Small Space Stream Summary for Matroid Center

Sagar Kale
EPFL, Lausanne, Switzerland
sagar.kale@epfl.ch

Abstract

In the matroid center problem, which generalizes the k-center problem, we need to pick a set of centers that is an independent set of a matroid with rank r. We study this problem in streaming, where elements of the ground set arrive in the stream. We first show that any randomized one-pass streaming algorithm that computes a better than Δ-approximation for partition-matroid center must use $\Omega(r^2)$ bits of space, where Δ is the aspect ratio of the metric and can be arbitrarily large. This shows a quadratic separation between matroid center and k-center, for which the Doubling algorithm [7] gives an 8-approximation using $O(k)$-space and one pass. To complement this, we give a one-pass algorithm for matroid center that stores at most $O(r^2 \log(1/\varepsilon)/\varepsilon)$ points (viz., stream summary) among which a $(7+\varepsilon)$-approximate solution exists, which can be found by brute force, or a $(17+\varepsilon)$-approximation can be found with an efficient algorithm. If we are allowed a second pass, we can compute a $(3+\varepsilon)$-approximation efficiently.

We also consider the problem of matroid center with z outliers and give a one-pass algorithm that outputs a set of $O((r^2+rz)\log(1/\varepsilon)/\varepsilon)$ points that contains a $(15+\varepsilon)$-approximate solution. Our techniques extend to knapsack center and knapsack center with z outliers in a straightforward way, and we get algorithms that use space linear in the size of a largest feasible set (as opposed to quadratic space for matroid center).

2012 ACM Subject Classification Theory of computation → Streaming models; Theory of computation → Facility location and clustering; Mathematics of computing → Matroids and greedoids

Keywords and phrases Streaming Algorithms, Matroids, Clustering

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2019.20

Category APPROX

Related Version A full version of the paper is available at `http://arxiv.org/abs/1810.06267`.

Funding This work was supported by ERC Starting Grant 335288-OptApprox.

Acknowledgements I thank Ashish Chiplunkar for his contributions, Maryam Negahbani for discussions, and anonymous reviewers for helpful comments.

1 Introduction

In the k-center problem, the input is a metric, and we need to select a set of k centers that minimizes the maximum distance between a point and its nearest center. Matroid center is a natural generalization of k-center, where, along with a metric over a set, the input also contains a matroid of rank r over the same set. We then need to choose a set of centers that is an independent set of the matroid that minimizes the maximum distance between a point and its nearest center. Then k-center is rank-k-uniform-matroid center. Examples of clustering problems where the set of centers needs to form an independent set of a partition matroid arise in content distribution networks (see Hajiaghayi et al. [16] and references therein). A partition matroid constraint can also be used to enforce fairness conditions such as having k_M centers of type M and k_W centers of type W. As another example, say the input points lie in a euclidean space, and we are required to output linearly independent centers, then

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019).
Editors: Dimitris Achlioptas and László A. Végh; Article No. 20; pp. 20:1–20:22

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

this is the linear-matroid center problem. Studying a combinatorial optimization problem in the streaming model is worthwhile not only in its own right, but also because it can lead to discovery of much faster algorithms[1].

In the streaming model, the input points arrive in the stream, and we are interested in designing algorithms that use space sublinear in the input size. We study the matroid center problem in the streaming model. By a clean reduction from the INDEX problem, we first show that any randomized one-pass streaming algorithm that computes a better than Δ-approximation for matroid center must use $\Omega(r^2)$ bits of space, where Δ is the aspect ratio of the metric (ratio of the largest distance to the smallest distance between two points), which can be arbitrarily large. Since the Doubling algorithm [7] gives an 8-approximation for k-center in one pass over the stream by storing at most k points, we get a quadratic separation between matroid center and k-center. We then give a one-pass algorithm that computes a $(7+\varepsilon)$-approximation using a *stream summary* of $O(r^2 \log(1/\varepsilon)/\varepsilon)$ points. The algorithm maintains an efficiently-updatable summary, and runs a brute-force step when the end of the stream is reached. We can replace the brute-force step by an efficient algorithm to get a $(17+\varepsilon)$-approximation. Alternatively, using a second pass, we can (efficiently) compute a $(3+\varepsilon)$-approximation. Our algorithms assume only oracle accesses to the metric and to the matroid.

In k-center or matroid center, even very few rogue points can wreck up the solution, which motivates the outlier versions where we can choose up to z points that our solution will not serve. McCutchen and Khuller [28] give a one-pass $(4+\varepsilon)$-approximation algorithm for k-center with z outliers that uses space $O(kz \log(1/\varepsilon)/\varepsilon)$. Building on their ideas, we give a $(15+\varepsilon)$-approximation one-pass algorithm for matroid center with z outliers, using a brute-force search through the summary as the last step, and a $(51+\varepsilon)$-approximation algorithm if we want an efficient implementation in the last step.

To the best of our knowledge, matroid center problems have not been considered in streaming. Chen, Li, Liang, and Wang [11] give an offline 3-approximation algorithm for matroid center and a 7-approximation algorithm for the outlier version; this approximation ratio is improved to 3 by Harris et al. [19]. These algorithms are not easily adaptable to the streaming setting if we are allowed only one pass, though, our two-pass algorithm for matroid center may be thought of as running multiple copies of Chen et al.'s 3-approximation algorithm. We mention that optimization problems over matroid or related constraints have been studied before in streaming [2, 3, 10].

The Doubling algorithm [7] gives an 8-approximation for k-center. Guha [15], using his technique of "stream-strapping", improves this to $2+\varepsilon$. We use the stream-strapping technique in this paper to reduce space-usage of our algorithms as well. Known streaming algorithms for k-center problems do not extend to the matroid center problems. Indeed, the gap between the space complexities of k-center and matroid center, exhibited by our lower bound, warrants the need for new ideas.

[1] This is demonstrated by Chakrabarti and Kale [3] who give streaming algorithms for submodular maximization problems that make *only* $2|E|$ total submodular-oracle calls ($\tilde{O}(|E|)$ total time) and achieve constant-factor approximations, where E is the ground set. On the other hand earlier fastest algorithms were greedy and potentially could make $\Omega(|E|^2)$ oracle calls. Trivially, $|E|$ oracle calls are needed for any non-trivial approximation.

1.1 Techniques

At the heart of many algorithms for k-center is Gonzalez's [13] furthest point heuristic that gives a 2-approximation. It first chooses an arbitrary point and adds it to the current set C of centers. Then it chooses a point that is farthest from C and adds it to C. This is repeated until C has k centers. Let C_E be the set of centers returned by this algorithm, and let p be the point that is farthest from C_E. Then $d(p, C_E)$ is the cost of the solution, whereas the set $C_E \cup \{p\}$ of size $k+1$ acts as a certificate that an optimum solution must have cost at least $d(p, C_E)/2$. This can be easily implemented in streaming if we are given a "guess" τ of OPT, i.e., the cost of an optimum solution. When we see a new point e in the stream, we add it to C if $d(e, C) > 2\tau$. Assuming that we know the aspect ratio Δ, we can do this for $2\log_{1+\varepsilon}\Delta$ guesses of OPT to get a $(2+\varepsilon)$-approximation as follows. Let R be the distance between first two points in the stream. Then maintain the set C as described above for guesses $\tau \in \{R/\Delta, (1+\varepsilon)R/\Delta, (1+\varepsilon)^2 R/\Delta, \ldots, R\Delta\}$. The stream-strapping technique reduces the number of active guesses to $O(\log(1/\varepsilon)/\varepsilon)$.

In extending this to matroid center, the biggest challenge is deciding which point to make a center. In a solution to k-center, if we replace a point by another point that is very close to it, then the cost can change only slightly, whereas if we do the same in a solution to matroid center, the solution might just become infeasible. Therefore, if we maintain a set C as earlier, it might quickly lose its independence in the matroid. The idea is to store, for each of the at most r points $c \in C$, a maximal independent set I_c of points close to c; here, by close we mean close in terms of the guess τ. This way, we store at most $r^2 + r$ points. Storing a maximal independent set for each point in C may seem wasteful, but our lower bound shows that it is necessary. Our first algorithmic insight is to show that this idea works for a correct guess. We show that if each optimum center s is in the span of an independent set I_c for a c that is close to s, then we can recover an independent set of small cost from the *summary* $\bigcup_{c \in C_E} I_c$. And as our second insight, we show how to extend the stream-strapping approach to reduce the number of active guesses, which helps us reduce the space usage. These ideas naturally combine with those of McCutchen and Khuller [28] and help us design an algorithm for matroid center with z outliers, but it is nontrivial to prove that the combination of these ideas works.

Knapsack center

In the knapsack center problem, each point e has a non-negative weight $w(e)$, and the goal is to select a set C of centers that minimizes the maximum distance between a point and its nearest center subject to the constraint that $\sum_{c \in C} w(c) \leqslant B$, where B is the *budget*. The k-center problem is a special case with unit weights and $B = k$. In the streaming setting, our algorithms for matroid center and matroid center with outliers can be extended to get constant approximations using space proportional to the size of a largest feasible set, i.e., $\max\{|S| : \sum_{e \in S} w(e) \leqslant B\}$. As described earlier, we maintain a set C of potential centers using the guess τ, and for each potential center c, we also maintain a smallest weight point, say s_c, in its vicinity. Then, in the end, the summary $\{s_c : c \in C\}$ contains a good solution. This idea works because replacing a center by a nearby point with a smaller weight does not affect the feasibility in the knapsack setting (which could destroy independence in the matroid setting).

1.2 Related Work

The k-center problem was considered in the '60s [17, 18]. It is NP-hard to achieve a factor of better than 2 [23], and polynomial-time 2-approximation algorithms exist [13, 21]. As mentioned earlier, Chen et al. [11] give a 3-approximation algorithm for matroid center

and a 7-approximation algorithm for the outlier version, and this approximation ratio is improved to 3 by Harris et al. [19]. Motivated by applications in content distribution networks, the matroid median problem is considered as well [16, 25]. The problem of k-center with outliers was first studied by Charikar et al. [8] who gave a 3-approximation algorithm. The approximation ratio was recently improved to 2 by Chakrabarty et al. [4]. We mention the work of Lattanzi et al. [26] that considers hierarchical k-center with outliers.

For knapsack center, a 3-approximation was given by Hochbaum and Shmoys [22]. For the outlier version of knapsack center, very recently, Chakrabarty and Negahbani [5] gave the first non-trivial approximation (a 3-approximation).

