Maximum Coverage in the Data Stream Model: Parameterized and Generalized

Andrew McGregor
mcgregor@cs.umass.edu
University of Massachusetts

David Tench
dtench@cs.umass.edu
University of Massachusetts

Hoa T. Vu
hvu2@sdsu.edu
San Diego State University

ABSTRACT

We present algorithms for the Max-k-Cover and Max-k-UniqueCover problems in the data stream model. The input to both problems are m subsets of a universe of size n and a value $k \in [m]$. In Max-k-Cover, the problem is to find a collection of at most k sets such that the number of elements covered by at least one set is maximized. In Max-k-UniqueCover, the problem is to find a collection of at most k sets such that the number of elements covered by exactly one set is maximized. These problems are closely related to a range of graph problems including matching, partial vertex cover, and capacitated maximum cut. In the stream model, we assume k is given and the sets are revealed online. Our goal is to design single-pass algorithms that use space that is sublinear in the input size. Our main algorithmic results are as follows.

- If sets have size at most d, there exist single-pass algorithms using $O(d^{d+1}k^d)$ space that solve both problems exactly. This is optimal up to logarithmic factors for constant d.
- If each element appears in at most r sets, we present single pass algorithms using $\tilde{O}(k^2 r/\epsilon^3)$ space that return a $1+\epsilon$ approximation in the case of Max-k-Cover and $2+\epsilon$ approximation in the case of Max-k-UniqueCover. We also present a single-pass algorithm using slightly more memory, i.e., $\tilde{O}(k^3 r/\epsilon^4)$ space, that $1+\epsilon$ approximates Max-k-UniqueCover.

In contrast to the above results, when d and r are arbitrary, any constant pass $1+\epsilon$ approximation algorithm for either problem requires $\Omega(\epsilon^{-2}m)$ space but a single pass $O(mk/\epsilon^2)$ space algorithm exists. In fact any constant-pass algorithm with an approximation better than $e^{1-1/k}$ requires $\Omega(m/k^2)$ space when d and r are unrestricted. En route, we also obtain an algorithm for the parameterized version of the streaming SetCover problem.

ACM Reference Format:
Andrew McGregor, David Tench, and Hoa T. Vu. 2020. Maximum Coverage in the Data Stream Model: Parameterized and Generalized. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Problem Description. We consider the Max-k-Cover and Max-k-UniqueCover problems in the data stream model. The input to both problems are m subsets of a universe of size n and a value $k \in [m]$. In Max-k-Cover, the problem is to find a collection of at most k sets such that the number of elements covered by at least one set is maximized. In Max-k-UniqueCover, the problem is to find a collection of at most k sets such that the number of elements covered by exactly one set is maximized. In the stream model, we assume k is provided but that the sets are revealed online and our goal is to design single-pass algorithms that use space that is sub-linear in the input size.

Max-k-Cover is a classic NP-Hard problem that has a wide range of applications including facility and sensor allocation [50], information retrieval [5], influence maximization in marketing strategy design [46], and the blog monitoring problem where we want to choose a small number of blogs that cover a wide range of topics [63]. It is well-known that the greedy algorithm, which greedily picks the set that covers the most number of uncovered elements, is a $e/(e-1)$ approximation and that unless $P = NP$, this approximation factor is the best possible [30].

Max-k-UniqueCover was first studied in the offline setting by Demaine et al. [25]. A motivating application for this problem was in the design of wireless networks where we want to place base stations that cover mobile clients. Each station could cover multiple clients but unless a client is covered by a unique station the client would experience too much interference. Demaine et al. [25] gave a polynomial time $O(\log k)$ approximation. Furthermore, they showed that Max-k-UniqueCover is hard to approximate within a factor $O(\log^\sigma n)$ for some constant σ under reasonable complexity assumptions. Erlebach and van Leeuwen [29] and Ito et al. [40] considered a geometric variant of the problem and Misra et al. [60] considered the parameterized complexity of the problem. This problem is also closely related to Minimum Membership Set Cover where one has to cover every element and minimizes the maximum overlap on any element [26, 51].

In the streaming set model, Max-k-Cover and the related SetCover problem[1] have both received a significant amount of attention [7, 15, 27, 36, 38, 39, 59, 63]. The most relevant result is a single-pass $2+\epsilon$ approximation using $\tilde{O}(k\epsilon^{-2})$ space [8, 59] although better approximation is possible in a similar amount of space if multiple passes are permitted [59] or if the stream is randomly ordered [61]. In this paper, we almost exclusively consider single-pass algorithms where the sets arrive in an arbitrary order. The unique coverage problem has not been studied in the data stream model although it, and Max-k-Cover, are closely related to various graph problems that have been studied.

[1] That is, find the minimum number of sets that cover the entire universe.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Relationship to Graph Streaming. There are two main variants of the graph stream model. In the *arbitrary order model*, the stream consists of the edges of the graph in arbitrary order. In the *adjacency list model*, all edges that include the same node are grouped together. Both models generalize naturally to hypergraphs where each edge could consists of more than two nodes. The arbitary order model has been more heavily studied than the adjacency list model but there has still been a significant amount of work in the latter model [6, 7, 11, 36, 41, 48, 55–57]. For further details, see a recent survey on work on the graph stream model [54].

To explore the relationship between Max-k-Cover and Max-k-UniqueCover and various graph stream problems, it makes sense to introduce to additional parameters beyond m (the number of sets) and n (the size of the universe). Specifically, throughout the paper we let d denote the maximum cardinality of a set in the input and let r denote the maximum multiplicity of an element in the universe where the *multiplicity* is the number of sets the element appears.[2] Then an input to Max-k-Cover and Max-k-UniqueCover can define a (hyper)graph in one of the following two natural ways:

1. *First Interpretation:* A sequence of (hyper-)edges on a graph with n nodes of maximum degree r (where the degree of a node v corresponds to how many hyperedges include that node) and m hyperedges where each hyperedge has size at most d. In the case where every set has size $d = 2$, the hypergraph is an *ordinary graph*, i.e., a graph where every edge just has two endpoints. With this interpretation, the graph is being presented in the arbitrary order model.
2. *Second Interpretation:* A sequence of adjacency lists (where the adjacency list for a given node includes all the hyperedges) on a graph with m nodes of maximum degree d and n hyperedges of maximum size r. In this interpretation, if every element appears in exactly $r = 2$ sets, then this corresponds to an ordinary graph where each element corresponds to an edge and each element corresponds to an edge. With this interpretation, the graph is being presented in the adjacency list model.

Under the first interpretation, the Max-k-Cover problem and the Max-k-UniqueCover problem when all sets have exactly size 2 naturally generalize the problem of finding a maximum matching in an ordinary graph in the sense that if there exists a matching with at least k edges, the optimum solution to either Max-k-Cover and Max-k-UniqueCover will be a matching. There is a large body of work on graph matchings in the data stream model [2, 12, 23, 24, 28, 31, 34, 35, 42, 43, 47–49, 53, 65] including work specifically on solving the problem exactly if the matching size is bounded [18, 20]. More precisely, Max-k-Cover corresponds to the partial vertex cover problem [52]: what is the maximum number of edges that can be covered by selecting k nodes. For larger sets, the Max-k-Cover and Max-k-UniqueCover are at least as hard as finding partial vertex covers and matching in hypergraphs.

Under the second interpretation, when all elements have multiplicity 2, Max-k-UniqueCover corresponds to finding the capacitated maximum cut, i.e., a set of at most k vertices such that the number of edges with exactly one endpoint in this set is maximized. In the offline setting, Ageev and Sviridenko [1] and Gaur et al. [33] presented a 2 approximation for this problem using linear programming and local search respectively. The (uncapacitated) maximum cut problem was been studied in the data stream model by Kapralov et al. [44, 45]; a 2-approximation is trivial in logarithmic space[3] but improving on this requires space that is polynomial in the size of the graph. The capacitated problem is a special case of the problem of maximizing a non-monotone sub-modular function subject to a cardinality constraint. This general problem has been considered in the data stream model [8, 13, 16, 37] but in that line of work it is assumed that there is oracle access to the function being optimized, e.g., given any set of nodes, the oracle will return the number of edges cut. Alaluf et al. [3] presented a $2 + \epsilon$ approximation in this setting, assuming exponential post-processing time. In contrast, our algorithm does not assume an oracle while obtaining a $1 + \epsilon$ approximation (and also works for the more general problem Max-k-UniqueCover).

1.1 Our Results

Our main results are the following single-pass stream algorithms[4]:

(A) Bounded Set Cardinality. If all sets have size at most d, there exists a $\tilde{O}(d^{d+1}k^d)$ space data stream algorithm that solves Max-k-UniqueCover and Max-k-Cover exactly. We show that this is nearly optimal in the sense that any exact algorithm requires $\Omega(k^d)$ space.

