

Chasing Positive Bodies

Sayan Bhattacharya* Niv Buchbinder† Roie Levin†‡ Thatchaphol Saranurak§

Abstract

We study the problem of chasing positive bodies in ℓ_1 : given a sequence of bodies $K_t = \{x^t \in \mathbb{R}_+^n \mid C^t x^t \geq 1, P^t x^t \leq 1\}$ revealed online, where C^t and P^t are nonnegative matrices, the goal is to (approximately) maintain a point $x_t \in K_t$ such that $\sum_t \|x_t - x_{t-1}\|_1$ is minimized. This captures the fully-dynamic low-recourse variant of any problem that can be expressed as a mixed packing-covering linear program and thus also the fractional version of many central problems in dynamic algorithms such as set cover, load balancing, hyperedge orientation, minimum spanning tree, and matching.

We give an $O(\log d)$ -competitive algorithm for this problem, where d is the maximum row sparsity of any matrix C^t . This bypasses and improves exponentially over the lower bound of $\sqrt{n}$ known for general convex bodies. Our algorithm is based on iterated information projections, and, in contrast to general convex body chasing algorithms, is entirely memoryless.

We also show how to round our solution dynamically to obtain the first fully dynamic algorithms with *competitive recourse* for all the stated problems above; i.e. their recourse is less than the recourse of every other algorithm on every update sequence, up to polylogarithmic factors. This is a significantly stronger notion than the notion of *absolute recourse* in the dynamic algorithms literature.

*Department of Computer Science, University of Warwick. Email: S.Bhattacharya@warwick.ac.uk. Supported by Engineering and Physical Sciences Research Council, UK (EPSRC) Grant EP/S03353X/1.

†Department of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University, Tel Aviv. Emails: niv.buchbinder@gmail.com, roiel@tauex.tau.ac.il. Supported in part by Israel Science Foundation grant 2233/19 and United States - Israel Binational Science Foundation grant 2022418.

‡Supported in part by a Fulbright Israel Postdoctoral Fellowship.

§Computer Science and Engineering Division, University of Michigan. Email: thsa@umich.edu. Supported by NSF CAREER grant 2238138.

1 Introduction

We study the problem of *chasing positive bodies* in ℓ_1 defined as follows. We are given a sequence of bodies $K_t = \{x^t \in \mathbb{R}_+^n \mid C^t x^t \geq 1, P^t x^t \leq 1\}$ revealed online, where C^t and P^t are matrices with non-negative entries. The goal is to (approximately) maintain a point $x^t \in K_t$ such that the total ℓ_1 -movement, $\sum_t \|x^t - x^{t-1}\|_1$, is minimized where $x^0 = 0$. More generally, given weight $w \in \mathbb{R}_+^n$, we want to minimize the weighted ℓ_1 -movement, $\sum_t w_i \sum_{i=1}^n |x_i^t - x_i^{t-1}|$. This captures the fully dynamic variant of any problem that can be expressed as a mixed packing-covering linear program, and thus also the fractional version of many central fully dynamic problems when the goal is to minimize *recourse*, i.e., the amount of change to the solution, as measured by ℓ_1 movement. Examples include set cover, load balancing (hyperedge orientation), minimum spanning tree, and matching.

A more general version our problem, called the *convex body chasing* problem, allows K_t to be an arbitrary convex body and the goal is to minimize total ℓ_p -movement for any $p \geq 1$. Friedman and Linial [FL93] introduced convex body chasing as a vast generalization of many online problems. This problem has been the subject of intensive study recently [BBE⁺20, ABC⁺19, BKL⁺20] which has culminated in algorithms with $O(n)$ competitive ratios [AGTG21, Sel20]. This bound is nearly tight since there is a lower bound of $\Omega(\max\{\sqrt{n}, n^{1-1/p}\})$ [BKL⁺20]. Unfortunately, because of this strong lower bound, algorithms for this general problem have generated no interesting applications to concrete combinatorial optimization problem.

In this paper, we show that for the special yet still expressive case of chasing positive bodies in ℓ_1 , we can bypass and exponentially improve upon the $\sqrt{n}$ lower bound. We then show that a solution to the positive body chasing problem can be rounded online to yield low-recourse fully dynamic algorithms for all the combinatorial problems mentioned above. Our algorithms have *competitive recourse* guarantees, a significantly stronger notion of recourse than the usual one used in the dynamic algorithm literature, which we will discuss soon.

1.1 Our Results

Our main contribution is the following theorem.

Theorem 1.1. *For any $\epsilon \in (0, 1]$, there is an $O(1/\epsilon \log(d/\epsilon))$ -competitive algorithm for chasing positive bodies in ℓ_1 such that $x^t \in K_t^{1+\epsilon} = \{x^t \in \mathbb{R}_+^n \mid C^t x^t \geq 1, P^t x^t \leq 1 + \epsilon\}$ at time t , and d is the maximal number of non-negative coefficients in a covering constraint.*

Note we give our algorithm ϵ -resource augmentation, i.e. we allow it to violate the packing constraints slightly. Alternatively, by scaling the solution, we may produce a solution that fully satisfies all packing constraints but violates the covering constraints up to an ϵ factor. As we discuss below, ϵ -resource augmentation does not matter in our applications since we lose additional approximation factors while rounding anyway. We compare the total movement of our algorithm that maintains a point in $K_t^{1+\epsilon}$ with an optimal solution that maintains a point in K_t . In several applications where $d = O(1)$, such as the fractional version of dynamic edge orientation and set cover with bounded frequency, our competitive ratio is completely independent of n . We also remark that we can handle static box constraints of the form $x \leq b$, where $b \in \mathbb{R}_+^n \cup \{\infty\}$ without any violation. See Appendix F for more details.

We complement Theorem 1.1 by the following lower bound:

Theorem 1.2. *No algorithm for positive body chasing can achieve competitive ratio better than*

$$\Omega\left(\max\left\{\min\left(\frac{1}{\epsilon\sqrt{\log(1/\epsilon)}}, \sqrt{n}\right), \log n\right\}\right).$$

The $\Omega(\log n)$ lower bound follows by the known bound for covering LPs [BN09b, GKL21]. The $\Omega(\min(1/(\epsilon\sqrt{\log(1/\epsilon)}), \sqrt{n}))$ lower bound follows by a reduction to the $\sqrt{n}$ lower bound for the general convex bodies chasing problem (Theorem 5.4 of in [BKL⁺20])¹. The full proof appears in Appendix E.

This lower bound implies that all the assumptions of Theorem 1.1 are crucial for the exponential improvement over the general convex body chasing case. In particular, it shows that the lower bound of $\Omega(\sqrt{n})$ for the general convex body chasing problem holds for the positive body chasing problem (without resource augmentation) as well. More generally, our linear dependence on $1/\epsilon$ is essentially tight. For example, when $\epsilon = \Theta(n^{-\delta})$ for a constant $0 < \delta < 1/2$, the competitive ratio is at least $\tilde{\Omega}(n^\delta)$ (hiding logarithmic terms), and in particular $O(\text{poly log } n)$ -competitive ratio is impossible. When $\epsilon = O(\frac{\log n}{n})$ the best algorithm is still the $O(n)$ -competitive algorithm for general convex body chasing [AGTG21, Sel20]. Finally, we remark that for ℓ_p -norms, when $p > 1$, the known lower bound of [FL93] forbids any $o(n^{1-1/p})$ -competitive algorithms even for chasing positive bodies and even with $\Omega(1)$ -resource augmentation.

Competitive Recourse for Dynamic Problems. An overarching goal of online and dynamic algorithms is to maintain near-optimal solutions to combinatorial problems as constraints change over time while minimizing the number of edits to the solution, a.k.a. recourse. An extensive line of work has produced low-recourse algorithms for Steiner tree under terminal updates [IW91, G GK16, GK14, LOP⁺15, GL20a], load balancing under job arrivals [AAPW01, GKS14, KLS23], set cover under element insertions/deletions [GKKP17, AAG⁺19, BHN19, GL20a, BHNW21, AS21], facility location under client updates [BLP22, GKLX20], and many fully dynamic graph problems under edge updates like edge orientation [BF99, SW20a, BBCG22], graph coloring [SW20b], maximal independent sets [AOSS18], and spanners [BKS12, BSS22].

The works mentioned above, and most others on dynamic algorithms, measure recourse in absolute terms. Such bounds are of the form “after T updates, the algorithm incurs recourse at most kT ”. We call these *absolute recourse* bounds. On the other hand, the online algorithms literature prefers *competitive* analysis, where one compares the performance of the algorithm to the best algorithm in hindsight. In this paper we give *competitive recourse* bounds: we say an algorithm has c -competitive recourse if it incurs recourse at most c times that of any other *offline* algorithm on every update sequence.

There are several advantages to studying competitive recourse over absolute recourse. An absolute recourse bound is a worst-case bound over *all* update sequences, whereas a competitive recourse bound is tailored to *each* update sequence. To illustrate this, consider an update sequence for set cover that repeatedly inserts and deletes the same element without changing the optimal solution. While competitive recourse algorithms would incur no recourse, an algorithm with absolute recourse guarantees might have large recourse proportional to these “irrelevant” updates. Hence, competitive recourse bounds can be much stronger than absolute bounds. For a concrete example, it is straightforward to obtain a fully dynamic algorithm for $(1 + \epsilon)$ -approximate matching with $O(1/\epsilon)$ absolute recourse by eliminating short augmenting paths, but no non-trivial algorithm with competitive recourse was known prior to our work.

For some problems small absolute recourse may not be possible at all. In the context of fully dynamic load balancing, [KLS23] show that no algorithm, even with full knowledge of the job arrival/departure sequence, can achieve competitive ratio $\alpha = o(\log(n))$ with less than $\Omega(n^{1/\alpha})$ absolute recourse. They write:

“To circumvent the negative result, one needs to consider a different measurement for recourse for the fully dynamic model.”

¹We would like to thank Mark Sellke for pointing out this reduction.

We argue in this paper that competitive recourse is the natural way around this obstacle.

Applications of Theorem 1.1 to Combinatorial Problems. In this paper, we give the first fully dynamic algorithms with polylogarithmic competitive recourse for central problems studied in the dynamic algorithms literature, including set cover, load balancing, bipartite matching and minimum spanning tree. The results are summarized in Table 1. In the table, our algorithms with α -approximation and c -competitive recourse guarantees have the following formal guarantee. For *every* update sequence, our algorithms maintain α -approximate solution with recourse at most c times the optimal recourse for maintaining an optimal solution. More generally, each of our algorithms even allow the following trade-off. Given parameter β , for *every* update sequence σ , it maintains a $(\alpha\beta)$ -approximate solution with total recourse at most $c \cdot \text{OPT}_{\text{REC}}^\beta(\sigma)$ where $\text{OPT}_{\text{REC}}^\beta(\sigma)$ denotes the optimal recourse for maintaining a β -approximate solution undergoing updates σ . This is useful because it is natural to expect that the optimal recourse becomes significantly smaller when we allow some approximation.

Problems	Approx.	Competitive recourse	Randomized vs. Deterministic	Ref.	Update type	Variables
Set cover	$O(\log n)$	$O(\log n \log f)$	Randomized	Cor. 4.2	elements	n elements, each in $\leq f$ sets
	$O(f)$	$O(f \log f)$	Deterministic	Cor. 4.2		
Load balancing	$2 + \epsilon$	$O\left(\frac{1}{\epsilon^3} \log n \cdot \log \frac{r}{\epsilon} \log \frac{1}{\epsilon}\right)$	Deterministic	Cor. 4.6	jobs	n jobs, each applicable to $\leq r$ machines
	$O\left(\frac{\log \log n}{\log \log \log n}\right)$	$O(\log r)$	Randomized	Cor. 4.6		
Bipartite matching	$1 + \epsilon$	$O\left(\frac{1}{\epsilon^4} \log n \log \frac{n}{\epsilon}\right)$	Randomized	Cor. 4.4	edges	n vertices
Minimum spanning tree	$2 + \epsilon$	$O\left(\frac{1}{\epsilon^3} \log n \log \frac{n}{\epsilon}\right)$	Randomized	Cor. 4.8	edges	n vertices

Table 1: Summary of our fully dynamic algorithms with competitive recourse. All randomized algorithms assume an oblivious adversary, and incur a polynomially-small additive term in the recourse bound which we omit. See Section 4 for details.

An important feature of our fractional algorithms is that their guarantees are independent of the type of dynamic update because Theorem 1.1 works so long as the feasible region at every step forms a positive body. For example, our fractional minimum spanning tree and bipartite matching algorithms work under both vertex and edge arrivals/departures, our fractional load balancing algorithm works under fine-grained updates to job-machine loads, and our fractional set cover algorithm works when set costs can be updated. Since our rounding schemes also work under these generalized types of updates, our final algorithms do as well.

We emphasize that our focus is minimizing recourse and not *update time*, the usual metric in the dynamic algorithms literature. Obtaining competitive recourse and small update time simultaneously is an interesting research direction.

1.2 Connections to Previous Work

Positive Body Chasing. Our result for chasing positive bodies directly generalizes the influential line of work on online covering problems [BN09b, GN14, GL20b]. In these, the convex bodies are nested and defined solely by covering constraints: for each time t , the body is $K_t = \{x \in \mathbb{R}_+^n \mid C^t x \geq 1\}$ where C^t is a non-negative matrix and $K_t \subseteq K_{t-1}$. The task is to maintain a monotonically increasing $x^t \in K_t$ while minimizing $\langle w, x \rangle$ for fixed $w \in \mathbb{R}_+^n$. This goal is equivalent to minimizing the ℓ_1 -movement weighted by w , since decreasing any

variable never helps to satisfy covering constraints. Even this special case of positive body chasing has been amazingly successful in unifying previous results and in resolving important open questions in competitive analysis. This includes, e.g., the classic ski rental problem, on-line set cover [AAA⁺09], weighted paging and variants [BBN12a, BBN12b, CLNT22], graph optimization problems related to connectivity and cuts [AAA⁺06, NPS11], the dynamic TCP-acknowledgement problem [KKR01, BJN07], metrical task systems [BBN10, BCLL21], and the k -server problem [BCL⁺18, BGMN19].

Another line of work [ABFP13, ABC⁺16] generalizes [BN09b, GN14] in a different way. Here the convex bodies K_t are nested covering constraints and $x^t \in K_t$ may only move monotonically. The goal in [ABFP13] is to minimize the maximum violation of a *fixed* set of packing constraints, and the goal of [ABC⁺16] is to minimize a non-decreasing convex function. This is different from our objective since we must handle an online sequence of both packing and covering constraints, but our solution x is allowed to move non-monotonically.

[BKS23] gave dynamic algorithms for maintaining $(1+\epsilon)$ -approximate solutions to mixed packing-covering linear programs. They showed how to handle *relaxing* updates (where the feasible region only grows) using small *update time*, and asked as an open problem if it is possible to do the same for *restricting updates* (where the feasible region only shrinks). Theorem 1.1 resolves the recourse version of this question: we can maintain an $(1+\epsilon)$ -approximate solution of the mixed packing-covering linear program with total absolute recourse $O(|x^*|_1 \cdot \frac{1}{\epsilon} \log \frac{n}{\epsilon})$, where x^* is any feasible solution at the end of the sequence of restricting updates. This follows because an offline algorithm can just move the solution to x^* from the beginning.

Competitive Recourse in Combinatorial Problems. Here, we only compare our algorithms from Table 1 with some of the few known dynamic algorithms with competitive recourse guarantees.

For the set cover problem, the online algorithms in [GN14] support element insertions and guarantee $\min\{O(\log n \log f), O(f \log f)\}$ -approximation, and these are the best possible. When every set has unit weight, the approximation ratio also translates to a competitive recourse bound, because no set ever leaves the solution. Our algorithm (Corollary 4.2) is the first fully dynamic competitive-recourse algorithm, even in the unweighted case.