Streaming

Charikar et al. [9] and Guha et al. [14] consider k-median with and without outliers in streaming. Guha [15] gives a $(2+\varepsilon)$-approximation one-pass algorithm for k-center that uses $O(k\log(1/\varepsilon)/\varepsilon)$ space, and McCutchen and Khuller [28] give a $(4+\varepsilon)$-approximation one-pass algorithm for k-center with z outliers that uses $O(kz\log(1/\varepsilon)/\varepsilon)$ space. The special cases of 1-center (or, the minimum enclosing ball problem) and 2-center in euclidean spaces have been considered [29, 24, 20] and better approximation ratios than the general k-center problem are known in streaming. Correlation clustering is studied in streaming by Ahn et al. [1]. Cohen-Addad et al. [12] give streaming algorithms for k-center in the sliding windows model, where we want to maintain a solution for only some number of the most recent points in the stream. Guha [15] also gives a space lower bound of $\Omega(n)$ for one-pass algorithms that give a better than 2 approximation for (even the special case of) 1-center by a simple reduction from INDEX, where n is the number of points.

k-center in different models

Chan et al. [6] consider k-center in the fully dynamic adversarial setting, where points can be added or deleted from the input, and the goal is to always maintain a solution by processing the input updates quickly. Malkomes et al. [27] study distributed k-center with outliers.

1.3 Organization of the Paper

We define the model and the problems in Section 2. Section 3 is on the lower bound. In Section 4, we give our important algorithmic ideas and discuss our algorithm for matroid center, and then in Section 5, we discuss the outlier version. In Appendix A, we give the improved space bounds.

2 Preliminaries

A matroid $\mathcal{M}$ is a pair $(E,\mathcal{I})$, where E is a finite set and is called the ground set of the matroid, and $\mathcal{I}$ is a collection of subsets of E that satisfies the following *axioms*:

1. $\emptyset \in \mathcal{I}$,
2. if $J \in \mathcal{I}$ and $I \subseteq J$, then $I \in \mathcal{I}$, and
3. if $I, J \in \mathcal{I}$ and $|I| < |J|$, then there exists $e \in J \setminus I$ such that $I \cup \{e\} \in \mathcal{I}$.

If a set $A \subseteq E$ is in $\mathcal{I}$, then it is called an *independent* set of the matroid $\mathcal{M}$, otherwise it is called a *dependent set*. A singleton dependent set is called a *loop*. *Rank* of a set A, denoted by $\text{rank}(A)$, is the size of a maximal independent set within A; note that rank is a well-defined function because of the third axiom, which is called the *exchange* axiom. Clearly,

for $A \subseteq B$, $\text{rank}(A) \leqslant \text{rank}(B)$. Rank of a matroid is the size of a maximal independent set within E. *Span* of a set A, denoted by $\text{span}(A)$, is the largest set that contains A and has the same rank as A (it can be shown that such a set is unique). We will also use *submodularity* of the rank function, i.e., for $A, B \subseteq E$,

$$\text{rank}(A \cup B) + \text{rank}(A \cap B) \leqslant \text{rank}(A) + \text{rank}(B)\,. \tag{1}$$

A matroid $(E, \mathcal{I})$ is a *partition* matroid if there exists a partition $\{E_1, E_2, \ldots, E_p\}$ of E and nonnegative integers $\ell_1, \ell_2, \ldots, \ell_p$, such that $\mathcal{I} = \{A \subseteq E : \forall i \in [p], |A \cap E_i| \leqslant \ell_i\}$. We say that ℓ_i is the *capacity* of part E_i. Observe that the rank of the matroid is $\sum_{i=1}^{p} \ell_i$.

A metric d over E is a (distance) function $d : E \times E \to \mathbb{R}_+$ that satisfies the following properties for all $e_1, e_2, e_3 \in E$:

1. $d(e_1, e_2) = 0$ if and only if $e_1 = e_2$,
2. $d(e_1, e_2) = d(e_2, e_1)$, and
3. $d(e_1, e_3) \leqslant d(e_1, e_2) + d(e_2, e_3)$; this property is called the *triangle inequality*.

We sometimes call elements in E points. For a point e and a positive number α, the closed ball of radius α around e, denoted by $\mathfrak{B}(e, \alpha)$, is the set $\{x \in E : d(e, x) \leqslant \alpha\}$. We overload d by defining $d(e, A) := \min_{x \in A} d(e, x)$ for $e \in E$ and $A \subseteq E$. The aspect ratio Δ of a metric is the ratio of the largest distance to the smallest in the metric, i.e., $\max_{x,y} d(x, y) / \min_{x,y} d(x, y)$.

The input for the matroid center problem is a matroid $\mathcal{M} = (E, \mathcal{I})$ of rank r and a metric d over E. The goal is to output an independent set S such that its cost $\max_{e \in E} d(e, S)$ is minimized. We are interested in algorithms that assume oracle (or black-box) accesses to the matroid and the metric. The algorithm can ask the matroid oracle whether a set is independent or not, and it can ask the metric oracle (or distance oracle) what the distance between given two points is. In the streaming model, elements of E arrive one by one, and we want to design an algorithm that uses small (sublinear in the input) space. The algorithm can query the oracles only with the elements of E. If the algorithm queries an oracle with an element not in E, then we say that it *fails*. A streaming algorithm can only remember a small part of the input, and the aforementioned restriction disallows plausible learning about forgotten elements indirectly from oracle calls. Also, an algorithm cannot just enumerate elements of E on the fly without looking at the stream, because it does not know the names of the elements in advance.

The input for matroid center with z outliers is also a matroid $\mathcal{M} = (E, \mathcal{I})$ and a metric d over E, but the goal is to output an independent set whose cost is computed with respect to $|E| - z$ closest points. Formally, cost of a set S is $\min\{\alpha \in \mathbb{R}_+ : |E \setminus (\bigcup_{s \in S} \mathfrak{B}(s, \alpha))| \leqslant z\}$.

We denote by OPT the cost of an optimum solution of the instance in the context.

3 Space Lower Bound for One Pass Matroid Center

We show that $\Omega(r^2)$ space is required to achieve better than Δ-approximation for a one-pass algorithm for matroid center. We reduce from the communication problem of INDEX. This reduction is based on the simple reduction for the maximum-matching-size problem: see Figure 1. In INDEX_N, Alice holds an N-bit string and Bob holds an index $I \in [N]$; Alice sends a message to Bob, who has to determine the bit at position I. It is known that Alice has to send a message of size at least $(1 - H_2(3/4))N \geqslant 2N/11$ for Bob to output correctly with a success probability of $3/4$, where H_2 is the binary entropy function.

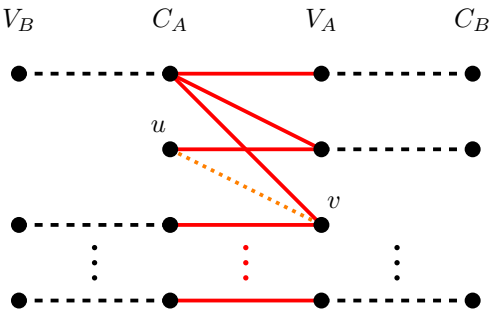

Figure 1 If we have a one-pass streaming algorithm that computes the size of a maximum matching of a k vertex bipartite graph using $o(k^2)$ space, then we can solve INDEX_N using $o(N)$ communication, which would be a contradiction. Alice and Bob agree on a bijection from $[N]$ to the edges of a complete bipartite graph $K_{k,k}$ and construct a graph G as follows. If ℓth bit is 1, Alice adds the corresponding edge (shown in solid red). If the index corresponds to the edge $\{u, v\}$ (shown as a dotted orange edge), Bob adds a new perfect matching between all but vertices u and v and $2k-2$ new vertices (shown as dashed black edges). Alice runs the matching-size estimation algorithm and sends the memory contents to Bob, who continues running it and computes the output. By design, if the index is 1, then maximum-matching-size is $2k-1$, otherwise it is $2k-2$, and an exact algorithm can distinguish between the two cases.

3.1 Reduction from Index to Partition-Matroid Center

We prove the following theorem.

▶ **Theorem 1.** *Any one-pass algorithm for partition-matroid center that outputs a better than Δ-approximation with probability at least $3/4$ must use at least $r^2/24$ bits of space.*

Proof. Assume, towards a contradiction, that there exists a one-pass algorithm for partition-matroid center that outputs a better than Δ-approximation using at most $r^2/24$ bits of space. Then we use it to solve the INDEX problem. Given an input for INDEX, Alice and Bob first construct a bipartite graph G just as described in Figure 1. Then they construct a partition-matroid center instance based on G. Before formalizing the construction, we emphasize that the metric does not correspond to the graph metric given by G, but each edge in G will become a point in the metric. The vertex set they use is union of four sets C_A, V_A, each of size q, and C_B, V_B, each of size $q-1$. Alice constructs a subset of edges between C_A and V_A based on her N-bit string, so we use $N = q^2$. We say that these edges are owned by Alice. If the index that Bob holds corresponds to an edge $\{u, v\}$ with $u \in C_A$ and $v \in V_A$, he adds a perfect matching M between $C_A \setminus \{u\}$ and V_B and a perfect matching M' between $V_A \setminus \{v\}$ and C_B. The edges in $M \cup M'$ are owned by Bob.

To each $u \in C_A \cup C_B$, we associate a cluster $C(u)$ of at most q points in the metric that we will construct, and to each $v \in V_A \cup V_B$, we associate a part $P(v)$ in the partition matroid with capacity 1. Thus, rank of the matroid $r = 2q-1$ because $|V_A \cup V_B| = 2q-1$. By our design, no two clusters will intersect and no two parts will intersect, i.e., $C(u) \cap C(u') = \emptyset$ for $u \neq u'$, and $P(v) \cap P(v') = \emptyset$ for $v \neq v'$. The metric is as follows. Any two points in the same cluster are a unit distance apart and any two points in two different clusters are distance Δ apart. This trivially forms a metric, because the clusters are disjoint. For each $u \in C_A$, Bob adds a point $p(u)$ in the cluster $C(u)$, so that it is nonempty. Add $P' := \{p(u) : u \in C_A\}$ as a part in the partition matroid with capacity 0, so no $p(u)$ can be a center. For each edge $\{u, v\}$ in G with $u \in C_A \cup C_B$ and $v \in V_A \cup V_B$, whoever owns that edge adds a point $p(\{u, v\})$ that goes in cluster $C(u)$ and part $P(v)$. Now, Alice runs the partition-matroid

center algorithm on the points she constructed. She can do this because she knows the metric and the part identity of each point, so she can simulate the distance and matroid oracles. Note that if the algorithm expects an explicit description of the partition matroid, Alice can also send along with each point the identity of the part to which it belongs and the capacity of the part (which is always 1 for her points). She then sends the memory contents to Bob, who continues running the algorithm on his points and computes the cost of the output. We note that Bob can also simulate the distance and matroid oracles. Any point he does not own corresponds to a red edge, and using the identity of that edge, he can figure out the part and cluster to which the point belongs.