(B) Bounded Multiplicity. If all elements occurs in at most r sets, we present the following algorithms:

- (B1) Max-k-UniqueCover: There exists a $2 + \epsilon$ approximation algorithm using $\tilde{O}(\epsilon^{-3}k^2 r)$ space.
- (B2) Max-k-UniqueCover: We show that the approximation factor can be improved to $1 + \epsilon$ at the expense of increasing the space use to $\tilde{O}(\epsilon^{-4}k^3 r)$.
- (B3) Max-k-Cover: There exists a $1 + \epsilon$ approximation algorithm using $\tilde{O}(\epsilon^{-3}k^2 r)$ space.

In contrast to the above results, when d and r are arbitrary, constant pass $1 + \epsilon$ approximation algorithm for either problem requires $\Omega(\epsilon^{-2}m)$ space [6].[5] We also generalize of lower bound for Max-k-Cover [59] to Max-k-UniqueCover to show that any constant-pass algorithm with an approximation better than $e^{1-1/k}$ requires $\Omega(m/k^2)$ space. We also present a single-pass algorithm with an $O(\log \min(k, r))$ approximation for Max-k-UniqueCover using $\tilde{O}(k^2)$ space, i.e., the space is independent of r and d but the approximation factor depends on r. This algorithm is a simple combination of a Max-k-Cover algorithm due to [59] and an algorithm for Max-k-UniqueCover in the offline setting due to Demaine et al. [25]. Finally, our Max-k-Cover result (B3) algorithm also yields a new multi-pass result for SetCover. See Section 4.4 for details.

1.2 Technical Summary and Comparisons

Technical summary. Our results are essentially streamable kernelization results, i.e., the algorithm "prunes" the input (in the case

[2] Note that d and r are dual parameters in the sense that if the input is $\{S_1, \ldots, S_m\}$ and we define $T_i = \{j : i \in S_j\}$ then $d = \max_j |S_j|$ and $r = \max_i |T_i|$.

[3] It suffices to count the number of edges M since there is always a cut whose size is between $M/2$ and M.

[4] Throughout we use $\tilde{O}$ to denote that logarithmic factors of m and n are being omitted.

[5] The lower bound result by Assadi [6] was for the case of Max-k-Cover but we will explain that it also applies in the case of Max-k-UniqueCover.

of Max-k-UniqueCover and Max-k-Cover this corresponds to ignoring some of the input sets) to produce a "kernel" in such a way that a) solving the problem optimally on the kernel yields a solution that is as good (or almost as good) as the optimal solution on the original input and b) the kernel is streamable and sufficiently smaller than the original input such that it is possible to find an optimal solution for the kernel in significantly less time than it would take to solve on the original input. In the field of fixed parameter tractability, the main requirement is that the kernel can be produced in polynomial time. In the growing body of work on streaming kernelization [17–19] the main requirement is that the kernel can then be constructed using small space in the data stream model. Our results fits in with this line of work and the analysis requires numerous combinatorial insights into the structure of the optimum solution for Max-k-UniqueCover and Max-k-Cover.

Our technical contributions can be outlined as follows.

- Results (A) and (B3) rely on various structural and combinatorial observations. At a high level, Result (A) uses the observation that each set of any Max-k-Cover or Max-k-UniqueCover solution intersects any maximal set of disjoint sets. The main technical step is to demonstrate that storing a small number of intersecting sets suffices to preserve the optimal solution.
- The $1+\epsilon$ and $2+\epsilon$ approximations for Max-k-Cover and Max-k-UniqueCover, i.e., results (B1) and (B3), are based on a very simple idea of first collecting the largest $O(rk/\epsilon)$ sets and then solving the problem optimally on these sets. This can be done in a space efficient manner using existing sketch for F_0 estimation in the case of Max-k-Cover and a new sketch we present the case of Max-k-UniqueCover. While the approach is simple, showing that it yields the required approximations requires some work and builds on a recent result by Manurangsi [52]. We also extend the algorithm to the model where sets can be inserted and deleted in a non-trivial way.

Comparison to related work. In the context of streaming algorithms, for the Max-k-Cover problem, McGregor and Vu [58] showed that any approximation better than $1/(1-1/e)$ requires $\Omega(m/k^2)$ space. For the more general problem of streaming submodular maximization subject to a cardinality constraint, Feldman et al. [32] very recently showed a stronger lower bound that any approximation better than 2 requires $\Omega(m)$ space. Our results provide a route to circumvent these bounds via parameterization on k, r, and d.

Result (B3) leads to a parameterized algorithm for streaming SetCover. This new algorithm uses $\tilde{O}(rk^2n^{\delta}+n)$ space which improves upon the algorithm by Har-Peled et al. [36] that uses $\tilde{O}(mn^{1/\delta}+n)$ space, where k is an upper bound for the size of the minimum set cover, in the case $rk^2 \ll m$. Both algorithms use $O(1/\delta)$ passes and yield an $O(1/\delta)$ approximation.

In the context of offline parameterized algorithms, Bonnet et al. [10] showed that Max-k-Cover is fixed-parameter tractable in terms of k and d. However, their branching-search algorithm is not streamable. Misra et al. [60] showed that the maximum unique coverage problem in which the aim is to maximize the number of uniquely covered elements u without any restriction on the number of sets in the solution is fixed-parameter tractable. This problem admits a kernel of size 4^u. On the other hand, they showed that the budgeted version of this problem (where each element has a profit and each set has a cost and the goal is maximize the profit subject to a budget constraint) is $W[1]$-hard when parameterized by the budget [6]. In this context, our result shows that a parameterization on both the maximum set size d and the budget k is possible (at least when all costs and profits are unit).

2 PRELIMINARIES

2.1 Notation and Parameters

Throughout the paper, m will denote the number of sets, n will denote the size of the universe, and k will denote the maximum number of sets that can be used in the solution. Given input sets $S_1, S_2, \ldots, S_m \subset [n]$, let

$$d = \max_i |S_i|$$

be the maximum set size and let

$$r = \max_j |\{i : j \in S_i\}|$$

be the maximum number of sets that contain the same element.

2.2 Structural Preliminaries

Given a collection of sets $C = \{S_1, S_2, \ldots, S_m\}$, we say a subcollection $C' \subset C$ is a *matching* if the sets in C' are mutually disjoint. C' is a maximal matching if there does not exist $S \in C \setminus C'$ such that S is disjoint from all sets in C'. The following simple lemma will be useful at various points in the paper.

Lemma 2.1. *For any input C, let $O \subset C$ be an optimal solution for either the* Max-k-Cover *or* Max-k-UniqueCover *problem. Let M_i be a maximal matching amongst the input set of size i. Then every set of size i in O of size intersects some set in M_i.*

Proof. Let $S \in O$ have size i. If it was disjoint from all sets in M_i then it could be added to M_i and the resulting collection would still be a matching. This violates the assumption that M_i is maximal. □

The next lemma extends the above result to show that we can potentially remove many sets from each M_i and still argue that there is an optimal solution for the original instance amongst the sets that intersect a set in some M_i.

Lemma 2.2. *Consider an input of sets of size at most d. For $i \in [d]$, let M_i be a maximal matching amongst the input set of size i and let M_i' be an arbitrary subset of M_i of size $\min(k+dk, |M_i|)$. Let D_i be the collection of all sets that intersect a set in M_i'. Then $\bigcup_i (D_i \cup M_i')$ contains an optimal solution to both the* Max-k-UniqueCover *and* Max-k-Cover *problem.*

Proof. If $|M_i| = |M_i'|$ for all $1 \le i \le d$ then the result follows from Lemma 2.1. Suppose that If not, let $j = \max\{i \in [d] : |M_i| > |M_i'|\}$. Let O be an optimal solution and let O_i be all the sets in O of size i. We know that every set in $O_d \cup O_{d-1} \cup \ldots \cup O_{j+1}$ is in

$$\bigcup_{i \ge j+1} (D_i \cup M_i') = \bigcup_{i \ge j+1} (D_i \cup M_i) .$$

[6] In the Max-k-UniqueCover problem that we consider, all costs and profits are one and the budget is k.

Hence, the number of elements (uniquely) covered by O is at most the number of elements (uniquely) covered by $O_d \cup O_{d-1} \cup \ldots \cup O_{j+1}$ plus kj since every set in $O_j \cup \ldots \cup O_1$ (uniquely) covers at most j additional elements. But we can (uniquely) cover at least the number of elements (uniquely) covered by $O_d \cup O_{d-1} \cup \ldots \cup O_{j+1}$ plus kj. This is because M_j contains $k + dk$ disjoint sets of size j and at least $k + dk - kd = k$ of these are disjoint from all sets in $O_d \cup O_{d-1} \cup \ldots \cup O_{j+1}$. Hence, there is a solution amongst $\bigcup_{i \geq j}(D_i \cup M_i')$ that is at least as good as O and hence is also optimal. □

2.3 Sketches and Subsampling

2.3.1 Coverage Sketch. Given a vector $x \in \mathbb{R}^n$, $F_0(x)$ is defined as the number of elements of x which are non-zero. If given a subset $S \subset \{1, \ldots, n\}$, we define $x_S \in \{0,1\}^n$ to be the characteristic vector of S (i.e., $x_i = 1$ iff $i \in S$) then given sets $S_1, S_2, \ldots$ note that $F_0(x_{S_1} + x_{S_2} + \ldots)$ is exactly the number of elements covered by $S_1 \cup S_2 \cup \ldots$. We will use the following result for estimating F_0.

