, for which $d(s, C) \leq 5r^*$ holds for any point $s \in S$.

Proof. For the runtime, Step 2 of Alg. 1 takes $O(k^2)$ to construct G as G contains at most $O(k)$ vertices and $O(k^2)$ edges. Then, the while-loop (Steps 7-17) iteration for at most $O(k)$ times as it removes at least one points from G , and each iteration takes $O(k)$ time. Therefore, the total runtime sums up to $O(k^2)$.

From Cond. (2) and (3) of Lem. 4, we immediately get $|C| \leq k$ and $|C \cap S_l| \leq k_l$, which indicates C is a feasible solution. It remains to bound the distance from any point $s \in S$ to C . For any $s \in S_l$, a point $i \in \Gamma_l$ must exist such that $d(s, i) \leq 2r^*$ holds according to the definition of Γ_l . If $i \in C$ holds, then we have $d(s, C) \leq 2r^*$. Otherwise, i.e. $i \notin C$, then there must exist an edge $e(i, j)$ with $j \in \Gamma_{3-l}$ in the auxiliary graph. According to Steps 8-15 in Alg. 1, at least one endpoint of $e(i, j)$, either i or j , must be added to C . Since $i \notin C$ by assumption, $j \in C$ holds. Moreover, the existence of edge $e(i, j)$ also means $d(i, j) \leq 3r^*$. So by triangle inequality, we have

$$d(s, C) \leq d(s, j) \leq d(s, i) + d(i, j) \leq 2r^* + 3r^* = 5r^*.$$

Therefore, regardless $i \in C$ holds or not, $d(s, C) \leq 5r^*$ holds for any $s \in S_l$ for any l . $\square$

Combining Case (1), (2) and (3), we eventually achieve a complete streaming algorithm for fair k -center provided that r^* is known. For the space complexity, the algorithm stores $m = 2$ independent center sets for each case where each independent center set has $O(k)$ points, so it consumes a space complexity of $O(k)$. We next analyze the update time of the algorithm. In the streaming stage, upon the arrival of each point i , the algorithm needs to verify its distance to every point within the current independent center sets, which contains $O(k)$ points. So the update time is $O(k)$. Combining Thm. 2, we have:

Corollary 1. *Provided r^* is known, fair k -center admits a streaming algorithm that achieves a ratio 5, consumes $O(k)$ memory and $O(k)$ update time.*

Moreover, as r^* is actually unknown, we need a $O(\log n)$ multiplicative factor over the memory complexity and overall runtime (Guo et al. 2025). So the algorithm consumes $O(k \log n)$ memory and $O(nk \log n)$ runtime in total.

Notably, the approximation ratio 5 is tight according to the example in the appendix. We can easily extend the algorithms respectively for Cases (2) and Cases (3) and obtain an algorithm for general m . Moreover, the streaming algorithm can be tuned to approximate the offline fair k -center with a ratio 3 (shown in the appendix).

4 Streaming Semi-structured Data Sets

In this section, we consider scenarios in which the data points are streamed following the demographic group order, i.e. all points belong to S_l arrive before S_{l+1} , $\forall l \in [m-1]$. We provide an improved algorithm achieving a ratio 3 for $m = 2$ that conforms to the state-of-art ratio for offline setting. Moreover, for general m , we show that the algorithm can be extended to achieve a ratio 4 (as shown in Appendix).

Our key idea of the improved algorithm for $m = 2$ is to obtain an extra point set Γ_{sub} in the streaming stage, such that points of Γ_{sub} can be used to replace the points of Γ_1 for the fairness constraint. In addition, to better suit Γ_{sub} , we compute a slightly improved independent center set Γ'_l instead of Γ_l except for Γ_1 . We first construct an independent set $\Gamma'_1 = \Gamma_1$ with $\lambda = 2r^*$ upon the data stream of S_1 . Then, upon the stream of S_2 , we select points using different designate approaches depending on whether $|\Gamma'_1| \leq k_1$ holds. The detailed algorithm is as in Alg. 2.