For the load balancing problem, our algorithms have interesting consequences even when we focus on special cases; Corollary 4.6 implies the first competitive-recourse algorithm for fully dynamic edge orientation in weighted graphs.² Previously, this was known only for unweighted graphs [BF99]. In the unweighted setting, our algorithm with $O(\log r)$ -competitive recourse is the first algorithm for fully dynamic hyperedge orientation with *constant* competitive recourse and non-trivial approximation when the rank $r = O(1)$. The previous result of [BBCG22] suffers a $O(\log n)$ factor in recourse.

[GTW14] gave an algorithm for maintaining a spanning tree (more generally a matroid base) when edges have acquisition costs and evolving holding costs. Their result can be used to obtain an $O(\log^2 n)$ competitive recourse algorithm for maintaining a spanning tree under edge updates but does not give guarantees about the cost of the tree at every step. Our algorithm (Corollary 4.8), in contrast, guarantees a $(2+\epsilon)$ approximation with $O_\epsilon(\log^2 n)$ competitive recourse.

The only other dynamic algorithm with competitive recourse guarantees we are aware of are [AMPST23] for vertex coloring and [ALPS16, ABL⁺20] for balanced graph partitioning. We hope that our generic framework will facilitate further results in this direction.

²This corresponds to the load balancing instance where each job is applicable to two machines.

1.3 Techniques and Overview

Our key technical contribution is an algorithm for chasing positive *halfspaces*, or positive bodies defined by a single covering or packing constraint (we explain the reduction from chasing bodies to chasing halfspaces in Appendix A). Our algorithm is the following. Given point x^{t-1} from the last round, our algorithm performs a projection in KL divergence onto the new feasible region (plus a small additive update for technical reasons). This is sometimes referred to as an *information projection* (e.g. [Csi84]). One can interpret many online covering algorithms as information projections [BGMN19]; our contribution is to extend the analysis for packing constraints which has until now remained elusive. Interestingly, in contrast to competitive algorithms for general convex body chasing ([AGTG21, Sel20]) this algorithm is completely memoryless, in that it only depends on x^{t-1} and the current violated constraint.

To analyze the algorithm, we write an auxiliary linear program whose optimal value is the optimal recourse of an offline algorithm with knowledge of all future constraints, and then fit a dual to our online algorithm’s solution. We construct this dual using the Lagrange multipliers of the convex programs we use to compute our projections. We start with a warmup analysis in Sections 3.1 to 3.3. The proof here is simple and contains the main ideas, but loses an additional aspect ratio term. To obtain the full result of Theorem 1.1, we need to overcome a significant barrier: one can show that no monotone dual solution can avoid a dependence on the aspect ratio (see e.g. [BN09a]). Nevertheless, we show that it is possible to carefully *decrease* some coordinates of the dual to achieve the desired bound. The details are in Section 3.4.

We turn to rounding our competitive fractional solution for common applications in Section 4. Our chasing positive bodies framework captures the task of maintaining fractional solutions to set cover (Section 4.1), bipartite matching (Section 4.2), load balancing (Section 4.3), and minimum spanning tree (Section 4.4) under dynamic updates. We show that we can also round these fractional solutions to true combinatorial solutions with bounded loss in both recourse and approximation. Our rounding algorithms in Sections 4.2 and 4.4 reuse a common recipe: we first maintain an object we call a *stabilizer*, which contains a good solution that itself has good competitive recourse (see Appendix C and Appendix D for details). We then run existing *absolute* recourse algorithms for maintaining a true solution within this stabilizer. Since the number of updates to the stabilizer is competitive with the recourse of the optimal offline solution, and the absolute recourse algorithms are competitive with respect to the number of updates to the stabilizer, overall this strategy yields a solution with good competitive recourse.

2 Preliminaries

Mathematical Notation. All logarithms in this paper are taken to be base e . For $w \in \mathbb{R}$, we use the notation $w_+ = \max(0, w)$. We use the symbol $\oplus$ to denote symmetric difference of sets. In the following definitions, let $x, y \in \mathbb{R}_+^n$ be vectors. The standard dot product between x and y is denoted $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$. We use a weighted generalization of KL divergence. Given a weight function w , define

$$\text{KL}_w(x \parallel y) := \sum_{i=1}^n w_i \left[x_i \log \left(\frac{x_i}{y_i} \right) - x_i + y_i \right].$$

It is known that $\text{KL}_w(x \parallel y) \geq 0$ for nonnegative vectors x, y (one can check this is true term by term above). Throughout this paper, we use the convention that for any $a, b \geq 0$, the expression

$$(a + \infty) \log \left(\frac{a + \infty}{b + \infty} \right) - a + b = 0. \quad (2.1)$$

These definitions agree with the respective limit behaviors.

Simplifying Positive Body Chasing Instances. In the positive body chasing problem, since $x^0 = 0$ and $x^t \geq 0$, then we have

$$\sum_{t \in [T]} \sum_{i \in [n]} w_i (x_i^t - x_i^{t-1})_+ \leq \sum_{t \in [T]} \sum_{i \in [n]} w_i \cdot |x_i^t - x_i^{t-1}| \leq 2 \cdot \sum_{t \in [T]} \sum_{i \in [n]} w_i (x_i^t - x_i^{t-1})_+.$$

Therefore, without loss of generality, we use the *upward recourse* as the measure of the movement of both our algorithm and the optimal algorithm.

Henceforth, we assume without loss of generality that at each time t , the positive body K_t is defined by a single covering constraint $\langle c^t, x \rangle \geq 1$ or packing constraint $\langle p^t, x \rangle \leq 1$. This reduction from chasing bodies to chasing halfspaces is standard (see e.g. [BBE⁺20]), but we spell out the main argument in Appendix A for completeness.

Let $c_i^{\max} = \max_t c_i^t$, $c_i^{\min} = \min_{t | c_i^t \neq 0} c_i^t$. We define the aspect ratio $\Delta = \max_i c_i^{\max} / c_i^{\min}$. Let $d^t = |\{i \mid c_i^t \neq 0\}|$ be the sparsity of the covering constraint. Let $d = \max_t \{d^t\}$ be the maximal row sparsity of any covering constraint.

3 An Algorithm for Chasing Positive Bodies

In this section we design a fractional algorithm for chasing positive bodies proving Theorem 1.1. As discussed in Section 2 we may assume without loss of generality that at each time t the positive body K_t is defined by a single covering or packing constraint.

3.1 The Algorithm:

The algorithm is given a parameter $\epsilon \in (0, 1]$. Whenever it gets a violated constraint, it projects back to the constraint using the following procedure.

- Initially, $x_i^0 = 0$ for all i . At any time $t = 1, 2, \dots, T$:
- When a violated covering constraint $\langle c^t, x \rangle < 1$ arrives, set x^t to be the solution to

$$\begin{aligned} \min_x \quad & \sum_{i | c_i^t \neq 0} w_i \cdot \left[\hat{x}_i \log \left(\frac{\hat{x}_i}{\hat{x}_i^{t-1}} \right) - \hat{x}_i \right] \\ \text{s.t.} \quad & \langle c^t, x \rangle \geq 1 \end{aligned} \quad (I).$$

where $\hat{x}_i := x_i + \frac{\epsilon}{4d^t \cdot c_i^t}$, $\hat{x}_i^{t-1} := x_i^{t-1} + \frac{\epsilon}{4d^{t-1} \cdot c_i^{t-1}}$, and $d^t = |\{i \mid c_i^t \neq 0\}|$ is the sparsity of the covering constraint.³ If for some x_i^t the coefficients $c_i^t = 0$, then we set $x_i^t = x_i^{t-1}$.

- When we are given a violated packing constraint $\langle p^t, x \rangle > 1 + \epsilon$, set x^t to be the solution to

$$\begin{aligned} \min_x \quad & \sum_{i | p_i^t \neq 0} w_i \left(x_i^t \log \left(\frac{x_i^t}{x_i^{t-1}} \right) - x_i^t \right) \\ \text{s.t.} \quad & \langle p^t, x \rangle \leq 1 + \epsilon \end{aligned} \quad (II).$$

If for some x_i^t the coefficients $p_i^t = 0$, then we set $x_i^t = x_i^{t-1}$.

We assume without loss of generality that we only get violated constraints. Let $C, P \subseteq \{1, \dots, T\}$ be the set of times in which covering/packing constraints appear respectively.

³Note: at time t we redefine $\hat{x}_i^{t-1}$, even though it was already defined at $t-1$ as $x_i^{t-1} + \epsilon / (3d^{t-1} \cdot c_i^{t-1})$. This is a slight abuse of notation but it makes our proofs cleaner, and also makes apparent the connection to KL divergence.

3.2 Analysis Framework

Let y^t, z^t be the Lagrange multipliers of the constraints (I) and (II) above. In our proof we use the KKT conditions at the optimal solution of this program for every time-step t :

$$\forall t \in C \quad \langle c^t, x^t \rangle = 1 \quad (3.1)$$

$$\forall t \in C, \forall i, c_i^t \neq 0 \quad w_i \log \left(\frac{\hat{x}_i^t}{\hat{x}_i^{t-1}} \right) = c_i^t y^t \quad (3.2)$$

$$\forall t \in P \quad \langle p^t, x^t \rangle = 1 + \epsilon \quad (3.3)$$

$$\forall t \in P, \forall i \quad w_i \log \left(\frac{x_i^t}{x_i^{t-1}} \right) = -p_i^t z^t \quad (3.4)$$

For convenience, we define $y^0 = z^0 = 0$.

The optimal upward recourse can be computed by following linear program $(\overline{P})$.

$(\overline{P}) : \min \sum_{t=1}^T \sum_{i=1}^n w_i \bar{\ell}_i^t$ s.t. $\forall t \in C : \quad \langle c^t, \bar{x}^t \rangle \geq 1$ $\forall t \in P : \quad \langle p^t, \bar{x}^t \rangle \leq 1$ $\forall i, t : \quad \bar{x}_i^t - \bar{x}_i^{t-1} \leq \bar{\ell}_i^t$ $\bar{x}_i^t, \bar{\ell}_i^t \geq 0$	$(\overline{D}) : \max \sum_{t \in C} \bar{y}^t - \sum_{t \in P} \bar{z}^t$ s.t. $\forall i, t \in C : \quad c_i^t \bar{y}^t - \bar{r}_i^t + \bar{r}_i^{t+1} \leq 0$ $\forall i, t \in P : \quad -p_i^t \bar{z}^t - \bar{r}_i^t + \bar{r}_i^{t+1} \leq 0$ $0 \leq \bar{r}_i^t \leq w_i$ $\forall t : \quad \bar{y}^t, \bar{z}^t \geq 0$
--	--

Our goal from now on is to construct a feasible solution to the dual program $(\overline{D})$ that bounds the total cost of our online algorithm.

We start with a weaker bound than the one that appears in Theorem 1.1 as a warmup.

3.3 Warmup: an $O(1/\epsilon \cdot \log(d\Delta/\epsilon))$ competitive bound

In this section, we show our first bound on the recourse of our algorithm.

Theorem 3.1 (Warmup Bound). *The total recourse of the algorithm is at most $O\left(\frac{1}{\epsilon} \cdot \log\left(\frac{d\Delta}{\epsilon}\right)\right)$ times the objective of $(\overline{D})$.*

Recall that $c_i^{\max} = \max_t c_i^t$ and $c_i^{\min} = \min_{t|c_i^t \neq 0} c_i^t$, as well as the aspect ratio $\Delta = \max_i c_i^{\max}/c_i^{\min}$.

We construct the following dual solution.

Let $A = \log\left(1 + \frac{4d\Delta}{\epsilon}\right)$. We set

$$\begin{aligned} \bar{y}^\tau &= y^\tau / A, \\ \bar{z}^\tau &= z^\tau / A, \\ \bar{r}_i^\tau &= w_i \cdot \left(1 - \frac{1}{A} \log\left(1 + \frac{4dc_i^{\max} \cdot x_i^{\tau-1}}{\epsilon}\right)\right) \end{aligned}$$

(which is useful to write as)

$$= w_i \cdot \left(1 - \frac{1}{A} \left(\log\left(x_i^{\tau-1} + \frac{\epsilon}{4d \cdot c_i^{\max}}\right) - \log\left(\frac{\epsilon}{4d \cdot c_i^{\max}}\right)\right)\right). \quad (3.5)$$

We start by showing that our dual is feasible.

Lemma 3.2. *The variables $(\bar{y}, \bar{z}, \bar{r})$ are a feasible dual solution to $(\bar{\mathcal{D}})$.*

Proof. For $\tau \in C$ we have

$$\bar{r}_i^\tau - \bar{r}_i^{\tau+1} = \frac{w_i}{A} \cdot \log \left(\frac{x_i^\tau + \frac{\epsilon}{4d \cdot c_i^{\max}}}{x_i^{\tau-1} + \frac{\epsilon}{4d \cdot c_i^{\max}}} \right) \geq \frac{w_i}{A} \log \left(\frac{x_i^\tau + \frac{\epsilon}{4d^\tau \cdot c_i^\tau}}{x_i^{\tau-1} + \frac{\epsilon}{4d^\tau \cdot c_i^\tau}} \right) = c_i^\tau \bar{y}^\tau.$$

The inequality follows since for any i and $\tau \in C$, we have $x_i^\tau \geq x_i^{\tau-1}$ and therefore, for any $a \geq b$ the inequality $\frac{x_i^\tau + a}{x_i^{\tau-1} + a} \leq \frac{x_i^\tau + b}{x_i^{\tau-1} + b}$ holds. Similarly, for $\tau \in P$ we have

$$\bar{r}_i^\tau - \bar{r}_i^{\tau+1} = \frac{w_i}{A} \cdot \log \left(\frac{x_i^\tau + \frac{\epsilon}{4d \cdot c_i^{\max}}}{x_i^{\tau-1} + \frac{\epsilon}{4d \cdot c_i^{\max}}} \right) \geq \frac{w_i}{A} \log \left(\frac{x_i^\tau}{x_i^{\tau-1}} \right) = -p_i^\tau \bar{z}^\tau.$$

The inequality follows since for any $\tau \in P$, we have $x_i^\tau \leq x_i^{\tau-1}$ and therefore, for any $a \geq 0$ the inequality $\frac{x_i^\tau}{x_i^{\tau-1}} \leq \frac{x_i^\tau + a}{x_i^{\tau-1} + a}$ holds. Finally,

$$0 \leq \bar{r}_i^\tau = w_i \cdot \left(1 - \frac{1}{A} \log \left(1 + \frac{4dc_i^{\max} \cdot x_i^{\tau-1}}{\epsilon} \right) \right) \leq w_i.$$

To see this, note that for all $\tau \in C$ and all $i \in [n]$ we have $0 \leq x_i^\tau \leq \frac{1}{c_i^\tau}$, otherwise the constraint in round τ is already satisfied upon arrival. Hence

$$0 \leq \log \left(1 + \frac{4dc_i^{\max} \cdot x_i^{\tau-1}}{\epsilon} \right) \leq \log \left(1 + \frac{4dc_i^{\max}}{\epsilon c_i^{\tau-1}} \right) \leq \log \left(1 + \frac{4d \cdot \Delta}{\epsilon} \right) = A. \quad \square$$

Next, to relate the movement of the algorithm to the dual objective, we need the following pair of crucial lemmas. First we bound the recourse in terms of just the y terms.

Lemma 3.3 (Bounding movement in terms of y). *The upward movement cost of the algorithm at any time $t \in C$ can be bounded as*

$$\sum_i w_i (x_i^t - x_i^{t-1})_+ \leq \left(1 + \frac{\epsilon}{4} \right) y^t.$$

Proof. From $1+x \leq e^x$ with $x = \log(b/a)$, we get the following convenient inequality (sometimes referred to as the ‘‘Poor man’s Pinkser’’ inequality). For all $a, b \geq 0$,

$$a - b \leq a \log \left(\frac{a}{b} \right).$$

Using this fact, we deduce

$$\begin{aligned} \sum_i w_i (x_i^t - x_i^{t-1})_+ &= \sum_{i | c_i^t \neq 0} w_i (\hat{x}_i^t - \hat{x}_i^{t-1})_+ \\ &\leq \sum_{i | x_i^t > x_i^{t-1}} w_i \cdot \hat{x}_i^t \cdot \log \left(\frac{\hat{x}_i^t}{\hat{x}_i^{t-1}} \right) \\ &\stackrel{(3.2)}{=} \sum_{i | x_i^t > x_i^{t-1}} \left(x_i^t + \frac{\epsilon}{4d^t \cdot c_i^t} \right) c_i^t y^t \\ &\leq \frac{\epsilon}{4} y^t + y^t \cdot \sum_i c_i^t x_i^t \\ &\stackrel{(3.1)}{=} \left(1 + \frac{\epsilon}{4} \right) y^t. \quad \square \end{aligned}$$

To bound the recourse as a function of the full dual objective, we also need argue that subtracting the z terms does not lose too much.