Theorem 2.1 ([9, 21]). *There exists an $\tilde{O}(\epsilon^{-2} \log \delta^{-1})$-space algorithm that, given a set $S \subseteq [n]$, can construct a data structure $\mathcal{M}(S)$, called an F_0* sketch *of S, that has the property that the number of distinct elements in a collection of sets $S_1, S_2, \ldots, S_t$ can be approximated up to a $1 + \epsilon$ factor with probability at least $1 - \delta$ given the collection of F_0 sketches $\mathcal{M}(S_1), \mathcal{M}(S_2), \ldots, \mathcal{M}(S_t)$.*

Note that if we set $\delta \ll 1/(\text{poly}(m) \cdot \binom{t}{k})$ in the above result we can try collection of k sets amongst $S_1, S_2, \ldots, S_t$ and get a $1 + \epsilon$ approximation for the coverage of each collection with high probability.

2.3.2 Unique Coverage Sketch. For unique coverage, our sketch of a set corresponds to subsampling the universe via some hash function $h : [n] \to \{0,1\}$ where h is chosen randomly such that for each i, $\Pr[h(i) = 1] = p$ for some appropriate value p. Specifically, rather processing an input set S, we process $S' = \{i \in S : h(i) = 1\}$. Note that $|S'|$ has size $p|S|$ in expectation. This approach was use by McGregor and Vu [59] in the context of Max-k-Cover and extends easily to Max-k-UniqueCover; see Appendix A. The consequence is that if there is a streaming algorithm that finds a t approximation, we can turn that algorithm into a $t(1 + \epsilon)$ approximation algorithm in which we can assume that $\text{OPT} = O(\epsilon^{-2} k \log m)$ with high probability[7] by running the algorithm on a subsampled sets rather than the original sets. Note that this also allows us to assume input sets have size $O(\epsilon^{-2} k \log m)$ since $|S'| \leq \text{OPT}$. Hence each "sketched" set can be stored in $B = O(\epsilon^{-2} k \log m \log n)$ bits.

2.3.3 Algorithm with $\Omega(m)$ Memory. We will use the above sketches in a more interesting context later in the paper, note that they immediately imply a trivial algorithmic result. Consider the naive algorithm that stores every set and finds the best solution; note that this requires exponential time. We note that since we can assume $\text{OPT} = O(\epsilon^{-2} k \log m)$, each set has size at most $O(\epsilon^{-2} k \log m)$. Hence, we need $\tilde{O}(\epsilon^{-2} mk)$ memory to store all the sets. This approach was noted in [59] in the context of Max-k-Cover but also apples to Max-k-UniqueCover. We will later explain that for a $1 + \epsilon$ approximation, the above trivial algorithm is optimal up to polylogarithmic factors for constant k.

3 EXACT ALGORITHMS

Let C be the input sets. In this section we will initially assume all input sets have size exactly d and will show that there exists a single-pass data stream algorithm that uses $\tilde{O}(d^{d+1} k^d)$ space and returns a collection of sets $C' \subset C$ such that the optimal solution for either the maximum coverage or unique coverage problem when restricted to C' is equal to the optimal solution with no such restriction. We will subsequently generalize this to the case when sets can have any size at most d. In this section we will assume that r can be unbounded, e.g., an element in the universe could appear in m of the input sets.

3.1 Warm-Up Idea

Appealing to Lemma 2.1, we know that sets in an optimal solution to maximum coverage or maximum unique coverage intersect with a maximal matching. Hence, a natural approach is to construct a maximimal matching A greedily as the sets arrive along with any set that intersects a set in A. If the maximal matching ever exceeds size k then we have an optimal solution to Max-k-Cover and Max-k-UniqueCover that covers dk elements and hence we can ensure $|A| \leq k$. However, a set in A could intersect with $\Omega(m)$ other sets in the worst case[8] The main technical step in the algorithm in the next section is a way to carefully store only some of the sets that intersect A such that we can bound the number of stored sets in terms of k and d and yet still assume that stored sets include an optimal solution to either Max-k-Cover or Max-k-UniqueCover.

3.2 Algorithm

(1) Let A and X_u (for all $u \in [n]$) be empty sets. Each will correspond to a collection of sets. Let $b = d(k-1)$.
(2) Process the stream and let S be the next set:
  (a) If S is disjoint from all sets in A and $|A| < k$, add S to A.
  (b) If $u \in S \cap S'$ for some $S' \in A$:
    (i) Add S to X_u if there does not exist a subset $T \subset (S \setminus \{u\})$ that occurs as subset of $(b+1)^{d-1-|T|}$ sets in X_u.
(3) Return the best solution in $C' = A \cup (\bigcup_u X_u)$.

3.3 Analysis

We start with the following combinatorial lemma[9].

Lemma 3.1. *Let $X = \{S_1, S_2, \ldots\}$ be a collection of distinct sets where each $S_i \subset [n]$ and $|S_i| = a$. Suppose for all $T \subseteq [n]$ with $|T| \leq a$ there exists at most*

$$\ell_{|T|} := (b+1)^{a-|T|}$$

sets in X that contain T. Furthermore suppose there exists a set T^* such that this inequality is tight. Then, for all $B \subset [n]$ disjoint from T^* with $|B| \leq b$ there exists $S_i \in X$ such that $T^* \subset S_i$ and $|S_i \cap B| = 0$.

[7] Throughout this paper, we say an algorithm is correct with high probability if the probability of failure is inversely polynomial in m.

[8] It can be bounded in terms of d and r however. Specifically, each set can intersect with at most $d(r-1)$ other sets. However, in this section we are assuming r is unbounded so this bound does not help us here.

[9] For the interested reader who is familiar with the relevant combinatorial results, we note that we can prove a similar lemma to the one here via the Sunflower Lemma [4, 62]. In particular, one can argue that there exists a sufficiently large sunflower amongst $\{S \in X : T^* \text{ is a subset of } S\}$ whose core includes T^*. With some small adjustment to the subsequent theorem, this would be sufficient for our purposes. However, we instead include this version of the lemma because it is simpler and self-contained.

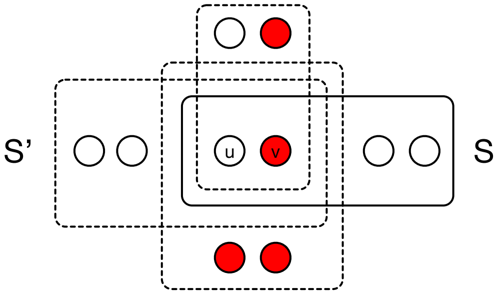

Figure 1: An example where all sets have size 4. Suppose the three dotted sets are currently stored in X_u. If S intersects u, it may not be added to X_u even if S is in an optimal solution O. In the above diagram, the elements covered by sets in $O \setminus \{S\}$ are shaded (note that the sets in O other than S are not drawn). In particular, if a subset T of $S \setminus \{u\}$ is a subset of many sets currently stored in X_u, it will not be added. For example, $T = \{v\}$ already occurs in the three subsets currently in X_u and, for the sake of a simple diagram, suppose 3 is the threshold for the maximum number of times a subset may appear in sets in X_u. Our analysis shows that there always exists a set S' in X_u that is "as good as" S in the sense that $S' \cap S = T \cup \{u\}$ and all the elements in $S' \setminus S$ are elements not covered by sets in $O \setminus \{S\}$.

Proof. If $|T^*| = a$ then $T^* \in X$ and this set satisfies the necessary conditions. Henceforth, assume $|T^*| < a$. Consider the $\ell_{|T^*|}$ sets in X that are supersets of T^*. Call this collection X'. For any $x \in B$, there are at most $\ell_{|T^*|+1}$ sets that include $T^* \cup \{x\}$. Since there are b choices for x, at most

$$b\ell_{|T^*|+1} = b(b+1)^{a-|T^*|-1} < (b+1)^{a-|T^*|} = \ell_{|T^*|}$$

sets in X' contains an element in B. Hence, at least one set in X does not contain any element in B. □

For any collection of sets F, let $f(F)$ be the maximum coverage of at most k sets in F and let $g(F)$ be the maximum unique coverage of at most k set in F.

Theorem 3.1. *The output of the algorithm satisfies $f(C') = f(C)$ and $g(C') = g(C)$.*

Proof. Let C_0 be the union of A and all sets that intersect a set in A, i.e.,

$$C_0 = \{S \in A\} \cup \{S \in C : |S \cap S'| > 0 \text{ for some } S' \in A\} .$$

Note that every set in the optimum solution of maximum coverage intersects with some set in A and hence $f(C_0) = f(C)$. For $i \geq 1$ consider,

$$C_i = C_0 \setminus \{\text{first } i \text{ sets in stream that are not in output } C'\} .$$

We will next argue that for any $i \geq 0$, $f(C_{i+1}) = f(C_i)$ and the theorem follows.

