2016 IEEE 57th Annual Symposium on Foundations of Computer Science

Online Algorithms for Covering and Packing Problems with Convex Objectives

Yossi Azar*, Niv Buchbinder†, T-H. Hubert Chan‡, Shahar Chen§, Ilan Reuven Cohen†, Anupam Gupta¶,
Zhiyi Huang‡, Ning Kang‡, Viswanath Nagarajan‖, Joseph (Seffi) Naor§, Debmalya Panigrahi**
*Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel.
Email: {azar@tau.ac.il, ilanrcohen@gmail.com}
†Department of Statistics and Operations Research, Tel Aviv University, Tel Aviv, Israel.
Email: niv.buchbinder@gmail.com
‡Department of Computer Science, The University of Hong Kong, Hong Kong, China.
Email: {hubert,zhiyi,nkang}@cs.hku.hk
§Department of Computer Science, Technion - Israel Institute of Technology, Haifa, Israel.
Email:{shahar.chen11@gmail.com, naor@cs.technion.ac.il}
¶Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA.
Email: anupamg@cs.cmu.edu
‖Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA.
Email: viswa@umich.edu
**Department of Computer Science, Duke University, Durham, NC 27708, USA.
Email: debmalya@cs.duke.edu

***Abstract*—We present online algorithms for covering and packing problems with (non-linear) convex objectives. The convex covering problem is defined as: $\min_{x\in\mathbb{R}_+^n} f(x)$ s.t. $Ax \geq 1$, where $f : \mathbb{R}_+^n \to \mathbb{R}_+$ is a monotone convex function, and A is an $m \times n$ matrix with non-negative entries. In the online version, a new row of the constraint matrix, representing a new covering constraint, is revealed in each step and the algorithm is required to maintain a feasible and monotonically non-decreasing assignment x over time. We also consider a convex packing problem defined as: $\max_{y\in\mathbb{R}_+^m} \sum_{j=1}^m y_j - g(A^T y)$, where $g : \mathbb{R}_+^n \to \mathbb{R}_+$ is a monotone convex function. In the online version, each variable y_j arrives online and the algorithm must decide the value of y_j on its arrival. This represents the Fenchel dual of the convex covering program, when g is the convex conjugate of f. We use a primal-dual approach to give online algorithms for these generic problems, and use them to simplify, unify, and improve upon previous results for several applications.**

***Index Terms*—online algorithm; convex optimization; primal-dual algorithm**

I. Introduction

In the area of online algorithms, the method of modeling a problem as a linear program, obtaining a fractional solution via a primal-dual algorithm, and then rounding it, has proved to be a very powerful general technique (see [4] for a survey). This framework is applicable to many central online problems, like the classic ski rental problem, online set-cover [5], generalized paging [6], [7], k-server and metrical task systems [8], [9], graph connectivity [10], [11], routing [12], load balancing and machine scheduling [13], [14], matching [15], and budgeted allocation [16]. Via this approach, not only can we unify previously known results, but we can also resolve important open questions in competitive analysis. The use of the online primal-dual method mirrors (and is inspired by) the use of linear programming relaxations for approximation algorithms, with one major difference. In the online setting, generic linear programming solvers cannot be used for even obtaining a fractional solution, since constraints are presented online. Hence, new algorithms have been developed for solving linear programs online [17], [18].

The basic setting of the online primal-dual framework is grounded on the well-studied covering/packing framework for combinatorial optimization problems. Covering/packing linear formulations are a special class of linear formulations in which all coefficients are non-negative. In a covering problem, the goal is to minimize a non-negative cost function subject to non-negative covering constraints. In a packing problem the goal is to maximize a non-negative profit function subject to non-negative packing constraints. Packing and covering form a primal-dual pair – for consistency, we refer to the covering problem as the primal problem and to the packing problem as the dual problem, although in some cases, it is the dual packing problem in which we are interested in. The covering/packing framework captures a large class of (relaxations) of well-

This paper presents results independently obtained by Y. Azar, I. R. Cohen, and D. Panigrahi [1], N. Buchbinder, S. Chen, A. Gupta, V. Nagarajan, and J. Naor [2], and T.-H. H. Chan, Z. Huang, and N. Kang [3].

Y. Azar and I. R. Cohen are supported in part by the Israel Science Foundation (grant No. 1506/16), by the I-CORE program (Center No. 4/11), and by the Blavatnik Fund. N. Buchbinder is supported in part by ISF grant 1585/15 and by BSF grant 2014414. S. Chen and J. Naor are supported in part by ISF grant 1585/15 and BSF grant 2014414. A. Gupta is supported in part by NSF awards CCF-1319811, CCF-1536002, CCF-1540541 and CCF-1617790. D. Panigrahi is supported in part by NSF Awards CCF-1527084 and CCF-1535972. T-H. H. Chan is supported in part by the Hong Kong RGC grant 17202715. Z. Huang is supported in part by the Hong Kong RGC grant 17202115.

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.24

IEEE computer society

studied combinatorial problems.

Although extremely successful, the online primal-dual approach has been mainly studied for linear objective functions. Yet, what about convex optimization problems? In the offline setting, the Ellipsoid and interior-point methods can solve these problems with linear convergence rates. But, online, it was not known how to solve convex optimization problems in general prior to this work. In contrast, specific online problems with non-linear convex/concave objectives have been studied in recent years, in areas such as energy-efficient scheduling [19], [20], [21], paging [22], network routing [23], combinatorial auctions [24], [14], matching [25], and budgeted allocation [26]. In this paper, we develop a general framework for a broad class of covering/packing programs with convex objective functions. The competitive ratios we obtain are optimal. This already improves and substantially generalizes previous results for problem classes such as mixed packing covering LPs considered in [13]. We then show how to round these fractional solutions online for specific optimization problems, obtaining improved results for non-linear optimization problems considered recently in machine scheduling, network routing, and combinatorial auctions.

