**Redesigning Mass Transit Systems to better
integrate with Mobility-on-Demand Systems or
Redesigning Mass Transit Systems to better
integrate with Mobility-on-Demand Systems via
The Batched Set Cover Problem**

Center for Transportation, Environment, and Community Health
Final Report

by

Juan C. Martínez Mori, Samitha Samaranayake

October 5, 2018

DISCLAIMER

1. Report No.	2.Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle	5. Report Date
Redesigning Mass Transit Systems to better integrate with Mobility-onDemand Systems *or* Redesigning Mass Transit Systems to better integrate with Mobility-on-Demand Systems via The Batched Set Cover Problem	October 5, 2018
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Juan C. Martínez Mori, Samitha Samaranayake	

9. Performing Organization Name and Address	10. Work Unit No.
School of Civil and Environmental Engineering Cornell University Ithaca, NY 14853	
	11. Contract or Grant No.
	69A3551747119

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
U.S. Department of Transportation 1200 New Jersey Avenue, SE Washington, DC 20590	Final Report 11/30/2016 – 8/31/2018
	14. Sponsoring Agency Code
	US-DOT

15. Supplementary Notes

16. Abstract

We introduce the batched set cover problem, which is a generalization of the online set cover problem. In this problem, the elements of the ground set that need to be covered arrive in batches. Our main technical contribution is a tight $\Omega(H_{\$\%\&'()})$ lower bound on the competitive ratio of any algorithm given an adversary that is required to produce batches of VC-dimension at least z, for some $z \in \mathbb{N}$. This restriction on the adversary is motivated by the fact that, in some real world applications, decisions are made after collecting batches of datßßa of high VCdimension. In particular, ridesharing systems rely on the batch assignment of trip requests to vehicles, and some related problems such as that of optimal congregation points for passenger pickups and dropoffs can be modeled as a batched set cover problem with VC-dimension typically greater than one. We note that while any online algorithm may be used to solve the batched set cover problem by artificially sequencing the elements in a batch, this procedure may neglect the rich information encoded in the complex interactions between the elements of a batch and the sets that contain them. Therefore, we propose a minor modification to an online algorithm found in [8] to obtain an algorithm that attempts to exploit such information. Unfortunately, we are unable to improve its analysis in a way that reflects this intuition. However, we present computational experiments that provide empirical evidence of a constant factor improvement in the competitive ratio. To the best of our knowledge, we are the first to use the VCdimension in the context of online (batched) covering problems.

17. Key Words	18. Distribution Statement
set cover, batched, online, primal-dual, VC-dimension, ridesharing	Public Access, arXiv, and a peer-reviewed publication to be submitted based on the content on which this report is developed.

19. Security Classif (of this report)	20. Security Classif. (of this page)	21. No of Pages	22. Price
Unclassified	Unclassified		

Form DOT F 1700.7 (8-69)

The Batched Set Cover Problem

Juan C. Martinez Mori[1] and Samitha Samaranayake[2]

[1]Systems, Cornell University
jm2638@cornell.edu
[2]School of Civil and Environmental Engineering, Cornell University
samitha@cornell.edu

Abstract

We introduce the *batched set cover problem*, which is a generalization of the online set cover problem. In this problem, the elements of the ground set that need to be covered arrive in batches. Our main technical contribution is a tight $\Omega(H_{m-2^z+1})$ lower bound on the competitive ratio of any fractional batched algorithm given an adversary that is required to produce batches of VC-dimension at least z, for some $z \in \mathbb{N}^0$. This restriction on the adversary is motivated by the fact that, in some real world applications, decisions are made after collecting batches of data of non-trivial VC-dimension. In particular, ridesharing systems rely on the batch assignment of trip requests to vehicles, and some related problems such as that of optimal congregation points for passenger pickups and dropoffs can be modeled as a batched set cover problem with VC-dimension greater than or equal to two. Furthermore, we note that while any online algorithm may be used to solve the batched set cover problem by artificially sequencing the elements in a batch, this procedure may neglect the rich information encoded in the complex interactions between the elements of a batch and the sets that contain them. Therefore, we propose a minor modification to an online algorithm found in [8] to obtain an algorithm that attempts to exploit such information. Unfortunately, we are unable to improve its analysis in a way that reflects this intuition. However, we present computational experiments that provide empirical evidence of a constant factor improvement in the competitive ratio. To the best of our knowledge, we are the first to use the VC-dimension in the context of online (batched) covering problems.

Keywords: set cover, batched, online, primal-dual, VC-dimension, ridesharing

1. Introduction

1.1. Background

Let $X = \{1, \cdots, n\}$ be a ground set of n elements and $S = \{S_1, \cdots, S_m\}$ be a collection of m subsets of X. A *set cover* is a sub-collection of S such that its union is X. The set cover problem is to find a minimum cardinality set cover of X. It is a classic NP-har5d problem that is also hard to approximate to within a factor of $(1 - \alpha) \ln n$ in polynomial time for any $\alpha > 0$[11,9].

In the online setting [2, 7, 8], the members of S are identified a priori, but the elements of the ground set that need to be covered, along with their respective set membership, are revealed sequentially. More precisely, the *online set cover problem* consists of a game between an algorithm and an oblivious adversary;

one which knows the algorithm but not the realization of any random choices[1]. The adversary produces, in advance, a sequence $\sigma = \sigma_1, \sigma_2, \cdots$ of elements of X, which it reveals to the algorithm one at a time. Upon the arrival of an element, the algorithm must either conclude that the element is already in the set cover, or irrevocably extend the set cover with a member of S containing the element.

Alon et al. [2] gave a deterministic $O(\log m \log n)$-competitive algorithm for the online set cover problem and a nearly matching lower bound for any online algorithm. Buchbinder and Naor [7, 8] later proposed a general scheme for the design and analysis of online algorithms, namely the primal-dual method[2] and used it to obtain new algorithms for the online set cover problem. Their algorithms generally consist of two phases: *i*) a deterministic $O(\log m)$ primal-dual subroutine for the fractional online set cover problem, which is optimal up to constant terms, and *ii*) a randomized rounding procedure whose expected cost is $O(\log n)$ times the cost of the fractional solution, ultimately producing randomized $O(\log m \log n)$-competitive algorithms. The rounding procedure can be derandomized, producing deterministic $O(\log m \log n)$-competitive algorithms.