Now we prove the correctness of the reduction. Say Bob holds the index corresponding to the edge $\{u, v\}$, where $u \in C_A$ and $v \in V_A$. If the index is 1, then $\{u, v\}$ exists in the graph, then opening centers at points corresponding to edges in $M \cup M' \cup \{u, v\}$ satisfies the partition matroid constraint and also for each $u \in C_A \cup C_B$, we have a center opened in $C(u)$, so the cost is 1. Let the index be 0. We want to show that there is no independent set of cost less than Δ. For a contradiction, assume there is such an independent set. Now, recall that $p(u)$ cannot be a center, so it has to be served by some center in $C(u)$, otherwise the cost will be Δ. Let $p(u)$ be served by some $p(\{u, v'\})$ for $v' \neq v$. Then $p(\{v', w\})$, where $\{v', w\} \in M'$, cannot be a center, because both $p(\{u, v'\}$ and $p(\{v', w\})$ belong to the part $P(v')$ with capacity 1. The point $p(\{v', w\})$ is the lone point in its cluster, and since it cannot be a center, the cost is Δ. If the algorithm is better than Δ-approximation, then Bob can distinguish between these two cases, and thus, solve INDEX_N using communication at most $r^2/24 \leqslant 4q^2/24 = N/6$ bits, which is a contradiction. ◀

After seeing the lower bound, a remark is in order. The difficulty in designing an algorithm is as follows. Even if we know that one center must lie in a ball of small radius centered at a known point, we do not know which points in that ball to store so as to recover an independent set of the matroid.

4 Matroid Center

Our algorithm for matroid center can be seen as a generalization of the algorithm by Hochbaum and Shmoys for k-center [21] adapted to the streaming setting. We first quickly describe the algorithm for k-center. Given an upper bound τ on the optimum cost, the algorithm stores a set C of up to k *pivots* such that distance between any two pivots is more than 2τ. When the algorithm sees a new point e in the stream such that distance between e and any pivot is more than 2τ, it makes e a pivot. The size of C cannot exceed k in this way, because τ is an upper bound on the optimum cost, so no two pivots are served by a single optimum center. Also, any other point is within distance 2τ of some pivot. In the end, the algorithm designates all pivots as centers. In generalizing this to matroid center, one obvious issue is that the set C of pivots constructed as above may not be an independent set for the given general matroid[2]. What we do know is that there has to be an optimum center within distance τ of each pivot. Formally, for $c \in C$, there exists s_c such that $d(c, s_c) \leqslant \tau$ and $\{s_c : c \in C\}$ is an independent set. For each pivot c, we maintain an independent set I_c of nearby points. We prove that it is enough to have each s_c be spanned by some $I_{c'}$ to get a good solution within $\bigcup_{c \in C} I_c$. Algorithm 1 gives a formal description.

[2] This is precisely why we call points in C "pivots" rather than "centers" in this paper.

Note that in Algorithm 1 if we try to add e to I_c under the condition that $d(e,c) \leqslant \tau$, then we may miss spanning some s_c. This will happen if $d(s_c, C) \in (\tau, 2\tau]$, where C is the set of pivots when s_c arrived. Using the condition $d(e,c) \leqslant \tau$ works if each s_c arrives after c though (we use it in the second pass of our two-pass algorithm).

Algorithm 1 One pass algorithm for matroid center.

```
1:  function MatroidCenter(τ,flag)
2:      Initialize pivot-set C ← ∅.
3:      for each point e in the stream do
4:          if there is a pivot c ∈ C such that d(e,c) ⩽ 2τ (pick arbitrary such c) then
5:              if I_c ∪ {e} is independent then
6:                  I_c ← I_c ∪ {e}.
7:          else if |C| = r then                ▷ We cannot have more pivots than the rank.
8:              Abort.      ▷ Because C ∪ {e} acts as a certificate that the guess is incorrect.
9:          else
10:             C ← C ∪ {e}.                                          ▷ Make e a pivot.
11:             If {e} is not a loop, I_e ← {e}, else I_e ← ∅.
12:     if flag = "brute force" then
13:         Find an independent set C'_B in ⋃_{c∈C} I_c such that d(c, C'_B) ⩽ 5τ for c ∈ C.
14:         If such C'_B does not exist, then abort, else return C'_B.
15:     return EfficientMatroidCenter(5τ, C, (I_c)_{c∈C}, M) (given in Algorithm 6
        in Appendix B).
```

First, we quickly bound the space usage.

▶ **Lemma 2.** *In any call to* MatroidCenter*, we store at most* $r^2 + r$ *points.*

Proof. The check on Line 7 ensures that $|C| \leqslant r$. For each pivot c, the size of its independent set I_c is at most r, hence the total number of points stored is at most $r^2 + r$. ◀

Consider a call to MatroidCenter with $\tau \geqslant \mathrm{OPT}$. Let C_E be the set of pivots at the end of the stream. As alluded to earlier, for an optimum independent set I^*, the following holds: for each $c \in C_E$, there exists $s_c \in I^*$ such that $d(c, s_c) \leqslant \tau$, and also $s_c \neq s_{c'}$ for $c \neq c'$, because $d(c, c') > 2\tau$. Now, we prove the following structural lemma that we need later.

▶ **Lemma 3.** *Let* $I_1, \ldots, I_t$ *and* $S = \{s_1, \ldots, s_u\}$ *be independent sets of a matroid such that there is an onto function* $f : [u] \to [t]$ *with the property that* s_i *is in the span of* $I_{f(i)}$ *for* $i \in [u]$*. Then there exists an independent set* B *such that* $|B \cap I_j| \geqslant 1$ *for* $j \in [t]$*.*

Proof. For each $\ell \in \{0, 1, \ldots, u\}$, we construct an independent set S_ℓ such that $|S_\ell| = u$, $|S_\ell \cap I_{f(j)}| \geqslant 1$ for $j \leqslant \ell$, and $s_{\ell+1}, \ldots, s_u \in S_\ell$, then S_u is our desired set B. Start with $S_0 = S$, and assume that we have constructed $S_0, S_1, \ldots, S_{\ell-1}$. If $s_\ell \in I_{f(\ell)}$, we are done, so let $s_\ell \notin I_{f(\ell)}$, then we claim that $\mathrm{rank}((S_{\ell-1} \setminus \{s_\ell\}) \cup I_{f(\ell)}) \geqslant u$. To see this, observe that $\mathrm{rank}(S_{\ell-1}) = u$, so by monotonicity of the rank function, $\mathrm{rank}(S_{\ell-1} \cup I_{f(\ell)}) \geqslant u$, but $s_\ell \in \mathrm{span}(I_{f(\ell)})$, so removing s_ℓ from $S_{\ell-1} \cup I_{f(\ell)}$ would not reduce its rank. We now give a formal argument for completeness. We have $(I_{f(\ell)} \cup \{s_\ell\}) \cup ((S_{\ell-1} \setminus \{s_\ell\}) \cup I_{f(\ell)}) = S_{\ell-1} \cup I_{f(\ell)}$, and $(I_{f(\ell)} \cup \{s_\ell\}) \cap ((S_{\ell-1} \setminus \{s_\ell\}) \cup I_{f(\ell)}) = I_{f(\ell)}$. By submodularity of the rank function (see (1) in Section 2), we have

$$\mathrm{rank}(S_{\ell-1} \cup I_{f(\ell)}) + \mathrm{rank}(I_{f(\ell)}) \leqslant \mathrm{rank}(I_{f(\ell)} \cup \{s_\ell\}) + \mathrm{rank}((S_{\ell-1} \setminus \{s_\ell\}) \cup I_{f(\ell)}) .$$

Let $q = \text{rank}(I_{f(\ell)})$. Since $s_\ell \in \text{span}(I_{f(\ell)})$, we have $\text{rank}(I_{f(\ell)} \cup \{s_\ell\}) = q$ and the above inequality gives

$$u + q \leqslant q + \text{rank}((S_{\ell-1} \setminus \{s_\ell\}) \cup I_{f(\ell)}),$$

which proves the claim. Now, $\text{rank}(S_{\ell-1} \setminus \{s_\ell\}) = u - 1 < \text{rank}((S_{\ell-1} \setminus \{s_\ell\}) \cup I_{f(\ell)})$, therefore there exists $a \in I_{f(\ell)}$ such that $S_\ell := (S_{\ell-1} \setminus \{s_\ell\}) \cup \{a\}$ is independent by the exchange axiom. ◀

▶ **Lemma 4** (Small stream summary for matroid center). *Consider a call to* MatroidCenter *with* $\tau \geqslant \text{OPT}$. *Then there exists an independent set* $B \subseteq \bigcup_{c \in C_E} I_c$ *such that* $d(e, B) \leqslant 7\tau$ *for any point* e *and* $d(c, B) \leqslant 5\tau$ *for any pivot* $c \in C_E$.

Proof. For $c \in C_E$, denote by s_c the optimum center that serves it, so $d(c, s_c) \leqslant \tau$. Let $c' \in C_E$ be such that we tried to add s_c to $I_{c'}$ either on Line 6 or on Line 11; note that c' may not be the same as c if we added it on Line 6. For an $x \in I^*$, let $a(x) \in C_E$ denote the pivot whose independent set $I_{a(x)}$ we tried to add x to. Either we succeeded, in which case $x \in I_{a(x)}$, or we failed, in which case $x \in \text{span}(I_{a(x)})$. In any case, by Lemma 3, for $\mathcal{A} := \{I_{a(x)} : x \in I^*\}$ there exists an independent set B such that $|I \cap B| \geqslant 1$ for all $I \in \mathcal{A}$.