A. The Convex Optimization Framework

The next convex covering problem is our primal problem.

$$\min_{x \in \mathbb{R}^n_+} \quad C(x) := f(x) \qquad \text{subject to} \quad Ax \geq \mathbf{1}. \tag{I.1}$$

Here, $f : \mathbb{R}^n_+ \to \mathbb{R}$ is a convex function and $A_{m \times n}$ is a non-negative matrix. The rows of A, the covering constraints, arrive online over time. At any point of time we must maintain a feasible fractional solution x, non-decreasing over time.

This convex program captures, e.g., the mixed covering-packing problem [13], the machine scheduling and online routing problems from [23], [20], and the nested set cover problem [27].

As in solving LPs online, we use a primal-dual technique where we simultaneously obtain a solution to both primal and dual programs. In the convex setting, the dual of (I.1) is the following packing problem:

$$\max_{y \in \mathbb{R}^m_+, z \in \mathbb{R}^n} \quad P(y) := \textstyle\sum_{j \in [m]} y_j - f^\star(z) \quad \text{subject to} \quad A^T y \leq z. \tag{I.2}$$

Here $f^\star$ is the Fenchel dual or conjugate function of f, which is always a convex function[*], and hence another convex optimization problem. In the online setting, at each step a new dual variable y_i arrives along with its coefficients. The dual values must be monotone: we fix a value of y_i upon arrival and cannot change it later, in particular in applications where the dual problem is our focus.

Interestingly, the dual has a natural interpretation of its own, as the problem of *social-welfare maximization with production costs*. Consider a seller who produces and sells items of n types in a combinatorial market with m buyers. The seller produces a quantity z_j of each item $j \in [n]$, at total cost $f^\star(z)$. Now, if y_j denotes the value obtained from the bundle of goods assigned to buyer j, and $Ay \leq z$ captures the constraints that no item is sold in greater quantities than its production, then we recover precisely the dual problem. This social-welfare maximization problem was studied by Blum *et al.* [24] and Huang *et al.* [14] for the "separable" special case in which the production cost of each item is a univariate convex function, and the total cost is the sum over the items: $f^\star(z) = \sum_i f_i^\star(z_i)$. This case is somewhat restrictive in that it does not capture production costs such as energy that are typically modeled as $f^\star(z) = (\sum z_i)^\alpha$ for some $\alpha > 1$. This is a non-separable function, and so not captured by previous work, but it falls neatly into our framework.

B. Our Results

Our main theorem for the general online covering/packing framework is for monotone convex functions with monotone gradients[†] (we denote the gradient ∇f). Our primal-dual approach simultaneously gives us solutions to both the primal and dual convex programs.

Theorem 1 (Online Convex Covering/Packing Theorem)**.** *Let $f : \mathbb{R}^n_+ \to \mathbb{R}_+$ be a function that is monotone, differentiable, convex, and ∇f is monotone. Let $p = \sup_{x \geq \mathbf{0}} \frac{\langle \nabla f(x), x \rangle}{f(x)}$. Then,*

(a) There exists an $O(p \log d)^p$-competitive algorithm that produces a monotone primal solution.

(b) There exists an $O(p \log(\rho d))^p$-competitive algorithm that produces a monotone dual solution.

Here d is the maximal row sparsity of the matrix and ρ is $\max_j \max_{i,i' | a_{i,j} \neq 0} \{ \frac{a_{i'j}}{a_{ij}} \}$.

The conditions of the theorem are satisfied, for example, by convex polynomials of degree p (with non-negative coefficients). One such example is $f(x) = \frac{1}{p}\|x\|_p^p$, whose Fenchel conjugate function is $f^\star = \frac{1}{q}\|x\|_q^q$ where $\frac{1}{p} + \frac{1}{q} = 1$. Intuitively, p bounds the growth of the objective around a given point and shows the (necessary) effects of discretization of the feasible region performed by the algorithm. In fact, our competitive ratios are the best possible for all p: see §III-A.

Observe that the dual result of Theorem 1(b) makes assumptions on the function f and then solves the dual problem using $f^\star$. However, if our goal is to solve the dual problem, e.g., for social-welfare maximization with production costs, we can instead impose conditions on the dual cost function $f^\star$, which may yield a better competitive ratio. Indeed, the assumptions on the primal and the dual yield incomparable families of convex functions.[‡] Our general result in this setting is the following:

[*] The Fenchel dual function is formally defined in (II.3); see, e.g., [28] for background and properties.

[†] Function $g : \mathbb{R}^n \to \mathbb{R}$ is monotone if for all x', x such that $x'_i \geq x_i$ for all $i \in [n]$, $g(x') \geq g(x)$. Monotonicity of the objective is clearly necessary, but that of its gradient is a technical assumption that might be avoidable.

[‡] For example, monotonicity of f and ∇f does not imply monotonicity of $\nabla f^\star$: e.g., if $f(x) = (x_1 + x_2)^2$, then $f^\star(z) = \max(z_1^2, z_2^2)$ which has non-monotone gradients. Hence, one can use Theorem 1 to solve the dual problem for this $f^\star$, but not Theorem 2.