1.2. Contributions

Herein we introduce the *batched set cover problem*, which is a generalization of the online set cover problem. However, as in [7, 8], our focus is on its fractional counterpart; this corresponds to phase *i*) of the primal-dual scheme. We immediately recover the integral case through the rounding procedures in phase *ii*), which we leave untouched. In essence, the batched set cover problem differs from the online set cover problem in that the adversary produces a sequence of batches of elements of X. Thus, the online set cover problem is a special case of the batched set cover problem in which each element revealed by the adversary is its own batch.

Note that the problem we consider is distinct from the capacitated online set cover problem with set requests treated by Bhawalkar et al. [5]. They argue that the uncapacitated problem is not meaningful because the elements in a batch can be thought of as arriving sequentially, whereas we argue that this is not always the case. Our main technical contribution is a tight lower bound on the competitive ratio of any fractional batched algorithm given a parametrized restriction on the adversary. Specifically, if we consider adversaries that are required to produce batches of Vapnik Chervonenkis (VC)-dimension [12] at least z, for some $z \in \mathbb{N}^0$, any fractional batched algorithm is $\Omega(H_{m-2^z+1})$-competitive. For $z > 0$, this bound is more generous (to the algorithm) than the $\Omega(H_m)$ bound of the online setting [7, 8] which we recover when $z = 0$.

In addition, we propose a minor modification to an online algorithm found in [8] to obtain a dedicated batched algorithm. The main idea is the simultaneous update of the dual variables that correspond to unsatisfied primal constraints, which is reminiscent of a primal-dual algorithm in [14] for the generalized Steiner tree problem. Unfortunately, we are unable to analyze this algorithm in a way that exhibits the effects of the more generous (parametrized) bound. Alternatively, we provide computational results that suggest that while the greedy strategy proposed in [5] is inoffensive for the worst case instance given $0 \le z \le 1$, it compromises on the competitive ratio obtained on the worst case instance given $z > 1$. Our experiments suggest that proposed algorithm improves on the competitive ratio obtained by the greedy strategy by some constant factor.

The significance of this problem stems from the fact that, in some real-world applications, decisions are made after collecting a batch of data. Moreover, in many of these applications, batches of data are rarely

---

[1] If the algorithm is deterministic, an oblivious adversary is equivalent to an adaptive adversary; one which makes requests adaptively in response to the algorithm [4].

[2] This scheme first appeared in the context of approximation algorithms [15].

produced by an absolute worst case adversary. The intent of our restricted adversarial model is to mimic the worst case instances that may effectively arise in the real world. For example, high-capacity ridesharing systems rely on the batch assignment of trip requests to vehicles [3], and some related problems such as that of optimal congregation points for passenger pickups and dropoffs can be modeled as a batched set cover problem. Intuitively, sequencing a batch of travel requests defeats the purpose of preparing the batch in the first place. Moreover, the batches that arise in this setting tend to have a VC-dimension greater than or equal to two, as the application revolves around exploiting the overlaps between distinct requests (see Section 2). Our results formalize this intuition. As listed in [5], further examples of applications of the batched set cover problem may be found in distributed computing, facility planning, and subscription markets.

To the best of our knowledge, we are the first to use the VC-dimension in the context of online (batched) covering problems. The VC-dimension has been used successfully in the context of approximation algorithms for (offline) set cover problems [6, 10] as well as in the context of improved running time bounds for unconstrained [1] and constrained [13] shortest path algorithms. In both of these settings, the algorithms exploit the low VC-dimension of the set systems on which they operate. Perhaps surprisingly, algorithms for the batched set cover problem may instead exploit the high VC-dimension of the set systems on which they operate, which we model as a restriction on the adversary. Intuitively, the reason is that the adversary is forced to reveal complex, intertwined batches. A dedicated algorithm attempts to exploit the richness of the information revealed, while a greedy algorithm is myopic to the interactions between the set memberships of the elements in a batch.

1.3. Organization

In Section 2 we formally introduce our problems and definitions. In Section 3 we consider bounds for the online fractional set cover problem. We present a known lower bound of $\Omega(H_m)$ on the competitive ratio of any online algorithm. While the tightness of this lower bound (up to constants) follows immediately from the existence of $O(H_m)$-competitive fractional algorithms [7, 8], we present an inductive proof that shows the tightness of the lower bound without the explicit need of a competitive algorithm. This technique is used in Section 4 to show the tightness of a $\Omega(H_{m-2^z+1})$-lower bound on the competitive ratio of any batched algorithm, given our restricted adversary parametrized by z. The reason for doing this is our argument that batching may offer a constant factor improvement in the competitive ratio. Hence, tightness up to constants is not informative enough for our purposes. In Section 5 we formalize the greedy strategy suggested in [5] and present our minor modification to an online algorithm found in [8]. We also present the results of our computational experiments.

2. Preliminaries

Let $x_j \in \{0,1\}$ be set to 1 if S_j is brought to the set cover and to 0 otherwise. Now, consider LP1, which describes the linear programming relaxation of the offline set cover problem. We refer to LP1 as the primal covering problem. Here, $c_j > 0$ refers to the cost of bringing some set $S_j \in S$ to the set cover, and the objective is to minimize the total cost incurred. In the unweighted case, $c_j = 1$ for all $j = 1, \cdots, m$. Constraints (1.1) ensure that every element $i = 1, \cdots, n$ in the ground set X is covered. Note that the set membership information of each element is encoded in its respective constraint.

(LP1)

$$minimize \sum_{\{j = 1, \cdots, m\}} c_j x_j$$
$$s.t.$$
$$(1.1) \sum_{\{j: i \in S_j\}} x_j \geq 1, \qquad i = 1, \ldots, n$$

5

$$x_j \geq 0, \qquad j = 1, \cdots, m$$

The primal covering problem has an associated dual packing problem, described in LP2. We refer to this primal-dual formulation throughout this work. We will refer to the collection of sets in S that individually contain $\sigma_i \in X$ by $S(\sigma_i)$.

(LP2)

$$maximize \sum\_\{i = 1, \cdots, n\} \ y_i$$
$$s.t.$$
$$(1.1)\sum\_\{i \in S_j\} \ y_i \leq c_j, \qquad j = 1, \ldots, m$$
$$y_i \geq 0, \qquad i = 1, \cdots, n$$

In the fractional online setting [7, 8] the objective function of LP1 is known a priori, but constraints (1.1) are revealed one by one. This corresponds to the algorithm identifying the costs of the sets in S a priori, but the adversary revealing a sequence $\sigma = \sigma_1, \sigma_2, \cdots$ of elements of X, along with their respective set membership, in an online fashion. Equivalently, the right hand side of the constraints (2.1) of LP2 are known a priori, but the variables involved in them and in the objective function are revealed one by one.