Lemma 3.4 (Bounding z in terms of y). *The dual vectors y and z satisfy*

$$\left(1 + \frac{\epsilon}{4}\right) \sum_{t \in C} y^t - (1 + \epsilon) \sum_{t \in P} z^t \geq 0 \quad (3.6)$$

Proof. By non-negativity of KL we have

$$\begin{aligned} \forall t \in C \quad 0 &\leq \sum_{i | c_i^t \neq 0} w_i \left[\hat{x}_i^t \log \left(\frac{\hat{x}_i^t}{\hat{x}_i^{t-1}} \right) - \hat{x}_i^t + \hat{x}_i^{t-1} \right] = \sum_{i | c_i^t \neq 0} w_i \left[\hat{x}_i^t \log \left(\frac{\hat{x}_i^t}{\hat{x}_i^{t-1}} \right) - x_i^t + x_i^{t-1} \right] \\ \forall t \in P \quad 0 &\leq \sum_{i | p_i^t \neq 0} w_i \left[x_i^t \log \left(\frac{x_i^t}{x_i^{t-1}} \right) - x_i^t + x_i^{t-1} \right]. \end{aligned}$$

Summing these over all times t we get:

$$\begin{aligned} 0 &\leq \sum_{t \in C} \sum_{i | c_i^t \neq 0} w_i \left[\hat{x}_i^t \cdot \log \left(\frac{\hat{x}_i^t}{\hat{x}_i^{t-1}} \right) - x_i^t + x_i^{t-1} \right] + \sum_{t \in P} \sum_{i | p_i^t \neq 0} w_i \left[x_i^t \log \left(\frac{x_i^t}{x_i^{t-1}} \right) - x_i^t + x_i^{t-1} \right] \\ &= \sum_{t \in C} \sum_{i | c_i^t \neq 0} w_i \hat{x}_i^t \cdot \log \left(\frac{\hat{x}_i^t}{\hat{x}_i^{t-1}} \right) + \sum_{t \in P} \sum_{i | p_i^t \neq 0} w_i \cdot x_i^t \log \left(\frac{x_i^t}{x_i^{t-1}} \right) - \sum_{i=1}^n w_i (x_i^T + x_i^0) \quad (3.7) \\ &\stackrel{(3.2) \ \& \ (3.4)}{=} \sum_{t \in C} y^t \sum_{i | c_i^t \neq 0} \left(c_i^t x_i^t + \frac{\epsilon}{4d^t} \right) - \sum_{t \in P} z^t \sum_{i | p_i^t \neq 0} p_i^t x_i^t - \sum_{i=1}^n w_i x_i^T \\ &\stackrel{(3.1) \ \& \ (3.3)}{\leq} \left(1 + \frac{\epsilon}{4}\right) \sum_{t \in C} y^t - (1 + \epsilon) \sum_{t \in P} z^t. \quad (3.8) \end{aligned}$$

Equation (3.7) follows from (2.1). Inequality (3.8) uses the fact that there are at most d^t indices i for which $c_i^t \neq 0$. $\square$

By a careful combination of Lemma 3.3 and Lemma 3.4, we get the following bound on the recourse in terms of the dual objective. We write the lemma in terms of a more general $\tilde{y}^\tau$ since we will need it later. For now, the reader should set $\alpha = 0$, i.e. $\tilde{y}^\tau = y^\tau$.

Lemma 3.5. *Let $\tilde{y}^\tau$ be such that $\sum_{\tau \in C} \tilde{y}^\tau \geq (1 - \alpha\epsilon) \sum_{\tau \in C} y^\tau$ for $\alpha \in [0, 1/10]$. Then,*

$$\sum_{t=1}^T \sum_i w_i (x_i^t - x_i^{t-1})_+ = O\left(\frac{1}{\epsilon}\right) \left[\sum_{t \in C} \tilde{y}^t - \sum_{t \in P} z^t \right] \quad (3.9)$$

Proof. Observe that

$$\begin{aligned} \left(1 + \frac{\epsilon}{4}\right) \sum_{t \in C} y^t &\leq \frac{1 + \frac{\epsilon}{4}}{1 - \alpha\epsilon} \cdot \sum_{t \in C} \tilde{y}^t \\ &\leq \left(1 + \frac{\epsilon}{4}\right) \cdot (1 + 2\alpha\epsilon) \cdot \sum_{t \in C} \tilde{y}^t \\ &\leq \left(1 + \frac{\epsilon}{4} + 2\alpha\epsilon + \frac{2\alpha\epsilon}{4}\right) \cdot \sum_{t \in C} \tilde{y}^t \\ &\leq \left(1 + \frac{\epsilon}{2}\right) \cdot \sum_{t \in C} \tilde{y}^t, \quad (3.10) \end{aligned}$$

where the first inequality follows from our assumption, and the second since $1/(1-x) \leq 1+2x$ for $x \in [0, \frac{1}{2}]$ and $\epsilon \leq 1$.

Summing Lemma 3.3 over all times t and adding $\frac{2+\epsilon}{\epsilon}$ of Lemma 3.4, we get:

$$\begin{aligned} \sum_i w_i (x_i^t - x_i^{t-1})_+ &\leq \sum_{t \in C} \left(1 + \frac{\epsilon}{4}\right) y^t + \frac{2+\epsilon}{\epsilon} \left[\sum_{t \in C} \left(1 + \frac{\epsilon}{4}\right) y^t - \sum_{t \in P} (1+\epsilon) z^t \right] \\ &\leq \sum_{t \in C} \left(1 + \frac{\epsilon}{2}\right) \tilde{y}^t + \frac{2+\epsilon}{\epsilon} \left[\sum_{t \in C} \left(1 + \frac{\epsilon}{2}\right) \tilde{y}^t - \sum_{t \in P} (1+\epsilon) z^t \right] \\ &= \frac{(2+\epsilon)(1+\epsilon)}{\epsilon} \left[\sum_{t \in C} \tilde{y}^t - \sum_{t \in P} z^t \right] \\ &= O\left(\frac{1}{\epsilon}\right) \left[\sum_{t \in C} \tilde{y}^t - \sum_{t \in P} z^t \right], \end{aligned}$$

where the second inequality is from (3.10) above. $\square$

Putting things together, we can conclude with the theorem of this section.

Proof of Theorem 3.1. By Lemma 3.5 we can bound the total movement of the algorithm by

$$\sum_{t=1}^T \sum_i w_i (x_i^t - x_i^{t-1})_+ = O\left(\frac{1}{\epsilon}\right) \cdot \left[\sum_{t \in C} y^t - \sum_{t \in P} z^t \right] = O\left(\frac{\log\left(\frac{d \cdot \Delta}{\epsilon}\right)}{\epsilon}\right) \cdot \left[\sum_{t \in C} \tilde{y}^t - \sum_{t \in P} \tilde{z}^t \right].$$

Since $(\bar{y}, \bar{z}, \bar{r})$ is a feasible dual solution, the theorem follows from weak duality. $\square$

3.4 Removing the Δ : An $O(1/\epsilon \log(d/\epsilon))$ competitive bound

We move to proving the more refined bound. The idea is once again to fit a dual to $\bar{\mathcal{D}}$, but this time we need to construct our dual solution more delicately. For one, we need to overcome the following significant barrier: the dual y can be computed monotonically online, however one can show that no monotone dual solution can avoid a dependence on Δ , see [BN09a]. Nevertheless, we show that it is possible to *decrease* some coordinates of the dual carefully, to achieve the following bound.

Theorem 3.6 (Δ -free Bound). *The total recourse of the algorithm is at most $O(1/\epsilon \cdot \log(d/\epsilon))$ times the objective of $(\bar{\mathcal{D}})$.*

We will construct a vector $\tilde{y}$ with the following properties.

Lemma 3.7. *Given y , there is a vector $\tilde{y}$ such that*

$$\forall 1 \leq s \leq t \leq T, \forall i \quad \sum_{\substack{\tau \in C: \\ \tau \in [s, t]}} c_i^\tau \tilde{y}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, t]}} p_i^\tau z^\tau \leq w_i \cdot \log\left(1 + \frac{40d^2}{\epsilon^2}\right) \quad (3.11)$$

$$\text{and} \quad \sum_{\tau \in C} \tilde{y}^\tau \geq \left(1 - \frac{\epsilon}{10}\right) \sum_{\tau \in C} y^\tau. \quad (3.12)$$

Once we have this, we can define our dual solution as follows. Let $A = \log\left(1 + \frac{40d^2}{\epsilon^2}\right)$. We set,

$$\begin{aligned} \forall s \in C \quad & \bar{y}^s = \tilde{y}^s / A, \\ \forall s \in P \quad & \bar{z}^s = z^s / A, \end{aligned}$$

$$\bar{r}_i^{T+1} = 0,$$

$$\forall s \in [T] \quad \bar{r}_i^s = \frac{1}{A} \cdot \max \left(0, \max_{t \geq s} \left\{ \sum_{\substack{\tau \in C: \\ \tau \in [s, t]}} c_i^\tau \bar{y}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, t]}} p_i^\tau \bar{z}^\tau \right\} \right).$$

Before proving Lemma 3.7, let us understand why this dual solution is feasible and implies Theorem 3.6.

Lemma 3.8. *The variables $(\bar{y}, \bar{z}, \bar{r})$ are a feasible dual solution to $(\bar{\mathcal{D}})$.*

Proof. We start by proving the feasibility of the dual solution. The constraints $\bar{r}_i^s \geq 0$ hold by construction. For a sequence $a_1, a_2, \dots, a_T$ of positive and negative numbers it holds that for any $s \in [T]$:

$$\begin{aligned} \max \left(0, \max_{t | s \leq t \leq T} \left\{ \sum_{\tau \in [s, t]} a_\tau \right\} \right) &\geq \max \left(a_s, a_s + \max_{t | s+1 \leq t \leq T} \left\{ \sum_{\tau \in [s+1, t]} a_\tau \right\} \right) \\ &= a_s + \max \left(0, \max_{t | s+1 \leq t \leq T} \left\{ \sum_{\tau \in [s+1, t]} a_\tau \right\} \right) \end{aligned}$$

By this observation and plugging $a_t = c_i^\tau \bar{y}^\tau$, if $t \in C$ or $a_t = -p_i^\tau \bar{z}^\tau$ for $t \in P$, we immediately get that $\bar{r}_i^s \geq c_i^\tau \bar{y}^\tau + \bar{r}_i^{s+1}$, or similarly $\bar{r}_i^s \geq -p_i^\tau \bar{z}^\tau + \bar{r}_i^{s+1}$.

Finally, by inequality (3.11) of Lemma 3.7 we get:

$$\bar{r}_i^s = \frac{1}{A} \cdot \max \left(0, \max_{t \geq s} \left\{ \sum_{\substack{\tau \in C: \\ \tau \in [s, t]}} c_i^\tau \bar{y}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, t]}} p_i^\tau \bar{z}^\tau \right\} \right) \leq \frac{1}{A} \cdot w_i \cdot \log \left(1 + \frac{40d^2}{\epsilon^2} \right) \leq w_i,$$

and the dual solution is feasible. $\square$

Our main theorem now follows by putting everything together.

Proof of Theorem 3.6. By (3.12) of Lemma 3.7 we have $\sum_{\tau \in C} \bar{y}^\tau \geq (1 - \frac{\epsilon}{10}) \sum_{\tau \in C} y^\tau$, so we can reuse Lemma 3.5 to bound the total recourse as

$$\sum_{t=1}^T \sum_i w_i (x_i^t - x_i^{t-1})_+ = O\left(\frac{1}{\epsilon}\right) \cdot \left[\sum_{t \in C} \bar{y}^t - \sum_{t \in P} \bar{z}^t \right] = O\left(\frac{\log(\frac{d}{\epsilon})}{\epsilon}\right) \cdot \left[\sum_{t \in C} y^t - \sum_{t \in P} z^t \right].$$

Since $(\bar{y}, \bar{z}, \bar{r})$ is a feasible dual solution, the theorem follows from weak duality. $\square$

The remainder of the section is devoted to proving Lemma 3.7. First we need an auxiliary lemma.

Lemma 3.9. *For any s and t such that $1 \leq s \leq t \leq T$, any $i \in [n]$, and any $S \subseteq C \cap [s, t]$, define $c_i^{\max}(S) = \max_{\tau \in S} \{c_i^\tau\}$. Then*

$$\frac{1}{w_i} \left(\sum_{\substack{\tau \in S: \\ \tau \in [s, t]}} c_i^\tau y^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, t]}} p_i^\tau z^\tau \right) \leq \log \left(1 + \frac{4d \cdot c_i^{\max}(S)}{\epsilon} \cdot x_i^t \right).$$

Proof. We have that

$$\frac{1}{w_i} \left(\sum_{\substack{\tau \in S: \\ \tau \in [s, t]}} c_i^\tau y^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, t]}} p_i^\tau z^\tau \right) = \sum_{\substack{\tau \in S: \\ \tau \in [s, t], c_i^\tau \neq 0}} \log \left(\frac{x_i^\tau + \frac{\epsilon}{4d^t \cdot c_i^\tau}}{x_i^{\tau-1} + \frac{\epsilon}{4d^t \cdot c_i^\tau}} \right) + \sum_{\substack{\tau \in P: \\ \tau \in [s, t]}} \log \left(\frac{x_i^\tau}{x_i^{\tau-1}} \right) \quad (3.13)$$

$$\begin{aligned} &\leq \sum_{\tau=s}^t \log \left(\frac{x_i^\tau + \frac{\epsilon}{4d \cdot c_i^{\max}(S)}}{x_i^{\tau-1} + \frac{\epsilon}{4d \cdot c_i^{\max}(S)}} \right) \\ &= \log \left(\frac{x_i^t + \frac{\epsilon}{4d \cdot c_i^{\max}(S)}}{x_i^{s-1} + \frac{\epsilon}{4d \cdot c_i^{\max}(S)}} \right) \\ &\leq \log \left(1 + \frac{4d \cdot c_i^{\max}(S)}{\epsilon} \cdot x_i^t \right). \end{aligned} \quad (3.14)$$

Equality (3.13) follows from (3.2) and (3.4). To see why inequality (3.14) follows, note that for any i and $\tau \in C$, it holds that $x_i^\tau \geq x_i^{\tau-1}$ and therefore, for any $a \geq b$ we have $\frac{x_i^\tau + a}{x_i^{\tau-1} + a} \leq \frac{x_i^\tau + b}{x_i^{\tau-1} + b}$. For any $\tau \in P$, $x_i^\tau \leq x_i^{\tau-1}$ and therefore, for any $a \geq 0$, we get $\frac{x_i^\tau}{x_i^{\tau-1}} \leq \frac{x_i^\tau + a}{x_i^{\tau-1} + a}$. Finally, for any $\tau \in (C \cap [s, t]) \setminus S$, we have that $x_i^\tau \geq x_i^{\tau-1}$ and therefore the corresponding summand is nonnegative. $\square$

With this, we are finally ready to prove the lemma.