Theorem 2 (Online Convex Packing Theorem). *Let $f^\star : \mathbb{R}_+^n \to \mathbb{R}_+$ be a convex polynomial function with non-negative coefficients, zero constant term and maximum degree q. Then, there exists an $O(q)$-competitive algorithm for the online convex packing problem.*

This competitive ratio is also optimal, even for the special case where $f^\star$ is "separable", i.e., the sum of single-variate convex functions $f^\star(z) = \sum_j g_j(z_j)$; see [14].

Prior to this work, analogous results for online covering/packing were only known for *linear* objectives. Competitive ratios of $O(\log n)$ for covering and $O(\log(\rho d))$ for packing were obtained in [17], the covering ratio being subsequently improved to $O(\log d)$ in [18]. For linear objectives, Theorem 1 obtains ratios of $O(\log d)$ for covering and $O(\log(\rho d))$ for packing, matching the best bounds.

C. Our Techniques

Our algorithm for convex covering and packing (Theorem 1) is based on a clean unified approach, that significantly simplifies even the well-studied linear case [18]. The value of the primal variables grows exponentially proportional both to their current value and their coefficient in the constraint, but divided by the gradient of f at the current solution. The dual is increased at a (carefully chosen) linear rate. This update rule is enough for obtaining the second part of Theorem 1 (which has a dependence on ρ, a parameter that depends on the matrix A). But it does not prove the first part, which is independent of the numbers. To obtain the improved primal result we keep "shadow duals" that are used to bound the primal cost more tightly. These dual variables are non-monotone and may be decreased at certain times. Decreasing the dual variables may potentially reduce the total dual value being used to bound the primal cost. However, doing this carefully and maintaining such "non-monotone" duals is crucial in obtaining the sharp $O(p \log d)^p$ competitive ratio.

The algorithm for convex packing (Theorem 2) achieves a much better competitive ratio when parameterized by the "smoothness parameter" of the dual cost function $f^\star$. This parametrization is also natural in applications such as welfare maximization. This algorithm increases duals linearly, while maintaining primal variables as the gradient of $f^\star$ at a scaled dual solution. The primal/dual updates and their analysis are quite different from those of Theorem 1: this is not surprising since the two algorithms are intended to perform well for (roughly) complementary classes of the function f. Moreover, as noted earlier, there are functions f and $f^\star$ for which the guarantees in both Theorem 1 and Theorem 2 are tight.

D. Applications

The general framework captures many existing and new applications. Due to space limitations, we only give an outline of these applications here and defer a detailed discussion to the full version of the paper.

(1) Mixed Covering and Packing LPs. Here we have an online covering problem, along with K different objective functions, and the goal is to minimize the p-norm of these K objective functions. As a direct application of our general covering framework (Theorem 1(a)), we obtain an $O(p \log d)$-competitive algorithm for this problem, where $d \leq n$ is the row-sparsity of the covering constraint matrix A. This yields an $O(\log K \log d)$-competitive algorithm for minimizing the maximum constraint (the ∞-norm), improving upon the previous best bound of $O(\log K \cdot \log(d\kappa\gamma))$ [13], where γ and κ are the max-to-min ratio of the entries in the covering and packing constraints respectively. For p-norms of the objective functions, no previous bound was known.

While our framework is designed for obtaining fractional solutions, it also yields new results for discrete problems via online rounding. Several examples follow.

(2) Social Welfare Maximization with Production Costs. As an application of our packing framework, we consider social welfare maximization in combinatorial auctions with production costs. For the problem with arbitrary valuation functions (assuming access via demand oracles), where the production cost function is allowed to be any convex degree-q polynomial production cost function with non-negative coefficients, we get an algorithm which is $O(q)$-competitive, with an additive loss in the social welfare that depends only on the production cost functions (but is independent of the number of buyers). [14] considered the separable case and characterized the optimal competitive ratio of any given cost function via a differential equation. For the special case of polynomial cost functions x^q, [14] explicitly solved the differential equation and obtained a competitive ratio of 4q. Our results in this paper, applying to the special case of separable degree-q polynomial functions, essentially gives the same algorithm and a competitive ratio of $O(q)$ [§], matching the results of [14] asymptotically. We stress that the results in this paper applies to the much more general non-separable cost functions.

(3) Unrelated Machine Scheduling with Startup Costs. In this problem, a set of n jobs arriving online have to be scheduled on m machines, where machine i has *startup cost* c_i and can run job j in *processing time* p_{ij}. The goal is to assign each job to a machine so as to simultaneously minimize the p-norm of the overall machine loads and the total startup cost of machines used. For this problem, we obtain an $(O(\log m \log(mn)), O(p^2 \log^{1/p}(mn)))$ bicriteria competitive ratio,[¶] which is almost tight. For the special case of total load (the 1-norm), we obtain a tighter bound of $(O(\log m \log n), O(1))$. For the special case of the ∞-norm (maximum load or makespan), the best competitive ratio previously obtained using tools specific to this case was $(O(\log m \log(mn)), O(\log m))$ [13] while our general framework yields a slightly worse bound of

[§] We have not optimized the constants for specific polynomials, as our focus has been on getting a general positive result for a large family of cost functions

[¶] A bi-criteria competitive ratio of (α, β) implies that the online algorithm produces a schedule of expected startup cost at most $\alpha\mathbf{C}$ and expected p-norm of machine loads at most $\beta\mathbf{L}$, if there exists a feasible solution with startup cost $\mathbf{C}$ and p-norm of loads $\mathbf{L}$.

$(O(\log m \log(mn)), O(\log^2 m (\log n)^{1/\log m}))$.[‖] For all other p-norms, including the 1-norm, no previous result was known, not only in the online setting but also offline [29], [30], [31].