Now, consider the following batched version of the set cover problem, which is also a game between an algorithm and an oblivious adversary. In the *batched set cover problem*, S is identified a priori, but the adversary produces a sequence $\beta = \beta_1, \beta_2, \cdots$ of batches of elements of X, which it reveals one batch at a time. For instance, $\beta_k = \{\sigma_{k,1}, \cdots, \sigma_{k,|\beta_k|}\}$, where $|\beta_k|$ denotes the size of the kth batch. When a batch arrives, all of its elements, along with their respective set membership information, are revealed simultaneously. The fractional batched setting is analogous to the fractional online setting, except constraints (1.1) appear in tandem. Equivalently, the variables involved in the objective function and constraints (2.1) are revealed in tandem. Note that the online setting is trivially recovered when each batch is a singleton. We refer to the union of sets in S that individually cover the elements in $\beta_k = \{\sigma_{k,1}, \cdots, \sigma_{k,|\beta_k|}\}$, namely $S(\sigma_{k,1}) \cup \cdots \cup S(\sigma_{k,|\beta_k|})$, by $S(\beta_k)$.

We define an instance I of the online set cover problem as a collection S together with the adversarial sequence. We introduce the following performance measures. The batched setting for both of these measures is analogous.

Definition 1. An online algorithm ALG^O is said to be c-competitive if for every instance I of the problem it outputs a solution of cost at most $c \cdot OPT(I)$ where $OPT(I)$ is the cost of the optimal offline solution.

Definition 2. An online adversary ADV^O is said to be c-advantaged if it produces an instance I such that every online algorithm ALG^O outputs a solution of cost at least $c \cdot OPT(I)$, where $OPT(I)$ is the cost of the optimal offline solution.

Our analysis in Section 4 relies on imposing a minimum on the VC-dimension of any batch β_k produced by the adversary. The VC-dimension was first proposed by Vapnik and Chernovekis [12], and it is a widely used measure of complexity in computational learning theory. We work with the following definitions.

Definition 3 (Set System). A set system (X, S) is a ground set X together with a collection S of subsets of X.

Definition 4 (Shattering). A subset $B \subseteq X$ is said to be *shattered* by S if $\{S \cap B : S \in S\} = P(B)$, where $P(B)$ is the power set of B.

Definition 5 (VC-dimension). The VC-dimension of a set system (X, S) is the cardinality of the largest subset $B \subseteq X$ to be shattered by S. We denote it by $VCD(X, S)$.

In particular, upon the arrival of a batch β_k we obtain a set system (β_k, S), where S is known a priori. Moreover, note that *i*) restricting the adversary to produce batches β_k with VC-dimension $VCD(\beta_k, S) \geq z$ is only meaningful when $m = |S| \geq 2^z$; otherwise the adversary is unable to produce any batches, and *ii*) by definition, any batch satisfying $VCD(\beta_k, S) \geq z$ also satisfies $|\beta_k| \geq z$. This is illustrated in Figure 1, which showcases how a batch β_k satisfying $VCD(\beta_k, S) \geq z$ can be constructed with $m = 2^z$ and $|\beta_k| = z$. Observe that in each of the cases, β_k is shattered since each of its subsets is the intersection of β_k with some $S \in S$. Of course, given z, there may be instances for which $m > 2^z$, or for which the adversary produces batches satisfying $|\beta_k| > z$, or both. We consider these cases in our analysis in Section 4.
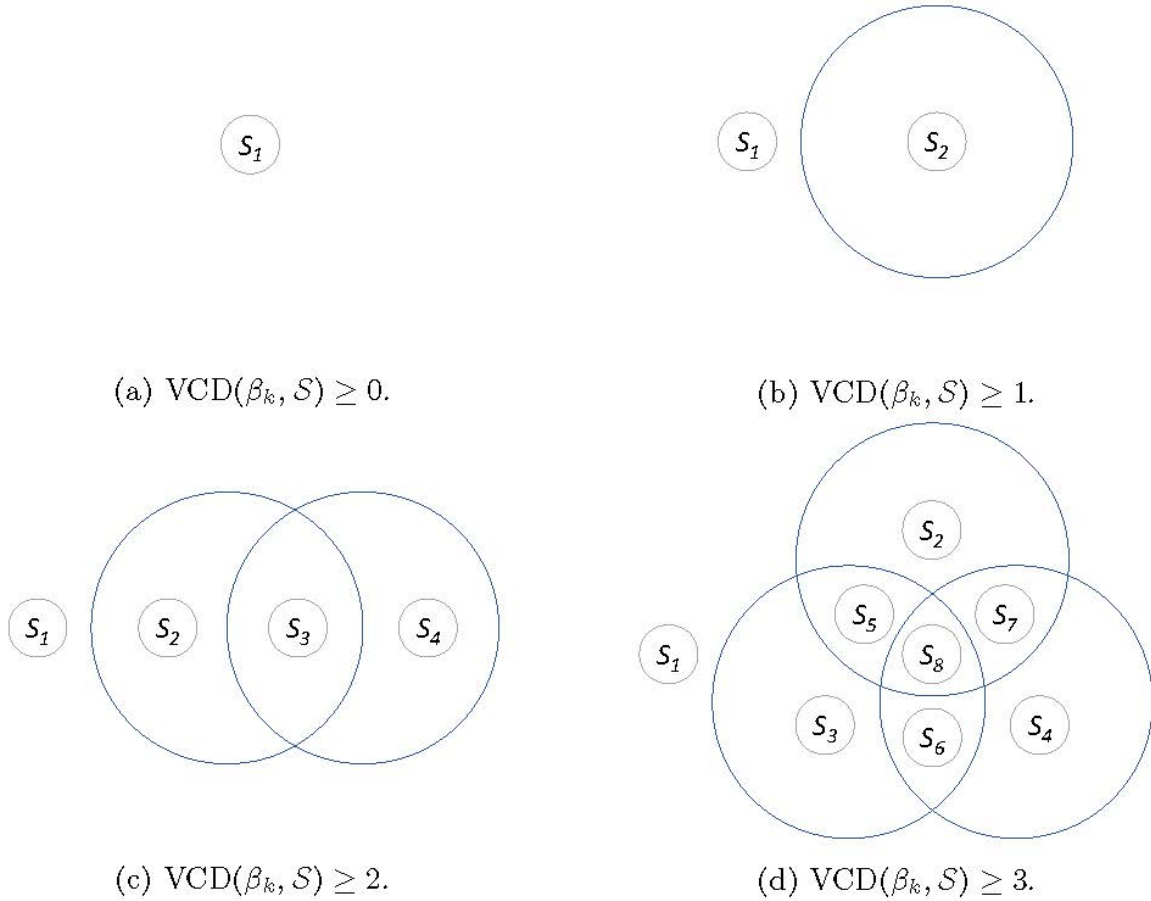

(a) VCD(β_k, S) ≥ 0.