Proof of Lemma 3.7. We obtain $\tilde{y}$ by inductively *decreasing* coordinates of the solution y produced by the algorithm. We prove by induction on ℓ that inequalities (3.11) and (3.12) hold until time ℓ , i.e.,

$$\forall 1 \leq s \leq t \leq \ell, \forall i \quad \sum_{\substack{\tau \in C: \\ \tau \in [s, t]}} c_i^\tau \tilde{y}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, t]}} p_i^\tau z^\tau \leq w_i \cdot \log \left(1 + \frac{40d^2}{\epsilon^2} \right) \quad (3.15)$$

$$\text{and} \quad \sum_{\substack{\tau \in C \\ \tau \leq \ell}} \tilde{y}^\tau \geq \left(1 - \frac{\epsilon}{10} \right) \sum_{\substack{\tau \in C \\ \tau \leq \ell}} y^\tau. \quad (3.16)$$

The base of the induction is $\ell = 0$ which holds trivially. Now suppose by induction that we have an assignment of $\tilde{y}^\tau$ for all $\tau \leq \ell - 1$ such that inequalities (3.15) and (3.16) hold until time $\ell - 1$ (and $\tilde{y}^\tau = 0$ for $\tau \geq \ell$). We show how to create an updated assignment $\tilde{y}_{new}^\tau$ such that these inequalities also hold for time ℓ .

If time $\ell \in P$, the inequalities hold already and there is no need to change $\tilde{y}^\tau$. Therefore, assume that $\ell \in C$. We construct $\tilde{y}_{new}$ as follows.

Algorithm 1 Constructing $\tilde{y}_{new}$

```

1: for  $i = 1, \dots, n$  do
2:   Initialize
       $R_i \leftarrow \left\{ \tau \in [\ell - 1] \mid c_i^\tau \geq \frac{10 \cdot d^\ell \cdot c_i^\ell}{\epsilon}, \tilde{y}^\tau > 0 \right\},$            // candidates for decrease
       $B \leftarrow c_i^\ell \cdot y^\ell.$                                            // budget
3:   while  $B > 0$  and  $R_i \neq \emptyset$  do
4:     Let  $\tau^*$  be the latest time in  $R_i$ .
5:     Update
       $B \leftarrow B - c_i^{\tau^*} \Delta \tilde{y}_i^{\tau^*}.$ 
6:     for every  $\tau \in [\ell - 1]$  do
7:       Update
       $\Delta \tilde{y}_i^\tau \leftarrow \min\{\tilde{y}^\tau, B/c_i^{\tau^*}\}.$ 
8:       If  $\Delta \tilde{y}_i^\tau = \tilde{y}^\tau$ , then also update  $R_i \leftarrow R_i \setminus \{\tau^*\}.$ 
9: Finally, set
       $\tilde{y}_{new}^\ell \leftarrow y^\ell,$ 
      and for all  $t \in [\ell - 1]$ 
       $\tilde{y}_{new}^\tau \leftarrow \tilde{y}^\tau - \max_{i \mid c_i^t \neq 0} \{\Delta \tilde{y}_i^\tau\}.$ 

```

By construction, at the end of while loop which always terminates, either $R_i = \emptyset$, or the total budget spent is $\Delta B = c_i^\ell y_i^\ell = \sum_{\tau \leq \ell-1} c_i^\tau \Delta \tilde{y}_i^\tau$. Throughout the execution, we always maintain that

$$c_i^\ell y_i^\ell - \sum_{\tau \leq \ell-1} c_i^\tau \Delta \tilde{y}_i^\tau \geq 0$$

and for all $\tau \in [\ell - 1]$, that $\Delta \tilde{y}_i^\tau \leq \tilde{y}^\tau$.

Since $c_i^\ell y_i^\ell - \sum_{\tau} c_i^\tau \Delta \tilde{y}_i^\tau \geq 0$ and for each $\tau \in R_i$, by definition $c_i^\tau \geq 10 \cdot d^\ell c_i^\ell / \epsilon$, we get that

$$\sum_{\tau \leq \ell-1} \Delta \tilde{y}_i^\tau \leq \frac{\epsilon}{10 \cdot d^\ell} \cdot \sum_{\tau \leq \ell-1} \frac{c_i^\tau \cdot \Delta \tilde{y}_i^\tau}{c_i^t} \leq \frac{\epsilon}{10 \cdot d^\ell} \cdot y^\ell. \quad (3.17)$$

Thus, even after the decreases of Algorithm 1, the total mass we add to $\tilde{y}$ is at least

$$\begin{aligned}
(\tilde{y}_{new}^\ell - \tilde{y}^\ell) + \sum_{\tau \leq \ell-1} (\tilde{y}_{new}^\tau - \tilde{y}^\tau) &= y^\ell - \sum_{\tau \leq \ell-1} \max_{i \mid c_i^t \neq 0} \{\Delta \tilde{y}_i^\tau\} \\
&\geq y^\ell - \sum_{i \mid c_i^\ell \neq 0} \sum_{\tau \leq \ell-1} \Delta \tilde{y}_i^\tau \\
&\geq y^\ell - \frac{\epsilon}{10 \cdot d^\ell} \sum_{i \mid c_i^\ell \neq 0} y^\ell \\
&= \left(1 - \frac{\epsilon}{10}\right) y^\ell.
\end{aligned}$$

Above, the first equality follows from the fact that we defined $\tilde{y}^\ell$ to be 0. The second inequality came from (3.17), and the third from the fact that at most d^ℓ coordinates of c^ℓ are nonzero.

Combining this with the inductive hypothesis (3.16),

$$\sum_{\tau \leq \ell} \tilde{y}_{new}^\tau = (\tilde{y}_{new}^\ell - \tilde{y}^\ell) + \sum_{\tau \leq \ell-1} (\tilde{y}_{new}^\tau - \tilde{y}^\tau) + \sum_{\tau \leq \ell-1} \tilde{y}^\tau \geq \left(1 - \frac{\epsilon}{10}\right) \sum_{\substack{\tau \in C \\ \tau \leq \ell}} y^\ell,$$

and hence inequality (3.16) is preserved.

It remains to prove that for any $i \in [n]$ and $s \leq t = \ell$, inequality (3.15) holds. Fix an index i and let $\tau_i^{\min}$ be the earliest time τ for which $\Delta \tilde{y}_i^\tau > 0$. We split into cases.

Case 1: $c_i^\ell = 0$. In this case

$$\sum_{\substack{\tau \in C: \\ \tau \in [s, \ell]}} c_i^\tau \tilde{y}_{new}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, \ell]}} p_i^\tau z^\tau = \sum_{\substack{\tau \in C: \\ \tau \in [s, \ell-1]}} c_i^\tau \tilde{y}_{new}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, \ell-1]}} p_i^\tau z^\tau \leq \sum_{\substack{\tau \in C: \\ \tau \in [s, \ell-1]}} c_i^\tau \tilde{y}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, \ell-1]}} p_i^\tau z^\tau.$$

and inequality (3.15) holds by the induction hypothesis. Assume in the remaining cases that $c_i^\ell > 0$.

Case 2: $R_i \neq \emptyset$ at the end of the loop and $s \leq \tau_i^{\min}$. In this case, the budget B was completely spent, i.e.

$$c_i^\ell y^\ell = \sum_{\substack{\tau \in C \\ \tau \in [\tau_i^{\min}, \ell-1]}} c_i^\tau \Delta \tilde{y}_i^\tau,$$

which means that

$$\sum_{\substack{\tau \in C: \\ \tau \in [\tau_i^{\min}, \ell]}} c_i^\tau \tilde{y}_{new}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [\tau_i^{\min}, \ell]}} p_i^\tau z^\tau \leq \sum_{\substack{\tau \in C: \\ \tau \in [\tau_i^{\min}, \ell-1]}} c_i^\tau \tilde{y}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [\tau_i^{\min}, \ell-1]}} p_i^\tau z^\tau.$$

Then, for any $s \leq \tau_i^{\min}$,

$$\sum_{\substack{\tau \in C: \\ \tau \in [s, \ell]}} c_i^\tau \tilde{y}_{new}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, \ell]}} p_i^\tau z^\tau \leq \sum_{\substack{\tau \in C: \\ \tau \in [s, \ell-1]}} c_i^\tau \tilde{y}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, \ell-1]}} p_i^\tau z^\tau,$$

and again, inequality (3.15) holds by the induction hypothesis.

Case 3: $R_i = \emptyset$ at the end of the loop or $s > \tau_i^{\min}$. In this case $[s, \ell] \cap R_i = \emptyset$ and therefore, for each $\tau \in [s, \ell]$ either $\tilde{y}_{new}^\tau = 0$, or $c_i^\tau \leq 10 \cdot d^\ell c_i^\ell / \epsilon$. Define

$$S = C \cap \{\tau \in [s, \ell], \tilde{y}^\tau > 0\},$$

$$c_i^{\max}(S) = \max_{\tau \in S} \{c_i^\tau\} \leq 10 \cdot d^\ell c_i^\ell / \epsilon.$$

Applying Lemma 3.9 with the set S above, we get that:

$$\begin{aligned} \frac{1}{w_i} \left(\sum_{\substack{\tau \in C \\ \tau \in [s, t]}} c_i^\tau \tilde{y}^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, t]}} p_i^\tau z^\tau \right) &= \frac{1}{w_i} \left(\sum_{\tau \in S} c_i^\tau y^\tau - \sum_{\substack{\tau \in P: \\ \tau \in [s, t]}} p_i^\tau z^\tau \right) \\ &\leq \log \left(1 + \frac{4d \cdot c_i^{\max}(S)}{\epsilon} \cdot x_i^t \right) \\ &\leq \log \left(1 + \frac{4d}{\epsilon} \cdot \frac{10 \cdot d^\ell c_i^\ell}{\epsilon} \cdot \frac{1}{c_i^\ell} \right) \\ &\leq \log \left(1 + \frac{40d^2}{\epsilon^2} \right). \end{aligned}$$

The second inequality follows from the definition of $c_i^{\max}(S)$. The last inequality follows from the fact that $x_i^\ell \leq 1/c_i^\ell$, since otherwise the constraint at time ℓ is already satisfied at time $\ell - 1$. This concludes the proof. $\square$

4 Rounding

In this section we study several applications of the chasing positive bodies problem. For each of these problems we show that a suitable fractional version of it fits into our framework. We then show how to round the fractional solution to an integral solution.

4.1 Dynamic Set Cover

In the set cover problem we are given a universe of n elements U and a collection of m sets $S_1, \dots, S_m \subseteq U$, where each set S_i has a cost $c(S_i) \geq 0$. The goal is to find a minimum cost collection of sets that covers all elements.

In the dynamic setting, at each time-step $t \in [T]$, an element gets inserted into/deleted from the universe U . Let U^t denote the state of U at time t ,⁴ and let OPT^t denote the cost of the optimal *fractional* set cover for U^t . We wish to maintain a set cover $\mathcal{S}^t$ of U^t whose cost $c(\mathcal{S}^t) := \sum_{S \in \mathcal{S}^t} c(S)$ is always within a factor $\beta \geq 1$ of OPT^t , such that the total recourse, $\sum_{t \in [T]} |\mathcal{S}^t \oplus \mathcal{S}^{t-1}|$, is as small as possible. The fractional version of the dynamic set cover problem fits naturally into our framework

$$K_t := \left\{ x^t \in \mathbb{R}_+^m \left| \begin{array}{l} \sum_{\substack{i \in [m]: \\ u \in S_i}} x_i^t \geq 1 \quad \forall u \in U^t, \\ \sum_{i \in [m]} c(S_i) \cdot x_i^t \leq \beta \cdot \text{OPT}^t \end{array} \right. \right\}.$$

Theorem 4.1 (Set Cover Rounding). *For any fractional solution x^t to dynamic set cover, we have:*

(1) *A deterministic dynamic rounding scheme such that*

$$\begin{aligned} \sum_{t \in [T]} |\mathcal{S}^t \oplus \mathcal{S}^{t-1}| &\leq O(f) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1, \\ \sum_{S \in \mathcal{S}^t} c(S) &\leq O(f) \cdot \sum_{i \in [m]} c(S_i) \cdot x_i^t \quad \forall t \in [T]. \end{aligned}$$

Here, f is an upper bound on the maximum frequency of any element at any time, i.e.,

$$f \geq \max_{u \in U^t} |\{i \in [m] : u \in S_i\}| \quad \forall t \in [T].$$

(2) *For every $\alpha \geq 1$, a randomized dynamic rounding scheme such that*

$$\mathbb{E} \left[\sum_{t \in [T]} |\mathcal{S}^t \oplus \mathcal{S}^{t-1}| \right] \leq (\alpha \log n) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1 + O\left(\frac{T}{n^\alpha}\right), \quad (4.1)$$

$$\mathbb{E} \left[\sum_{S \in \mathcal{S}^t} c(S) \right] \leq O(\alpha \log n) \cdot \sum_{i \in [m]} c(S_i) \cdot x_i^t \quad \forall t \in [T]. \quad (4.2)$$

Here, $n \geq \max_t |U^t|$ is an upper bound on the number of elements in the universe at any time.

⁴We assume that each element in U^t is covered by at least one set in $\{S_1, \dots, S_m\}$. This ensures the existence of a valid set cover.

The proof, which we defer to Appendix B, is an adaptation of standard offline rounding techniques to the dynamic setting. Note that in this problem the row sparsity of the covering constraints of K_t is at most $f \geq \max_{u \in U^t} |\{i \in [m] : u \in S_i\}|$. Hence, by setting $\epsilon = 1$ in Theorem 1.1, we can maintain a fractional solution x^t that satisfies the covering constraints, at any time t has cost at most $O(\beta) \cdot \text{OPT}^t$, and has a $O(\log f)$ -competitive total recourse with respect to any offline algorithm maintaining a fractional solution of cost $\leq \beta \cdot \text{OPT}^t$. Also note that if we set α to be a constant, then the additive term in (4.1) becomes $1/\text{poly}(n)$ per time-step. Combining this with Theorem 4.1 we get:

Corollary 4.2. *For any $\beta \geq 1$ and any update-sequence σ consisting of T updates, there exist:*

- (1) *A deterministic dynamic algorithm that maintains a set cover $\mathcal{S}^t$ with cost $O(\beta \cdot f \cdot \text{OPT}^t)$ at all times, whose total recourse is at most $O(f \log f) \cdot \text{OPT}_{\text{REC}}^\beta(\sigma)$.*
- (2) *A randomized dynamic algorithm that maintains a set cover $\mathcal{S}^t$ with expected cost $O(\beta \cdot \log n \cdot \text{OPT}^t)$ at all times, whose expected total recourse is at most $O(\log f \cdot \log n) \cdot \text{OPT}_{\text{REC}}^\beta(\sigma) + \frac{T}{\text{poly}(n)}$.*

Here, $\text{OPT}_{\text{REC}}^\beta(\sigma)$ is the optimal total recourse for any offline algorithm which maintains a fractional set cover of cost at most $\beta \cdot \text{OPT}^t$ throughout the update-sequence σ .

We observe that we can easily extend our analysis to allow for other forms of updates, such as changing the cost of a set.

4.2 Dynamic Bipartite Matching

In the maximum bipartite matching problem, we are given a bipartite graph $G = (V, E)$ with $|V| = n$ nodes. A matching $M \subseteq E$ is subset of edges that do not share any common endpoint, and the goal is to find a matching M of maximum size. In the dynamic setting, at each time-step $t \in [T]$ an edge gets inserted into/deleted from the set E . Let E^t denote the state of E at time t , and let OPT^t denote the value of the maximum fractional matching at time t . We wish to maintain a matching M^t in $G^t = (V, E^t)$ whose size is always within a constant factor $\beta \leq 1$ of OPT^t , such that the total recourse, $\sum_{t \in [T]} |M^t \oplus M^{t-1}|$, is as small as possible. The fractional version of the dynamic maximum matching problem fits naturally into our framework with the following

$$K_t := \left\{ x^t \in \mathbb{R}_+^{\binom{|V|}{2}} \mid \begin{array}{ll} \sum_{e \in \partial(v)} x_e^t \leq 1 & \forall v \in V, \\ x_e^t = 0 & \forall e \in \binom{V}{2} \setminus E^t, \\ \sum_{e \in E^t} x_e^t \geq \beta \cdot \text{OPT}^t. \end{array} \right\}. \quad (4.3)$$

Note that the vector x^t has an entry for every *potential edge* (i.e., unordered pair of nodes) $e \in \binom{V}{2}$. If a potential edge e is currently not present in the graph, then we set $x_e^t = 0$. The notation $\partial(v)$ denotes the set of all potential edges with one endpoint in $v \in V$.