Let O be an optimum solution in C_i and let $\{S\} = C_i \setminus C_{i+1}$. If $S \notin O$ then clearly $f(C_{i+1}) = f(C_i)$ since $O \subseteq C_{i+1}$. If $S \in O$ but not in C_{i+1} then let $u \in S$ be the node for which we contemplated adding S to X_u but didn't because of the additional requirements.

Claim 1. *There exists S' in X_u such that $f((O \setminus \{S\}) \cup \{S'\}) = f(C_i)$ as required.*

Proof of Claim. If S was not added to X_u there exists a subset of $T^* \subset (S \setminus \{u\})$ that is a subset of $(b+1)^{d-1-|T^*|}$ sets in X_u. Let X be the collection of sets of size $a = d-1$ formed by removing u from each of the sets in X_u. Note that X satisfies the assumptions of Lemma 3.1. Let B be the set of at most $b = d(k-1)$ elements in the set

$$B = \{v : v \in S'' \text{ for some } S'' \in O\} \setminus S .$$

By Lemma 3.1, there exists a set S' in X such that $T^* \subset S'$ and $|(S' \setminus T^*) \cap B| = 0$. Hence, $f((O \setminus \{S\}) \cup \{S'\})$. □

The proof for unique coverage, i.e., $g()$, is identical. □

Lemma 3.2. *The space used by the algorithm is $\tilde{O}(d^{d+1}k^d)$.*

Proof. Recall that one of the requirements for a set S to be added to X_u is that the number of sets in X_u that are supersets of any subset of $S \setminus \{u\}$ of size t is at most $(b+1)^{d-1-t}$. This includes the empty subset and since every set in X_u is a superset of the empty set, we deduce that

$$|X_u| \leq (b+1)^{d-1-0} = (b+1)^{d-1} .$$

Since $|A| \leq k$, the number of sets that are stored is at most

$$\begin{aligned} |A| + \sum_{u \in \cup_{S \in A}} |X_u| &\leq |A| + d|A| \cdot (b+1)^{d-1} \\ &\leq |A| + d|A| \cdot O((dk)^{d-1}) \\ &= O((dk)^d) . \end{aligned}$$

□

3.4 Generalization to Sets of Different Size

In the case where sets may have any size at most d, we run the algorithm described in Section 3.2 in parallel for stream sets of each size $t \in [d]$. By appealing to Lemma 2.2, we know that an optimal maximum overage or maximum unique coverage intersects with the union of maximal matchings of sets of size t for each t. We again rely on Lemma 3.1 to establish that we can store only some of the sets that intersect these matchings and still retain an optimal solution to either coverage problem. We describe the algorithm below.

(1) Let A_t and $X_{u,t}$ (for all $u \in [n]$ and $t \in [d]$) be empty sets. Each will correspond to a collection of sets. Let $b = d(k-1)$.
(2) Process the stream and let S be the next set and let $t = |S|$:
  (a) If S is disjoint from all sets in A_t and
$$|A_t| < \begin{cases} dk + k & \text{if } t < d \\ k & \text{if } t = d \end{cases}$$
  then add S to A_t.
  (b) If $u \in S \cap S'$ for some $S' \in A_t$:
    (i) Add S to $X_{u,t}$ if there does not exist a subset $T \subset (S \setminus \{u\})$ that occurs as subset of $(b+1)^{t-1-|T|}$ sets in X_u.
(3) Return $C'' = \left(\bigcup_t A_t\right) \cup \left(\bigcup_{u,t} X_{u,t}\right)$.

Theorem 3.2. *The output of the algorithm satisfies $f(C'') = f(C)$ and $g(C'') = g(C)$.*

Proof. Let

$$C_0 = \bigcup_t \left(\{S \in A_t\} \cup \{S \in C : |S \cap S'| > 0 \text{ for some } S' \in A\}\right) .$$

Define C_i, O, and S as in the proof of Theorem 3.1. By Lemma 2.2, $f(C_0) = f(C)$ since there is an optimum solution of maximum coverage in which every set intersects with some set A_t. Let $u \in S$ be the node which prevented us from adding S to $X_{u,t}$. We now prove an analog of Claim 1 which implies that for any $i \geq 0, f(C_{i+1}) = f(C_i)$.

Claim 2. *There exists S' in $X_{u,t}$ such that $f((O \setminus \{S\}) \cup \{S'\}) = f(C_i)$ as required.*

Proof of Claim. If S was not added to $X_{u,t}$ there exists of a subset of $T^* \subset (S \setminus \{u\})$ that is a subset of $(b+1)^{t-1-|T^*|}$ sets in X_u. Let X be the collection of sets of size $a = t-1$ formed by removing u from each of the sets in X_u. X satisfies the assumptions of Lemma 3.1. Let B be the set of at most $b = d(k-1)$ elements in the set

$$B = \{v : v \in S'' \text{ for some } S'' \in O\} \setminus S .$$

By Lemma 3.1, there exists a set S' in X such that $T^* \subset S'$ and $|(S' \setminus T^*) \cap B| = 0$. Hence, $f((O \setminus \{S\}) \cup \{S'\})$ □

Again, the proof is identical for unique coverage. □

Lemma 3.3. *The space used by the algorithm is $\tilde{O}(d^{d+1}k^d)$.*

Proof. For all t, $|X_{u,t}| \leq (b+1)^{t-1}$. Since $|A_d| \leq k$ and $|A_t| = O(dk)$ for $t < k$, the number of sets stored is at most:

$$\sum_{t=1}^{d} \left(|A_t| + \sum_{u \in \bigcup_{S \in A_t}} |X_{u,t}| \right)$$
$$\leq O(d^2k + d^2k(1 + (b+1) + \ldots + (b+1)^{d-2}) + dk(b+1)^{d-1})$$
$$= O((dk)^d) .$$

□

We summarize the result as a theorem.

Theorem 3.3. *There exists a single-pass, $\tilde{O}(d^{d+1}k^d)$-space algorithm that yields an exact solution to* Max-k-Cover *and* Max-k-UniqueCover.

4 APPROXIMATION ALGORITHMS

In this section, we present a variety of different approximation algorithms where the space used by the algorithm is independent of d but, in some cases, may depend on r.

4.1 Unique Coverage: $2 + \epsilon$ Approximation

In this section, we present a $2 + \epsilon$ approximation for unique coverage. The algorithm is simple but the analysis is non-trivial. The algorithm stores the ηk largest sets where $\eta = \lceil r/\epsilon \rceil$ and finds the best unique coverage achievable by selecting at most k of these sets.

We will present an algorithm with a $1 + \epsilon$ approximation in the next subsection with the expense of an extra k/ϵ factor in the space use. However, the algorithm in this section is appealing in the sense that it is much simpler and can be extended to insertion-deletion streams. The analysis of this approach may also be of independent interest.

Let C' be the ηk sets of largest size. To find the best solution C'' amongst C', we use the unique coverage sketches presented in the Section 2. Note that to find the ηk largest sets we just store the sizes of sets sketched so far along with their unique coverage sketches. Finally, we return the best solution C'' using most k sets in C' based on the unique coverage sketches that we store. Recall that each unique coverage sketch requires $\tilde{O}(k/\epsilon^2)$ space. We have the following result.

Theorem 4.1. *There exists a randomized single-pass algorithm using $\tilde{O}(\epsilon^{-2}\eta k) = \tilde{O}(\epsilon^{-3}k^2 r)$ space algorithm that $2+\epsilon$ approximates* Max-k-UniqueCover.

Proof. Let the sizes of the ηk largest sets be (with arbitrarily tie-breaking) be $d_1 \geq d_2 \geq \ldots \geq d_{\eta k}$ and let

$$d^* := \frac{d_1 + \ldots + d_k}{k} \quad \text{and} \quad d' := \frac{d_{k+1} + \ldots + d_{\eta k}}{(\eta - 1)k} .$$

Let O be an optimal collection of sets for Max-k-UniqueCover. First, we observe that for each set $S \in O \setminus C'$, we have that $|S| \leq d_{\eta k} \leq d'$. Hence,

$$\text{OPT} \leq f(O \cap C') + \sum_{S \in O \setminus C'} |S| \leq h(C'') + kd' .$$

where $h()$ is a function of a collection of sets that returns the number of elements that are covered by exactly one of these sets. Thus, if $kd' < 0.5\,\text{OPT}$, then it is immediate that the number of elements uniquely covered by our solution is $h(C'') > 0.5\,\text{OPT}$.

Now we consider the case $kd' \geq 0.5\,\text{OPT}$. For the sake of analysis, consider randomly partitioning C' into a set C'_1 of size k and $C'_2 = C' \setminus C'_1$. Observe that

$$\text{E}\left[h(C'_1)\right]$$
$$= \sum_{S \in C'} \text{E}\left[\text{\# of elements uniquely covered by } S \text{ in } C'_1\right]$$
$$= \sum_{S \in C'} \sum_{u \in S} \Pr\left[S \in C'_1 \text{ and } u \text{ is uniquely covered in } C'_1\right]$$
$$\geq \sum_{S \in C'} \sum_{u \in S} \left(\Pr\left[S \in C'_1\right] - \sum_{S' \in C \setminus \{S\} : u \in S'} \Pr\left[S \in C'_1, S' \in C'_1\right] \right)$$
$$\geq \sum_{S \in C'} |S| \left(\epsilon/r - (r-1)(\epsilon/r)^2\right)$$
$$\geq \eta k \cdot d' \left(\epsilon/r - r(\epsilon/r)^2\right) \geq kd'\,(1-\epsilon) \geq (1-\epsilon)\,\text{OPT}/2 .$$

□

We note that it is possible to improve the result slightly in the case $r = 2$ by setting $\eta = \sqrt{2/\epsilon}$. This results in saving a factor of $O(\epsilon^{-1/2})$ in the space. See Appendix B for details.