Then we prove that with only the representative points of $\Gamma'_1 \cup \Gamma'_2$ and Γ_{sub} , our post-streaming algorithm can compute an approximation solution of ratio 3.

Lemma 5. *For Γ'_1 and Γ'_2 produced by the above algorithm, we have: (1) $|\Gamma'_1| + |\Gamma'_2| \leq k$; (2) $|\Gamma'_2| \leq k_2$ always holds; (3) There exists a subset $\Gamma''_1 \subseteq \Gamma'_1$ and accordingly a subset $\Gamma'_{sub} = \{\sigma(c) | c \in \Gamma''_1\} \subseteq \Gamma_{sub}$, such that $|\Gamma'_1 \setminus \Gamma''_1 \cup \Gamma'_{sub} \cap S_1| \leq k_1$ holds, where $\sigma(c)$ is a possible replacement of c in Γ_{sub} , i.e. $d(c, \sigma(c)) \leq r^*$.*

Proof. For Cond. (1), according to the algorithm, any pair of two points $p, q \in \bigcup \Gamma'_l, l = 1, 2$, we have $d(p, q) > 2r^*$. So every pair of points must appear in two different clusters in the optimal solution. Thus, $|\bigcup_{l=1}^2 \Gamma'_l| \leq k$ holds.

Algorithm 2: Improved streaming stage against semi-structural streams

Input: A stream of points $S = \bigcup_l S_l$ in which all points of S_1 arrive before the points of S_2 , and $\lambda = 2r^*$.

Output: The sets of representative points.

```

1 Set  $\Gamma_{sub} = \emptyset$  and  $\Gamma'_l = \emptyset$  for  $l = 1, 2$ ;
2 upon each arriving point  $i \in S_1$  do
3   if  $d(i, \Gamma'_1) > \lambda$  then
4     Set  $\Gamma'_1 \leftarrow \Gamma'_1 \cup \{i\}$ ;           // Recall that
      $d(i, \emptyset) = \infty$ .
5 upon each arriving point  $i \in S_2$  do
6   if  $|\Gamma'_1| \leq k_1$  then
7     if  $d(i, \Gamma'_1) > 3\lambda/2$  and  $d(i, \Gamma'_2) > \lambda$  then
8       Set  $\Gamma'_2 \leftarrow \Gamma'_2 \cup \{i\}$ ;
9   else
10    if  $d(i, \Gamma'_1 \cup \Gamma'_2) > \lambda$  then
11      Set  $\Gamma'_2 \leftarrow \Gamma'_2 \cup \{i\}$ ;
12    if there exists a point  $j \in \Gamma'_1$  that has no
      replacement in  $\Gamma_{sub}$  and  $d(i, j) \leq \lambda/2$  then
13      Set  $\Gamma_{sub} \leftarrow \Gamma_{sub} \cup \{i\}$ ;
14 Return  $\Gamma'_1, \Gamma'_2$ , and  $\Gamma_{sub}$ .
```

For Cond. (2), we have $|\Gamma'_2| \leq k - |\Gamma'_1| < k - k_1 = k_2$ if $|\Gamma'_1| > k_1$. Otherwise, as Γ'_1 covers all points of the clusters centered at points of S_1 in the optimal solution within a radius $3r^*$. Then, the remaining points to be covered in the algorithm appears in only the optimal clusters centered at points of S_2 , which are at most k_2 clusters. On the other hand, every pair of points in Γ'_2 must appear in different optimal clusters. So $|\Gamma'_2| \leq k_2$.