(b) VCD(β_k, S) ≥ 1.

(c) VCD(β_k, S) ≥ 2.

(d) VCD(β_k, S) ≥ 3.

Figure 1: Construction of S *and* β_k *satisfying* $VCD(\beta_k, S) \geq z$, *for* $z = 0,1,2,3$ *with minimum possible cardinality requirements on* $m = |S|$ *and* $|\beta_k|$. *The construction for* $z > 3$ *is analogous. The nodes labeled* $S_1, \cdots, S_m$ *represent the sets in S, whereas the circles around them represent the elements of* $\beta_k \subseteq X$ *they contain (i.e., constraints in LP1)*

In a ridesharing context, we may interpret each S as a possible congregation point (e.g., an intersection in a street network), whereas each constraint corresponds to the set of compatible congregation points (e.g., within walking distance) for each trip origin or destination. Note that the only reasons why we would have $VCD(\beta_k, S) \leq 1$ are if *i*) $|\beta_k| \leq 1$, or *ii*) $|\beta_k| > 1$ but all the travel request form either a collection of proper subsets or a collection completely disjoint subsets of the possible congregation points. Given

7

sufficiently high and heterogeneous demand, batches with $VCD(\beta_k, S) \leq 1$ are unlikely to arise; the batches that arise look more like the those in Figure 1 (c) and (d).

3. Fractional Online Set Cover

Lemma 1 (Variation of Buchbinder and Naor [8]). There exists an instance I^* of the unweighted fractional online set cover problem such that any online algorithm is $\Omega(H_m)$-competitive on this instance.

Proof. Consider the following instance I^*, which is particular to the sequence σ produced by an adversary in response to an arbitrary online algorithm ALG^O. Let $\sigma_1 \in \bigcap_{S_j \in S} S_j$. Upon the arrival of σ_1, ALG^O must satisfy $\sum_{j:\sigma_1 \in S_j} x_j \geq 1$. Thus, it must let $x_j \geq \frac{1}{m}$ for at least one $S_j \in S$. Refer to such $S_j \in S$ as S^1 and to its corresponding variable as x^1. Now, let $\sigma_2 \in \bigcap_{S_j \in S \setminus S^1} S_j$. Upon the arrival of σ_2, ALG^O must satisfy $\sum_{j:\sigma_2 \in S_j} x_j \geq 1$. Thus, it must let $x_j \geq \frac{1}{m-1}$ for at least one $S_j \in S \setminus S^1$. Again, refer to such $S_j \in S \setminus S^1$ as S^2 and to its corresponding variable as x^2. In general, an adversary may continue revealing elements σ_i satisfying

$$\sigma_i \in \bigcap\_\{S_j \in S \setminus \bigcup_{i' < i} S^{i'}\} S_j,$$

forcing the algorithm to let $x_j \geq \frac{1}{m-i+1}$ for at least one $S_j \in S \setminus \bigcup_{i' < i} S^{i'}$. Refer to such $S_j \in S \setminus \bigcup_{i' < i} S^{i'}$ as S^i and to its corresponding variable as x^i. After m steps, the total cost $ALG^O(I^*)$ incurred by the algorithm, namely $x^1 + \cdots + x^m$, is at least

$$\frac{1}{m} + \frac{1}{m-1} + \cdots + \frac{1}{2} + 1 = H_m.$$

Meanwhile, the total cost $OPT(I^*)$ incurred by an optimal offline solution is 1, which corresponds to simply letting $x^m = 1$. Thus, $H_m \leq \frac{ALG^O(I^*)}{OPT(I^*)}$. Q.E.D.

Note that I^* depends on ALG^O only in the sense that the particular adversarial sequence σ produced is a response to the particular algorithm; the lower bound on the competitive factor, on the other hand, is independent of the algorithm. Thus, we may parametrize the instance I^* in Lemma 1 by $m = |S|$ to obtain the instance class $I^*(m)$. In other words, $I^*(m)$ refers to the instances that produce a lower bound of H_m on the competitive factor of any online algorithm as a result of the adversary following the strategy in the proof of Lemma 1. If we vary m, we obtain the family of instance classes $I^* = \{I^*(m): m \in \mathbb{Z}_+\}$.

The tightness of this lower bound (up to constants) is an immediate result of the existence of $O(\log m)$-competitive algorithms for the fractional online set cover problem [7, 8]. In Lemma 3 we present a different approach to show that this lower bound is tight, this time without relying on a particular algorithm. We use Lemma 2 whose proof is in the Appendix.

Lemma 2. $H_r \geq \frac{1}{2}(H_{r-t} + H_t)$ for any $r \geq t \geq 0$.

Lemma 3. There exists an online algorithm ALG^O for the unweighted fractional online set cover problem such that any adversary is $O(H_m)$-advantaged. In particular, this bound is matched by any adversary that follows the strategy in the family of instance classes I^*, described in the proof of Lemma 1.

Proof. By Lemma 1, there exists an adversary that is H_m-advantaged, namely one that follows the strategy of instance class $I^*(m)$. We need to show that this is indeed the best an adversary can be guaranteed to

achieve. We prove this by strong induction on m and by focusing on an arbitrary adversary ADV^O. We will make use of the existence of a randomized algorithm ALG^O that, in principle, produces specific outcomes with non-zero probability.

Base Case ($m = 1$): When ADV^O reveals any element σ_1, it must be the case that $\sigma_1 \in S_1$. Then, ALG^O must let $x = 1$, achieving a competitive factor of H_1, as in $I^*(1)$.