The main result in this section is summarized below. The proof of Theorem 4.3, which we defer to Appendix C, follows from an adaptation of a standard rounding technique [ACC⁺18] to our setting.

Theorem 4.3 (Maximum Matching Rounding). *For any fractional solution x^t to dynamic bipartite matching, and any $\delta \in (0, 1)$, $\alpha \geq 1$, there exists a randomized dynamic rounding scheme such that:*

$$\mathbb{E} \left[\sum_{t \in [T]} |M^t \oplus M^{t-1}| \right] \leq O \left(\frac{\alpha \log n}{\delta^3} \right) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1 + O \left(\frac{T}{\delta \cdot n^\alpha} \right). \quad (4.4)$$

$$\mathbb{E}[|M^t|] \geq (1 - \delta) \cdot \sum_{e \in E^t} x_e^t \quad \forall t \in [T]. \quad (4.5)$$

Note that in this problem the row-sparsity of the covering constraint in K_t is at most $\binom{n}{2} = \Theta(n^2)$. Thus, using the online algorithm from Theorem 1.1, we can maintain a fractional matching x^t that satisfies $\sum_{e \in E^t} x_e^t \geq (1 - \epsilon)\beta \cdot \text{OPT}^t$ at all times t , and has a $O((1/\epsilon) \cdot \log(n/\epsilon))$ -competitive total recourse with respect to any offline algorithm which maintains a fractional matching of value $\geq \beta \cdot \text{OPT}^t$. Setting $\delta = \epsilon$ and α to be some large constant, and combining this with Theorem 4.3, we now get:

Corollary 4.4. *For any $\beta, \epsilon \in (0, 1]$, and any update-sequence σ with T updates, there exists a randomized dynamic algorithm that maintains a matching M^t of expected size at least $(1 - \epsilon)\beta \cdot \text{OPT}^t$ at all times, with expected total recourse $O((1/\epsilon^4) \cdot \log n \cdot \log(n/\epsilon)) \cdot \text{OPT}_{\text{REC}}^\beta(\sigma) + \frac{T}{\epsilon \cdot \text{poly}(n)}$. Here, $\text{OPT}_{\text{REC}}^\beta(\sigma)$ is the optimal total recourse for any offline algorithm which maintains a fractional matching of value at least $\beta \cdot \text{OPT}^t$ throughout the update-sequence σ .*

We remark that it is straightforward to extend Theorem 4.3 and Corollary 4.4 to weighted bipartite graphs, at the cost of incurring an exponential in $(1/\epsilon)$ factor overhead in recourse, using a well-known reduction due to Bernstein et al. (see Theorem 3.1 in [BDL21]). We can also easily extend our analysis to also allow for vertex updates as well.

4.3 Dynamic Load Balancing on Unrelated Machines

In the load balancing problem, we are given a set M of m machines, and a set J of jobs. Each job $j \in J$ can only be processed by a nonempty subset $M(j) \subseteq M$ of machines. If we assign job j to machine $i \in M(j)$, then the machine incurs a load of p_{ij} . Our goal is to come up with a valid assignment $\psi : J \rightarrow M$ of jobs to machines so as to minimize the “makespan”, which is defined to be the maximum load on any machine and is given by $\text{Obj}(\psi) := \max_{i \in M} \left(\sum_{j \in J: \psi(j)=i} p_{ij} \right)$.

In the dynamic setting, at each time-step t a job gets inserted into/deleted from the set J . (Our analysis also works for other types of updates, such as changing the set $M(j)$ for a job j .) Let J^t denote the set of jobs at time t . Let $\text{OPT}_{\text{int}}^t$ denote the optimal *integral* makespan at time t . We wish to maintain a valid assignment $\psi^t : J^t \rightarrow M$ such that $\text{Obj}(\psi^t) \leq \beta \cdot \text{OPT}_{\text{int}}^t$ for some $\beta \geq 1$. We incur a “recourse” of one whenever a job gets reassigned from one machine to another, or whenever a job gets inserted into/deleted from the set J . Overloading the notation $\oplus$, which denotes the symmetric difference between two sets, we define the recourse incurred at time t to be $\psi^{t-1} \oplus \psi^t := |J^t \oplus J^{t-1}| + |\{j \in J^t \cap J^{t-1} : \psi^t(j) \neq \psi^{t-1}(j)\}|$. We wish to ensure that the total recourse, given by $\sum_{t \in [T]} \psi^t \oplus \psi^{t-1}$, is as small as possible. Let $\mathcal{J}$ denote the collection of all possible jobs that can ever be present, so that we have $J^t \subseteq \mathcal{J}$ for all t , and let $n := |\mathcal{J}|$. The fractional version of this dynamic problem fits into our framework, where the set K_t consists of all $x^t \in \mathbb{R}_+^{|\mathcal{J}| \times |M|}$ that satisfy the following constraints:

$$\sum_{i \in M} x_{ij}^t \geq 1 \quad \forall j \in J^t. \quad (4.6)$$

$$x_{ij}^t = 0 \quad \forall i \in M, j \in J^t \text{ with } p_{ij} > \text{OPT}_{\text{int}}^t. \quad (4.7)$$

$$x_{ij}^t = 0 \quad \forall i \in M, j \in \mathcal{J} \setminus J^t. \quad (4.8)$$

$$x_{ij}^t = 0 \quad \forall j \in \mathcal{J}, i \in M \setminus M(j). \quad (4.9)$$

$$\sum_{j \in J^t} p_{ij} \cdot x_{ij}^t \leq \beta \cdot \text{OPT}_{\text{int}}^t \quad \forall i \in M. \quad (4.10)$$

The main result in this section is summarized below. The proof of Theorem 4.5 follows almost

immediately from a recent result by Krishnaswamy et al. [KLS23]. Parts 1 and 2 of the theorem below follow from Appendix B and Section 4 of [KLS23], respectively.⁵

Theorem 4.5 (Load Balancing Rounding by [KLS23]). *Consider any fractional solution x^t to the dynamic load balancing problem on unrelated machines. Then we have:*

(1) *For any $\delta > 0$, a deterministic dynamic rounding scheme such that:*

$$\sum_{t \in [T]} \psi^t \oplus \psi^{t-1} \leq O\left(\frac{1}{\delta^4} \log \frac{1}{\delta} \cdot \log n\right) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1, \quad (4.11)$$

$$\text{Obj}(\psi^t) \leq (2 + \delta)\beta \cdot \text{OPT}_{\text{int}}^t \quad \forall t \in [T]. \quad (4.12)$$

(2) *A randomized dynamic rounding scheme such that:*

$$\mathbb{E}\left[\sum_{t \in [T]} \psi^t \oplus \psi^{t-1}\right] \leq O(1) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1 + \frac{T}{\text{poly}(n)}, \quad (4.13)$$

$$\mathbb{E}[\text{Obj}(\psi^t)] = O\left(\frac{\log \log n}{\log \log \log n}\right) \cdot \text{OPT}_{\text{int}}^t \quad \forall t \in [T]. \quad (4.14)$$

The row-sparsity of the covering constraints in K_t is $r := \max_{j \in \mathcal{J}} |M(j)|$. Thus, by Theorem 1.1, we can maintain a fractional solution x^t that satisfies the four constraints (4.6)-(4.9) as well as the constraint $\sum_{j \in \mathcal{J}} p_{ij} \cdot x_{ij}^t \leq (1 + \epsilon)\beta \cdot \text{OPT}_{\text{int}}^t$ at all times t . Furthermore this solution has total recourse that is $O((1/\epsilon) \log(r/\epsilon))$ -competitive with respect to any offline algorithm which maintains a feasible fractional solution to all five constraints (4.6)-(4.10). Setting $\delta = \epsilon$ and combining this with Theorem 4.5, we get:

Corollary 4.6. *For any $\beta \geq 1$, $\epsilon \in (0, 1]$, and any update-sequence σ with T updates, there exist:*

- (1) *A deterministic dynamic algorithm that maintains an assignment ψ^t with makespan at most $(2 + \epsilon)\beta \cdot \text{OPT}_{\text{int}}^t$ at all times, with total recourse $O((1/\epsilon^5) \log(1/\epsilon) \log n \log(r/\epsilon)) \cdot \text{OPT}_{\text{REC}}^\beta(\sigma)$.*
- (2) *A randomized dynamic algorithm that maintains an assignment ψ^t with expected makespan $O(\frac{\log \log n}{\log \log \log n}) \cdot \text{OPT}_{\text{int}}^t$ for all t , with expected total recourse $O(\log(r)) \cdot \text{OPT}_{\text{REC}}^\beta(\sigma) + \frac{T}{\text{poly}(n)}$.*

Here, $\text{OPT}_{\text{REC}}^\beta(\sigma)$ is the optimal total recourse for any offline algorithm which maintains a feasible fractional solution to all the five constraints (4.6)-(4.10) throughout the update-sequence σ , and $r := \max_{j \in \mathcal{J}} |M(j)|$ is the maximum number of machines that are willing to process any given job.

4.4 Dynamic Minimum Spanning Tree

In the minimum spanning tree (MST) problem, we are given a connected graph $G = (V, E)$ with $|V| = n$ nodes, and a cost $c_e \geq 0$ associated with each edge. Our goal is to compute a spanning tree $\mathcal{T} = (V, E_{\mathcal{T}})$ of G of minimum total cost $\sum_{e \in E_{\mathcal{T}}} c_e$.

In the dynamic setting, at each time-step $t \in [T]$ an edge gets inserted into/deleted from the set E , subject to the condition that the input graph $G = (V, E)$ remains connected. Let E^t denote the state of E at time t , and let OPT^t denote the value of the minimum cost spanning tree of $G^t = (V, E^t)$. We wish to maintain a spanning tree $\mathcal{T}^t = (V, E_{\mathcal{T}}^t)$ in G^t whose cost is always

⁵The final result on load balancing of [KLS23] handles only job arrivals because of the limitation of their online fractional algorithm. However, their rounding schemes work in the fully dynamic setting.

within a factor $\beta \geq 1$ of OPT^t , such that the total recourse, $\sum_{t \in [T]} |E_{\mathcal{T}}^t \oplus E_{\mathcal{T}}^{t-1}|$ is as small as possible. The fractional version of the dynamic MST problem fits naturally into our framework with the following

$$K_t := \left\{ x^t \in \mathbb{R}_+^{\binom{V}{2}} \mid \begin{array}{ll} \sum_{e \in \partial^t(S)} x_e^t \geq 1 & \forall \emptyset \subset S \subset V, \\ x_e^t = 0 & \forall e \in \binom{V}{2} \setminus E^t, \\ \sum_{e \in E^t} c_e \cdot x_e^t \leq \beta \cdot \text{OPT}^t. \end{array} \right\}. \quad (4.15)$$

Note that the vector x^t has an entry for every *potential edge* $e \in \binom{V}{2}$. If a potential edge e is currently not present in the graph, then we set $x_e^t = 0$. The notation $\partial^t(S)$ denotes the set of all edges in G^t crossing the cut S . We say that an $x^t \in \mathbb{R}_+^{\binom{V}{2}}$ is a *fractional MST* in G^t , with cost at most $\beta \cdot \text{OPT}^t$, iff $x^t \in K_t$ as per (4.15) above. The main result in this section is summarized in the theorem below. We defer the proof of Theorem 4.7 to Appendix D.

Theorem 4.7 (MST Rounding). *Consider any fractional solution x^t to the dynamic MST problem, and any $\delta > 0, \alpha \geq 1$. Then there exists a randomized dynamic rounding scheme which maintains a spanning tree $\mathcal{T}^t = (V, E_{\mathcal{T}}^t)$ of G^t such that:*

$$\mathbb{E} \left[\sum_{t \in [T]} |E_{\mathcal{T}}^t \oplus E_{\mathcal{T}}^{t-1}| \right] \leq O\left(\frac{\alpha \log n}{\delta^2}\right) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1 + \frac{T}{O(n^\alpha)}, \quad (4.16)$$

$$\sum_{e \in E_{\mathcal{T}}^t} c_e \leq (2 + \delta) \cdot \sum_{e \in E^t} c_e \cdot x_e^t \quad \forall t \in [T]. \quad (4.17)$$

In this problem, the row-sparsity of the covering constraints is at most $\binom{n}{2} = \Theta(n^2)$. Thus, by Theorem 1.1, we can maintain a fractional MST x^t in G^t that has cost $\sum_{e \in E^t} c_e \cdot x_e^t \leq (1 + \epsilon)\beta \cdot \text{OPT}^t$, and has a $O((1/\epsilon) \log(n/\epsilon))$ -competitive total recourse with respect to any offline algorithm which maintains a fractional MST of cost at most $\beta \cdot \text{OPT}^t$ at all times. Setting $\delta = \epsilon$ and α to be some large constant, and combining this with Theorem 4.7, we get:

Corollary 4.8. *For any $\beta \geq 1, \epsilon \in (0, 1]$, and any update-sequence σ with T updates, there exists a randomized dynamic algorithm that maintains a spanning tree of cost at most $(2 + \epsilon)\beta \cdot \text{OPT}^t$ at all times, with expected total recourse $O((1/\epsilon^3) \cdot \log n \cdot \log(n/\epsilon)) \cdot \text{OPT}_{\text{REC}}^\beta(\sigma) + \frac{T}{\text{poly}(n)}$. Here, $\text{OPT}_{\text{REC}}^\beta(\sigma)$ is the optimal total recourse for any offline algorithm which maintains a fractional MST of cost at most $\beta \cdot \text{OPT}^t$ throughout the update-sequence σ .*

We remark that we can easily extend our analysis to allow for other forms of updates, such as changing the cost of an edge that is being inserted, or vertex insertions/deletions.

A Reducing Positive Bodies to Positive Halfspaces

In this section we show that it is enough to assume that at each time t , each positive body K_t is defined by a single covering or packing constraint.

Suppose we have an algorithm $\mathcal{A}$ for chasing *halfspaces* that satisfies covering constraints fully and satisfies packing constraints up to ϵ : the algorithm produces x^t such that for all $t \in C$ we have $\langle c^t, x \rangle \geq 1$ and for all $t \in P$ we have $\langle p^t, x \rangle \leq 1 + \epsilon$.

The following is an algorithm $\mathcal{A}'$ for the that satisfies packing constraints up to violation δ . Given a body $K_t \neq \emptyset$, while there a constraint of K_t violated by at least $\delta/10$, feed that constraint to algorithm $\mathcal{A}$ run with parameter $\epsilon = \delta/20$. With every such fix, algorithm $\mathcal{A}$ moves at least $\Omega(\delta/w)$, where w is the maximal non-zero coefficient in the constraint. Since there exists

a finite-movement optimal solution that satisfies all the constraints, and the competitive ratio of algorithm $\mathcal{A}$ that chases half-spaces is also finite, we get that the number of such rounds is finite.

Hence, after finitely many rounds, all the constraints in K_t are satisfied up to $\delta/10$. By scaling up the solution by $(1 - \delta/10)^{-1}$ (in all times t), we may ensure that all covering constraints are fully satisfied, and packing constraints are violated by up to δ . Furthermore, if $\mathcal{A}$ is c -competitive for positive halfspace chasing, then the solution output by $\mathcal{A}$ is at most c that of the optimal algorithm that chases the bodies K_t . Hence algorithm $\mathcal{A}'$ is also $O(c)$ competitive for positive body chasing.