For Cond. (3), if $|\Gamma'_1| \leq k_1$ holds after streaming then the proof is done. So we need only to prove the case for $|\Gamma'_1| > k_1$. Clearly, there exists a subset of Γ'_1 contains at least $|\Gamma'_1| - k_1$ points, say Γ''_1 , appearing in optimal clusters centered at points of S_2 . According to Alg. 2, at least one point $j \in S_2$ for each point $i \in \Gamma''_1$ exists with $d(i, j) \leq r^*$, and is added as $\sigma(i)$ to Γ_{sub} as a replacement point of i , collectively composing the set Γ'_{sub} . So $|\Gamma'_1 \setminus \Gamma''_1 \cup \Gamma'_{sub} \cap S_1| \leq k_1$ holds. $\square$

Following the above lemma, we can immediately obtain a 3-approximation solution by simply computing $C = \Gamma_2 \cup \Gamma'_1 \setminus \Gamma''_1 \cup \Gamma'_{sub}$ as the desired center set.

5 Experimental Results

In this section, we conduct an empirical evaluation of our algorithms utilizing both simulated and real-world datasets, compared with three previous approximation algorithms that serve as baselines. All experiments are averaged over at least 20 iterations and conducted on a Linux machine equipped with 12th Gen Intel(R) Core(TM) i9-12900K CPU and 32 GB RAM by Python 3.8. A detailed description of the experimental setup is introduced below.

5.1 Experimental setting

Datasets We employ both simulated and real-world datasets to evaluate our approximation algorithm. We summarize the detailed datasets in Tab. 1.

Real-world datasets: We apply our algorithms to seven real-world datasets: Wholesale, Student, Bank, CreditCard, Adult, SushiA and CelebA, following the most recent related works (Jones, Nguyen, and Nguyen 2020; Chen et al. 2019). Consistent with previous studies, we utilize meaningful numerical features for clustering and incorporate selected binary categorical attributes to construct datasets with fair constraints across all datasets.

Simulated datasets: we provide two datasets that serve as complementary benchmarks: the simulated dataset provides an exact optimal solution, whereas the large-scale dataset includes more time for file I/O operations, making it ideal for measuring computational efficiency. First, the dataset (called *SimuA*) is used to assess the *empirical approximation ratio* of our algorithms and the baseline methods. Inspired by previous research (Kleindessner, Awasthi, and Morgenstern 2019), we used their method to construct a simulated dataset with a known optimal solution for the fair k -center problem. Secondly, we leverage the implementation from Chiplunkar, Kale, and Ramamoorthy (2020) to generate a **100G** dataset (called *SimuB*), which allows us to evaluate the runtime performance of different algorithms.

Constraints Settings For the simulated dataset, we configured the parameters to compare both the average and maximum values of the approximation ratios for these algorithms and to verify the approximation ratio of our algorithms. To ensure fair center selection, we aligned the number of required centers from each group with the proportional size of that group. Following the fairness principle of disparate impact as outlined by Feldman et al. (2015), we restricted the selection to k_l data points from the l th group to serve as centers. We then evaluated the clustering quality by varying the number of k across these datasets.

Algorithms Compared with existing streaming fair k -center clustering algorithms, this experiment includes an extended algorithm from the offline 3-approximation algorithm by Jones, Nguyen, and Nguyen (2020) (denoted as ExJones), a two-pass streaming 3-approximation algorithm by Chiplunkar, Kale, and Ramamoorthy (2020) (denoted as two-pass 3-Approx), and a one-pass streaming 7-approximation algorithm (Lin, Guo, and Jia 2024) (denoted as 7-Approx). Moreover, we propose two algorithms in this paper: 1) a one-pass 5-approximation algorithm (denoted as **5-Approx**) on the general metric; 2) a one-pass 3-approximation algorithm with $m = 2$ (denoted as **3-Approx**) for semi-structured data streaming in our experiments.

Metrics We use the *cost* metric, as defined in Sec. 2, to compare the quality of clustering across these datasets based on their average values. In addition, we adopted the second to measure runtime.

Dataset	#Record	#Dim.
Wholesale	440	6
Student	649	16
Bank	4,521	7
SushiA	5,000	10
CreditCard	30,000	19
Adult	32,561	6
CelebA	202,599	15,360
SimulatedA	4,000,000	1,000
SimulatedB	[2k,4k,6k,8k,10k]	2

Table 1: Datasets Summary.