Inductive Step: Assume, by way of strong induction, that the statement is true for $m = 2, \cdots, w$. We need to show that the statement is true for $m = w + 1$. By Lemma 1, there exists an adversary that is H_{w+1}-advantaged, namely one that follows the strategy of instance class $I^*(w + 1)$. Now, consider the case in which ADV^O deviates from the strategy of $I^*(w + 1)$ on some arbitrary step i. Let σ_{i^*} be the ith element according to the strategy of $I^*(w + 1)$. If ADV^O reveals an element σ_i such that $S(\sigma_i) \cap S(\sigma_{i^*})$ is empty, the cost of the optimal solution increases by 1, which by Lemma 2 irrevocably decreases the advantage of ADV^O. Therefore, suppose that ADV^O reveals an element σ_i such that $S(\sigma_i) \cap S(\sigma_{i^*})$ is non-empty. Then, with non-zero probability ALG^O disregards all $S_j \in S(\sigma_i) \setminus S(\sigma_{i^*})$, if any, making such deviation futile. Thus, safely assume that ADV^O instead reveals an element σ_i such that $S(\sigma_i) \subset S(\sigma_{i^*})$. Let $t = |S(\sigma_i)|$, $r = |S(\sigma_{i^*})|$, and note that $r > t$. Let $i' > i$ be the first step after step i such that $S(\sigma_i) \cap S(\sigma_{i'}) \neq \emptyset$. As before, with non-zero probability ALG^O disregards all $S_j \in S(\sigma_{i'}) \setminus S(\sigma_i)$, if any, so safely assume that ADV^O reveals an element $\sigma_{i'}$ such that $S(\sigma_{i'}) \subset S(\sigma_i)$. Then, by the inductive hypothesis, given that element σ_i satisfied $S(\sigma_i) \subset S(\sigma_{i^*})$, the best ADV^O can do is to recreate $I^*(t)$ on the remainder of the steps, starting with i'. In particular, the best ADV^O can do is to reveal an element $\sigma_{i'}$ such that $S(\sigma_{i'}) \subset S(\sigma_i)$ and $|S(\sigma_{i'})| + 1 = |S(\sigma_i)|$. A symmetric argument can be made about concurrently recreating $I^*(r - t)$ on the remainder of the steps, which is disjoint from $I^*(t)$ after the ith step and hence increases the offline solution by one. However, by Lemma 2, this achieves a strictly lower competitive advantage for ADV^O. Q.E.D.

4. Fractional Batched Set Cover

4.1. General Case

Lemma 4. There exists an instance I^* of the unweighted fractional batched set cover problem such that any batched algorithm is $\Omega(H_m)$-competitive on this instance.

Lemma 5. There exists a batched algorithm ALG^B for the unweighted fractional batched set cover problem such that any adversary is $O(H_m)$-advantaged.

Lemma 4 follows from the fact that the fractional online set cover problem is a special case of the fractional batched set cover problem, together with Lemma 1. In Section 4.2 we consider the case in which the adversary is imposed a minimum VC-dimension for any batch β_k produced. In the proof of Lemma 7, we mention why an adversary never benefits from producing batches with a VC-dimension larger than the minimum required. This, together with Lemma 3, yields Lemma 5.

4.2. Restricted Adversary

Our intent now is to characterize instance classes that distinguish the fractional batched set cover problem from the fractional online set cover problem. In particular, we restrict the adversary in that it is forced to produce batches β_k satisfying $VCD(\beta_k, S) \geq z$, for some $z \in \mathbb{N}^0$. Given z, we assume that $m = |S| \geq 2^z$ and $\beta_k \geq z$, as described in Section 2.

Lemma 6. There exists an instance I_z^* of the unweighted fractional batched set cover problem, with an adversary satisfying $VCD(\beta_k, S) \geq z$ for any batch β_k, such that any batched algorithm is $\Omega(H_{m-2^z+1})$-competitive on this instance.

Proof. This proof is analogous to that of Lemma 1. Consider the following instance I_z^*, which is particular to the sequence β produced by an adversary satisfying $VCD(\beta_k, S) \geq z$ for any batch β_k, in response to an arbitrary batched algorithm ALG^B. In the following, our ordering of $S_1, \cdots, S_m \in S$ is arbitrary.

Let $\beta_1 = \{\sigma_{1,1}, \cdots, \sigma_{1,z}, \sigma_{1,z+1}\}$ such that $S(\{\sigma_{1,1}, \cdots, \sigma_{1,z}\})$ is as in the diagram in Figure 1 on sets $S_1, \cdots, S_{2^z-1}, S_{2^z}$ with $\beta_1 \cap S_{2^z} = \beta_1$, in addition to each $\sigma_{1,q} \in \beta_1$ being also contained in each $S_j \in \{S_{2^z}, S_{2^z+1}, \cdots, S_m\}$. This last property cannot decrease the VC-dimension, so β_1 is a valid batch. Then, because of the constraint corresponding to $\sigma_{1,z+1} \in \beta_1$, ALG^B must let $x_j \geq \frac{1}{m-2^z+1}$ for at least one $S_j \in \{S_{2^z}, \cdots, S_m\}$. For clarity and without loss of generality, assume such S_j is S_{2^z}.

Then, let $\beta_2 = \{\sigma_{2,1}, \cdots, \sigma_{2,z}, \sigma_{2,z+1}\}$ such that $S(\{\sigma_{2,1}, \cdots, \sigma_{2,z}\})$ is as in the diagrams in Figure 1 on sets $S_2, \cdots, S_{2^z}, S_{2^z+1}$ with $\beta_2 \cap S_{2^z+1} = \beta_2$, in addition to each $\sigma_{2,q} \in \beta_2$ being also contained in each $S_j \in \{S_{2^z+1}, \cdots, S_m\}$. Then, because of the constraint corresponding to $\sigma_{2,z+1} \in \beta_2$, ALG^B must let $x_j \geq \frac{1}{m-2^z-1+1}$ for at least one $S_j \in \{S_{2^z+1}, \cdots, S_m\}$. For clarity and without loss of generality, assume such S_j is S_{2^z+1}.

In general, an adversary may continue revealing $\beta_k = \{\sigma_{k,1}, \cdots, \sigma_{k,z}, \sigma_{k,z+1}\}$ such that $S(\{\sigma_{k,1}, \cdots, \sigma_{k,z}\})$ is as in the diagrams in Figure 1 on sets $S_k, \cdots, S_{2^z+k-2}, S_{2^z+k-1}$ with $\beta_k \cap S_{2^z+k-1} = \beta_k$, in addition to each $\sigma_{k,q} \in \beta_k$ being also contained in each $S_j \in \{S_{2^z+k-1}, \cdots, S_m\}$. Then, because of the constraint corresponding to $\sigma_{k,z+1} \in \beta_k$, ALG^B must let $x_j \geq \frac{1}{m-2^z-(k-1)+1}$ for at least one $S_j \in \{S_{2^z+k-1}, \cdots, S_m\}$. For clarity and without loss of generality, assume such S_j is S_{2^z+k-1}.