(4) Capacity Constrained Facility Location (CCFL). We are given m potential facility locations, each with an opening cost c_i and a capacity u_i. Now, n clients arrive online, each client $j \in [n]$ having an assignment cost a_{ij} and a demand b_{ij} for each facility $i \in [m]$. The online algorithm must open facilities (paying the opening costs c_i) and assign each arriving client j to an open facility i (paying the assignment cost a_{ij}, and incurring a load p_{ij} on facility i). The *congestion* of an assignment is the maximum load on any facility. The objective in CCFL is to simultaneously minimize the sum of opening costs and assignment costs, and the maximum congestion of any facility. We obtain an $O(\log^2 m \log mn)$-competitive randomized online algorithm. This competitive ratio is worse by a logarithmic factor than the best result known [13], but follows easily from our general framework.

(5) Capacitated Multicast Problem (CMC). This is a common generalization of CCFL and the *online multicast problem* [10]. There are m edge-disjoint rooted trees $T_1, \cdots, T_m$ corresponding to multicast trees in some network. Each tree T_i has a *capacity* u_i, and each edge $e \in \cup_{i=1}^m T_i$ has an opening cost c_e. A sequence of n clients arrive online, and each must be assigned to one of these trees. Each client j has a tree-dependent load of p_{ij} for tree T_i, and is connected to exactly one vertex π_{ij} in tree T_i. Thus, if client j is assigned to tree T_i then the load of T_i increases by p_{ij}, and all edges on the path in T_i from π_{ij} to its root must be opened. The objective is to minimize the total cost of opening the edges, subject to the capacity constraints that the total load on tree T_i is at most u_i. Solving a natural fractional convex relaxation, and then applying a suitable randomized rounding to it, we get an $O(\log^2 m \log mn)$-competitive randomized online algorithm that violates each capacity by an $O((d + \log^2 m) \log mn)$ factor; here d is the maximum depth of the trees $\{T_i\}_{i=1}^m$. The capacitated multicast problem with depth $d = 2$ trees generalizes the CCFL problem, in which case we recover the above result for CCFL.

(6) Online Set Cover with Set Requests (SCSR). We are given a universe U of n *resources*, and a collection of m *facilities*, where each facility $i \in [m]$ is specified by (i) a subset $S_i \subseteq U$ of resources (ii) opening cost c_i and (iii) capacity u_i. The resources and facilities are given up-front. Now, a sequence of k *requests* arrive over time. Each request $j \in [k]$ requires some subset $R_j \subseteq U$ of resources. The request has to be served by assigning it to some collection $F_j \subseteq [m]$ of facilities whose sets collectively cover R_j, i.e., $R_j \subseteq \cup_{i \in F_j} S_i$. Note that these facilities have to be open, and we incur their opening cost. Moreover, if facility i is used to serve client j, this contributes to the load of facility i, and this total load must be at most the capacity u_i. This problem was considered recently by Bhawalkar *et al.* [27]. Using an approach identical to that for the CCFL problem, we get an $O(\log^2 m \log mnk)$-competitive randomized online algorithm that violates each capacity by an $O(\log^2 m \log mnk)$ factor. Again this factor is weaker than the best result known by a logarithmic factor, but directly follows from our general framework.

II. Preliminaries

For a positive integer n, we denote $[n] := \{1, 2, \ldots, n\}$. We use x to denote a column vector. For two vectors a and b of the same dimension, we write $a \geq b$ if each coordinate of a is at least the corresponding coordinate of b. We use $\langle a, b \rangle = a^\intercal b$ to denote the dot product of a and b. Let $\mathbf{0}$ and $\mathbf{1}$ denote the all zero's and all one's vectors, respectively. A function $g : \mathbb{R}_+^n \to \mathbb{R}_+^m$ is *monotone* if $a \leq b$ implies that $g(a) \leq g(b)$. For $i \in [n]$, we use e_i to denote the unit vector whose ith coordinate is 1.

Fenchel Conjugate: Given a function $f : \mathbb{R}_+^n \to \mathbb{R}_+$, its Fenchel conjugate $f^\star : \mathbb{R}_+^n \to \mathbb{R}_+$ is defined as

$$f^\star(z) = \sup_{x \in \mathbb{R}_+^n} \big\{ \langle x, z \rangle - f(x) \big\}. \qquad \text{(II.3)}$$

It is well known (e.g., [28]) that the supremum is achieved at a gradient of f, and thus $\langle x, z \rangle - f(x)$ is the negative of the y-intercept of a hyperplane that passes through point x and has gradient z. The Fenchel conjugate can also be interpreted as representing f by a function that relates slopes of tangents to their y-intercept. As an example, if $f(x) = \frac{1}{p} x^p$ is a degree-p polynomial ($p > 1$), then $f^\star(z) = (1 - \frac{1}{p}) z^{\frac{p}{p-1}}$ is a degree-$\frac{p}{p-1}$ polynomial. For the rest of this paper, we will focus on the following type of functions f.

Assumption 3 (Nice function)**.** *Function $f : \mathbb{R}_+^n \to \mathbb{R}_+$ is convex, monotone, differentiable, and $f(\mathbf{0}) = 0$.*

In this case, the conjugate function $f^\star$ satisfies the following properties:

1) The conjugate $f^\star$ is non-negative, monotone, convex, and $f^\star(\mathbf{0}) = 0$.
2) If f is lower semi-continuous, $f^{\star\star} = f$.

We will use the following properties of nice functions.