Now, we will bound the cost of B. See Figure 2. Consider any point e in the stream. Let

- $c(e) \in C_E$ be such that $d(e, c(e)) \leqslant 2\tau$,
- $s_{c(e)}$ be the optimum center that serves $c(e)$, so $d(c(e), s_{c(e)}) \leqslant \tau$,
- $a(s_{c(e)}) \in C_E$ be the pivot whose independent set we tried to add $s_{c(e)}$, so $d(s_{c(e)}, a(s_{c(e)}))$ is at most 2τ,
- $c'(e)$ be an arbitrary point in $I_{a(s_{c(e)})} \cap B$, so $d(a(s_{c(e)}), c'(e)) \leqslant 2\tau$ because $c'(e) \in I_{a(s_{c(e)})}$.

Then by triangle inequality, $d(e, B)$ is at most

$$d(e, c'(e)) \leqslant d(e, c(e)) + d(c(e), s_{c(e)}) + d(s_{c(e)}, a(s_{c(e)})) + d(a(s_{c(e)}), c'(e)) \leqslant 2\tau + \tau + 2\tau + 2\tau,$$

which is 7τ; this proves the first part of the lemma.

For any $c \in C_E$, we can bound $d(c, B)$ in a similar way. Let s_c be the optimum center that serves c, and similarly define $a(s_c)$ to be the pivot such that $d(s_c, a(s_c)) \leqslant 2\tau$. Also, let c' be the point in B such that $d(a_{s_c}, c') \leqslant 2\tau$. This gives that $d(c, B) \leqslant d(c, c') \leqslant 5\tau$. ◀

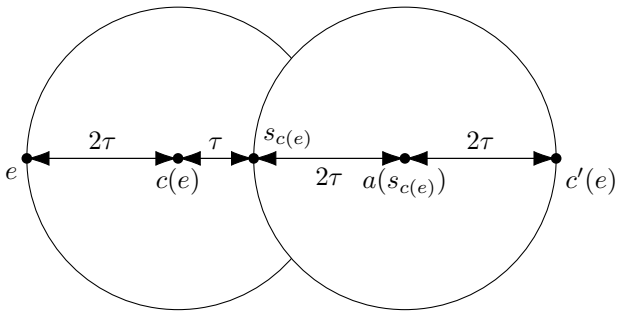

Figure 2 To see how to bound the cost of the independent set B, let e be any point in the stream, $c(e) \in C_E$ be the pivot close to e, $s_{c(e)}$ be the optimum center that covers $c(e)$, $a(s_{c(e)}) \in C_E$ be the pivot close to $s_{c(e)}$, and $c'(e)$ be a point in B that covers $a(s_{c(e)})$.

Before proving our main theorem, we need the following guarantee on the efficient offline 3-approximation algorithm denoted by EfficientMatroidCenter. This algorithm is based on the offline algorithm for matroid center by Chen et al. [11]. We give it as input $\alpha = 5\tau$,

the set C_E of pivots, their independent sets $(I_c)_{c\in C_E}$, and the underlying matroid $\mathcal{M}$ with the promise on the input that there is an independent set $B \subseteq \bigcup_{c\in C_E} I_c$ such that for $c \in C_E$, it holds that $d(c, B) \leqslant 5\tau = \alpha$.

▶ **Theorem 5.** *If* EfficientMatroidCenter *does not fail, then it outputs a set C' such that $d(c, C') \leqslant 3\alpha$ for each $c \in C_E$. If the input promise holds, then* EfficientMatroidCenter *does not fail.*

Proof. This theorem is proved as Theorem 22 in the appendix. See Appendix B. ◀

Now we prove the main result.

▶ **Theorem 6.** *There is an efficient $(17 + \varepsilon)$-approximation one-pass algorithm for matroid center that stores at most $2(r^2 + r) \log_{(1+\varepsilon/17)} \Delta$ points. With a brute force algorithm, one can get a $(7 + \varepsilon)$-approximation.*

Proof. The algorithm is as follows. Let δ be the distance between the first two points. Then for $2 \log_{1+\varepsilon/17} \Delta$ guesses τ of OPT starting from δ/Δ to $\delta\Delta$, we run MatroidCenter(τ, flag). We return the set of centers returned by the instance corresponding to the smallest guess τ. Lemma 2 gives the desired space bound.

Case 1. flag = "brute force".

Suppose the algorithm returned C'_B. Lemma 4 guarantees that for $\tau \in [\mathrm{OPT}, (1 + \varepsilon/17)\,\mathrm{OPT})$, the algorithm will not abort. Then, by the check on Line 14, cost of C'_B is at most $7\tau \leqslant (7 + \varepsilon)\,\mathrm{OPT}$.

Case 2. flag = "efficient algorithm".

Let the algorithm returned C'. Theorem 5 guarantees that for $\tau \in [\mathrm{OPT}, (1+\varepsilon/17)\,\mathrm{OPT})$, the algorithm will not abort. By Theorem 5 for any $c \in C_E$, we have $d(c, C') \leqslant 15\tau$. Since we forget only the points within distance 2τ of C_E, we get that for any point e in the stream, $d(e, C') \leqslant 17\tau \leqslant (17 + \varepsilon)\,\mathrm{OPT}$. ◀

We make some remarks.

▶ Remark 7. We do need to know the rank of the matroid (or an upper bound), otherwise we cannot control the space usage. The instances run using a very small guess may store a very large number of pivots without the check on Line 7.

▶ Remark 8. We can decrease the space usage to $O(r^2 \log(1/\varepsilon)/\varepsilon)$ points using the parallelization ideas of Guha [15]. To make the ideas work, we do need some properties of matroids. We give the details in Appendix A.

▶ Remark 9. By running $\binom{|E|}{2}$ guesses, EfficientMatroidCenter can be used to get an offline 3-approximation algorithm for a more general version of matroid center, where the cost is computed with respect to a subset C_E of E and any point in E can be a center.

4.1 Extension to Knapsack Center

Recall that in the knapsack center problem, each point e has a non-negative weight $w(e)$, and the goal is to select a set C of centers that minimizes the maximum distance between a point and its nearest center subject to the constraint that $\sum_{c\in C} w(c) \leqslant B$, where B is the *budget*. We modify Algorithm 1 slightly to give an algorithm for knapsack center using space r factor smaller than the matroid case, where, in this case, r is the size of a largest feasible set. We make sure that all I_c variables are singletons, so the algorithm stores at most $2r$ points. Instead of the if condition on Line 5, we replace the point x in I_c by e if $w(x) > w(e)$.

This idea works because replacing a point by a nearby point with a smaller weight does not affect the feasibility in the knapsack setting (which could destroy independence in the matroid setting). Let C_E be the set of pivots at the end of the stream. By almost the same argument as in the proof of Lemma 4, we get the following.

▶ **Lemma 10.** *Let $\tau \geqslant \mathrm{OPT}$. Then there exists a feasible set $K \subseteq \bigcup_{c \in C_E} I_c$ such that $d(e, K) \leqslant 7\tau$ for any point e and $d(c, K) \leqslant 5\tau$ for any pivot $c \in C_E$.*

For the efficient version, we then use the 3-approximation algorithm by Hochbaum and Shmoys [22].

▶ **Theorem 11.** *There is an efficient $(17+\varepsilon)$-approximation one-pass algorithm for knapsack center that stores at most $4r \log_{(1+\varepsilon/17)} \Delta$ points, where r is the size of a largest feasible set. With a brute force algorithm, one can get a $(7+\varepsilon)$-approximation.*

4.2 An Efficient Two Pass Algorithm

This algorithm is a streaming two-pass simulation of the offline 3-approximation algorithm of Chen et al. [11] for matroid center. We describe the algorithm and give the analysis below.

In our one-pass algorithm, i.e. Algorithm 1, say we are promised that for any pivot c, the optimum center that serves it appears after c. Then it is enough to try to add e to I_c whenever $d(e, c) \leqslant \tau$; we call this a modified check. Let C_E be the set of pivots in the end, then $(I_c)_{c \in C_E}$ form a partition such that if we pick one point from each I_c to get set B, we can serve each point in C_E using B with cost at most τ. With the modified check, for $c, c' \in C_E$ such that $c \neq c'$, the optimum points s_c and $s_{c'}$ that serve them are also different because $d(c, c') > 2\tau$. Now, $s_c \in \mathrm{span}(I_c)$ due to the promise that s_c arrived after c, and Lemma 3 gives us the required independent set B. We then define a partition matroid $\mathcal{M}_C$ with partition $(I_c)_{c \in C_E}$ and capacities 1 and solve the matroid intersection problem on $\mathcal{M}_c$ and $\mathcal{M}$ restricted to $\bigcup_{c \in C_E} I_c$ and get the output C'. Existence of B guarantees that $|C'| = |C_E|$, thus we are able to serve all points in C_E at a cost of τ. Since the points we forget are within distance 2τ of C_E, our total cost is at most 3τ by triangle inequality. We can get rid of the assumption that s_c arrives after c by having a second pass through the stream. We give a formal description in Algorithm 2.

As in the one-pass algorithm, we run $2 \log_{1+\varepsilon/3} \Delta$ guesses τ of OPT. We return the set of centers returned by the instance corresponding to the smallest guess. For $\tau \in [\mathrm{OPT}, (1+\varepsilon/3)\,\mathrm{OPT})$, the algorithm will not abort due to existence of the independent set B (which we argued earlier). This gives us the following theorem.

▶ **Theorem 12.** *There is an efficient $(3+\varepsilon)$-approximation two-pass algorithm for matroid center that stores at most $2(r^2 + r) \log_{(1+\varepsilon/3)} \Delta$ points.*

5 Matroid Center with Outliers

We first present a simplified analysis of McCutchen and Khuller's algorithm [28] for k-center with z outliers. This abstracts their ideas and sets the stage for the matroid version that we will see later.