After the $m - 2^z + 1$ possible steps, the total cost $ALG^B(I_z^*)$ incurred by the algorithm, namely $x_{2^z} + \cdots + x_m$, is at least

$$H_{m-2^z+1} = \frac{1}{m - 2^z + 1} + \frac{1}{m - 2^z} + \cdots + \frac{1}{2} + 1.$$

Meanwhile, the total cost $OPT(I_z^*)$ incurred by an optimal offline solution is 1, which corresponds to simply letting $x^{m-2^z+1} = 1$. Thus, $H_{m-2^z+1} \leq \frac{ALG^O(I_z^*)}{OPT(I_z^*)}$. In simple terms, the adversary may capitalize on $S_j \in \{S_{2^z}, \cdots, S_m\}$ while assuming a sunk cost $S_j \in \{S_1, \cdots, S_{2^z-1}\}$. Q.E.D.

As in Section 3, we parametrize the instances I_z^* in Lemma 6 by m. Thus, given z, we obtain the family of instance classes $I_z^* = \{I_z^*(m) : m \in \mathbb{Z}_+\}$. We recover the general case when $z = 0$. Analogous to Lemma 3, Lemma 7 shows the lower bound given z is tight. Its proof can be found in the Appendix.

Lemma 7. There exists a batched algorithm ALG^B for the unweighted fractional online set cover problem such that any adversary satisfying $VCD(\beta_k, S) \geq z$ for any batch β_k is $O(H_{m-2^z+1})$-advantaged. In particular, this bound is matched by any adversary that follows the strategy in the family of instance classes I_z^*, described in the proof of Lemma 6.

5. Batched Algorithms

5.1. Analysis

Note that since any batch β_k could be artificially decomposed into a sequence of $|\beta_k|$ elements that are 'revealed' one at a time, any competitive algorithm for online set cover would produce a competitive feasible solution; we refer to this as the 'trivial' approach. More precisely, the trivial approach consists of two phases: *i)* some (possibly randomized) subroutine that executes a mapping $f: \beta_k \to (\sigma_{k,1'}, \cdots, \sigma_{k,b_k'})$, where $\sigma_{k,q'}$ is the q'th element of the artificial sequence, followed by *ii)* any competitive algorithm for the online set cover problem.

An example of such approach is the $O(\log m)$-competitive primal-dual subroutine in Algorithm 1, which we refer to as $ALG^{B,T}$ followed by any $O(\log n)$ rounding technique (i.e., the second phase of the primal-dual method). $ALG^{B,T}$ is a minor modification of the $O(\log m)$-competitive Algorithm 2 in Section 4.2 of [8] for the batched setting. Note that $d = max_{k,q}|S(\sigma_{k,q})| \leq m$. Its correctness follows immediately from Theorem 4.2 of [8]. For conciseness, we only mention that the proof relies on showing three claims together with weak duality: *i)* the algorithm produces a primal feasible solution to LP1, *ii)* the algorithm produces a dual feasible solution to LP2, and *iii)* the objective value of LP1 is bounded above by $O(\log m)$ times the objective value of LP2. Clearly, the complete algorithm is $O(\log m \log n)$-competitive.

Algorithm 1: $ALG^{B,T}$: 'Trivial' batched algorithm.

/* Upon the arrival of batch $k := \{\sigma_{k,1}, \cdots, \sigma_{k,|\beta_k|}\}$:   */

1. **for** $\sigma_{k,q} \in f(\{\sigma_{k,1}, \cdots, \sigma_{k,|\beta_k|}\})$ **do**
2.   **while** $\sum_{j:\sigma_{k,q}\in S_j} x_j < 1$ **do**
3.   Increase $y_{k,q}$ continuously
4. $x_j \leftarrow \frac{1}{d}[\exp(\frac{\ln(1+d)}{c_j} \cdot \sum_{k',q':\sigma_{k',q'}\in S_j} y_{k',q'}) - 1], \forall j: \sigma_{k,q} \in S_j$
5.   **end**
6. **end**

Nevertheless, unless for all $|\beta_k| = 1$ for all k, the trivial approach may fail to leverage the rich information that is possibly implicit in the fact that all elements of a given batch are revealed simultaneously. We refer to an algorithm that attempts to leverage any such information as a 'dedicated' algorithm. We obtain such an algorithm if we replace $ALG^{B,T}$ with Algorithm 2, which we refer to as $ALG^{B,D}$.

Algorithm 2: $ALG^{B,D}$: 'Dedicated' batched algorithm.

/* Upon the arrival of batch $k := \{\sigma_{k,1}, \cdots, \sigma_{k,|\beta_k|}\}$:   */

1. **while** $\exists \sigma_{k,q}$ such that $\sum_{j:\sigma_{k,q}\in S_j} x_j < 1$ **do**
2.   Increase $y_{k,q}$ continuously, $\forall q: \sum_{j:\sigma_{k,q}\in S_j} x_j < 1$
3. $x_j \leftarrow \frac{1}{d}[\exp(\frac{\ln(1+d)}{c_j} \cdot \sum_{k',q':\sigma_{k',q'}\in S_j} y_{k',q'}) - 1] \forall j: \exists \sigma_{k,q} \in S_j$
4. **end**

Note the difference in how the dual variables are updated: sequentially in $ALG^{B,T}$ and simultaneously in $ALG^{B,D}$. This is reminiscent of the approach of increasing multiple variables at once in a primal-dual algorithm by [14] for the generalized Steiner tree problem. As expected, $ALG^{B,D}$ is also $O(\log m)$-

competitive[3]; it is also a minor modification of the $O(\log m)$-competitive Algorithm 2 in Section 4.2 of [8], and its correctness also follows immediately from Theorem 4.2 of [8]. Unfortunately, we are unable to improve its analysis in a way that reflects the intuition that batching should improve the competitive factor obtained under certain conditions. For example, we expect batching to help in the families of instances I_z^*, shown in Section 4 to produce a more generous lower bound on the competitive factor of any algorithm due to the VC-dimension requirement on the batches produced by the adversary. We leave presenting such an analysis as an open problem. As an alternative, in Section 5.2 we present the results of computational experiments that compare the performance (i.e., competitive factor) of $ALG^{B,T}$ and $ALG^{B,D}$ on instances of I_z^*.