Lemma 4 (Bounded Growth)**.** *Suppose that function f satisfies Assumption 3 and there is some $p \geq 1$ such that $\langle \nabla f(x), x \rangle \leq p \cdot f(x)$ for all $x \in \mathbb{R}_+^n$. Then, the following statements hold.*

(a) For $\delta \geq 1$, for $x \in \mathbb{R}_+^n$, $f(\delta x) \leq \delta^p f(x)$.
(b) For all $x \in \mathbb{R}_+^n$, $f^\star(\nabla f(x)) = \langle x, \nabla f(x) \rangle - f(x) \leq (p-1) \cdot f(x)$.
(c) If $p > 1$ then for any $0 < \gamma \leq 1$, $z \in \mathbb{R}_+^n$, $f^\star(\gamma z) \leq \gamma^{\frac{p}{p-1}} \cdot f^\star(z)$ and $f^\star(\gamma \cdot \nabla f(z)) \leq \gamma^{\frac{p}{p-1}} \cdot (p-1) \cdot f(z)$.
(d) If $p = 1$ then for any $0 < \gamma \leq 1$, $z \in \mathbb{R}_+^n$, $f^\star(\gamma \cdot \nabla f(z)) \leq 0$.

Proof. For statement (a), define the function $g : [1, +\infty) \to \mathbb{R}$ by $g(\theta) := \ln f(\theta x)$. Then, $g'(\theta) = \frac{\langle \nabla f(\theta x), x \rangle}{f(\theta x)} \leq \frac{p}{\theta}$, where the last inequality follows because $\langle f(\theta x), \theta x \rangle \leq p f(\theta x)$.

[‖] Note that the $(\log n)^{1/\log m}$ is a superconstant only if n is super-exponential in m. So, for polynomial instances, the competitive ratio is $(O(\log m \log(mn)), O(\log^2 m))$, which only has an additional $\log m$ term in the competitive ratio of the makespan compared to [13].

Integrating over $\theta \in [1, \delta]$ gives $g(\delta) - g(1) \leq p \ln \delta$, which is equivalent to $f(\delta x) \leq \delta^p f(x)$.

For statement (b), observe that $f^\star(\nabla f(x)) = \sup_{w \in \mathbb{R}^n_+} h(w)$, where $h(w) := \langle w, \nabla f(x) \rangle - f(w)$. Hence, $\nabla h(w) = \nabla f(x) - \nabla f(w)$. Since f is a convex function, h is a concave function. Also $\nabla h(x) = 0$. It follows that the (global) supremum of h is at $w = x$, and so $f^\star(\nabla f(x)) = \langle x, \nabla f(x) \rangle - f(x)$. This is at most $(p-1) \cdot f(x)$ from the assumption on f.

For statement (c), for $0 < \gamma \leq 1$, let $\delta := (\frac{1}{\gamma})^{\frac{1}{p-1}} \geq 1$,

$$\begin{aligned} f^\star(\gamma \mathbf{z}) &= \sup_{\mathbf{x} \in R^n_+} \{\langle \mathbf{x}, \gamma \mathbf{z} \rangle - f(\mathbf{x})\} \\ &= \gamma^{\frac{p}{p-1}} \sup_{\mathbf{x} \in R^n_+} \{\langle \delta \mathbf{x}, \mathbf{z} \rangle - \delta^p f(\mathbf{x})\} \\ &\leq \gamma^{\frac{p}{p-1}} \sup_{\mathbf{x} \in R^n_+} \{\langle \delta \mathbf{x}, \mathbf{z} \rangle - f(\delta \mathbf{x})\} \\ &= \gamma^{\frac{p}{p-1}} f^\star(\mathbf{z}) \ , \end{aligned}$$

where the inequality follows from statement (a). This proves the first part of (c). The second part follows by combining this with (b).

For statement (d), we have $p = 1$ and:

$$\begin{aligned} f^\star(\gamma \cdot \nabla f(z)) &= \sup_{\mathbf{x} \in R^n_+} \{\langle \mathbf{x}, \gamma \cdot \nabla f(z) \rangle - f(\mathbf{x})\} \\ &\leq \gamma \cdot \sup_{\mathbf{x} \in R^n_+} \{\langle \mathbf{x}, \nabla f(z) \rangle - f(\mathbf{x})\} \\ &= \gamma \cdot f^\star(\nabla f(z)) \ \leq \ 0 \end{aligned}$$

The first inequality is since $f \geq 0$ and $\gamma \leq 1$, and the second inequality is by part (b). □

Convex Covering/Packing Framework: We consider the following convex covering problem:

$$\min_{x \in \mathbb{R}^n_+} \quad C(x) := f(x) \quad \text{subject to} \quad Ax \geq \mathbf{1}. \qquad \text{(II.4)}$$

Above, $f : \mathbb{R}^n_+ \to \mathbb{R}$ is a monotone convex function and $A_{m \times n}$ is non-negative. Let $f^\star$ be the Fenchel dual function of f. The (Fenchel) dual problem of (II.4) is the following packing problem**:

$$\max_{y \in \mathbb{R}^m_+} \quad P(y) := \textstyle\sum_{j \in [m]} y_j - f^\star\big(A^T y\big). \qquad \text{(II.5)}$$

For definitions of the online versions of the packing/covering problems, see the Introduction. The following lemma is a standard weak duality result.