B Proof of Theorem 4.1

Theorem 4.1 (Set Cover Rounding). *For any fractional solution x^t to dynamic set cover, we have:*

(1) *A deterministic dynamic rounding scheme such that*

$$\begin{aligned} \sum_{t \in [T]} |\mathcal{S}^t \oplus \mathcal{S}^{t-1}| &\leq O(f) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1, \\ \sum_{S \in \mathcal{S}^t} c(S) &\leq O(f) \cdot \sum_{i \in [m]} c(S_i) \cdot x_i^t \quad \forall t \in [T]. \end{aligned}$$

Here, f is an upper bound on the maximum frequency of any element at any time, i.e.,

$$f \geq \max_{u \in U^t} |\{i \in [m] : u \in S_i\}| \quad \forall t \in [T].$$

(2) *For every $\alpha \geq 1$, a randomized dynamic rounding scheme such that*

$$\mathbb{E} \left[\sum_{t \in [T]} |\mathcal{S}^t \oplus \mathcal{S}^{t-1}| \right] \leq (\alpha \log n) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1 + O\left(\frac{T}{n^\alpha}\right), \quad (4.1)$$

$$\mathbb{E} \left[\sum_{S \in \mathcal{S}^t} c(S) \right] \leq O(\alpha \log n) \cdot \sum_{i \in [m]} c(S_i) \cdot x_i^t \quad \forall t \in [T]. \quad (4.2)$$

Here, $n \geq \max_t |U^t|$ is an upper bound on the number of elements in the universe at any time.

Proof. We prove each part separately.

Proof of (1): Initialize $\mathcal{S}^0 = \emptyset$. Subsequently, while handling the update at time t , initialize $\mathcal{S}^t = \mathcal{S}^{t-1}$. Next, whenever $x_i^t \geq 1/f$ for some $S_i \notin \mathcal{S}^t$, add S_i to $\mathcal{S}^t$. In contrast, whenever the LP-value $x_i^t \leq 1/2f$ for some $S_i \in \mathcal{S}^t$, remove S_i from $\mathcal{S}^t$.

Thus, we always have $\{S_i : x_i \geq 1/2f\} \supseteq \mathcal{S} \supseteq \{S_i : x_i \geq 1/f\}$. Since x is a valid fractional set cover, this ensures that $\mathcal{S}$ is a valid integral set cover with cost at most $2f \cdot \sum_{i=1}^m c(S_i) \cdot x_i^t$.

To bound the recourse, note that we insert/delete a set S_i in $\mathcal{S}$ (thereby incurring a recourse of one) only after its LP-value has changed by at least $1/2f$. In other words, the recourse of our rounding scheme is at most $2f$ times the recourse incurred by the underlying dynamic algorithm which maintains the fractional solution x .

Proof of (2): We maintain two collections of sets $\mathcal{R}^t$ and $\mathcal{B}^t$, and output as our solution $\mathcal{S}^t := \mathcal{R}^t \cup \mathcal{B}^t$. (The reader may think of $\mathcal{R}$ as being the *randomly sampled* sets, and $\mathcal{B}$ as the

backup sets.) For each $i \in [m]$, the set S_i has an exponential random variable χ_i with parameter $\lambda := \alpha \log n$ that is sampled once and fixed at the outset of the dynamic process. Throughout the sequence of updates, any every time-step we define $\mathcal{R}^t := \{S_i \in \mathcal{S} : x_i^t \geq \chi_i\}$. For each element $u \in U$, let $S(u)$ denote a minimum cost set in $\{S_1, \dots, S_m\}$ which contains u . Define $\mathcal{B}^t = \{S(u) \mid u \text{ uncovered by } \mathcal{R}^t\}$.

Clearly $\mathcal{R}^t \cup \mathcal{B}^t$ is a feasible set cover; it remains to analyze the approximation ratio and the recourse. We do so separately for $\mathcal{R}^t$ and $\mathcal{B}^t$.

To analyze $\mathcal{R}^t$, note that for all $i \in [m]$, we have

$$\Pr[S_i \in \mathcal{R}^t] = \Pr[x_i^t \geq \chi_i] = 1 - \exp(-x_i^t \cdot \alpha \log n) \leq x_i^t \cdot \alpha \log n.$$

Summing over $i \in [m]$, we bound the expected cost of $\mathcal{R}^t$ as $\mathbb{E}[c(\mathcal{R}^t)] \leq \sum_{i \in [m]} c(S_i) \cdot x_i^t \cdot \alpha \log n$.

For the recourse of $\mathcal{R}$, observe that due to the t^{th} update, the set S_i contributes a recourse of one iff χ_i lies in an interval of length $|x_i^t - x_i^{t-1}|$. As the exponential distribution is memoryless, this event occurs with probability at most $1 - \exp(-|x_i^t - x_i^{t-1}| \cdot \alpha \log n) \leq |x_i^t - x_i^{t-1}| \cdot \alpha \log n$. Summing over all $i \in [m]$, the expected recourse of $\mathcal{R}$ due to the t^{th} update is given by:

$$\mathbb{E}[|\mathcal{R}^t \oplus \mathcal{R}^{t-1}|] \leq \alpha \log n \cdot \|x^t - x^{t-1}\|_1.$$

Moving on to $\mathcal{B}^t$, consider any element $u \in U^t$. We have:

$$\begin{aligned} \Pr[u \text{ not covered by } \mathcal{R}^t] &= \prod_{i \in [m]: u \in S_i} \Pr[S_i \notin \mathcal{R}^t] = \prod_{i \in [m]: u \in S_i} e^{-x_i^t \cdot \alpha \log n} \\ &= \exp \left\{ -\alpha \log n \cdot \left(\sum_{i \in [m]: u \in S_i} x_i^t \right) \right\} \leq \frac{1}{n^\alpha}, \end{aligned}$$

where the last inequality holds since x_i^t is a feasible fractional set cover. Since $|U^t| \leq n$, it follows that the expected number of uncovered (with respect to $\mathcal{R}^t$) elements in U^t is at most $n \cdot 1/n^\alpha \leq 1/n^{\alpha-1}$. Each of these uncovered elements contributes one set to $\mathcal{B}^t$. Accordingly, the expected recourse of $\mathcal{B}$ for the t^{th} update is at most $\mathbb{E}[|\mathcal{B}^t \oplus \mathcal{B}^{t-1}|] \leq \mathbb{E}[|\mathcal{B}^t| + |\mathcal{B}^{t-1}|] = O(1/n^{\alpha-1})$.

To bound the expected cost of $\mathcal{B}^t$, note that each set $B \in \mathcal{B}^t$ is added because of an element $u \in U^t$ that is uncovered by $\mathcal{R}^t$. Since $B = S(u)$ is a cheapest set covering u and x_i^t is a feasible fractional set cover for U^t , we have $c(B) \leq \sum_{i \in [m]} c(S_i) \cdot x_i^t$. By the argument above, we now derive that:

$$\mathbb{E}[c(\mathcal{B}^t)] \leq \mathbb{E}\left[|\mathcal{B}^t| \cdot \sum_{i \in [m]} c(S_i) \cdot x_i^t\right] \leq \frac{1}{n^{\alpha-1}} \cdot \sum_{i \in [m]} c(S_i) \cdot x_i^t = O(\alpha \log n) \sum_{i \in [m]} c(S_i) \cdot x_i^t.$$

To summarize, the total expected cost of $\mathcal{S}^t = \mathcal{R}^t \cup \mathcal{B}^t$ is at most $O(\alpha \log n) \cdot \sum_{i \in [m]} c(S_i) \cdot x_i^t$, and the expected recourse per update is at most $\alpha \log n \cdot \|x^t - x^{t-1}\|_1 + O(1/\alpha)$. $\square$

C Proof of Theorem 4.3

Theorem 4.3 (Maximum Matching Rounding). *For any fractional solution x^t to dynamic bipartite matching, and any $\delta \in (0, 1)$, $\alpha \geq 1$, there exists a randomized dynamic rounding scheme such that:*

$$\mathbb{E}\left[\sum_{t \in [T]} |M^t \oplus M^{t-1}|\right] \leq O\left(\frac{\alpha \log n}{\delta^3}\right) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1 + O\left(\frac{T}{\delta \cdot n^\alpha}\right). \quad (4.4)$$

$$\mathbb{E}[|M^t|] \geq (1 - \delta) \cdot \sum_{e \in E^t} x_e^t \quad \forall t \in [T]. \quad (4.5)$$

Our rounding scheme consists of two steps. In the first, we maintain a subgraph H^t of G^t , which we call a *stabilizer*, such that the recourse of H^t is bounded with respect to the ℓ_1 movement of x^t , and H^t contains a matching whose size is very close to $\text{Obj}(x^t) := \sum_{e \in E} x_e^t$. In the second step, we maintain a $(1 - \delta)$ -approximate maximum matching in H^t whose recourse is small with respect to the number of updates to H^t .

In the first step, we show:

Lemma C.1. *We can maintain a subgraph $H = (V, E_H)$ of G^t such that:*

- (1) $\mathbb{E}[\mu(H^t)] \geq (1 - \delta) \cdot \text{Obj}(x^t)$ at all times t , where $\mu(H^t)$ is the maximum matching size in H^t .
- (2) $\mathbb{E}\left[\sum_t |E_H^{t-1} \oplus E_H^t|\right] \leq O\left(\frac{\alpha \log n}{\delta^2}\right) \cdot \sum_t \|x^t - x^{t-1}\|_1 + O\left(\frac{T}{n^\alpha}\right).$

We defer the proof of Lemma C.1 for a moment, while we show how to complete the proof of the theorem.

Lemma C.2. *We can maintain a $(1 - \delta)$ -approximate maximum matching M^t in H^t , with $O(1/\delta)$ absolute recourse per update in H^t .*

Proof. We maintain a matching M^t in H^t that does not admit any augmenting path of length less than $1 + \Theta(1/\delta)$. Such a matching M^t is a $(1 - \delta)$ -approximate maximum matching in H^t . If an update in H^t leads to the creation of an augmenting path of length less than $1 + \Theta(1/\delta)$ with respect to M^t , then we modify M^t by augmenting along that path. This implies an absolute recourse of $O(1/\delta)$ per update in H^t . $\square$

Theorem 4.3 now follows by combining Lemma C.2 and Lemma C.1.

Proof of Lemma C.1. We construct the stabilizer H^t as follows. Fix the parameter

$$\kappa := \lceil (100(\alpha + 4) \log n) / \delta^2 \rceil. \quad (C.1)$$

At preprocessing, for each $e = (u, v) \in \binom{V}{2}$ and each $i \in [\kappa]$, we draw a value $\chi_{e,i} \in [0, 1]$ uniformly and independently at random. Define the collection of multi-edges

$$F^t := \left\{ e_i : e \in \binom{V}{2}, i \in [\kappa], \text{ and } x_e > \chi_{e,i} \right\},$$

(note that F^t is always a subset of the edge set of G^t) and let z be the weighted multigraph that assigns each edge in F^t the value $((1 + \delta)\kappa)^{-1}$. Define Γ^t to be the event that z^t represents a valid fractional matching in F^t . Finally, set the edge set of H^t to be F^t if Γ^t holds, otherwise set H^t to be an arbitrary maximum matching in G^t .

Before proving Lemma C.1, we first show that z^t is a high-value fractional matching with good probability. For convenience, let

$$\text{Obj}(z^t) := \sum_{e_i \in F^t} z_{e_i}^t.$$

Also, for each edge $e \in E^t$ and $i \in [\kappa]$, let $X_{e,i} \in \{0, 1\}$ be the indicator for the event that $e_i \in E_F^t$. Note that

$$\mathbb{E}[X_{e,i}] = \Pr[x_e > \chi_{e,i}] = x_e^t. \quad (C.2)$$

Claim C.3. *For all time-steps t , event Γ^t holds with probability at least $1 - 1/n^{\alpha+3}$.*

Proof. Fix any node $v \in V$. Let $\deg^t(v, F)$ denote the number of multi-edges incident on v in F . Since $z_{e_i} = (1 + \delta)^{-1} \cdot (1/\kappa)$ for all $e_i \in E_F^t$, it suffices to show that $\deg^t(v, F) \leq (1 + \delta)\kappa$ with sufficiently high probability. Observe that (C.2) implies

$$\mathbb{E}[\deg^t(v, F)] = \sum_{e \in E^t(v), i \in [\kappa]} \mathbb{E}[X_{e,i}] = \sum_{e \in E^t(v)} \kappa \cdot x_e^t \leq \kappa. \quad (\text{C.3})$$

Recall the value of κ from (C.1). Applying a Chernoff bound,

$$\Pr [\deg^t(v, F) \leq (1 + \delta) \cdot \kappa] \geq 1 - 1/n^{\alpha+4}. \quad (\text{C.4})$$

The claim follows by a union bound over all n nodes in G^t . $\square$

Claim C.4. *For all time-steps t , we have $\mathbb{E}[\text{Obj}(z^t) \mid \Gamma^t] \geq (1 - \Theta(\delta)) \cdot \text{Obj}(x^t) - 1/n^{\alpha+1}$.*

Proof. From (C.2) and the fact that $z_{e_i}^t = ((1 + \delta)\kappa)^{-1}$,

$$\mathbb{E}[\text{Obj}(z^t)] = \sum_{e \in E^t} \mathbb{E} \left[\sum_{i \in [\kappa]: e_i \in E_F^t} z_{e_i}^t \right] = \sum_{e \in E^t} (1 + \delta)^{-1} \cdot x_e^t \geq (1 - \Theta(\delta)) \cdot \text{Obj}(x^t) \quad (\text{C.5})$$

Next, since $|E_F^t| \leq \kappa n^2$, it follows that with probability 1

$$\text{Obj}(z^t) = |E_F^t| \cdot (1 + \delta)^{-1} (1/\kappa) \leq n^2. \quad (\text{C.6})$$

By the law of total probability

$$\mathbb{E}[\text{Obj}(z^t) \mid \Gamma^t] \geq \mathbb{E}[\text{Obj}(z^t) \mid \Gamma^t] \cdot \Pr [\Gamma^t] = \mathbb{E}[\text{Obj}(z^t)] - \mathbb{E}[\text{Obj}(z^t) \mid \overline{\Gamma^t}] \cdot \Pr [\overline{\Gamma^t}].$$

Using (C.5), (C.6), and Claim C.3, we conclude

$$\mathbb{E}[\text{Obj}(z^t) \mid \Gamma^t] \geq (1 - \Theta(\delta)) \cdot \text{Obj}(x^t) - \frac{n^2}{n^{\alpha+3}} = (1 - \Theta(\delta)) \cdot \text{Obj}(x^t) - 1/n^{\alpha+1}. \quad \square$$

We are now ready to prove part (1) of Lemma C.1. The claim is trivially true if G^t is the empty graph. Otherwise, by the law of total probability,

$$\begin{aligned} \mathbb{E}[\mu(H^t)] &= \mathbb{E}[\mu(F^t) \mid \Gamma^t] \cdot \Pr [\Gamma^t] + \mathbb{E}[\mu(G^t) \mid \overline{\Gamma^t}] \cdot \Pr [\overline{\Gamma^t}] \\ &\geq \mathbb{E}[\text{Obj}(z^t) \mid \Gamma^t] \cdot \Pr [\Gamma^t] + \text{Obj}(x^t) \cdot \Pr [\overline{\Gamma^t}] \end{aligned} \quad (\text{C.7})$$

$$\geq (1 - \Theta(\delta)) \cdot \text{Obj}(x^t) - 1/n^{\alpha+1} \quad (\text{C.8})$$

$$\geq (1 - \Theta(\delta)) \cdot \text{Obj}(x^t). \quad (\text{C.9})$$

Step (C.7) follows from the fact that the bipartite matching polytope has no integrality gap, step (C.8) from Claim C.4, and step (C.9) since G^t is nonempty and hence $1/n^{\alpha+1} \leq \delta \cdot \text{Obj}(x^t)$.

We conclude with the proof of part (2). The recourse of H^t is at most the recourse of F^t , plus n^2 for every time-step that Γ^t does not hold (because in this case we need to replace the graph with a maximum matching in G^t and back which requires at most $2 \cdot \binom{n}{2} \leq n^2$ edge insertions). Since Γ^t holds with probability $1 - 1/n^{\alpha+3}$, the expected contribution of switching to G^t and back is at most $1/n^\alpha$ per time-step.