5.2. Computational Experiments

Figure 2 (a) and (b) present the results of computational experiments that compare the worst-case performance (i.e., competitive factor) of $ALG^{B,T}$ and $ALG^{B,D}$ obtained on instances of I_z^*, respectively, for various values of z and m. We discretize both algorithms with a step size of $\epsilon = 0.001$. We justify the use of instances of I_z^* by the fact that it is not some arbitrary family of instance classes; Lemma 6 and Lemma 7 imply it produces tight bounds. Also, note that because of the symmetric nature of the batches β_k that arise in I_z^*, the particular order produced by the mapping $f \colon \beta_k \to (\sigma_{k,1'}, \cdots, \sigma_{k,|\beta_k|'})$ in $ALG^{B,T}$ is irrelevant for evaluating the worst-case performance of the trivial approach so long as $\sigma_{k,z+1}$ is the last element 'revealed'. This notion does not translate to $ALG^{B,D}$, since the dual variable updates occur simultaneously. Also, recall that for instances in I_z the optimal offline solution is one.
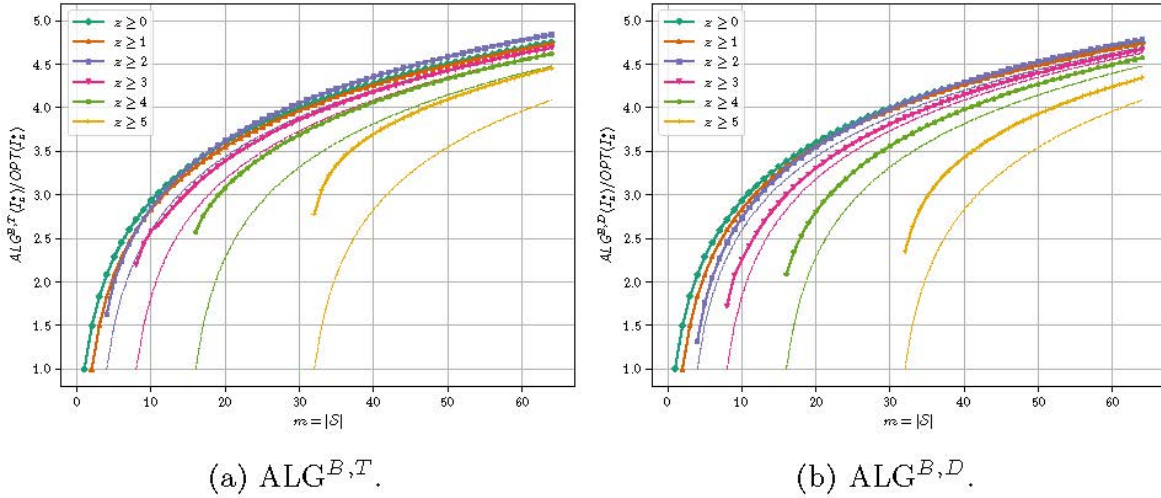

(a) ALG^{B,T}. (b) ALG^{B,D}.

Figure 2: Competitive ratio of (a) ALG^{B,T} and (b) ALG^{B,D}, obtained on instances of I_z^* for $z = 0,1,2,3,4$. The solid line indicates the competitive ratio obtained by the algorithm, while the dashed line indicates the $\Omega(H_{m-2^z+1})$ lower bound on the competitive ratio of any batched algorithm.

Note that for $z = 0,1$, both algorithms match the theoretical lower bound. This is expected, as in both of these cases, per the description of I_z^*, all the batches are singletons. On the other hand, for $z > 1$, neither of the algorithms match said lower bound. However, it can be observed that $ALG^{B,D}$ is closer than $ALG^{B,T}$ to the theoretical lower bound on the competitive ratio; this is not surprising as the algorithm has only be shown to be optimal up to constants. The improvement is more significant as z increases, though it is only

---

[3] To be precise, both algorithms are $2\ln(1 + d)$-competitive.

by a constant factor; both curves remain logarithmic. These results provide empirical evidence of the benefits of batching when the VC-dimension is high.

Acknowledgments

References

1. Abraham, I., Delling, D., Fiat, A., Goldberg, A.V., Werneck, R.F.: VC-dimension and shortest path algorithms. In: International Colloquium on Automata, Languages, and Programming. pp. 690–699. Springer (2011)
2. Alon, N., Awerbuch, B., Azar, Y.: The online set cover problem. In: Proceedings of the 35th annual ACM Symposium on the Theory of Computation. pp. 100–105 (2003)
3. Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D.: On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences 114(3), 462–467 (2017).
4. Ben-David, S., Borodin, A., Karp, R., Tardos, G., Wigderson, A.: On the power of randomization in on-line algorithms. Algorithmica 11(1), 2–14 (1994)
5. Bhawalkar, K., Gollapudi, S., Panigrahi, D.: Online set cover with set requests. In: LIPIcs-Leibniz International Proceedings in Informatics. vol. 28. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2014).
6. Bronnimann, H., Goodrich, M.T.: Almost optimal set covers in finite vc-dimension. Discrete & Computational Geometry 14(4), 463–479 (1995)
7. Buchbinder, N., Naor, J.: Online primal-dual algorithms for covering and packing problems. In: Proceedings of the 13th Annual European Symposium on Algorithms. pp. 689–701 (2005)
8. Buchbinder, N., Naor, J.: The design of competitive online algorithms via a primal-dual approach. Foundations and Trends in Theoretical Computer Science 3(2-3), 93–263 (2009).
9. Dinur, I., Steurer, D.: Analytical approach to parallel repetition. In: Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing. pp. 624–633. STOC '14, ACM (2014).
10. Even, G., Rawitz, D., Shahar, S.M.: Hitting sets when the vc-dimension is small. Information Processing Letters 95 (2), 358–362 (2005)
11. Feige, U.: A threshold of lnn for approximating set cover. Journal of the ACM 45(4), 634–652 (1998).