Lemma 5 (Weak Duality)**.** *For any $x \in \mathbb{R}^n_+$ such that $Ax \geq 1$ and any $y \in \mathbb{R}^m_+$, we have $C(x) \geq P(y)$, which holds even if f is not convex.*

Proof. Let x, y be feasible solutions to problems (II.4) and (II.5) respectively. Then,

$$\begin{aligned} f(x) &\geq f(x) + y^t\,(\mathbf{1} - Ax) \\ &= \sum_{j \in [m]} y_j - \big(x^t A^T y - f(x)\big) \\ &\geq \sum_{j \in [m]} y_j - \sup_{x \in \mathbb{R}^n_+} \big(x^t A^T y - f(x)\big) \\ &= \sum_{j \in [m]} y_j - f^\star\big(A^T y\big). \end{aligned}$$

The first inequality follows by the feasibility of x and since $y \geq 0$. The second inequality follows as $x \geq 0$. □

III. Online Covering Framework

We give here a primal-dual algorithm for the online convex packing/covering problem, and prove Theorem 1. The algorithm is presented in Figure III and is very simple; in fact, even simpler than the optimal algorithm for the linear case [18].

Let x denote the primal solution and y the dual solution. The algorithm is presented in Figure III. The primal update rule is particularly simple (see the first three steps): effectively, it is a multiplicative increase in the primal variables, where the rate of increase of x_i is inversely proportional to the i^{th} coordinate of the gradient at x. The dual update in Step 4 is just as simple: it increases the current dual variable at constant rate; here $0 < \delta < 1$ is a parameter which we optimize later. The monotonicity of the dual is immediate.

Finally, we also maintain "shadow dual" variables denoted by z in Step 5 which are used only for the primal analysis.†† Note that this step both increases some shadow duals and decreases others. This non-monotonicity of the shadow dual is essential for proving the primal bound in Theorem 1(a).

For the analysis, we denote by $x^\tau, y^\tau, \cdots$ the values of the variables at time τ. Let $\overline{x}$ be the final value of x at the end of the execution. We first analyze the algorithm with the more complex update step 5 and then state the minor changes required to analyze the dual.

Observation 6 (Feasible Solutions)**.** *The algorithm maintains a feasible monotonically non-decreasing primal solution x. Also, the duals y and z are both feasible.*

To analyze the competitive ratio we first state the following claim whose proof is deferred to the full version of the paper.

Claim 7 (Lower-bounding x variables)**.** *For a variable x_i, let $T_i = \{j \mid a_{ji} > 0\}$ and let S_i be any subset of T_i. If the current dual variable z_k increases at rate r (and other dual variables may decrease), then*

$$x_i^\tau \geq \frac{1}{\max_{j \in S_i}\{a_{ji}\} \cdot d} \left(\exp\left(\frac{1}{r \cdot \nabla_i f(\overline{x})} \sum_{j \in S_i} a_{ji} z_j^\tau \right) - 1 \right). \qquad \text{(III.7)}$$

Proof of Theorem 1. First, we prove the competitive ratio of the primal algorithm. For this, we first claim that for any time τ, the variables z satisfy $A^t z \leq \delta \nabla f(\overline{x})$, i.e. for each i, $\sum_{j=1}^k a_{ji} z_j^\tau \leq \delta\, \nabla_i f(\overline{x})$. Indeed, while processing constraint

** Variable z appearing in (I.2) is redundant here.

†† The shadow variables z are unrelated to the z-variables in (I.2).

Fractional Algorithm: When the k^{th} request $\sum_{i=1}^{n} a_{ki} x_i \geq 1$ arrives:

1) Let τ be a continuous variable denoting the current time.
2) While the new constraint is unsatisfied, i.e., $\sum_{i=1}^{n} a_{ki} x_i < 1$, increase τ at rate 1 and:
3) **Change primal variables:**
   - For each i with $a_{ki} > 0$, increase each x_i at rate
   $$\frac{\partial x_i}{\partial \tau} = \frac{a_{ki}\, x_i + \frac{1}{d}}{\nabla_i f(x)}. \tag{III.6}$$
   Here d is an upper bound on the row sparsity of the matrix and $\nabla_i f(x)$ is the i^{th}-coordinate of the gradient $\nabla f(x)$.
4) **Change dual variables:**
   - Increase y_k at rate $w = \frac{\delta}{\log(1+d\cdot\rho)}$. Here ρ is an upper bound on the maximum-to-minimum ratio of positive entries in any column of A.
5) **Change "shadow" dual variables (only for primal analysis):**
   - Increase z_k at rate $r = \frac{\delta}{\log(1+2d^2)}$.
   - If for variable x_i we have $\sum_{j=1}^{k} a_{ji} z_j = \delta \cdot \nabla_i f(\overline{x})$, then
     - Let $m_i^\star = \arg\max_{j=1}^{k} \{a_{ji} \mid z_j > 0\}$.
     - Increase $z_{m_i^\star}$ at rate $-\frac{a_{ki}}{a_{m_i^\star i}} \cdot r$. (Note: this change occurs only if a_{ki} is strictly positive.)

Fig. III.1. Algorithm for Convex Covering

k, if $\sum_{j=1}^{k} a_{ji} z_j^\tau < \delta \cdot \nabla_i f(\overline{x})$ for column i we are trivially satisfied. Suppose that during the processing of constraint k, we have $\sum_{j=1}^{k} a_{ji} z_j^\tau = \delta\, \nabla_i f(\overline{x})$ for some dual constraint i and time τ. Now the dual decrease part of the algorithm kicks in, and the rate of increase in the left-hand side of the dual constraint is at most:

$$\frac{d}{d\tau}\left(\sum_{j=1}^{k} a_{ji} z_j^\tau\right) = a_{ki} \cdot r - a_{m_i^\star i} \cdot \frac{a_{ki}}{a_{m_i^\star i}} \cdot r = 0.$$