Extension to Insert/Delete Streams. We now explain how the above approach can be extended to the case where sets may be inserted and deleted. In this setting, it is not immediately obvious how to select the largest ηk sets; the approach used when sets are only inserted does not extend. Note that in this model we can set m to be that maximum number of sets that have been inserted and not deleted at any prefix of the stream rather than the total number of sets inserted/deleted.

However, we can extend the result as follows. Suppose the sketch of a set for approximating maximum unique coverage requires B bits; recall from Section 2.3 that $B = k\epsilon^{-2}\,\mathrm{polylog}(n, m)$ suffices. We can encode such a sketch of a set S as an integer $i(S) \in [2^B]$. Suppose we know that exactly ηk sets have size at least some threshold t. We will remove this assumption shortly. Consider the vector $x \in [N]$ where $N = 2^B$ that is initially 0 and then is updated by a stream of set insertions/deletions as follows:

1. When S is inserted, if $|S| \geq t$, then $x_{i(S)} \leftarrow x_{i(S)} + 1$.
2. When S is deleted, if $|S| \geq t$, then $x_{i(S)} \leftarrow x_{i(S)} - 1$.

At the end of this process $x \in \{0, 1, \ldots, , m\}^{2^B}$, $\ell_1(x) = \eta k$, and reconstruct the sketches of largest ηk sets given x. Unfortunately, storing x explicitly in small space is not possible since, while we are promised that at the end of the stream $\ell_1(x) = \eta k$, during the stream it could be that x is an arbitrary binary string with m one's and this requires $\Omega(m)$ memory to store. To get around this, it is sufficient to maintain a linear sketch of x itself that support sparse recovery. For our purposes, the CountMin Sketch [22] is sufficient although other approaches are possible. The CountMin Sketch allows x to reconstructed probability $1 - \delta$ using a sketch of size

$$O(\log N + \eta k \log(\eta k/\delta) \log m) = O(\eta k \epsilon^{-2}\,\mathrm{polylog}(n, m)) .$$

To remove the assumption that we do not know t in advance, we consider values:

$$t_0, t_1, \ldots, t_{\lceil \log_{1+\epsilon} m \rceil} \text{ where } t_i = (1+\epsilon)^i .$$

We define vector $x^0, x^1, \ldots \in \{0, 1, \ldots, , m\}^{2^B}$ where x^i is only updated when a set of size $\leq t_i$ but $> t_{i-1}$ is inserted/deleted. Then there exists i such that $\leq \eta k$ sets have size $\leq t_{i-1}$ and the sketches of these sets can be reconstructed from $x^0, \ldots, x^{t_{i-1}}$. To ensure we have ηk sets, we may need some additional sketches corresponding to sets of size $> t_{i-1}$ and $\leq t_i$ but unfortunately there could be m such sets and we are only guaranteed recover of x^{t_i} when it is sparse. However, if this is indeed the case we can still recover enough entries of x^{t_1} by first subsampling the entries at the appropriate rate (we can guess sampling rate $1, 1/2, 1/2^2, \ldots 1/m$) in the standard way. Note that we can keep track of $\ell_1(x^i)$ exactly for each i using $O(\log m)$ space.

4.2 Unique Coverage: $1 + \epsilon$ Approximation

The approximation factor in the previous section can be improved to $1 + \epsilon$ at the expense of an extra factor of k/ϵ in the space. Recall in Section 3.1 that there exists an algorithm for solving Max-k-UniqueCover exactly by storing $O(kdr)$ sets, i.e., with $\tilde{O}(kd^2r)$ space. Combining this with the Subsampling Framework discussed in Section 2.3.1, we may assume $d \leq \mathrm{OPT} = O(\epsilon^{-2} k \log m)$. This immediately implies the following theorem.

Theorem 4.2. *There exists a randomized one-pass algorithm using $\tilde{O}(\epsilon^{-4}k^2r)$ space that finds a $1 + \epsilon$ approximation of* Max-k-UniqueCover.

Note that the same approach would work for Max-k-Cover but we present a better result in Section 4.4.

4.3 Unique Coverage: $O(\log \min(k, r))$ Approx.

We now present an algorithm whose space does not depend on r but the result comes at the cost of increasing the approximation factor to $O(\log(\min(k, r)))$. It also has the feature that the running time is polynomial in k in addition to being polynomial in m and n.

The basic idea is as follows: We consider an existing algorithm that first finds a 2 approximation for the Max-k-Cover problem. Let the corresponding solution be C'. The algorithm then finds the best solution of Max-k-UniqueCover among the sets in C'.

Let z^* be a guess such that $(1 - \epsilon)\,\mathrm{OPT}^* \leq z^* \leq \mathrm{OPT}^*$ where OPT^* is the value of the optimal Max-k-Cover.

1. Initialize $T = \varnothing$ which will store sets from the stream.
2. For each set S in the stream, if $|T| < k$ and
$$|(\cup_{A \in T} A) \cup S| - |\cup_{A \in T} A| \geq z^*/(2k) ,$$
then add S to T and store S in the memory.
3. Return the best solution Q (in terms of unique coverage) among the sets in T.

The following theorem captures the above algorithm.

Theorem 4.3. *There exists a randomized one-pass, $\tilde{O}(k^2)$-space, algorithm that with high probability finds a $O(\log \min(k, r))$ approximation of* Max-k-UniqueCover.

Proof. It has been shown in previous work [8, 59] that T is a $2 + \epsilon$ approximation of Max-k-Cover. Demaine et al. [25] proved that Q is an $O(\log \min(k, r))$ approximation of Max-k-UniqueCover. In fact, they presented a polynomial time algorithm to find Q from T such that the number of uniquely covered elements is at least

$$\Omega(1/\log k) \cdot |\cup_{A \in T} A| \geq \Omega(1/\log k) \cdot 1/2 \cdot \mathrm{OPT}^* \geq \Omega(1/\log k) \cdot \mathrm{OPT} .$$

We note that $\mathrm{OPT}^* \leq k\,\mathrm{OPT}$. Otherwise, one can find a set that covers more than OPT elements which is a contradiction.

The above algorithm needs to keep track of the elements being covered by T at all points during the stream. This requires $\tilde{O}(\mathrm{OPT}^*) = \tilde{O}(k\,\mathrm{OPT})$ space. Furthermore, storing the sets in T needs $\tilde{O}(k\,\mathrm{OPT})$ space. Finally, guessing z^* entails a $O(\epsilon^{-1} \log \mathrm{OPT})$ factor. Thus, the algorithm uses $\tilde{O}(\epsilon^{-1} k\,\mathrm{OPT})$ space which could be translated into another algorithm that uses $\tilde{O}(\epsilon^{-3}k^2)$ space after using the subsampling framework. For the purpose of the proving the claimed approximation factor we can set ϵ to a small constant. □

4.4 Maximum Coverage and Set Cover

In this section, we generalize the approach of Manurangsi [52] and combine that with F_0-sketch to obtain a $1 + \epsilon$ approximation using $\tilde{O}(\epsilon^{-3}k^2r)$ space for the maximum coverage problem.

Manurangsi [52] showed that for the maximum k-vertex cover problem, the $\Theta(k/\epsilon)$ vertices with highest degrees form a $1 + \epsilon$ approximation kernel. That is, there exist k vertices among those that cover $(1 - \epsilon)\,\mathrm{OPT}$ edges. We now consider a set system in which an element belongs to at most r sets (this can also be viewed as a hypergraph where each set corresponds to a vertex and each element corresponds to a hyperedge; we then want to find k vertices that touch as many hyperedges as possible).

We begin with the following lemma that generalizes the aforementioned result in [52]. We may assune that $m \gg Crk/\epsilon$ for some large constant C; otherwise, we can store all the sets.

Lemma 4.1. *Suppose* $m > \lceil rk/\epsilon \rceil$. *Let* K *be the collection of* $\lceil rk/\epsilon \rceil$ *sets with largest sizes (tie-broken arbitrarily). There exist* k *sets in* K *that cover* $(1-\epsilon)\,\mathrm{OPT}$ *elements.*

Proof. Let O denote the collection of k sets in some optimal solution. Let $O^{in} = O \cap K$ and $O^{out} = O \setminus K$. We consider a random subset $Z \subset K$ of size $|O^{out}|$. We will show that the sets in $Z \cup O^{in}$ cover $(1-\epsilon)\,\mathrm{OPT}$ elements in expectation; this implies the claim.