5.1 McCutchen and Khuller's Algorithm

As usual, we start with a guess τ for the optimum cost. The algorithm maintains a set C of pivots such that $|\mathfrak{B}(c, 2\tau)| \geqslant z + 1$ for any $c \in C$, so the optimum has to serve at least one of these nearby points. (Recall that $\mathfrak{B}(e, \alpha) = \{x \in E : d(e, x) \leqslant \alpha\}$.) When a new point

Algorithm 2 Two pass algorithm for matroid center.

function MATROIDCENTER2P(τ)
  $C \leftarrow \emptyset$.
  for each point e in the stream **do** ▷ First pass.
    if $d(e, C) \geqslant 2\tau$ **then**
      $C \leftarrow C \cup \{e\}$.
      If $\{e\}$ is not a loop, $I_e \leftarrow \{e\}$, else $I_e \leftarrow \emptyset$.
  for each point e in the stream **do** ▷ Second pass.
    if $\exists c \in C$ such that $d(e, c) \leqslant \tau$ (there can be at most one such c) **then**
      if $I_c \cup \{e\}$ is independent **then**
        $I_c \leftarrow I_c \cup \{e\}$.
  Let $\mathcal{M}_C = (\bigcup_{c \in C} I_c, \mathcal{I}_C)$ be a partition matroid with partition $\{I_c : c \in C\}$ and capacities 1.
  Let $\mathcal{M}'$ be the matroid $\mathcal{M}$ restricted to $\bigcup_{c \in C} I_c$.
  $C' \leftarrow$ MATROID-INTERSECTION($\mathcal{M}_C$, $\mathcal{M}'$)
  if $|C'| < |C|$ **then**
    Return fail with C as certificate.
  Return C'.

arrives, it is ignored if it is within distance 4τ of C. Otherwise it is added to the set F of "free" points. As soon as the size of F reaches $(k - |C| + 1)z + 1$, we know for sure that, for a correct guess, the optimum will have to serve the free points with at most $k - |C|$ clusters, and one of those clusters will have more than z points by the generalized pigeonhole principle. Hence, there *must* exist a free point that has at least z other points within distance 2τ in F, because its cluster diameter is at most 2τ. This gives us a new pivot $c \in F$ with its support points. We remove those points in F that are within distance 4τ of c and continue to the next element in the stream. In the end, we will be left with at most $(k - |C| + 1)z$ free points, and they are served by at most $k - |C|$ optimum centers. On these remaining free points, we run an offline 2-approximation algorithm for $(k - |C|)$-center with z outliers, e.g., that of Chakrabarty et al.[4]. Algorithm 3 gives a formal description. We note that we do not need the sets A_c for $c \in C$ in the algorithm, but we need them in the analysis.

Let us bound the space usage first. The variable C contains at most k pivots, otherwise we abort on Section 5.1, and Section 5.1 make sure that the variable F contains at most $(k + 1)z + 1$ points. In total, we store at most $(k + 1)z + 1$ points at any moment.

▶ **Lemma 13.** *For $\tau \geqslant$ OPT, K-CENTER-Z-OUTLIERS(τ) stores at most $(k + 1)z + 1$ points, and the cost of C' returned by K-CENTER-Z-OUTLIERS(τ) is at most 4τ.*

Proof. Let C_E be the set of pivots and F_E be the set of free points when the stream ended, and let $|C_E| = \ell_E$. We claim that for any $c \neq c'$, where $c, c' \in C_E$, $e \in A_c$, and $e' \in A_{c'}$, we have $d(e, e') > 2\tau$. We now prove this claim. Assume without loss of generality that c was made a pivot before c' by the algorithm. So points within distance 4τ of c were removed from F. Any point that existed in F after this removal, in particular e', must be farther than 4τ from c. This implies that

$$4\tau < d(c, e') \leqslant d(c, e) + d(e, e'), \quad \text{and} \quad d(e, e') > 2\tau\,,$$

because $d(c, e) \leqslant 2\tau$. Now, we know that for $c \in C_E$, there exists $x_c \in A_c$ that has to be served by an optimum center, say s_c, because $|A_c| > z$, so not all of the points in A_c can be outliers. By the earlier claim, for $c \neq c'$, we have $d(x_c, x_{c'}) > 2\tau$ implying that $s_c \neq s_{c'}$ and

Algorithm 3 McCutchen and Khuller's algorithm [28] for k-center with z outliers.

```
1: function K-CENTER-Z-OUTLIERS(τ)
2:     Pivot-set C ← ∅, free-point set F ← ∅, and ℓ ← 0.
3:     for each point e in the stream do
4:         if d(e, C) > 4τ then
5:             F ← F ∪ {e}.
6:         if |F| = (k − ℓ + 1)z + 1 then        ▷ there is a new pivot among the free points;
7:             Let c ∈ F be s.t. |𝔅(c, 2τ) ∩ F| ⩾ z + 1     ▷ such c exists for a correct guess.
8:             If such c does not exist, then abort.
9:             C ← C ∪ {c}.
10:            F ← F \ 𝔅(c, 4τ).
11:            A_c ← {c}∪ arbitrary subset of 𝔅(c, 2τ) \ {c} of size z.
12:            ℓ ← ℓ + 1.
13:            If ℓ = k + 1, then abort.                ▷ guess is wrong.
14:    C_F ← 2-approx for (k − ℓ)-center with z outliers on F by an efficient offline algorithm.
15:    return C′ ← C ∪ C_F.
```

$\ell_E \leqslant k$. Also note that none of these optimum centers can serve a point in F_E, because by triangle inequality

$$d(s_c, F_E) \geqslant d(c, F_E) - d(c, x_c) - d(x_c, s_c) > 4\tau - 2\tau - \tau = \tau$$

for $c \in C_E$. This shows that all but z points in F_E have to be served by at most $k - \ell_E$ optimum centers with cost at most τ. For each of these optimum centers, there exists a free point in F_E within distance τ. So there exists a set B_F of $k - \ell_E$ points in F_E, such that B_F covers all but at most z points of F_E with cost 2τ. So a 2-approximation algorithm recovers $k - \ell_E$ centers with cost at most 4τ. Observing that we only forget points in the stream that are within distance 4τ of some pivot in C_E finishes the proof. ◀

By running K-CENTER-Z-OUTLIERS(τ) for at most $O(\log(1/\varepsilon)/\varepsilon)$ geometrically-increasing active guesses, we get the $(4+\varepsilon)$-approximation algorithm for k-center with z outliers. This analysis is based on that of McCutchen and Khuller [28].

5.2 Matroid Center with Outliers

It is now possible to naturally combine the ideas used for matroid center and those used for k-center with z outliers to develop an algorithm for matroid center with z outliers.

Whenever the free-point set becomes large enough, we create a pivot c and an independent set I_c to which we try to add all free points within distance 4τ of c. We do the same for a new point e in the stream that is within distance 4τ of some pivot $c \in C$, i.e., we try to add it to I_c keeping I_c independent in the matroid. Otherwise $d(e, C) > 4\tau$, so we make it a free point. The structural property of matroids that we proved as Lemma 3 then enables us to show that $\bigcup_{c \in C} I_c$ and the set of free points make a good summary of the stream. See Algorithm 4 for a formal description. Here, we note that we do not need the sets A_c for $c \in C$ in the algorithm if flag is set to "brute force", but we need them in the analysis in any case.

Let C_E be the set of pivots and F_E be the set of free points when the stream ended, and let $\ell_E = |C_E|$.

Algorithm 4 One-pass algorithm for matroid center with outliers.

function MATROID-CENTER-Z-OUTLIERS(τ, flag)
  Pivot-set $C \leftarrow \emptyset$, free-point set $F \leftarrow \emptyset$, and $\ell \leftarrow 0$.
  for each point e in the stream **do**
    if $\exists c \in C$ such that $d(e, c) \leqslant 4\tau$ **then**
      If $I_c \cup \{e\}$ is independent, then $I_c \leftarrow I_c \cup \{e\}$.
    else
      $F \leftarrow F \cup \{e\}$.
    if $|F| = (r - \ell + 1)z + 1$ **then**
      Let $c \in F$ be s. t. $|\mathfrak{B}(c, 2\tau) \cap F| \geqslant z + 1$ (if not, we guessed wrong, so abort).
      $C \leftarrow C \cup \{c\}$.
      $A_c \leftarrow \{c\}$ and if $\{c\}$ is not a loop, $I_c \leftarrow \{c\}$, else $I_c \leftarrow \emptyset$.
      $\ell \leftarrow \ell + 1$ (if ℓ becomes $r + 1$ here, we guessed wrong, so abort).
      for each $x \in F \cap \mathfrak{B}(c, 4\tau)$ **do**
        $F \leftarrow F \setminus \{x\}$.
        If $I_c \cup \{x\}$ is independent, then $I_c \leftarrow I_c \cup \{x\}$.
        If $|A_c| \leqslant z$, then $A_c \leftarrow A_c \cup \{x\}$.
  if flag = "brute force" **then**
    Find an independent set C'_B in $F \cup \bigcup_{c \in C} I_c$ by brute force such that cost of C'_B is $\leqslant 11\tau$ with respect to C and $\leqslant 9\tau$ with respect to all but at most z points of F.
    If such C'_B does not exist, abort, else return C'_B.
  if flag = "efficient" **then**
    Run the offline 3-approximation algorithm by Harris et al. [19] for matroid center with z outliers to get an independent set C' of centers in $F \cup \bigcup_{c \in C}(A_c \cup I_c)$ such that cost of C' is $\leqslant 47\tau$ with respect to C and $\leqslant 45\tau$ with respect to all but z points of F.
    If such C' does not exist, abort, else return C'.

▶ **Lemma 14** (Small summary for matroid center with outliers). *For $\tau \geqslant \mathrm{OPT}$, Algorithm 4 stores at most $O(r^2 + rz)$ points, and there exists an independent set $B \subseteq F_E \cup \bigcup_{c \in C_E} I_c$ such that cost of B is at most 15τ; also $d(c, B) \leqslant 11\tau$ for any pivot $c \in C_E$, and B covers all but at most z points of F_E with cost at most 9τ.*

Proof. Let I^* be an optimum independent set of centers. By the same argument as in the proof of Lemma 13, the following claim is true. For any $c \neq c'$, where $c, c' \in C_E$, $e \in A_c$, and $e' \in A_{c'}$, we have $d(e, e') > 2\tau$. Now, we know that for $c \in C_E$, there exists $x_c \in A_c$ that has to be served by an optimum center, say s_c, because $|A_c| > z$. By the earlier claim, for $c \neq c'$, we have $d(x_c, x_{c'}) > 2\tau$ implying that $s_c \neq s_{c'}$ and $\ell_E \leqslant r$. Let $I^*_{C_E} = \{s_c : c \in C_E\}$ be the set of optimum centers that serve some $x_c \in A_c$ for $c \in C_E$. None of the optimum centers in $I^*_{C_E}$ can serve a point in F_E, because $d(s_c, F_E) > \tau$ for $c \in C_E$. This shows that all but z points in F_E have to be served by at most $r - \ell_E$ optimum centers with cost at most τ. Since $|I_c| \leqslant r$ for any c in the variable C, size of $\bigcup_{c \in C} I_c$ is always bounded by r^2. Also, the check on the size of F ensures that $|F| \leqslant (r + 1)z + 1$, so total number of points stored is at most $O(r^2 + rz)$ at any moment.