Now consider the update when primal constraint k arrives and τ is the current time. Let $U(\tau)$ denote the set of indices i for which we have $a_{ki} > 0$ and $\sum_{j=1}^{k} a_{ji} z_i^\tau = \delta\, \nabla_i f(\overline{x})$. So $|U(\tau)| \leq d$, the row-sparsity of A. Moreover, define $S_i := \{j \mid a_{ji} > 0, z_j^\tau > 0\}$ for every $i \in U(\tau)$. Clearly, $\sum_{j \in S_i} a_{ji} z_j^\tau = \sum_{j=1}^{k} a_{ji} z_j^\tau = \delta \cdot \nabla_i f(\overline{x})$. Plugging in $r = \frac{\delta}{\ln(1+2d^2)}$ into Claim 7, and using the fact that $\sum_i a_{ki} x_i^\tau < 1$, we get for every $i \in U(\tau)$,

$$\frac{1}{a_{ki}} > x_i^\tau \geq \frac{1}{\max_{j \in S_i}\{a_{ji}\} \cdot d}\left(\exp\left(\ln(1+2d^2)\right) - 1\right),$$

and after simplifying we get $\frac{a_{ki}}{a_{m_i^\star i}} = \frac{a_{ki}}{\max_{j \in S_i}\{a_{ji}\}} \leq \frac{1}{2d}$. As a result, we can bound the rate of change in the dual expression $\sum_{j=1}^{k} z_j$ at any time τ:

$$\begin{aligned}\frac{d\left(\sum_{j=1}^{k} z_j\right)}{d\tau} &= r - \sum_{i \in U(\tau)} \frac{a_{ki}}{a_{m_i^\star i}} \cdot r \\ &\geq r\left(1 - \sum_{i \in U(\tau)} \frac{1}{2d}\right) \geq \frac{1}{2} r \end{aligned} \tag{III.8}$$

where the last inequality follows as $|U(\tau)| \leq d$. On the other hand, when processing constraint k during the execution of the algorithm, the rate of increase of the primal objective f is:

$$\begin{aligned}\frac{df(x^\tau)}{d\tau} &= \sum_i \nabla_i f(x^\tau) \frac{\partial x_i^\tau}{\partial \tau} = \sum_{i | a_{ki} > 0} \nabla_i f(x^\tau)\left(\frac{a_{ki} x_i^\tau + \frac{1}{d}}{\nabla_i f(x^\tau)}\right) \\ &= \sum_{i | a_{ki} > 0}\left(a_{ki} x_i^\tau + \frac{1}{d}\right) \leq 2. \end{aligned} \tag{III.9}$$

The final inequality uses the fact that the covering constraint is unsatisfied, and that d is at least the number of non-zeroes in the vector a_k. From (III.8) and (III.9) we can now bound the following primal-dual ratio:

$$\frac{d\left(\sum_{j=1}^{k} z_j^\tau\right)}{df(x^\tau)} \geq \frac{r}{4} = \frac{\delta}{4 \ln\left(1+2d^2\right)}. \tag{III.10}$$

Thus, if $\overline{z}$ is the final dual solution we get,

$$\sum_{i=1}^{m} \overline{z}_i \geq \frac{\delta}{4 \ln\left(1+2d^2\right)} \cdot f(\overline{x}). \tag{III.11}$$

To complete the proof of Theorem 1, we use Lemma 4. We have for any $0 \leq \delta \leq 1$:

$$f^*(A^t \overline{z}) \leq f^*(\delta \cdot \nabla f(\overline{x})) \leq \begin{cases} \delta^{\frac{p}{p-1}} \cdot (p-1) \cdot f(\overline{x}) & \text{if } p > 1 \\ 0 & \text{if } p = 1 \end{cases}$$

The first inequality is by monotonicity of $f^\star$ and our claim that $A^t z \leq \delta \nabla f(\overline{x})$. The second inequality is by Lemma 4 parts (c-d). Combining with (III.11), $D =$

$$\sum_{i=1}^{m} \overline{z}_i - f^*(A^t \overline{z}) \geq \left(\frac{\delta}{4 \ln\left(1+2d^2\right)} - \delta^{\frac{p}{p-1}} \cdot (p-1)\right) \cdot f(\overline{x})$$

if $p > 1$, and $D \geq \frac{\delta}{4 \ln(1+2d^2)} f(\overline{x})$ if $p = 1$. The optimal choice of δ is $\frac{1}{(4p \ln(1+2d^2))^{p-1}}$. Plugging in this value we get:

$$f(\overline{x}) \leq \left(4p \ln\left(1+2d^2\right)\right)^p D \leq \left(4p \ln\left(1+2d^2\right)\right)^p f(OPT).$$

Hence the proof of the primal algorithm.

Next, we prove the primal-dual competitive ratio using the monotone dual update in step 4. Applying Claim 7 with $w = \frac{\delta}{\ln(1+d\rho)}$ (instead of r) with the final (feasible) primal and dual solutions $\overline{x}, \overline{y}$ and with $S_i := \{j \mid a_{ji} > 0\}$ to get

$$\overline{x}_i \geq \frac{1}{\max_{j \in S_i}\{a_{ji}\} \cdot d} \left(\exp\left(\frac{\ln(1+d\rho)}{\delta \cdot \nabla_i f(\overline{x})} \sum_{j \in S_i} a_{ji} \overline{y}_j \right) - 1 \right).$$