5.2 Experimental analysis

In Fig. 1, we first report the empirical approximation ratios of the algorithms on a simulated dataset with known optimal solutions. This allows us to compare the experimental objective values of our algorithms and the baselines against the optimal objective. We then evaluate the clustering cost of these algorithms on real-world datasets. Finally, we report the runtime(s) performance on large-scale simulated datasets, focusing on file I/O operations in our algorithms compared to other baselines.

Approximation Factor We compare algorithm performance by evaluating the relative solution ratio with the provided optimal solution on the simulated dataset. The ratio of the evaluation result can be called the empirical approximation ratio, and the maximum value represents the worst-case cost in all the experimental results. We run the code of constructing the simulated dataset shared by Kleindessner, Awasthi, and Morgenstern (2019) for their algorithm setting $m = 2$, $k = 100$ and varying the size of the simulated datasets $|S|$ from 2,000 to 10,000. For each size of the dataset, we perform 10 runs on 10 kinds of fairness with 10 different constraint ratios.

We observe that all algorithms align well with their theoretical bounds presented in this paper and the baselines. Notably, the 3-Approx algorithm demonstrates the best performance, even outperforming the two-pass 3-Approx algorithm (Chiplunkar, Kale, and Ramamoorthy 2020). This aligns with our theoretical results, as the algorithm provides a strong approximation guarantee. When the dataset size $|S| = 2,000$, the maximal empirical approximation ratio of the 7-Approx algorithm achieves an approximation ratio greater than 5, validating the suitability of the dataset for testing these algorithms. Compared with the Jones’ method, our algorithm has a lower empirical approximation ratio, which demonstrates the advantage of our algorithm’s result. Consequently, these algorithms exhibit superior performance on ideal datasets, aligning well with their theoretical guarantees.

Comparison with Baselines We report the clustering costs achieved by our algorithms and three baselines across seven real-world datasets in Tab. 2. The parameter k is set proportionally to the dataset size: datasets above the dividing line use 1% of the total number of records, while those below use 1%, which depends on the variation in dataset scales.

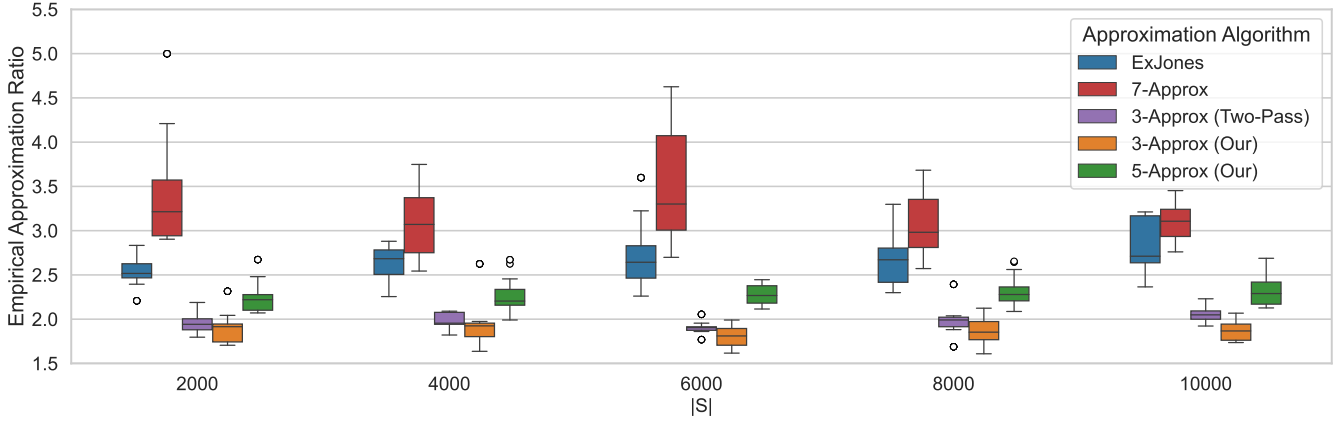


Figure 1: Empirical approximation ratio ($cost/r^*$) of our algorithms in comparison with other baselines.