When we first process a new point e in the stream, we either try to add it to some I_c or to F. If e is never removed from F, then $e \in F_E$, otherwise, we try to add it to some I_c. The same argument applies to any $x \in I^*$, so if $x \notin F_E$, then we did try to add it to some I_c. For an $x \in I^* \setminus F_E$, let $a(x) \in C_E$ denote the pivot whose independent set $I_{a(x)}$ we tried to add x to.

By Lemma 3, for $\mathcal{A} := \{I_{a(x)} : x \in I^* \setminus F_E\} \cup \{\{x\} : x \in I^* \cap F_E\}$, there exists an independent set B such that $|I \cap B| \geqslant 1$ for all $I \in \mathcal{A}$. Since for $x \in I^* \cap F_E$ the singleton $\{x\} \in \mathcal{A}$, the set B must contain $\{x\}$. For a free point e served by an optimum center s such that we tried to add s to some I_c, we have that $d(e, B) \leqslant d(e, s) + d(s, c) + d(c, B) \leqslant \tau + 4\tau + 4\tau = 9\tau$, which means that B serves all but z points of F_E with cost at most 9τ. Now, we claim that for any point e in the stream, $d(e, B) \leqslant 15\tau$. We just saw that if $e \in F_E$ is served by an optimum center, then $d(e, B) \leqslant 9\tau$, so assume that $e \notin F_E$, that means there is a $c \in C_E$ such that $d(e, c) \leqslant 4\tau$; denote this c by $c(e)$. See Figure 3. Let $s_{c(e)}$ be the optimum center that serves an $x_{c(e)} \in A_{c(e)}$ (recall that such a point exists because $|A_{c(e)}| > z$). So $d(c(e), s_{c(e)}) \leqslant 3\tau$, and $a(s_{c(e)}) \in C_E$ was the pivot such that $d(s_{c(e)}, a(s_{c(e)})) \leqslant 4\tau$. Let $c'(e)$ be an arbitrary point in $I_{a(s_{c(e)})} \cap B$, whose existence is guaranteed by the property of B. We have $d(a(s_{c(e)}), c'(e)) \leqslant 4\tau$, because $c'(e) \in I_{a(s_{c(e)})}$. Then by triangle inequality,

$$\begin{aligned} d(e, B) &\leqslant d(e, c'(e)) \\ &\leqslant d(e, c(e)) + d(c(e), x_{c(e)}) + d(x_{c(e)}, s_{c(e)}) + d(s_{c(e)}, a(s_{c(e)})) + d(a(s_{c(e)}), c'(e)) \\ &\leqslant 4\tau + 2\tau + \tau + 4\tau + 4\tau = 15\tau\,, \end{aligned}$$

hence, cost of B is at most 15τ.

For any $c \in C_E$, we can bound $d(c, B)$ in a similar way. Let s_c be the optimum center that serves an $x_c \in A_c$. Define $a(s_c)$ to be the pivot such that $d(s_c, a(s_c)) \leqslant 4\tau$. Also, let c' be the point in B such that $d(a_{s_c}, c') \leqslant 4\tau$. This gives that $d(c, B) \leqslant d(c, c') \leqslant 11\tau$. We already established that B covers all but at most z points of F_E with cost at most 9τ. The proof is now complete. ◀

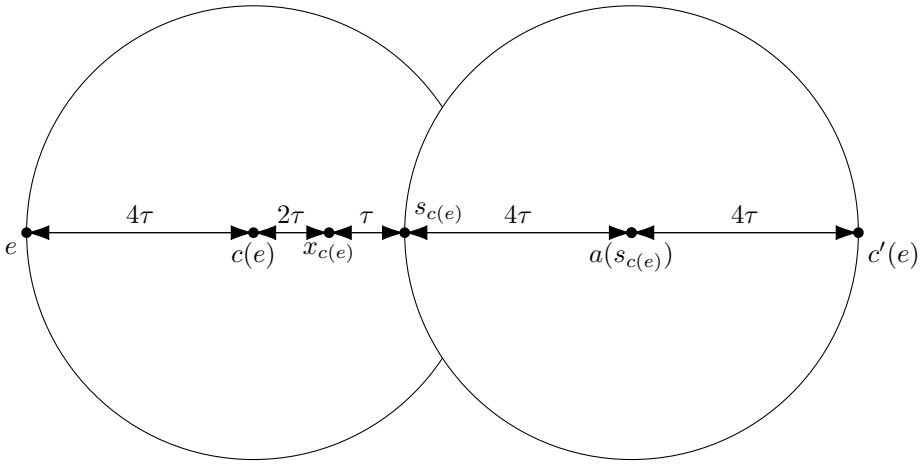

Figure 3 To see how to bound the cost of the independent set B, let e be any point in the stream, $c(e) \in C_E$ be the pivot close to e, $x_{c(e)}$ be the point in the support $A_{c(e)}$ of $c(e)$ that an optimum center serves, $s_{c(e)}$ be the optimum center that serves $x_{c(e)}$, $a(s_{c(e)}) \in C_E$ be the pivot close to $s_{c(e)}$, and $c'(e)$ be a point in B that covers $a(s_{c(e)})$.

▶ **Theorem 15.** *There is an efficient $(51 + \varepsilon)$-approximation one-pass algorithm for matroid center with z outliers that stores at most $O((r^2 + rz) \log \Delta/\varepsilon)$ points. With a brute force algorithm, one can get a $(15 + \varepsilon)$-approximation.*

Proof. We run $O(\log \Delta/\varepsilon)$ parallel copies of MATROID-CENTER-Z-OUTLIERS(τ, flag) and return the output of the copy for the smallest unaborted guess. We claim that the copy corresponding to guess $\tau' \in [\mathrm{OPT}, (1+\varepsilon/50)\,\mathrm{OPT})$, call it $\mathbb{I}(\tau')$, will not abort. Denote by C_E, F_E, and $(I_c)_{c\in C_E}$ contents of the corresponding variables in $\mathbb{I}(\tau')$ at the end of the stream (we will not abort mid-stream because $\tau' \geqslant \mathrm{OPT}$).

By Lemma 14, $F_E \cup \bigcup_{c\in C_E} I_c$ contains a solution that has cost $11\tau'$ with respect C_E and $9\tau'$ with respect to all but at most z of F_E. These checks can be performed by the brute force algorithm. Since any instance for guess τ forgets only those points within distance 4τ of its pivots, the brute force algorithm outputs a $(15+\varepsilon)$-approximation.

By Lemma 14, there exists a solution of cost $\leqslant 15\tau'$, and the efficient 3-approximation algorithm for matroid center with z outliers will return a solution C' with cost at most $45\tau'$. Note that C' has to cover at least one point from A_c for each $c \in C_E$, hence $d(c, C') \leqslant 47\tau'$. Since we forget points only within distance $4\tau'$ of C_E, we get the desired approximation ratio. ◀

References

1 Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation Clustering in Data Streams. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2237–2246, 2015.

2 Ashwinkumar Badanidiyuru Varadaraja. Buyback problem: approximate matroid intersection with cancellation costs. In *Proceedings of the 38th international colloquium conference on Automata, languages and programming - Volume Part I*, ICALP'11, pages 379–390, 2011.

3 Amit Chakrabarti and Sagar Kale. Submodular maximization meets streaming: matchings, matroids, and more. *Mathematical Programming*, 154(1):225–247, 2015. doi:10.1007/s10107-015-0900-7.

4 Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. The Non-Uniform k-Center Problem. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016*, pages 67:1–67:15, 2016.

5 Deeparnab Chakrabarty and Maryam Negahbani. Generalized Center Problems with Outliers. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107, pages 30:1–30:14, 2018.

6 T-H. Hubert Chan, Arnaud Guerqin, and Mauro Sozio. Fully Dynamic k-Center Clustering. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 579–587, 2018.

7 Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental Clustering and Dynamic Information Retrieval. In *Proc. 29th Annual ACM Symposium on the Theory of Computing*, STOC '97, pages 626–635, 1997.

8 Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for Facility Location Problems with Outliers. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 642–651, 2001.

9 Moses Charikar, Liadan O'Callaghan, and Rina Panigrahy. Better Streaming Algorithms for Clustering Problems. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 30–39. ACM, 2003.

10 Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. Streaming Algorithms for Submodular Function Maximization. In *Proc. 42nd International Colloquium on Automata, Languages and Programming*, pages 318–330, 2015.

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

12 Vincent Cohen-Addad, Chris Schwiegelshohn, and Christian Sohler. Diameter and k-Center in Sliding Windows. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55, pages 19:1–19:12, 2016.

13 Teofilo F. Gonzalez. Clustering to Minimize the Maximum Intercluster Distance. *Theor. Comput. Sci.*, 38:293–306, 1985.

14 S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):515–528, May 2003.

15 Sudipto Guha. Tight Results for Clustering and Summarizing Data Streams. In *Proc. 12th International Conference on Database Theory*, ICDT '09, pages 268–275, 2009.

16 MohammadTaghi Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. Budgeted Red-blue Median and Its Generalizations. In *Proceedings of the 18th Annual European Conference on Algorithms: Part I*, ESA'10, pages 314–325. Springer-Verlag, 2010. URL: `http://dl.acm.org/citation.cfm?id=1888935.1888972`.

17 S. L. Hakimi. Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. *Oper. Res.*, 12(3):450–459, June 1964. `doi:10.1287/opre.12.3.450`.

18 S. L. Hakimi. Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems. *Oper. Res.*, 13(3):462–475, June 1965. `doi:10.1287/opre.13.3.462`.

19 David G. Harris, Thomas Pensyl, Aravind Srinivasan, and Khoa Trinh. A Lottery Model for Center-Type Problems with Outliers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 10:1–10:19, 2017. `doi:10.4230/LIPIcs.APPROX-RANDOM.2017.10`.

20 Behnam Hatami and Hamid Zarrabi-Zadeh. A Streaming Algorithm for 2-Center with Outliers in High Dimensions. *Comput. Geom.*, 60:26–36, 2017.

21 Dorit S. Hochbaum and David B. Shmoys. A Best Possible Heuristic for the k-Center Problem. *Math. Oper. Res.*, 10(2):180–184, May 1985. `doi:10.1287/moor.10.2.180`.

22 Dorit S. Hochbaum and David B. Shmoys. A Unified Approach to Approximation Algorithms for Bottleneck Problems. *J. ACM*, 33(3):533–550, May 1986. `doi:10.1145/5925.5933`.