Let $\chi(Z)$ denote the set of elements covered by the sets in Z. Let $[\mathcal{E}]$ denote the indicator variable for event $\mathcal{E}$. We rewrite

$$|\chi(Z \cup O^{in})| = |\chi(O^{in})| + |\chi(Z)| - |\chi(O^{in}) \cap \chi(Z)| .$$

Furthermore, the probability that we pick a set S in K to add to Z is

$$p := \frac{|O^{out}|}{|K|} \leq \frac{k}{kr/\epsilon} = \frac{\epsilon}{r} .$$

Next, we upper bound $\mathrm{E}\left[|\chi(O^{in}) \cap \chi(Z)|\right]$. We have

$$\begin{aligned} \mathrm{E}\left[|\chi(O^{in}) \cap \chi(Z)|\right] &\leq \sum_{u \in \chi(O^{in})} \sum_{S \in K : u \in S} \Pr\left[S \in Z\right] \\ &\leq \sum_{u \in \chi(O^{in})} rp \leq |\chi(O^{in})| \cdot \epsilon . \end{aligned}$$

We lower bound $\mathrm{E}\left[|\chi(Z)|\right]$ as follows.

$$\begin{aligned} &\mathrm{E}\left[|\chi(Z)|\right] \\ &\geq \mathrm{E}\left[\sum_{S \in K}\left(|S|[S \in Z] - \sum_{S' \in K \setminus \{S\}} |S \cap S'|[S \in Z \wedge S' \in Z]\right)\right] \\ &\geq \sum_{S \in K}\left(|S|p - \sum_{S' \in K \setminus \{S\}} |S \cap S'|p^2\right) \\ &\geq \sum_{S \in K}\left(|S|p - r|S|p^2\right) \geq p(1-pr)\sum_{S \in K}|S| \geq p(1-\epsilon)\sum_{S \in K}|S| . \end{aligned}$$

In the above derivation, the second inequality follows from the observation that $\Pr\left[S \in Z \wedge S' \in Z\right] \leq p^2$. The third inequality is because $\sum_{S' \in K \setminus \{S\}} |S \cap S'| \leq r|S|$ since each element belongs to at most r sets.

For all $S \in K$, we must have have

$$|S| \geq \frac{\sum_{Y \in O^{out}} |Y|}{|O^{out}|} \geq \frac{|\chi(O^{out})|}{|O^{out}|} .$$

Thus,

$$\begin{aligned} \mathrm{E}\left[|\chi(Z)|\right] \geq p\,(1-\epsilon)\,|K|\frac{|\chi(O^{out})|}{|O^{out}|} &= p\,(1-\epsilon)\,\frac{|\chi(O^{out})|}{p} \\ &= (1-\epsilon)|\chi(O^{out})| . \end{aligned}$$

Putting it together,

$$\begin{aligned} \mathrm{E}\left[|\chi(Z \cup O^{in})|\right] &\geq |\chi(O^{in})| + (1-\epsilon)|\chi(O^{out})| - |\chi(O^{in})| \cdot \epsilon \\ &\geq (1-\epsilon)\,\mathrm{OPT} . \end{aligned}$$

□

With the above lemma in mind, the following algorithm's correctness is immediate.

(1) Store F_0-sketches of the kr/ϵ largest sets, where the failure probability of the sketches is set to $\frac{1}{\mathrm{poly}(n)\binom{m}{k}}$.
(2) At the end of the stream, return the k sets with the largest coverage based on the estimates given by the F_0-sketches.

We restate our result as a theorem.

Theorem 4.4. *There exists a randomized one-pass,* $\tilde{O}(k^2 r/\epsilon^3)$*-space, algorithm that with high probability finds a* $1+\epsilon$ *approximation to* Max-k-Cover.

Application to Parameterized Set Cover. We parameterize the set cover problem as follows. Given a set system, either A) output a set cover of size αk if $\mathrm{OPT} \leq k$ where α the approximation factor or B) correctly declare that a set cover of size k does not exist.

Theorem 4.5. *There exists a randomized,* $O(1/\delta)$*-pass,* $\tilde{O}(rk^2 n^{1/\delta} + n)$*-space, algorithm that with high probability finds a* $O(1/\delta)$ *approximation of the parameterized set cover problem.*

Proof. In each pass, we run the algorithm in Theorem 4.4 with parameters k and $\epsilon = 1/n^{\delta/3}$ on the remaining uncovered elements. The space use is $\tilde{O}(rk^2 n^{1/\delta} + n)$. Here, we need additional $\tilde{O}(n)$ space to keep track of the remaining uncovered elements.

Note that if $\mathrm{OPT} \leq k$, after each pass, the number of uncovered elements is reduced by a factor $1/n^{\delta/3}$. This is because if n' is the number of uncovered elements at the beginning of a pass, then after that pass, we cover all but at most $n'/n^{\delta/3}$ of those elements. After i passes, the number of remaining uncovered elements is $O(n^{1-i\delta/3})$; we therefore use at most $O(1/\delta)$ passes until we are done. At the end, we have a set cover of size $O(k/\delta)$.

If after $\omega(1/\delta)$ passes, there are still remaining uncovered elements, we declare that such a solution does not exist. □

Our algorithm improves upon the algorithm by Har-Peled et al. [36] that uses $\tilde{O}(mn^{1/\delta} + n)$ space for when $rk^2 \ll m$ and also yields an $O(1/\delta)$ approximation.

Extension to Insert/Delete Streams. The result can be extended to the case where sets are inserted and deleted using the same approach as that used for unique coverage.

5 LOWER BOUNDS

5.1 Lower Bounds for Exact Solutions

As observed earlier, any exact algorithm for either the Max-k-Cover or Max-k-UniqueCover problem on an input where all sets have size d will return a matching of size k if one exists. However, by a lowerbound due to Chitnis et al. [18] we know that determining if there exists a matching of size k in a single pass requires $\Omega(k^d)$ space. This immediately implies the following theorem.

Theorem 5.1. *Any single-pass algorithm that solves* Max-k-Cover *or* Max-k-UniqueCover *exactly with probability at least* $9/10$ *requires* $\Omega(k^d)$ *space.*

5.2 Lower bound for a $e^{1-1/k}$ approximation

The strategy is similar to previous work on Max-k-Cover [58, 59]. However, we need to argue that the relevant probabilistic construction works for all collections of fewer than k sets since the unique coverage function is not monotone. This extra argument will also allow us to show that the lower bound also applies to bi-criteria approximation in which we are allowed to pick more than k sets (this is not the case for Max-k-Cover).

We make a reduction from the communication problem k-player set disjointness, denoted by $\mathsf{DISJ}(m, k)$. In this problem, there are k players where the ith player has a set $S_i \subseteq [m]$. It is promised that exactly one of the following two cases happens a) NO instance: All the sets are pairwise disjoint and b) YES instance: There is a unique element $v \in [m]$ such that $v \in S_i$ for all $i \in [k]$ and all other elements belong to at most one set. The (randomized) communication complexity, for some large enough constant success probability, of the above problem in p-round, one-way model is $\Omega(m/(pk))$ even if the players may use public randomness [14]. We can assume that $|S_1 \cup S_2 \cup \ldots \cup S_k| \geq m/4$ via a padding argument.

Theorem 5.2. *Any constant-pass randomized algorithm with an approximation better than $e^{1-1/k}$ for* Max-k-UniqueCover *requires $\Omega(m/k^2)$ space.*

Proof. Consider a sufficiently large n where k divides n. For each $i \in [m]$, let $\mathcal{P}_i$ be a random partition of $[n]$ into k sets $V_1^i, \ldots, V_k^i$ such that an element in the universe $U = [n]$ belongs to exactly one of these sets uniformly at random. In particular, for all $i \in [m]$ and $v \in U$,

$$\Pr\left[v \in V_j^i \wedge (\forall j' \neq j, v \notin V_{j'}^i)\right] = 1/k \ .$$

The partitions are chosen independently using public randomness before receiving the input. For each player j, if $i \in S_j$, then they put V_j^i in the stream. Note that the stream consists of $\Theta(m)$ sets.

If the input is a NO instance, then for each $i \in [m]$, there is at most one set V_j^i in the stream. Therefore, for each element $v \in [n]$ and any collection of $\ell \leq k$ sets $V_{j_1}^{i_1}, \ldots, V_{j_\ell}^{i_\ell}$ in the stream,

$$\begin{aligned}\Pr\left[v \text{ is uniquely covered by } V_{j_1}^{i_1}, \ldots, V_{j_\ell}^{i_\ell}\right] &= \ell/k \cdot (1 - 1/k)^{\ell-1} \\ &\leq \ell/k \cdot e^{-(\ell-1)/k} \ .\end{aligned}$$

Therefore, in expectation, $\mu_\ell := \mathrm{E}\left[h(\{V_{j_1}^{i_1}, \ldots, V_{j_\ell}^{i_\ell}\})\right] \leq \ell/k \cdot e^{-(\ell-1)/k} n$ where $h()$ is the number of elements that are uniquely covered. By an application of Hoeffding's inequality,

$$\begin{aligned}&\Pr\left[h(\{V_{j_1}^{i_1} \cup \ldots \cup V_{j_\ell}^{i_\ell}\}) > \mu_\ell + \epsilon e^{-(k-1)/k} \cdot n\right] \\ \leq &\exp\left(-2\epsilon^2 e^{-2(\ell-1)/k} n\right) \\ \leq &\exp\left(-\Omega(\epsilon^2 n)\right) \leq \frac{1}{m^{10k}} \ .\end{aligned}$$

The last inequality follows by letting $n = \Omega(\epsilon^{-2} k \log m)$. The following claim shows that for large k, in expectation, picking k sets is optimal in terms of unique coverage.