12. Vapnik, V.N., Chervonenkis, A.Y.: On the uniform convergence of relative frequencies of events to their probabilities. In: Measures of complexity, pp. 11–30. Springer (2015)
13. Vera, A., Banerjee, S., Samaranayake, S.: Computing constrained shortest-paths at scale (2017)
14. Williamson, D.P., Goemans, M.X., Mihail, M., Vazirani, V.V.: A primal-dual approximation algorithm for generalized steiner network problems. Combinatorica 15(3), 435–454 (1995).
15. Williamson, D.P., Shmoys, D.B.: The Design of Approximation Algorithms. Cambridge University Press, New York, NY, USA, 1st edn. (2011)

Appendix

Proof of Lemma 2

Proof. Assume, without loss of generality, that $r - t \geq t$. Note that

$$H_r = \frac{1}{r} + \frac{1}{r-1} + \cdots + \frac{1}{t+1} + H_t.$$

Likewise, note that

$$\frac{1}{2}(H_{r-t} + H_t) = \frac{1}{2}(\frac{1}{r-t} + \frac{1}{r-t-1} + \cdots + \frac{1}{2} + 1 + \frac{1}{t} + \frac{1}{t-1} + \cdots + \frac{1}{2} + 1)$$

$$= \frac{1}{2}(\frac{1}{r-t} + \frac{1}{r-t-1} + \cdots + \frac{1}{t+1}) + \frac{1}{t} + \frac{1}{t-1} + \cdots + \frac{1}{2} + 1$$

$$= \frac{1}{2}(\frac{1}{r-t} + \frac{1}{r-t-1} + \cdots + \frac{1}{t+1}) + H_t.$$

Clearly,

$$\frac{1}{r} + \frac{1}{r-1} + \cdots + \frac{1}{t+1} > \frac{1}{2}(\frac{1}{r-t} + \frac{1}{r-t-1} + \cdots + \frac{1}{t+1}),$$

so the proof is complete. Q.E.D.

Proof of Lemma 7

Proof. This proof is analogous to that of Lemma 3. By Lemma 6 there exists an adversary satisfying $VCD(\beta_k, S) \geq z$ for any batch β_k that is H_{m-2^z+1}-advantaged, namely one that follows the strategy of instance class $I_z^*(m)$. We need to show that this is indeed the best an adversary can be guaranteed to achieve. We prove this by strong induction on m and by focusing on an arbitrary batched adversary ADV^B. We will make use of the existence of a randomized algorithm ALG^B that, in principle, produces specific outcomes with non-zero probability.

Base Case $(1 \leq m < 2^z)$: If $m < 2^z$, the statement is vacuously true because the adversary cannot produce any batches.

Base Case $(m = 2^z)$: When ADV^B reveals any batch $\beta_1 = \{\sigma_{1,1}, \cdots, \sigma_{1,z}\}$, it must be the case that $S(\beta_1)$ is as in the diagrams in Figure 1 on sets $S_1, \cdots, S_{m=2^z}$. Then, with non-zero probability, ALG^B disregards any $S_j \notin \{S_m\}$. In such case, ALG^B lets $x_m = 1$, achieving a competitive factor of H_1. This is in agreement with the description of $I_z^*(1)$.

Inductive Step. Assume, by way of strong induction, that the statement is true for $m = 2^z + 1, \cdots, w$. We need to show that the statement is true for $m = w + 1$. By Lemma 6, there exists an adversary that is H_{w+1-2^z+1}-advantaged, namely one that follows the strategy of instance class $I_z^*(w + 1)$. Now, consider the case in which ALG^B deviates from the strategy of $I_z^*(w + 1)$ on some arbitrary step k. Let $\beta_k = \{\sigma_{k,1}, \cdots, \sigma_{k,l}\}$, where $l \geq z$ and let $\beta_{k^*} = \{\sigma_{k^*,1}, \cdots, \sigma_{k^*,z}, \sigma_{k^*,z+1}\}$ be the kth batch according to the strategy of $I_z^*(w + 1)$. Further, denote $S^{\cap \beta_k} = \cap_{q=1}^l S(\sigma_{k,q})$, with $t = |S^{\cap \beta_k}|$, as well as $S^{\cap \beta_{k^*}} = \cap_{q=1}^z S(\sigma_{k^*,q})$ with $r = |S^{\cap \beta_{k^*}}|$. If $S^{\cap \beta_k} \cap S^{\cap \beta_{k^*}} = \emptyset$, then the cost of the optimal offline solution increases by 1, which by Lemma 2 irrevocably decreases the advantage of ADV^B. Therefore, suppose that ADV^B reveals a batch β_k such that $S^{\cap \beta_k} \cap S^{\cap \beta_{k^*}} \neq \emptyset$. Moreover, with non-zero probability ALG^B disregards all $S_j \in S(\beta_k) \setminus S^{\cap \beta_k}$, so safely assume this is the case for the rest of the proof. Now, with non-zero probability ALG^B disregards all $S_j \in S^{\cap \beta_k} \setminus S^{\cap \beta_{k^*}}$, if any, making such deviation futile. Thus, safely assume that $S^{\cap \beta_k} \subset S^{\cap \beta_{k^*}}$, implying that $t < r$. Let $k' > k$ be the first step after step k such that $S^{\cap \beta_k} \cap S(\beta_{k'}) \neq \emptyset$. As before, with non-zero probability ALG^B disregards all $S_j \in S(\beta_{k'}) \setminus S^{\cap \beta_{k'}}$, so safely assume that this is the case for the rest of the proof. Then, with non-zero probability ALG^B also disregards any $S_j \in S^{\cap \beta_{k'}} \setminus S^{\cap \beta_k}$, if any, so safely assume that $S^{\cap \beta_{k'}} \subset S^{\cap \beta_k}$. Then, by the inductive hypothesis, given that batch β_k satisfied $S^{\cap \beta_k} \subset S^{\cap \beta_{k^*}}$, the best ADV^B can do is to recreate $I_z^*(t)$ on the remainder of the steps, starting with k'. In particular, this requires $S^{\cap \beta_{k'}} \subset S^{\cap \beta_k}$ and $|S^{\cap \beta_{k'}}| + 1 = |S^{\cap \beta_k}|$. A symmetric argument can be made about concurrently recreating $I_z^*(r - t)$ on the remainder of the steps, which is disjoint from $I_z^*(t)$ after the kth step and hence increases the offline solution by one. However, by Lemma 2, this achieves a strictly lower competitive advantage for ADV^B.

Note that ADV^B cannot be guaranteed to benefit from producing batches β_k satisfying $VCD(\beta_k, S) > z$ because with non-zero probability ALG^B disregards any $S_j \in S(\beta_k) \setminus S^{\cap \beta_k}$, making such deviation futile with non-zero probability. In fact, a larger VC-dimension would involve more sets, possibly making $t = |S^{\cap \beta_k}|$ (and hence the attainable competitive advantage) smaller. Q.E.D.