23 Wen-Lian Hsu and George L. Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979. `doi:10.1016/0166-218X(79)90044-1`.

24 Sang-Sub Kim and Hee-Kap Ahn. An improved data stream algorithm for clustering. *Computational Geometry*, 48(9):635–645, 2015. `doi:10.1016/j.comgeo.2015.06.003`.

25 Ravishankar Krishnaswamy, Amit Kumar, Viswanath Nagarajan, Yogish Sabharwal, and Barna Saha. The Matroid Median Problem. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 1117–1130, 2011.

26 Silvio Lattanzi, Stefano Leonardi, Vahab Mirrokni, and Ilya Razenshteyn. Robust Hierarchical k-Center Clustering. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, ITCS '15, pages 211–218, 2015.

27 Gustavo Malkomes, Matt J Kusner, Wenlin Chen, Kilian Q Weinberger, and Benjamin Moseley. Fast Distributed k-Center Clustering with Outliers on Massive Data. In *Advances in Neural Information Processing Systems 28*, pages 1063–1071. Curran Associates, Inc., 2015. URL: `http://papers.nips.cc/paper/5997-fast-distributed-k-center-clustering-with-outliers-on-massive-data.pdf`.

28 Richard Matthew McCutchen and Samir Khuller. Streaming Algorithms for k-Center Clustering with Outliers and with Anonymity. In *Proc. 11th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 165–178, 2008.

29 Hamid Zarrabi-Zadeh and Asish Mukhopadhyay. Streaming 1-Center with Outliers in High Dimensions. In *Proceedings of the 21st Annual Canadian Conference on Computational Geometry, Vancouver, British Columbia, Canada, August 17-19, 2009*, pages 83–86, 2009.

A Handling the Guesses

We extend the ideas of Guha [15] and McCutchen and Khuller [28] to run $O(\log(1/\varepsilon)/\varepsilon)$ active guesses. Although, to make this idea work for matroids, we do need a property of matroids (see Lemma 16). The way to do this is to start with a lower bound R on the optimum and spawn instances, which we call *original* instances, $\mathbb{I}(\tau)$ for guesses $\tau = R, R(1+\varepsilon), \ldots, R(1+\varepsilon)^{\beta} = R\alpha/\varepsilon$, for some α that depends on the basic algorithm that we use, e.g., for matroid center, we will use $\alpha = 2+\varepsilon$. When a guess τ' fails, we replace an instance $\mathbb{I} = \mathbb{I}(\tau)$ for $\tau \leqslant \tau'$ with a new instance, which we call its *child* instance, $\mathbb{I}_N = \mathbb{I}(\tau(1+\varepsilon)^{\beta})$. In the new instance $\mathbb{I}_N$, we treat the summary that we maintained for $\mathbb{I}(\tau)$ as the initial stream. Since the new guess in $\mathbb{I}_N$ is about $1/\varepsilon$ times larger than the old guess in $\mathbb{I}$, the distance between a point that we forgot and the summary stored by $\mathbb{I}$ is about ε times the new guess. Therefore, the cost analysis does not get much affected for a correct guess. If we forgot an optimum center, a nearby point in the summary can act as its replacement. This statement is obvious for a uniform matroid, because all points are treated the same way within the matroid, but it is not true for general matroids; in fact, as exhibited by our lower bound, it is not true even for partition matroids. So with each point in the summary, we pass to the new instance an independent set I_o. The following simple lemma shows that if an optimum center x is in the span of I_o, and if we construct I_c for a new pivot c such that $I_o \subseteq \text{span}(I_c)$, then I_c also spans the optimum center.

▶ **Lemma 16.** *Let I and J be independent sets of a matroid such that $J \subseteq \text{span}(I)$. If $e \in \text{span}(J)$, then $e \in \text{span}(I)$.*

Proof. Let $\text{rank}(I) = q$. Towards a contradiction, let $\text{rank}(I \cup \{e\}) = q+1$. Since $J \subseteq \text{span}(I)$, $\text{rank}(I \cup J) = q$. Now, $e \in \text{span}(J)$, so $\text{rank}(I \cup J \cup \{e\}) = q$, i.e., $\text{rank}(I \cup \{e\}) \leqslant q$, which gives us the desired contradiction. ◀

A.1 A Smaller Space Algorithm for Matroid Center

We modify the function MatroidCenter(τ,flag) from earlier to accept a starting stream and an independent set for each point in the starting stream: MatroidCenter(τ, C_o, $(J_{c_o})_{c_o \in C_o}$,flag). Before processing any new points in the stream we process the points in C_o as follows. When processing a $c_o \in C_o$, if $d(c_o, C) \leqslant 2\tau$, try to add points in J_{c_o} to I_c. Otherwise create a new pivot c in C and initialize $I_c = J_{c_o}$. Once C_o is processed, we continue with the stream and work exactly as in MatroidCenter(τ). We give complete pseudocode in Algorithm 5.

For an instance $\mathbb{I}(\tau)$ let $C_o(\tau)$ be the initial summary and $\mathcal{J}(\tau)$ be the collection of independent sets that we passed to it, and let $E(\tau)$ be the part of the actual stream that it processed. Also, let $\mathbb{I}(\tau_o)$ be the instance for $\tau_o = \varepsilon\tau/(2+\varepsilon)$ from which $\mathbb{I}(\tau)$ was spawned.

▶ **Lemma 17.** *Let e be a point that arrived before the substream $E(\tau)$. Then e has a nearby* representative *$\rho_e \in C_o(\tau)$ such that $d(e, \rho_e) \leqslant \varepsilon\tau$ and also the independent set J_{ρ_e} corresponding to ρ_e spans e.*

Proof. We prove this claim by induction on the number of ancestors. For an original instance, the claim holds trivially, because no point arrived before. Otherwise, there are two cases: either $e \in E(\tau_o)$ or e arrived before $E(\tau_o)$. If $e \in E(\tau_o)$, then by the logic of the algorithm, there exists $a(e) \in C_o(\tau)$ such that $d(e, a(e)) \leqslant 2\tau_o = 2\varepsilon\tau/(2+\varepsilon) \leqslant \varepsilon\tau$, and also we tried

Algorithm 5 One pass algorithm for matroid center with smaller space.

Let R be the minimum distance for some two points in the first $r+1$ points in the stream.
for $\tau \in \{R, R(1+\varepsilon), \ldots, R(1+\varepsilon)^{\beta} = (2+\varepsilon)R/\varepsilon\}$ in parallel **do**
    MatroidCenter(τ, $\emptyset$, $\emptyset$).
    if an instance with guess τ is aborted **then**
        for all active $\mathbb{I}(\tau')$ with guess $\tau' \leqslant \tau$, current pivots C_o, and independent sets $(J_{c_o})_{c_o \in C_o}$ **do**
            Replace it with the child instance MatroidCenter($\tau'(1+\varepsilon)^{\beta}$, C_o, $(J_{c_o})_{c_o \in C_o}$, flag).
Return the set C' of centers returned by the active instance with the smallest guess.

function MatroidCenter(τ, C_o, $(J_{c_o})_{c_o \in C_o}$, flag)
    $C \leftarrow \emptyset$.
    for each point c_o in C_o **do**
        if $\exists c \in C$ such that $d(c_o, c) \leqslant 2\tau$ (pick arbitrary such c if there are several) **then**
            for $e_o \in J_{c_o}$ **do**
                if $I_c \cup \{e_o\}$ is independent **then**
                    $I_c \leftarrow I_c \cup \{e_o\}$.
        else
            $C \leftarrow C \cup \{c_o\}$.
            $I_{c_o} \leftarrow J_{c_o}$.
    # Processing of the old pivots finished, continue with the actual stream.
    for each point e in the stream **do**
        if there is a pivot $c \in C$ such that $d(e, c) \leqslant 2\tau$ (pick arbitrary such c) **then**
            if $I_c \cup \{e\}$ is independent **then**
                $I_c \leftarrow I_c \cup \{e\}$.
        else if $|C| = r$ **then** ▷ We cannot have more pivots than the rank.
            Abort. ▷ Because $C \cup \{e\}$ acts as a certificate that the guess is incorrect.
        else
            $C \leftarrow C \cup \{e\}$. ▷ Make e a pivot.
            If $\{e\}$ is not a loop, $I_e \leftarrow \{e\}$, else $I_e \leftarrow \emptyset$.
    if flag = "brute force" **then**
        Find an independent set C'_B in $\bigcup_{c \in C} I_c$ such that $d(c, C'_B) \leqslant (5+2\varepsilon)\tau$ for $c \in C$.
        If such C'_B does not exist, then abort, else return C'_B.
    return EfficientMatroidCenter($(5+2\varepsilon)\tau$, C, $(I_c)_{c \in C}$, $\mathcal{M}$).

to add e to $I_{a(e)}$ (that became $J_{a(e)}$ for the next instance $\mathbb{I}(\tau)$). Otherwise, by induction hypothesis, there is a point $e' \in C_o(\tau_o)$ such that $d(e, e') \leqslant \varepsilon\tau_o$ and $J_{e'}$ spans e. Now, let $\rho_{e'} \in C_o(\tau)$ be such that $d(e', \rho_{e'})) \leqslant 2\tau_o$ (such $\rho_{e'}$ must exist by logic of the algorithm). Using triangle inequality and the above inequality that $d(e, e') \leqslant \varepsilon\tau_o$, we get

$$d(e, \rho_{e'}) \leqslant d(e, e') + d(e', \rho_{e'}) \leqslant \varepsilon\tau_o + 2\tau_o = (2+\varepsilon)\tau_o = (2+\varepsilon)\frac{\varepsilon\tau}{(2+\varepsilon)} = \varepsilon\tau\,.$$

Moreover, in the instance $\mathbb{I}(\tau_o)$, we tried to add all points in $J_{e'}$ to $I_{\rho_{e'}}$, so by Lemma 16, $e \in \mathrm{span}(I_{\rho_{e'}})$ (see that $I_{\rho_{e'}}$ became $J_{\rho_{e'}}$ for the next instance $\mathbb{I}(\tau)$), which proves the claim. ◀

▶ **Theorem 18.** *There is an efficient* $((17+7\varepsilon)(1+\varepsilon))$*-approximation one-pass algorithm for matroid center that stores at most* $O(r^2 \log(1/\varepsilon)/\varepsilon)$ *points. With a brute force algorithm, one can get a* $((7+3\varepsilon)(1+\varepsilon))$*-approximation.*

Proof. Space usage is easy to analyze. At any time, we have at most $O(\log_{1+\varepsilon}(1/\varepsilon)) = O(\log(1/\varepsilon)/\varepsilon)$ active instances and each instance stores at most $O(r^2)$ points.