Lemma 5.1. *The function $g(\ell) = \ell/k \cdot e^{-(\ell-1)/k} n$ is increasing in the interval $(-\infty, k]$ and decreasing in the interval $[k, +\infty)$.*

Proof. We take the partial derivative of g with respect to ℓ

$$\frac{\partial g}{\partial \ell} = \frac{e^{(1-\ell)/k}(k - \ell)}{k^2} \cdot n$$

and observe that it is non-negative if and only if $\ell \leq k$. □

By appealing to the union bound over all $\binom{m}{1} + \ldots + \binom{m}{k-1} + \binom{m}{k} \leq O(m^{k+1})$ possible collections $\ell \leq k$ sets, we deduce that with high probability, for all collections of $\ell \leq k$ sets $S_1, \ldots, S_\ell$,

$$\begin{aligned}h(\{S_1, \ldots, S_\ell\}) &\leq \mu_\ell + \epsilon e^{-(k-1)/k} \cdot n \\ &\leq \ell/k \cdot e^{-(\ell-1)/k} n + \epsilon e^{-(k-1)/k} \cdot n \\ &\leq (1 + \epsilon) e^{-1+1/k} n \ .\end{aligned}$$

If the input is a YES instance, then clearly, the maximum k-unique coverage is n. This is because there exists i such that $i \in S_1 \cap \ldots \cap S_k$ and therefore $V_1^i, \ldots, V_k^i$ are in the stream and these sets uniquely cover all elements.

Therefore, any constant pass algorithm that finds a $(1+2\epsilon)e^{1-1/k}$ approximation of Max-k-UniqueCover for some large enough constant success probability implies a protocol to solve $\mathsf{DISJ}(m, k)$. Thus, $\Omega(m/k^2)$ space is required. □

Remark. Since $g(\ell)$ is decreasing in the interval $[k, m]$, the lower bound also holds for bi-criteria approximation where the algorithm is allows to pick more than k sets.

5.3 Lower bound for $1 + \epsilon$ approximation

Assadi [6] presents a $O\left(m/\epsilon^2\right)$ lower bound for the space required to compute a $1 + \epsilon$ approximation for Max-k-Cover when $k = 2$, even when the stream is in a random order and is allowed constant passes. This is accomplished via a reduction to multiple instances of the Gap-Hamming Distance problem on a hard input distribution, where an input with high maximum coverage corresponds to a YES answer for some Gap-Hamming Distance instance, and a low maximum coverage corresponds to a NO answer for all GHD instances. This hard distribution has the additional property that high maximum coverage inputs also have high maximum unique coverage, and low maximum coverage inputs have low maximum unique coverage. Therefore, the following corollary holds:

Corollary 5.1. *Any constant-pass randomized algorithm with an approximation factor $1+\epsilon$ for* Max-k-UniqueCover *requires $\Omega(m/\epsilon^2)$ space.*

ACKNOWLEDGMENTS

This work was partially supported by NSF grants CCF-1934846, CCF-1908849, CCF-1907738 and CCF-1637536.

REFERENCES

[1] A. A. Ageev and M. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *J. Comb. Optim.*, 8(3):307–328, 2004.

[2] K. J. Ahn and S. Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. *Inf. Comput.*, 222:59–79, 2013.

[3] N. Alaluf, A. Ene, M. Feldman, H. L. Nguyen, and A. Suh. Optimal streaming algorithms for submodular maximization with cardinality constraints. In *ICALP*, volume 168 of *LIPIcs*, pages 6:1–6:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[4] R. Alweiss, S. Lovett, K. Wu, and J. Zhang. Improved bounds for the sunflower lemma. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 624–630, 2020.

[5] A. Anagnostopoulos, L. Becchetti, I. Bordino, S. Leonardi, I. Mele, and P. Sankowski. Stochastic query covering for fast approximate document retrieval. *ACM Trans. Inf. Syst.*, 33(3):11:1–11:35, 2015.

[6] S. Assadi. Tight space-approximation tradeoff for the multi-pass streaming set cover problem. In *PODS*, pages 321–335. ACM, 2017.

[7] S. Assadi, S. Khanna, and Y. Li. Tight bounds for single-pass streaming complexity of the set cover problem. In *STOC*, pages 698–711. ACM, 2016.

[8] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: massive data summarization on the fly. In *KDD*, pages 671–680. ACM, 2014.

[9] Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan. Counting distinct elements in a data stream. In *RANDOM*, volume 2483 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2002.

[10] É. Bonnet, V. T. Paschos, and F. Sikora. Parameterized exact and approximation algorithms for maximum k-set cover and related satisfiability problems. *RAIRO Theor. Informatics Appl.*, 50(3):227–240, 2016.

[11] V. Braverman, R. Ostrovsky, and D. Vilenchik. How hard is counting triangles in the streaming model? In *ICALP (1)*, volume 7965 of *Lecture Notes in Computer Science*, pages 244–254. Springer, 2013.

[12] M. Bury and C. Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 263–274, 2015.

[13] A. Chakrabarti and S. Kale. Submodular maximization meets streaming: matchings, matroids, and more. *Math. Program.*, 154(1-2):225–247, 2015.

[14] A. Chakrabarti, S. Khot, and X. Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *IEEE Conference on Computational Complexity*, pages 107–117. IEEE Computer Society, 2003.

[15] A. Chakrabarti and A. Wirth. Incidence geometries and the pass complexity of semi-streaming set cover. In *SODA*, pages 1365–1373. SIAM, 2016.

[16] C. Chekuri, S. Gupta, and K. Quanrud. Streaming algorithms for submodular function maximization. In *ICALP (1)*, volume 9134 of *Lecture Notes in Computer Science*, pages 318–330. Springer, 2015.

[17] R. Chitnis and G. Cormode. Towards a theory of parameterized streaming algorithms. In *14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany*, pages 7:1–7:15, 2019.

[18] R. Chitnis, G. Cormode, H. Esfandiari, M. Hajiaghayi, A. McGregor, M. Monemizadeh, and S. Vorotnikova. Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In *SODA*, pages 1326–1344. SIAM, 2016.

[19] R. H. Chitnis, G. Cormode, H. Esfandiari, M. Hajiaghayi, and M. Monemizadeh. Brief announcement: New streaming algorithms for parameterized maximal matching & beyond. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 56–58, 2015.

[20] R. H. Chitnis, G. Cormode, M. T. Hajiaghayi, and M. Monemizadeh. Parameterized streaming: Maximal matching and vertex cover. In *SODA*, pages 1234–1251. SIAM, 2015.

[21] G. Cormode, M. Datar, P. Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *IEEE Trans. Knowl. Data Eng.*, 15(3):529–540, 2003.

[22] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005.

[23] M. Crouch and D. S. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, pages 96–104, 2014.

[24] M. S. Crouch, A. McGregor, and D. Stubbs. Dynamic graphs in the sliding-window model. In *Algorithms - ESA 2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, pages 337–348, 2013.

[25] E. D. Demaine, U. Feige, M. Hajiaghayi, and M. R. Salavatipour. Combination can be hard: Approximability of the unique coverage problem. *SIAM J. Comput.*, 38(4):1464–1483, 2008.

[26] M. Dom, J. Guo, R. Niedermeier, and S. Wernicke. Minimum membership set covering and the consecutive ones property. In *SWAT*, volume 4059 of *Lecture Notes in Computer Science*, pages 339–350. Springer, 2006.

[27] Y. Emek and A. Rosén. Semi-streaming set cover. *ACM Trans. Algorithms*, 13(1):6:1–6:22, 2016.

[28] L. Epstein, A. Levin, J. Mestre, and D. Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM J. Discrete Math.*, 25(3):1251–1265, 2011.

[29] T. Erlebach and E. J. van Leeuwen. Approximating geometric coverage problems. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 1267–1276, 2008.

[30] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[31] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2):207–216, 2005.

[32] M. Feldman, A. Norouzi-Fard, O. Svensson, and R. Zenklusen. The one-way communication complexity of submodular maximization with applications to streaming and robustness. In *STOC*, pages 1363–1374. ACM, 2020.

[33] D. R. Gaur, R. Krishnamurti, and R. Kohli. Erratum to: The capacitated max k-cut problem. *Math. Program.*, 126(1):191, 2011.

[34] A. Goel, M. Kapralov, and S. Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 468–485, 2012.

[35] V. Guruswami and K. Onak. Superlinear lower bounds for multipass graph processing. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, Palo Alto, California, USA, 5-7 June, 2013*, pages 287–298, 2013.