To bound the recourse of F^t , note that each copy of an edge $e_i \in E_F^t \oplus E_F^{t-1}$ iff the random variable $\chi_{e,i}$ (which is uniform in $[0, 1]$) lies in an interval of length $|x_e^t - x_e^{t-1}|$, and hence

$$\mathbb{E}[|E_F^t \oplus E_F^{t-1}|] = \kappa \cdot \sum_{e \in \binom{V}{2}} |x_e^t - x_e^{t-1}| = \kappa \cdot \|x^t - x^{t-1}\|_1.$$

Summing over all time-steps t and putting everything together, the total expected recourse of H^t is at most

$$\mathbb{E} \left[\sum_t |E_H^t \oplus E_H^{t-1}| \right] \leq \kappa \cdot \sum_t \|x^t - x^{t-1}\|_1 + \sum_t 1/n^\alpha. \quad \square$$

D Proof of Theorem 4.7

Theorem 4.7 (MST Rounding). *Consider any fractional solution x^t to the dynamic MST problem, and any $\delta > 0, \alpha \geq 1$. Then there exists a randomized dynamic rounding scheme which maintains a spanning tree $\mathcal{T}^t = (V, E_{\mathcal{T}}^t)$ of G^t such that:*

$$\mathbb{E} \left[\sum_{t \in [T]} |E_{\mathcal{T}}^t \oplus E_{\mathcal{T}}^{t-1}| \right] \leq O \left(\frac{\alpha \log n}{\delta^2} \right) \cdot \sum_{t \in [T]} \|x^t - x^{t-1}\|_1 + \frac{T}{O(n^\alpha)}, \quad (4.16)$$

$$\sum_{e \in E_{\mathcal{T}}^t} c_e \leq (2 + \delta) \cdot \sum_{e \in E^t} c_e \cdot x_e^t \quad \forall t \in [T]. \quad (4.17)$$

We follow the same paradigm as in Appendix C of maintaining a stabilizer, which in this case is a graph that contains a low-cost spanning tree whose recourse is bounded with respect to the ℓ_1 movement of x , and then giving an absolute recourse algorithm with respect to the updates to the stabilizer.

In the first step, we show:

Lemma D.1. *We can maintain a subgraph $H = (V, E_H)$ of G^t such that:*

(1) $\mathbb{E}[\mu(H^t)] \leq (2 + \delta) \sum_{e \in E^t} c_e \cdot x_e^t$, where $\mu(H^t)$ is the expected cost of the minimum spanning tree of H^t .

(2) $\mathbb{E} \left[\sum_t |E_H^{t-1} \oplus E_H^t| \right] \leq O \left(\frac{\alpha \log n}{\delta^2} \right) \cdot \sum_t \|x^t - x^{t-1}\|_1 + O \left(\frac{T}{n^\alpha} \right)$.

Once again, we show to complete the proof before giving the stabilizer construction.

Lemma D.2. *We can maintain a minimum spanning tree $\mathcal{T}$ of F with $O(1)$ absolute recourse per edge insertion/deletion in F .*

Proof. When edge e gets deleted from F , if $e \notin E_{\mathcal{T}}$, we don't make any changes to $\mathcal{T}$, otherwise replace e in $E_{\mathcal{T}}$ with the cheapest edge crossing its fundamental cut. When an edge e is added to F , add e to $E_{\mathcal{T}}$ then delete from $E_{\mathcal{T}}$ the cheapest edge along the fundamental cycle associated with e . In both cases $\mathcal{T}$ remains the MST after the update, and incurs absolute recourse $O(1)$. $\square$

Combining Lemma D.1 and Lemma D.2 yields Theorem 4.7. We turn to showing how to build the stabilizer.

Proof of Lemma D.1. Let $\gamma > 0$ be a sufficiently large constant. For every edge $e \in \binom{V}{2}$, define

$$p_e^t := \min \left\{ 1, \frac{100\gamma(\alpha + 3) \log n \cdot x_e^t}{\delta^2} \right\} \quad (D.1)$$

At preprocessing, for each edge $e \in \binom{V}{2}$ we draw a value $\chi_e \in [0, 1]$ uniformly and independently at random, and define the edge set $F^t := \left\{ e \in \binom{V}{2} : p_e^t > \chi_e \right\}$ (note that F^t is always a subset of the live edges E^t). Let Γ^t be the event that F^t contains a spanning tree of cost at most

$(2 + \delta) \cdot \sum_{e \in E^t} c_e \cdot x_e^t$. Finally, define the edge set of H^t to be F^t if Γ^t holds, otherwise set H^t to be an arbitrary minimum spanning tree of G^t .

Part (1) of Lemma D.1 is immediate by construction (since the integrality gap of the cut LP formulation of minimum spanning is at most 2), and it remains to prove part (2). To this end, we show that H^t contains a cheap spanning tree with good probability.

Claim D.3. *For all time-steps t , event Γ^t holds with probability at least $1 - 1/n^{\alpha+3}$.*

Proof. We follow the proof of Lemma 3.8 in [CQ18]. Let (G^t, x^t) be the weighted graph with edge weights defined by x^t . Define the *strength* of an edge e , denoted by κ_e^t , to be the maximum value of k such that a maximal k -connected vertex-induced subgraph of G^t contains e .

The mincut in (G^t, x^t) has value ≥ 1 , since x^t is feasible to (4.15), and therefore $\kappa_e^t \geq 1$ for all $e \in E^t$. For each edge $e \in E^t$, we now define

$$q_e^t := \min \left\{ 1, \frac{\gamma(\alpha + 3) \log n \cdot c_e \cdot x_e^t}{\delta^2 \cdot \sum_{e' \in E^t} c_{e'} \cdot x_{e'}^t} \right\} \text{ and } r_e^t := \max\{p_e^t, q_e^t\}. \quad (\text{D.2})$$

Now, consider the weighted graph (Z^t, z^t) with edge set $E_Z^t := \{e \in \binom{V}{2} : r_e^t > \chi_e\}$, where the weight of edge $e \in E_Z^t$ is $z_e^t := (1 + \delta) \cdot x_e^t / r_e^t$. Let $\mathcal{E}^t$ be the event that the weighted graph (Z^t, z^t) has mincut at least 1 and has $\sum_{e \in E_Z^t} c_e \cdot z_e^t \leq (1 + \delta) \cdot \sum_{e \in E^t} c_e \cdot x_e^t$. Since the cut LP formulation of minimum spanning tree has integrality gap at most 2, event $\mathcal{E}^t$ in turn means that Z^t contains an integral spanning tree $\mathcal{T}'$ of cost at most $2(1 + \delta) \cdot \sum_{e \in E^t} c_e \cdot x_e^t$.

Following the analysis in the proof of Lemma 3.8 in [CQ18], we derive that $\mathcal{E}^t$ holds with probability at least $1 - 1/n^{\alpha+3}$. Conditioned on $\mathcal{E}^t$ holding, consider any edge $e^* \in E_Z^t \setminus E_F^t$. We have $p_{e^*}^t \leq \chi_{e^*} < r_{e^*}^t$. Since $\chi_{e^*} \in [0, 1]$, recalling the values of $p_{e^*}^t$ and $r_{e^*}^t$ from (D.1) and (D.2), we infer that

$$\frac{100\gamma(\alpha + 3) \log n \cdot x_{e^*}^t}{\delta^2} \leq \frac{\gamma(\alpha + 3) \log n \cdot c_{e^*} \cdot x_{e^*}^t}{\delta^2 \cdot \sum_{e \in E^t} c_e \cdot x_e^t}.$$

Rearranging the terms in the above inequality, we get $c_{e^*} \geq 100 \cdot \sum_{e \in E^t} c_e \cdot x_e^t$, which means that e^* is not in $\mathcal{T}'$, the spanning tree of cost $2(1 + \delta) \cdot \sum_{e \in E^t} c_e \cdot x_e^t$ in Z^t . In other words, the edges of $\mathcal{T}$ are also contained in F^t , which was the claim to be proven. $\square$

We conclude with the proof of part (2) of Lemma D.1. The recourse of H^t is at most the recourse of F^t , plus n^2 for every time-step that Γ^t does not hold (because in this case we need to replace the graph with a spanning tree in G^t and back which requires at most $2 \cdot \binom{n}{2} \leq n^2$ edge insertions). Since Γ^t holds with probability $1 - 1/n^{\alpha+3}$, the expected contribution of switching to G^t and back is at most $1/n^\alpha$ per time-step.

To bound the recourse of F^t , note that each edge $e \in E_F^t \oplus E_F^{t-1}$ iff the random variable χ_e (which is uniform in $[0, 1]$) lies in an interval of length $|p_e^t - p_e^{t-1}| = O(\alpha \log n / \delta^2) |x_e^t - x_e^{t-1}|$, and hence

$$\mathbb{E}[|E_F^t \oplus E_F^{t-1}|] = O\left(\frac{\alpha \log n}{\delta^2}\right) \cdot \sum_{e \in \binom{V}{2}} |x_e^t - x_e^{t-1}| = O\left(\frac{\alpha \log n}{\delta^2}\right) \cdot \|x^t - x^{t-1}\|_1.$$

Summing over all time-steps t and putting everything together, the total expected recourse of H^t is at most

$$\mathbb{E}\left[\sum_t |E_H^t \oplus E_H^{t-1}|\right] \leq O\left(\frac{\alpha \log n}{\delta^2}\right) \cdot \sum_t \|x^t - x^{t-1}\|_1 + \sum_t O(1/n^\alpha). \quad \square$$

E Lower Bound

In this section we show that resource augmentation is essential in order to get a sub-polynomial competitive ratio for positive body chasing.

Theorem 1.2. *No algorithm for positive body chasing can achieve competitive ratio better than*

$$\Omega \left(\max \left\{ \min \left(\frac{1}{\epsilon \sqrt{\log(1/\epsilon)}}, \sqrt{n} \right), \log n \right\} \right).$$

The crux of our bound is the following lemma.

Lemma E.1. *No algorithm for positive body chasing can achieve competitive ratio better than*

$$\max \left\{ \frac{\sqrt{n}(1 - 10\epsilon \cdot \sqrt{n \log n})}{10}, 0 \right\}.$$

Let us first see how this implies Theorem 1.2.

Proof of Theorem 1.2. First note that the $\Omega(\log n)$ lower bound for fractional online set cover (see e.g. [BN09b]) implies an $\Omega(\log n)$ bound for positive body chasing, even with ϵ resource augmentation when $\epsilon \in (0, 1]$.

It remains to argue the remaining part of the bound. If $\epsilon \leq 1/(20\sqrt{n \log n})$, then Lemma E.1 implies a lower bound of $\Omega(\sqrt{n})$. If on the other hand $\epsilon > 1/(20\sqrt{n \log n})$, then the same construction from Lemma E.1 on a subspace of dimension $n' = 1/(900 \cdot \epsilon^2 \log(1/\epsilon)) \leq n$ implies a lower bound of $\Omega(\sqrt{n'}) = \Omega(1/\epsilon \sqrt{\log(1/\epsilon)})$. $\square$

Finally, we prove the main lemma.

Proof of Lemma E.1. Assume that $n = 2^k$ for sufficiently large k . Also, assume $\epsilon \leq 1/10\sqrt{n \log n}$, otherwise there is nothing to prove. Let h^i be the i^{th} row of the Hadamard matrix of order n . Define $S = \sqrt{n}(2\sqrt{\log n} + 1)$, as well as $(x^0, s^0) = (2\sqrt{(\log n)/n} \cdot \vec{1}, \sqrt{n})$. Note that $S = \|(x^0, s^0)\|_1$.

We execute the following instance for M many phases. In each phase the adversary chooses a uniformly random vector $b \in \{-1, 1\}^n$. A phase is composed of n time-steps, where the body at time $t \in [n]$ is defined as

$$K_t = \left\{ (x, s) \in R_+^{n+1} \left| \begin{array}{l} \langle h^t, x \rangle + s + \sum_{i=1}^n x_i \geq b^t + S + \langle h^t, x^0 \rangle \quad \text{if } b^t = 1 \\ \langle h^t, x \rangle + s + \sum_{i=1}^n x_i \leq b^t + S + \langle h^t, x^0 \rangle \quad \text{if } b^t = -1. \end{array} \right. \right\}$$

We claim that this is an instance of positive body chasing. Let us first check that the constraint of each K_t is a covering/packing constraint. For the left-hand side of each constraint, the coefficients of each variable are at least zero because entries of h^t are in $\{+1, -1\}$ and so the coefficient of x_i is $h_i^t + 1 \geq 0$. Also, the coefficient of s is 1. For the right-hand side, we have that $b^t + S + \langle h^t, x^0 \rangle \geq 0$ because, by the definition of the Hadamard matrix, $\langle h^t, x^0 \rangle$ equals to 0 when $t > 0$ and equals to $\|x^0\|$ when $t = 0$. It remains to show that $K_t \neq \emptyset$. Observe that $(x^0, s^0 + b^t) \in K_t$ and, in fact, the constraint is tight: $\langle h^t, x_0 \rangle + (s^0 + b^t) + \sum_{i=1}^n x_i = b^t + S + \langle h^t, x^0 \rangle$ because $S = \sum_i x_i + s_0$.

Movement cost of ALG: We argue that any online algorithm must pay at least $n/2 \cdot (1 - 10\epsilon \cdot \sqrt{n \log n})$ per phase in expectation.

For convenience, define $L^t = \langle h^t, x^t \rangle + s + \sum_i x_i$ to be the left-hand side of the constraints above, and let $R^t = S + \langle h^t, x^0 \rangle$. The algorithm needs to satisfy covering constraints fully, and packing constraints up to $(1 + \epsilon)$; if $b^t = 1$, the algorithm must choose x^t such that

$$L^t \geq R^t + 1,$$

and if $b^t = -1$, the algorithm must choose x^t such that

$$L^t \leq (R^t - 1)(1 + \epsilon).$$

The term L^t must change by at least half the distance between $R^t + 1$ and $(R^t - 1)(1 + \epsilon)$ in expectation, i.e.

$$\mathbb{E}[|L^t - L^{t-1}|] \geq \frac{(R^t + 1) - (R^t - 1)(1 + \epsilon)}{2} = 1 - \frac{\epsilon(R^t - 1)}{2}.$$

Since $L^t = \langle l^t, (x^t, s^t) \rangle$ where l^t is a vector with coefficients in $[0, 2]$, this implies that

$$\mathbb{E}[\|(x^t, s^t) - (x^{t-1}, s^{t-1})\|_1] \geq \frac{1}{2} - \frac{\epsilon(R^t - 1)}{4} = \frac{1 - 10\epsilon \cdot \sqrt{n \log n}}{2}.$$

Therefore, the expected cost of any algorithm during a single phase of n time-steps is at least $n/2 \cdot (1 - 10\epsilon \cdot \sqrt{n \log n})$.

Movement of OPT: We argue that OPT pays at most $5\sqrt{n}$ in expectation per phase. We begin by assuming that OPT always returns to (x^0, s^0) at the end of a phase, since this can only increase the cost of OPT.

First, note that since $(x^0, s^0 + b^t) \in K_t$, then the optimal solution may move at each time t to this point, and return at the end to (x^0, s^0) . The total movement cost of this solution is bounded by $4 \sum_{i=1}^n |b^i| = 4n$. However, we claim that a better solution is available to OPT with high probability.

Let $\mathcal{E}$ be the event that $\|H^{-1}b\|_\infty \leq 2\sqrt{(\log n)/n}$. Because

$$\|H^{-1}b\|_2 \leq \|H^{-1}\| \cdot \|b\|_2 \leq \frac{1}{\sqrt{n}} \cdot \sqrt{n} = 1,$$

we have that $\|H^{-1}b\|_1 \leq \sqrt{n} \cdot \|H^{-1}b\|_2 \leq \sqrt{n}$. Define

$$\begin{aligned} x^* &= x^0 + H^{-1}b = 2\sqrt{\frac{\log n}{n}} \cdot \vec{1} + H^{-1}b, \\ s^* &= S - \sum_i x_i^* \geq \sqrt{n} - \|H^{-1}b\|_1 \end{aligned}$$

Now for all $t \in [n]$, since $\langle h^t, x^* \rangle = \langle h^t, x^0 \rangle + b^t$ by definition of x^* , we have by design

$$\langle h^t, x^* \rangle + s^* + \sum_{i=1}^n x_i^* = b^t + S + \langle h^t, x^0 \rangle.$$

Additionally, if $\mathcal{E}$ holds, then using the fact that $\|H^{-1}b\|_1 \leq \sqrt{n}$, the point $(x^*, s^*) \in R_+^{n+1}$ and thus $(x^*, s^*) \in \bigcap_{t=1}^n K_t$. Consequently, OPT can move directly from (x^0, s^0) to (x^*, s^*) and back at the end of the phase, paying at most $4\sqrt{n}$.