Dataset	ExJones	7-Approx	5-Approx	3-Approx (Semi)	3-Approx (Two pass)
Wholesale	1.36	1.28	<u>1.04</u>	0.84	0.61
Student	<u>1.85</u>	1.85	1.90	1.79	1.73
CelebA	24.36	26.27	<u>21.34</u>	21.34	21.25
Bank	<u>0.49</u>	0.93	0.61	0.40	0.43
SushiA	1.42	1.66	<u>1.41</u>	1.51	1.48
Credit	0.93	1.89	1.03	0.94	0.81
Adult	0.65	1.08	<u>0.62</u>	0.52	0.56

Table 2: Cost comparison on the real-world datasets. (The underline highlights the best results of the general one-pass algorithms. In addition, the **bold** indicates the best result when semi-structured data streaming is included alongside the one-pass streaming algorithms.)

This same proportion is also applied to the center constraints across different groups. All datasets are normalized prior to running the algorithms, and for CelebA, the first 1,000 samples are used in the experiments.

In Tab. 2, we observe that 3-Approx (Semi) outperforms most of the other one-pass algorithms, while the two-pass 3-approximation algorithm keeps the better performance than the one-pass streaming algorithms on the seven datasets. We attribute this to the fact that our algorithm is a modification of the original center selection method based on streaming. By combining streaming techniques with the structure of *independent center set*, it effectively identifies more meaningful center points. As a result, it achieves lower clustering costs than the other baselines under the one-pass streaming setting (shown as the underline value). We also observe that the results reported by 3-Approx (two-pass) consistently outperform all the one-pass streaming algorithms. This can be explained that the two-pass 3-approximation algorithm has the advantage of refining the center set in the second pass, enabling it to identify more accurate center points.

Streaming Runtime Using the method from Chiplunkar, Kale, and Ramamoorthy (2020), we generate an artificial 100 GB dataset containing 4,000,000 items and 1,000 features. In Fig. 2, we evaluate the performance at run-time of our algo-

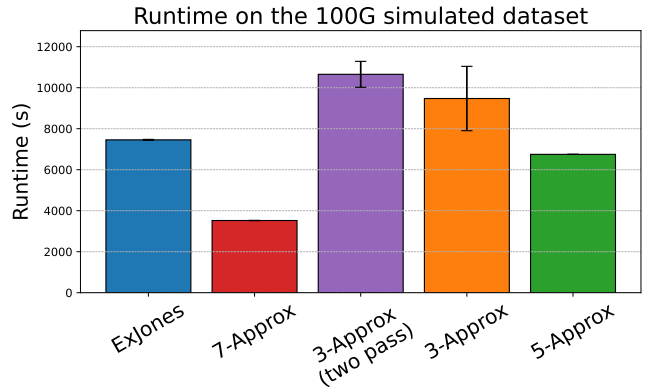


Figure 2: Runtime on the 100G simulated dataset.

gorithms and baseline methods in this dataset. Our results show that 5-Approx algorithm runs faster than Chiplunkar, Kale, and Ramamoorthy (2020) and Jones, Nguyen, and Nguyen (2020), which aligns with the theoretical expectations, as all of our algorithms are one-pass streaming methods. Among these methods, 7-Approx (Lin, Guo, and Jia 2024) achieves the fastest runtime, because it selects fewer points during the streaming process and omits post-processing steps, thereby reducing computational overhead.

6 Conclusion

In this paper, we first devise a one-pass streaming algorithm with an approximation ratio of 5 and a memory complexity of $O(k \log n)$. We demonstrate that such approximation ratio and memory usage are optimal for the metric space. Observing the broad applications of semi-structured data streams, we present a 3-approximation for $m = 2$ and a 4-approximation for general m . Lastly, extensive experiments were conducted to demonstrate the performance gain of our algorithms compared to baselines including the state-of-the-art method.