[36] S. Har-Peled, P. Indyk, S. Mahabadi, and A. Vakilian. Towards tight bounds for the streaming set cover problem. In *PODS*, pages 371–383. ACM, 2016.

[37] C. Huang, N. Kakimura, and Y. Yoshida. Streaming algorithms for maximizing monotone submodular functions under a knapsack constraint. In *APPROX-RANDOM*, volume 81 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[38] P. Indyk, S. Mahabadi, R. Rubinfeld, J. Ullman, A. Vakilian, and A. Yodpinyanee. Fractional set cover in the streaming model. In *APPROX-RANDOM*, volume 81 of *LIPIcs*, pages 12:1–12:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[39] P. Indyk and A. Vakilian. Tight trade-offs for the maximum k-coverage problem in the general streaming model. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, pages 200–217, 2019.

[40] T. Ito, S. Nakano, Y. Okamoto, Y. Otachi, R. Uehara, T. Uno, and Y. Uno. A 4.31-approximation for the geometric unique coverage problem on unit disks. *Theor. Comput. Sci.*, 544:14–31, 2014.

[41] J. Kallaugher, A. McGregor, E. Price, and S. Vorotnikova. The complexity of counting cycles in the adjacency list streaming model. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, pages 119–133, 2019.

[42] M. Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1679–1697, 2013.

[43] M. Kapralov, S. Khanna, and M. Sudan. Approximating matching size from random streams. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 734–751, 2014.

[44] M. Kapralov, S. Khanna, and M. Sudan. Streaming lower bounds for approximating MAX-CUT. In *SODA*, pages 1263–1282. SIAM, 2015.

[45] M. Kapralov, S. Khanna, M. Sudan, and A. Velingker. $(1 + \omega(1))$-approximation to MAX-CUT requires linear space. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1703–1722, 2017.

[46] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.

[47] C. Konrad. Maximum matching in turnstile streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 840–852, 2015.

[48] C. Konrad, F. Magniez, and C. Mathieu. Maximum matching in semi-streaming with few passes. In *APPROX-RANDOM*, volume 7408 of *Lecture Notes in Computer Science*, pages 231–242. Springer, 2012.

[49] C. Konrad and A. Rosén. Approximating semi-matchings in streaming and in two-party communication. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 637–649, 2013.

[50] A. Krause and C. Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, pages 1650–1654. AAAI Press, 2007.

[51] F. Kuhn, P. von Rickenbach, R. Wattenhofer, E. Welzl, and A. Zollinger. Interference in cellular networks: The minimum membership set cover problem. In *COCOON*, volume 3595 of *Lecture Notes in Computer Science*, pages 188–198. Springer, 2005.

[52] P. Manurangsi. A note on max k-vertex cover: Faster fpt-as, smaller approximate kernel and improved approximation. In *2nd Symposium on Simplicity in Algorithms, SOSA@SODA 2019, January 8-9, 2019 - San Diego, CA, USA*, pages 15:1–15:21, 2019.

[53] A. McGregor. Finding graph matchings in data streams. *APPROX-RANDOM*, pages 170–181, 2005.

[54] A. McGregor. Graph stream algorithms: a survey. *SIGMOD Record*, 43(1):9–20, 2014.

[55] A. McGregor and S. Vorotnikova. Planar matching in streams revisited. In *APPROX-RANDOM*, volume 60 of *LIPIcs*, pages 17:1–17:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[56] A. McGregor and S. Vorotnikova. Triangle and four cycle counting in the data stream model. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 445–456, 2020.

[57] A. McGregor, S. Vorotnikova, and H. T. Vu. Better algorithms for counting triangles in data streams. In *PODS*, pages 401–411. ACM, 2016.
[58] A. McGregor and H. T. Vu. Better streaming algorithms for the maximum coverage problem. In *ICDT*, volume 68 of *LIPIcs*, pages 22:1–22:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
[59] A. McGregor and H. T. Vu. Better streaming algorithms for the maximum coverage problem. *Theory of Computing Systems*, pages 1–25, 2018.
[60] N. Misra, H. Moser, V. Raman, S. Saurabh, and S. Sikdar. The parameterized complexity of unique coverage and its variants. *Algorithmica*, 65(3):517–544, 2013.
[61] A. Norouzi-Fard, J. Tarnawski, S. Mitrovic, A. Zandieh, A. Mousavifar, and O. Svensson. Beyond 1/2-approximation for submodular maximization on massive data streams. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 3826–3835. PMLR, 2018.
[62] A. Rao. Coding for sunflowers. *CoRR*, abs/1909.04774, 2019.
[63] B. Saha and L. Getoor. On maximum coverage in the streaming model & application to multi-topic blog-watch. In *SDM*, pages 697–708. SIAM, 2009.
[64] J. P. Schmidt, A. Siegel, and A. Srinivasan. Chernoff-hoeffding bounds for applications with limited independence. *SIAM J. Discrete Math.*, 8(2):223–250, 1995.
[65] M. Zelke. Weighted matching in the semi-streaming model. *Algorithmica*, 62(1-2):1–20, 2012.

A THE SUBSAMPLING FRAMEWORK

Assuming we have v such that $\mathrm{OPT}/2 \leq v \leq \mathrm{OPT}$. Let $h : [n] \to \{0,1\}$ be a hash function that is $\Omega(\epsilon^{-2} k \log m)$-wise independent. We run our algorithm on the subsampled universe $U' = \{u \in U : h(u) = 1\}$. Furthermore, let

$$\Pr\left[h(u) = 1\right] = p = \frac{ck \log m}{\epsilon^2 v}$$

where c is some sufficiently large constant. Let $S' = S \cap U'$ and let OPT' be the optimal unique coverage value in the subsampled set system. The following result is from our previous work [59]. We note that the proof is the same except that the indicator variables now correspond to the events that an element being uniquely covered (instead of being covered).

Lemma A.1. *With probability at least* $1 - 1/\mathrm{poly}(m)$*, we have that*

$$p\,\mathrm{OPT}(1+\epsilon) \geq \mathrm{OPT}' \geq p\,\mathrm{OPT}(1-\epsilon)$$

Furthermore, if $S_1, \ldots, S_k$ *satisfies* $UC(\{S_1', \ldots, S_k'\}) \geq p\,\mathrm{OPT}(1-\epsilon)/t$ *then*

$$UC(\{S_1, \ldots, S_k\}) \geq \mathrm{OPT}(1/t - 2\epsilon)\ .$$

We could guess $v = 1, 2, 4, \ldots, n$. One of the guesses must be between $\mathrm{OPT}/2$ and OPT which means $\mathrm{OPT}' = O(\epsilon^{-2} k \log m)$. Furthermore, if we find a $1/t$ approximation on the subsampled universe, then that corresponds to a $1/t - 2\epsilon$ approximation in the original universe. We note that as long as $v \leq \mathrm{OPT}$ and h is $\Omega(\epsilon^{-2} k \log m)$-wise independent, we have (see [64], Theorem 5):

$$\begin{aligned} &\Pr\left[UC(\{S_1', \ldots, S_\ell'\}) = p \cdot UC(\{S_1, \ldots, S_\ell\}) \pm \epsilon p\,\mathrm{OPT}\right] \\ &\geq 1 - \exp\left(-\Omega(k \log m)\right) \\ &\geq 1 - 1/m^{\Omega(k)}\ . \end{aligned}$$

This gives us Lemma A.1 even for when $v < \mathrm{OPT}/2$. However, if $v \leq \mathrm{OPT}/2$, then OPT' may be larger than $O(\epsilon^{-2} k \log m)$, and we may use too much memory. To this end, we simply terminate those instantiations. Among the instantiations that are not terminated, we return the solution given by the smallest guess.

B IMPROVING THEOREM 4.1 WHEN $r = 2$

Specifically, in the proof of suffices to set $\eta = \sqrt{2/\epsilon}$ change the bound of for $\mathrm{E}\left[h(C_1')\right]$ in the proof of Theorem 4.1 as follows:

$$\begin{aligned} \mathrm{E}\left[h(C_1')\right] &= \sum_{S \in C'} \mathrm{E}\left[\#\text{ of elements uniquely covered by } S \text{ in } C_1'\right] \\ &\geq \sum_{S \in C'} \sum_{i \in S} \frac{k}{\eta k} \frac{(\eta - 1)k}{\eta k - 1} \\ &= \frac{1}{\eta} \cdot \left(\frac{(\eta - 1)k}{\eta k - 1}\right) \sum_{S \in C'} |S| \\ &> \frac{\eta - 1}{\eta^2} \left(kd^* + (\eta - 1)kd'\right) \\ &\geq \mathrm{OPT}(1/\eta - 1/\eta^2 + (1 - 1/\eta)^2 0.5) \\ &= \mathrm{OPT}(0.5 - 0.5/\eta^2) \\ &= (0.5 - \epsilon)\,\mathrm{OPT}\ . \end{aligned}$$

The space used in resulting algorithm scales with $\epsilon^{-2.5}$ rather than ϵ^{-3} as implied by the analysis for general r.