Consider the instance $\mathbb{I}(\tau')$ for which we returned on Line 7 in Algorithm 5, and suppose the outputs were C' or C'_B (depending on "flag"). We note that some active copy *will* return, because τ cannot keep on increasing indefinitely. E.g., consider τ larger than the maximum distance between any two points. Let C_E be the contents of the variable C in $\mathbb{I}(\tau')$ at the end of the stream. Then we know that costs of C'_B and C' are at most $(5+2\varepsilon)\tau'$ and $(15+6\varepsilon)\tau'$ with respect to C_E due to the check that we do on Line 31 and by Theorem 5 for EfficientMatroidCenter. By Lemma 17, any point that arrived before $E(\tau')$ is within distance $\varepsilon\tau'$ of $C_o(\tau')$, and each point in $C_o(\tau')$ is within distance $2\tau'$ of C_E, which shows that costs of C'_B and C' are at most $(7+3\varepsilon)\tau'$ and $(17+7\varepsilon)\tau'$ with respect to the whole stream (by triangle inequality). Next, we show that $\tau' \leqslant (1+\varepsilon)\,\mathrm{OPT}$, and that will finish the proof.

Consider the guess $\tau \in (\mathrm{OPT}, (1+\varepsilon)\,\mathrm{OPT}]$. If τ was never active, that means $\tau' \leqslant \mathrm{OPT}$, and we are done. Otherwise, τ was active, and we will prove that it was not aborted. Since $\tau \leqslant \mathrm{OPT}$, we will not abort mid-stream in $\mathbb{I}(\tau)$, so let C_E be the set of pivots at the end of the stream in $\mathbb{I}(\tau)$. We will show that there is an independent set B such that cost of B with respect to C_E is at most $(5+2\varepsilon)\tau$. By Line 31 and by Theorem 5 for EfficientMatroidCenter, this would imply that $\mathbb{I}(\tau)$ cannot abort.

From here on, the proof follows that of Lemma 4. Let $c \in C_E$. Denote by s_c the optimum center that serves it, so $d(c, s_c) \leqslant \tau$. If $s_c \in E(\tau)$, then $s_c \in \mathrm{span}(I_{c'})$ for some $c' \in C_E$ and $d(s_c, c') \leqslant 2\tau$. Otherwise, s_c arrived before $E(\tau)$. Let ρ_{s_c} be the representative of s_c whose existence is guaranteed by Lemma 17, so $d(s_c, \rho_{s_c}) \leqslant \varepsilon\tau$. Then let $c' \in C_E$ be such that $d(\rho_{s_c}, c') \leqslant 2\tau$ and $J_{\rho_{s_c}}$ is spanned by $I_{c'}$. Thus, by triangle inequality

$$d(s_c, c') \leqslant d(s_c, \rho_{s_c}) + d(\rho_{s_c}, c') \leqslant \varepsilon\tau + 2\tau = (2+\varepsilon)\tau\,, \tag{2}$$

and by Lemma 16, s_c is spanned by $I_{c'}$. Denote by $\mathcal{A}$ the collection of such $I_{c'}$'s. Now, by Lemma 3, there exists an independent set B such that $|I \cap B| \geqslant 1$ for all $I \in \mathcal{A}$. Pick c_p from $I_{c'} \cap B$. Either $c_p \in E(\tau)$ or it arrived before. In any case, again using Lemma 17, we have $d(c_p, c') \leqslant (2+\varepsilon)\tau$ (we use this below), and

- $d(c, s_c) \leqslant \tau$, because s_c is the optimum center that covers c,
- $d(s_c, c') \leqslant (2+\varepsilon)\tau$, by Inequality (2), and
- $d(c', c_p) \leqslant (2+\varepsilon)\tau$.

Thus, by triangle inequality, $d(c, B) \leqslant (5+2\varepsilon)\tau$. So $\mathbb{I}(\tau)$ will not abort. This finishes the proof. ◀

Reducing the space usage for matroid center with z outliers can be done by naturally combining the techniques above and those in Section 5.2. We define a similar overloading matroid-center-z-outliers(τ, C_o, $(J_{c_o})_{c_o \in C_o}$, F_o, flag), where F_o contains the set of free points in $\mathbb{I}(\tau_o)$ when it aborted and this function was called with the updated guess τ. We skip the details and state the following theorem without proof.

▶ **Theorem 19.** *There is an efficient* $(51+\varepsilon)$*-approximation one-pass algorithm for matroid center with* z *outliers that stores at most* $O((r^2+rz)\log(1/\varepsilon)/\varepsilon)$ *points. With a brute force algorithm, one can get a* $(15+\varepsilon)$*-approximation.*

A.2 Extension to Knapsack Center

In Section 4.1, we saw how to modify Algorithm 1 to get an algorithm for knapsack center that stores at most $2r$ points, where r is the size of a largest feasible set. Using the same idea, algorithms for two-pass matroid center, matroid center with outliers, and smaller space matroid center, which are Algorithms 2, 4 and 6, can be extended to the knapsack center without losing the approximation ratio and with a space r factor smaller than the matroid case. For the outlier version of knapsack center, to get an efficient algorithm, we use the 3-approximation algorithm by Chakrabarty and Negahbani [5]. So we get the following theorems, where r is the size of a largest feasible set.

▶ **Theorem 20.** *There is an efficient $(17+\varepsilon)$-approximation one-pass algorithm for knapsack center that stores at most $O(r\log(1/\varepsilon)/\varepsilon)$ points. With a brute force algorithm, one can get a $(7+\varepsilon)$-approximation.*

▶ **Theorem 21.** *There is an efficient $(51+\varepsilon)$-approximation one-pass algorithm for knapsack center with z outliers that stores at most $O(rz\log(1/\varepsilon)/\varepsilon)$ points. With a brute force algorithm, one can get a $(15+\varepsilon)$-approximation.*

B An Implementation of Efficient Matroid Center

We now give an implementation of EfficientMatroidCenter. The input consists of α, C_E, X, such that $C_E \subseteq X$, and the underlying matroid $\mathcal{M}$ defined over X. Furthermore, the promise is that there is an independent set $B \subseteq X$ such that for each $c \in C_E$, we have $d(c,B) \leqslant \alpha$. Our implementation is based on the algorithm of Chen et al. [11] for matroid center. We show that it outputs a set C' such that, assuming the promise, $d(c,C') \leqslant 3\alpha$ for $c \in C_E$.

Algorithm 6 Efficient algorithm for matroid center based on the algorithm by [11].

1: **function** EfficientMatroidCenter(α, C_E, X, $\mathcal{M}$)
2: Initialize: $C \leftarrow \emptyset$.
3: **while** there is an unmarked point e in C_E **do**
4: $C \leftarrow C \cup \{e\}$, $B_e \leftarrow \mathfrak{B}(e,\alpha) \cap X$, and mark all points in $\mathfrak{B}(e,2\alpha) \cap C_E$.
5: Let $\mathcal{M}_C = (\cup_{c\in C} B_c, \mathcal{I}_C)$ be a partition matroid with partition $\{B_c : c \in C\}$ and capacities 1.
6: Let $\mathcal{M}'$ be the matroid $\mathcal{M}$ restricted to $\cup_{c\in C} B_c$.
7: $C' \leftarrow$ matroid-intersection($\mathcal{M}_C$, $\mathcal{M}'$)
8: **if** $|C'| < |C|$ **then**
9: Return fail.
10: Return C'.

▶ **Theorem 22.** *If* EfficientMatroidCenter *does not fail, then it outputs a set C' such that $d(c,C') \leqslant 3\alpha$ for each $c \in C_E$. If the input promise holds, then* EfficientMatroidCenter *does not fail.*

Proof. In this proof, we refer by C the contents of the variable C after the while loop ended, and let c_E be any arbitrary point in C_E. Define the function Marker : $C_E \to C$ such that Marker$(c_E) \in C$ is the "marker" of c_E, i.e., we marked c_E when processing Marker(c_E). In the end, all c_E's are marked, so Marker is a valid function. By the logic on Line 4, we have that

$$d(c_E, \text{Marker}(c_E)) \leqslant 2\alpha\,. \tag{3}$$

Let EfficientMatroidCenter does not fail, then $|C'| \geqslant |C|$ and C' satisfies the partition matroid constraint of $\mathcal{M}_C$. By definition of $\mathcal{M}_C$, $\operatorname{rank}(\mathcal{M}_C) = |C|$, hence $|C'| \leqslant |C|$, which implies that $|C'| = |C|$. Therefore, for each $c \in C$, the set C' must contain exactly one element in $\mathfrak{B}(c, \alpha)$ and $d(c, C') \leqslant \alpha$, in particular, $d(\text{Marker}(c_E), C') \leqslant \alpha$. This, triangle inequality, and Inequality (3) gives

$$d(c_E, C') \leqslant d(c_E, \text{Marker}(c_E)) + d(\text{Marker}(c_E), C') \leqslant 2\alpha + \alpha = 3\alpha\,,$$

which proves the first part of the statement of the lemma. We prove the second part next.

Assume that the promise holds. Then let B be the set such that cost of B is at most α with respect to C_E, in particular, with respect to C. For $c \in C$, define $\text{Coverer}(c) \in B$ to be an arbitrarily chosen "coverer" of c, i.e.,

$$d(c, \text{Coverer}(c)) \leqslant \alpha\,. \tag{4}$$

Then the set $B' := \{\text{Coverer}(c) : c \in C\}$ is a subset of B, so it is independent in $\mathcal{M}$. Now, for $c, c' \in C$, such that $c \neq c'$, we have $\text{Coverer}(c) \neq \text{Coverer}(c')$ by Inequality (4) because $d(c, c') > 2\alpha$. This implies that $|B'| = |C|$. Next, $\text{Coverer}(c) \in B' \cap B_c$ for each $c \in C$, hence the set B' is also independent in $\mathcal{M}_C$. Therefore $B' \in \mathcal{M}_C \cap \mathcal{M}'$, and matroid-intersection returns an independent set of size $|C|$, i.e., it does not fail. ◀

▶ Remark 23. By running $\binom{|X|}{2}$ guesses, EfficientMatroidCenter can be used to get an offline 3-approximation algorithm for a more general version of matroid center, where the cost is computed with respect to a subset C_E of X and any point in X can be a center.