References

- Anegg, G.; Vargas Koch, L.; and Zenklusen, R. 2022. Techniques for Generalized Colorful k -Center Problems. In *30th Annual European Symposium on Algorithms (ESA 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- Angelidakis, H.; Kurpisz, A.; Sering, L.; and Zenklusen, R. 2022. Fair and Fast k -Center Clustering for Data Summarization. In *International Conference on Machine Learning*, 669–702. PMLR.
- Backurs, A.; Indyk, P.; Onak, K.; Schieber, B.; Vakilian, A.; and Wagner, T. 2019. Scalable fair clustering. In *International Conference on Machine Learning*, 405–413. PMLR.
- Bandyapadhyay, S.; Inamdar, T.; Pai, S.; and Varadarajan, K. 2019. A Constant Approximation for Colorful k -Center. *Leibniz International Proceedings in Informatics, LIPIcs*, 144.
- Ceccarello, M.; Pietracaprina, A.; and Pucci, G. 2019. Solving k -center Clustering (with Outliers) in MapReduce and Streaming, almost as Accurately as Sequentially. *Proceedings of the VLDB Endowment*, 12(7): 766–778.
- Chen, D. Z.; Li, J.; Liang, H.; and Wang, H. 2016. Matroid and knapsack center problems. *Algorithmica*, 75: 27–52.
- Chen, X.; Fain, B.; Lyu, L.; and Munagala, K. 2019. Proportionally fair clustering. In *International Conference on Machine Learning*, 1032–1041. PMLR.
- Chen, X.; Ji, S.; Wu, C.; Xu, Y.; and Yang, Y. 2024. An approximation algorithm for diversity-aware fair k -supplier problem. *Theoretical Computer Science*, 983: 114305.
- Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair clustering through fairlets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5036–5044.
- Chiplunkar, A.; Kale, S.; and Ramamoorthy, S. N. 2020. How to solve fair k -center in massive data models. In *Proceedings of the 37th ICML*, 1877–1886.
- Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 259–268.
- Guo, L.; Jia, C.; Liao, K.; Lu, Z.; and Xue, M. 2025. Near-Optimal Algorithms for Instance-Level Constrained k -Center Clustering. *IEEE Transactions on Neural Networks and Learning Systems*.
- Jia, X.; Sheth, K.; and Svensson, O. 2022. Fair colorful k -center clustering. *Mathematical Programming*, 192(1-2): 339–360.
- Jones, M.; Nguyen, H.; and Nguyen, T. 2020. Fair k -centers via maximum matching. In *International Conference on Machine Learning*, 4940–4949. PMLR.
- Kale, S. 2019. Small Space Stream Summary for Matroid Center. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*, 145: 20.
- Kleindessner, M.; Awasthi, P.; and Morgenstern, J. 2019. Fair k -center clustering for data summarization. In *Proceedings of the 36th ICML*, 3448–3457. PMLR.
- Lin, Z.; Guo, L.; and Jia, C. 2024. Streaming Fair k -Center Clustering over Massive Dataset with Performance Guarantee. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 105–117. Springer.
- Mahabadi, S.; and Trajanovski, S. 2024. Core-sets for Fair and Diverse Data Summarization. *Advances in Neural Information Processing Systems*, 36.
- Matthew McCutchen, R.; and Khuller, S. 2008. Streaming Algorithms for k -Center Clustering with Outliers and with Anonymity. *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, 165–178.
- Nguyen, H. L.; Nguyen, T.; and Jones, M. 2022. Fair Range k -center. *arXiv preprint arXiv:2207.11337*.
- Wu, X.; Feng, Q.; Huang, Z.; Xu, J.; and Wang, J. 2024. New Algorithms for Distributed Fair k -Center Clustering: Almost Accurate as Sequential Algorithms. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 1938–1946.