It remains to bound the probability of $\mathcal{E}$. The matrix H^{-1} has entries in $\{-1/n, 1/n\}$, so each entry of $H^{-1}b$ is a random walk on the line with step size $1/n$.

Fact E.2 (Chernoff Bound). *If $Z = \sum_{i=1}^n X_i$ where X_i are independent random variables such that $\Pr[X_i = 1] = \Pr[X_i = -1] = 1/2$, then*

$$\Pr[|Z| \geq \sqrt{an \log n}] \leq 2n^{-a/2}.$$

By this fact together with a union bound, event $\|H^{-1}b\|_\infty \leq 2\sqrt{(\log n)/n}$ holds with probability $1 - 2/n$. Hence, the expected cost of OPT in total over the phase is $(1 - 2/n) \cdot 4\sqrt{n} + 2/n \cdot 4n \leq 5\sqrt{n}$, for sufficiently large n .

To conclude, summing over M phases and accounting for OPT moving to (x^0, s^0) before the first phase, the competitive ratio of any algorithm is at least

$$\frac{\mathbb{E}[\text{ALG}]}{\mathbb{E}[\text{OPT}]} \geq \frac{M \cdot n/2 \cdot (1 - 10\epsilon \cdot \sqrt{n \log n})}{5 \cdot M \cdot \sqrt{n} + \|(x^0, s^0)\|_1} \geq \frac{\sqrt{n}(1 - 10\epsilon \cdot \sqrt{n \log n})}{10} \cdot (1 - o(1)). \quad \square$$

F Handling Box Constraints

In this section we show that we can handle box constraints, a special case of packing constraints, without violation. Box constraints are “static” packing constraints of the form $x_j \leq b_j$.

In general, we allow to add to the polytope at time t the box constraints $x \leq b$, where $b \in R_+^n \cup \{\infty\}$. Thus, the more general polytope at time t is:

$$K_t^{1+\epsilon} = \{x^t \in R_+^n \mid x \leq b, C^t x^t \geq 1, P^t x^t \leq 1 + \epsilon\}$$

We claim that we can simply eliminate these new constraints by adding additional covering constraints. For each covering constraint $\sum_j c_{ij} x_j \geq 1$, and every subset of variables S , we add the constraint:

$$\sum_{j \notin S} c_{ij} x_j \geq \max \left\{ 0, 1 - \sum_{j \in S} c_{ij} \cdot b_j \right\}$$

It is easy to verify that if $x \in R_+^n$ satisfies these new constraints, then the vector x' , where $x'_j = \min\{x_j, b_j\}$ satisfies the original constraints along with the box constraints. On the other hand, any x'' that satisfies the box constraints satisfies all the new covering constraints.

Thus, our algorithm maintains a feasible point to $K_t^{1+\epsilon}$ but with box constraints removed, and the new covering constraints added instead. The algorithm maintains at time t the vector x' , where $x'_j = \min\{x_j, b_j\}$, and its total movement cost is no more than that of x .

Finally, we remark that although the number of such constraints is exponential, it is easy to verify if a solution x satisfies these new constraints by checking that x satisfies the covering constraint with $S = \{j \mid x_j \geq b_j\}$.

References

- [AAA⁺06] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. A general approach to online network optimization problems. *ACM Transactions on Algorithms*, 2(4):640–660, 2006. → cited on page 4
- [AAA⁺09] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. The online set cover problem. *SIAM J. Comput.*, 39(2):361–370, 2009. → cited on page 4
- [AAG⁺19] Amir Abboud, Raghavendra Addanki, Fabrizio Grandoni, Debmalaya Panigrahi, and Barna Saha. Dynamic set cover: improved algorithms and lower bounds. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23–26, 2019*, pages 114–125. ACM, 2019. → cited on page 2

- [AAPW01] Baruch Awerbuch, Yossi Azar, Serge A. Plotkin, and Orli Waarts. Competitive routing of virtual circuits with unknown duration. *J. Comput. Syst. Sci.*, 62(3):385–397, 2001. → cited on page 2
- [ABC⁺16] Yossi Azar, Niv Buchbinder, T.-H. Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph Naor, and Debmalaya Panigrahi. Online algorithms for covering and packing problems with convex objectives. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016*, pages 148–157, 2016. → cited on page 4
- [ABC⁺19] C. J. Argue, Sébastien Bubeck, Michael B. Cohen, Anupam Gupta, and Yin Tat Lee. A nearly-linear bound for chasing nested convex bodies. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 117–122. SIAM, 2019. → cited on page 1
- [ABFP13] Yossi Azar, Umang Bhaskar, Lisa Fleischer, and Debmalaya Panigrahi. Online mixed packing and covering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013*, pages 85–100, 2013. → cited on page 4
- [ABL⁺20] Chen Avin, Marcin Bienkowski, Andreas Loukas, Maciej Pacut, and Stefan Schmid. Dynamic balanced graph partitioning. *SIAM Journal on Discrete Mathematics*, 34(3):1791–1812, 2020. → cited on page 4
- [ACC⁺18] Moab Arar, Shiri Chechik, Sarel Cohen, Cliff Stein, and David Wajc. Dynamic matching: Reducing integral algorithms to approximately-maximal fractional algorithms. In *45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107 of *LIPIcs*, pages 7:1–7:16, 2018. → cited on page 16
- [AGTG21] C. J. Argue, Anupam Gupta, Ziyi Tang, and Guru Guruganesh. Chasing convex bodies with linear competitive ratio. *J. ACM*, 68(5):32:1–32:10, 2021. → cited on page 1, 2, 5
- [ALPS16] Chen Avin, Andreas Loukas, Maciej Pacut, and Stefan Schmid. Online balanced repartitioning. In *Distributed Computing: 30th International Symposium, DISC 2016, Paris, France, September 27-29, 2016. Proceedings*, pages 243–256. Springer, 2016. → cited on page 4
- [AMPST23] Yossi Azar, Chay Machluf, Boaz Patt-Shamir, and Noam Touitou. Competitive vertex recoloring: (online disengagement). *Algorithmica*, pages 1–27, 2023. → cited on page 4
- [AOSS18] Sepehr Assadi, Krzysztof Onak, Baruch Schieber, and Shay Solomon. Fully dynamic maximal independent set with sublinear update time. In *Proceedings of the 50th Annual ACM SIGACT Symposium on theory of computing*, pages 815–826, 2018. → cited on page 2
- [AS21] Sepehr Assadi and Shay Solomon. Fully dynamic set cover via hypergraph maximal matching: An optimal approximation through a local approach. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference)*, volume 204 of *LIPIcs*, pages 8:1–8:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. → cited on page 2
- [BBCG22] Suman K Bera, Sayan Bhattacharya, Jayesh Choudhari, and Prantar Ghosh. A new dynamic algorithm for densest subhypergraphs. In *Proceedings of the ACM Web Conference 2022*, pages 1093–1103, 2022. → cited on page 2, 4
- [BBE⁺20] Nikhil Bansal, Martin Böhm, Marek Eliás, Grigorios Koumoutsos, and Seeun William Umboh. Nested convex bodies are chaseable. *Algorithmica*, 82(6):1640–1653, 2020. → cited on page 1, 6
- [BBN10] Nikhil Bansal, Niv Buchbinder, and Joseph Naor. Metrical task systems and the k-server problem on HSTs. In *ICALP (1)*, pages 287–298, 2010. → cited on page 4
- [BBN12a] Nikhil Bansal, Niv Buchbinder, and Joseph Naor. A primal-dual randomized algorithm for weighted paging. *J. ACM*, 59(4):19, 2012. → cited on page 4
- [BBN12b] Nikhil Bansal, Niv Buchbinder, and Joseph Naor. Randomized competitive algorithms for generalized caching. *SIAM J. Comput.*, 41(2):391–414, 2012. → cited on page 4
- [BCL⁺18] Sébastien Bubeck, Michael B. Cohen, Yin Tat Lee, James R. Lee, and Aleksander Madry. k-server via multiscale entropic regularization. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 3–16, 2018. → cited on page 4
- [BCLL21] Sébastien Bubeck, Michael B. Cohen, James R. Lee, and Yin Tat Lee. Metrical task systems on trees via mirror descent and unfair gluing. *SIAM J. Comput.*, 50(3):909–923, 2021. → cited on page 4
- [BDL21] Aaron Bernstein, Aditi Dudeja, and Zachary Langley. A framework for dynamic matching in weighted graphs. In *53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 668–681. ACM, 2021. → cited on page 17

- [BF99] Gerth Stølting Brodal and Rolf Fagerberg. Dynamic representations of sparse graphs. In *Algorithms and Data Structures: 6th International Workshop, WADS'99 Vancouver, Canada, August 11–14, 1999 Proceedings 6*, pages 342–351. Springer, 1999. → cited on page 2, 4
- [BGMN19] Niv Buchbinder, Anupam Gupta, Marco Molinaro, and Joseph (Seffi) Naor. k -servers with a smile: Online algorithms via projections. In *Accepted to the Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, 2019. → cited on page 4, 5
- [BHN19] Sayan Bhattacharya, Monika Henzinger, and Danupon Nanongkai. A new deterministic algorithm for dynamic set cover. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9–12, 2019*, pages 406–423. IEEE Computer Society, 2019. → cited on page 2
- [BHNW21] Sayan Bhattacharya, Monika Henzinger, Danupon Nanongkai, and Xiaowei Wu. Dynamic set cover: Improved amortized and worst-case update time. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2537–2549. SIAM, 2021. → cited on page 2
- [BJN07] Niv Buchbinder, Kamal Jain, and Joseph (Seffi) Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *Proc. of the 15th Annual European Symp.*, pages 253–264, 2007. → cited on page 4
- [BKL⁺20] Sébastien Bubeck, Bo'az Klartag, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. Chasing nested convex bodies nearly optimally. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 1496–1508. SIAM, 2020. → cited on page 1, 2
- [BKS12] Surender Baswana, Sumeet Khurana, and Soumojit Sarkar. Fully dynamic randomized algorithms for graph spanners. *ACM Transactions on Algorithms (TALG)*, 8(4):1–51, 2012. → cited on page 2
- [BKS23] Sayan Bhattacharya, Peter Kiss, and Thatchaphol Saranurak. Dynamic algorithms for packing-covering lps via multiplicative weight updates. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1–47. SIAM, 2023. → cited on page 4
- [BLP22] Sayan Bhattacharya, Silvio Lattanzi, and Nikos Parotsidis. Efficient and stable fully dynamic facility location. In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS)*, 2022. → cited on page 2
- [BN09a] Niv Buchbinder and Joseph Naor. The design of competitive online algorithms via a primal-dual approach. *Foundations and Trends in Theoretical Computer Science*, 3(2-3):93–263, 2009. → cited on page 5, 10
- [BN09b] Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing. *Math. Oper. Res.*, 34(2):270–286, 2009. → cited on page 2, 3, 4, 26
- [BSS22] Sayan Bhattacharya, Thatchaphol Saranurak, and Pattara Sukprasert. Simple dynamic spanners with near-optimal recourse against an adaptive adversary. In *30th Annual European Symposium on Algorithms (ESA)*, volume 244 of *LIPICs*, pages 17:1–17:19, 2022. → cited on page 2
- [CLNT22] Christian Coester, Roie Levin, Joseph (Seffi) Naor, and Ohad Talmon. Competitive algorithms for block-aware caching. In *SPAA '22: 34th ACM Symposium on Parallelism in Algorithms and Architectures, Philadelphia, PA, USA, July 11 - 14, 2022*, pages 161–172. ACM, 2022. → cited on page 4
- [CQ18] Chandra Chekuri and Kent Quanrud. Fast approximations for metric-tsp via linear programming. *CoRR*, abs/1802.01242, 2018. → cited on page 25
- [Csi84] Imre Csiszár. Sanov property, generalized i-projection and a conditional limit theorem. *The Annals of Probability*, pages 768–793, 1984. → cited on page 5
- [FL93] Joel Friedman and Nathan Linial. On convex body chasing. *Discrete & Computational Geometry*, 9:293–321, 1993. → cited on page 1, 2
- [GGK16] Albert Gu, Anupam Gupta, and Amit Kumar. The power of deferral: Maintaining a constant-competitive Steiner tree online. *SIAM J. Comput.*, 45(1):1–28, 2016. → cited on page 2
- [GK14] Anupam Gupta and Amit Kumar. Online Steiner tree with deletions. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5–7, 2014*, pages 455–467. SIAM, 2014. → cited on page 2
- [GKKP17] Anupam Gupta, Ravishankar Krishnaswamy, Amit Kumar, and Debmalaya Panigrahi. Online and dynamic algorithms for set cover. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19–23, 2017*, pages 537–550. ACM, 2017. → cited on page 2

- [GKL21] Anupam Gupta, Gregory Kehne, and Roie Levin. Random order online set cover is as easy as offline. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, pages 1253–1264. IEEE, 2021. → cited on page 2
- [GKLX20] Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. On the facility location problem in online and dynamic models. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. → cited on page 2
- [GKS14] Anupam Gupta, Amit Kumar, and Cliff Stein. Maintaining assignments online: Matching, scheduling, and flows. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 468–479. SIAM, 2014. → cited on page 2
- [GL20a] Anupam Gupta and Roie Levin. Fully-dynamic submodular cover with bounded recourse. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1147–1157. IEEE, 2020. → cited on page 2
- [GL20b] Anupam Gupta and Roie Levin. The online submodular cover problem. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1525–1537. SIAM, 2020. → cited on page 3
- [GN14] Anupam Gupta and Viswanath Nagarajan. Approximating sparse covering integer programs online. *Math. Oper. Res.*, 39(4):998–1011, 2014. → cited on page 3, 4
- [GTW14] Anupam Gupta, Kunal Talwar, and Udi Wieder. Changing bases: Multistage optimization for matroids and matchings. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 563–575, 2014. → cited on page 4
- [IW91] Makoto Imase and Bernard M. Waxman. Dynamic Steiner tree problem. *SIAM J. Discret. Math.*, 4(3):369–384, 1991. → cited on page 2
- [KKR01] A. R. Karlin, C. Kenyon, and D. Randall. Dynamic TCP acknowledgement and other stories about $e/(e-1)$. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 502–509, 2001. → cited on page 4
- [KLS23] Ravishankar Krishnaswamy, Shi Li, and Varun Suriyanarayana. Online unrelated-machine load balancing and generalized flow with recourse. In *STOC*, 2023. → cited on page 2, 18
- [ŁOP⁺15] Jakub Łącki, Jakub Oćwieja, Marcin Pilipeczuk, Piotr Sankowski, and Anna Zych. The power of dynamic distance oracles: Efficient dynamic algorithms for the Steiner tree. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 11–20, 2015. → cited on page 2
- [NPS11] Joseph Naor, Debmalaya Panigrahi, and Mohit Singh. Online node-weighted Steiner tree and related problems. In *IEEE 52nd Annual Symp. on Foundations of Computer Science, FOCS*, pages 210–219, 2011. → cited on page 4
- [Sel20] Mark Sellke. Chasing convex bodies optimally. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020*, pages 1509–1518. SIAM, 2020. → cited on page 1, 2, 5
- [SW20a] Saurabh Sawlani and Junxing Wang. Near-optimal fully dynamic densest subgraph. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 181–193, 2020. → cited on page 2
- [SW20b] Shay Solomon and Nicole Wein. Improved dynamic graph coloring. *ACM Transactions on Algorithms (TALG)*, 16(3):1–24, 2020. → cited on page 2