Note that $\overline{x}_i \leq \frac{1}{\min_{j \in S_i}\{a_{ji}\}}$ (since all constraints that contain x_i would be satisfied at the upper bound). Simplifying and using that $\rho \geq \frac{\max_{j \in S_i}\{a_{ji}\}}{\min_{j \in S_i}\{a_{ji}\}}$ we get, $\sum_{j \in S_i} a_{ji}\overline{y}_j \leq \delta \cdot \nabla_i f(\overline{x})$, or equivalently $A^t \overline{y} \leq \delta \cdot \nabla f(\overline{x})$. Using the same arguments as above we get, $\sum_{i=1}^m \overline{y}_i \geq \frac{\delta}{2\ln(1+d\rho)} \cdot f(\overline{x})$. Plugging it in we get,

$$\begin{aligned} D &= \sum_{i=1}^{m} \overline{y}_i - f^*(A^t \overline{y}) \\ &\geq \left(\frac{\delta}{2\ln(1+d\cdot\rho)} - \delta^{\frac{p}{p-1}} \cdot (p-1) \right) \cdot f(\overline{x}). \end{aligned}$$

Optimizing, we get $\delta = \frac{1}{(2p\ln(1+d\cdot\rho))^{p-1}}$ yielding a competitive ratio $O(p\log(d\rho))^p$ for the dual. □

A. A Lower Bound for Online Convex Covering

In this section we adapt the example in Azar *et al.* [13] for the ℓ_∞ norm to the ℓ_p norm and showing that our competitive ratio for the convex covering problem is optimal up to constants.

Theorem 8 (Lower bound for ℓ_p^p-norm objective function)**.** *There exists an instance of the online covering problem with an ℓ_p^p-norm objective function f for which* any *online algorithm is $\Omega(p\log d)^p$-competitive, where d is the maximal sparsity of the covering matrix.*

Proof. For parameters p and d, the example has $2^{p+1} - 2$ pairwise disjoint sets (blocks) of d variables each. We use B_i to refer to the ith block. The objective function f is a polynomial of degree p composed of 2^p sums of variables each raised to the power of p. To define the sums let us consider a complete binary tree of depth p. Each node in this tree except the root corresponds to a unique block. Each term in f corresponds to a path from the root to a leaf node k, and is of the form $\left(\sum_{x \in \cup_{i \in Q_k} B_i} x\right)^p$ where Q_k is the set of blocks encountered on the path (excluding the root, which is not a block).

In [13], there is a procedure for revealing covering constraints of sparsity at most $2d$ of two blocks such that:

- the sum of variables in the online algorithm for one of the blocks is at least $H_d/2$, where H_d refers to the dth harmonic number, and
- there is a feasible solution that sets a single variable in the other block to 1.

First, we apply the procedure to the two children of the root. Next, the procedure is applied to the children of the block with the larger sum of variables ($\geq H_d/2$) in the algorithmic solution. This continues until we reach one of the leaves. Then, there is path from root to leaf where p blocks each have a total variable value of at least $H_d/2$ in the algorithmic solution. On the other hand, the optimal solution only sets one variable to 1 in each root to leaf path. Therefore, the competitive ratio of the algorithm is at least $\frac{\left(p\frac{H_d}{2}\right)^p}{2^p} = \Omega(p\log d)^p$. □

B. An Application: ℓ_p-norm Set Cover

In this section we consider a generalization of the online set-cover problem introduced in [5]. In our problem we are given n sets $\{S_j\}_{j=1}^n$ over some ground set U. Apart from the set system, we are also given K cost functions $b_k : [n] \rightarrow \mathbb{R}_+$ for $k \in [K]$. Elements from U arrive online and must be covered by some set upon arrival; the decision to select a set into the solution is irrevocable. The goal is to maintain a set-cover that minimizes the ℓ_p norm of the K cost functions. We use the above fractional online algorithm along with a rounding scheme (similar to [23]) to obtain the following.

Theorem 9 (ℓ_p-norm Set Cover)**.** *There is an $O\left(\frac{p^3}{\log p}\log d \log r\right)$-competitive randomized online algorithm for set cover minimizing the ℓ_p-norm of multiple cost-functions. Here d is the maximum number of sets containing any element, and $r = |U|$ is the number of elements.*

Proof. We use the following convex relaxation. There is a variable x_j for each set $j \in [n]$ which denotes whether this set is chosen.

$$\begin{aligned} \min \quad & f(x) \;=\; \sum_{k=1}^{K} \left(\sum_{j=1}^{n} b_{kj} \cdot x_j \right)^p + \sum_{j=1}^{n} \left(\sum_{k=1}^{K} b_{kj}^p \right) \cdot x_j \\ s.t. \quad & \sum_{j : e \in S_j} x_j \geq 1, \qquad \forall e \in U \\ & x \geq 0. \end{aligned}$$

Since f satisfies $\langle \nabla f(x), x \rangle \leq p\, f(x)$, we can use our framework to solve this fractional convex covering problem online, with a cost at most $O(p\log d)^p$ times the optimal fractional solution. Moreover, if $C^\star$ denotes the p^{th} power of the optimal value of the objective of the given set cover instance, then the optimal value of the above fractional relaxation is at most $2C^\star$, and hence the value of our fractional online solution x is $f(x) = O(p\log d)^p \cdot C^\star$.

To get an integer solution, we use a simple online randomized rounding algorithm. For each set $j \in [n]$, define X_j to be a $\{0,1\}$-random variable with $\Pr[X_j = 1] = \min\{4p\log r \cdot x_j, 1\}$. This can easily be implemented online. It is easy to see by a Chernoff bound that for each element e, it is not covered with probability at most $\frac{1}{r^{2p}}$. If an element e is not covered by this rounding, we choose the set minimizing $\min_{j=1}^n\{\sum_{k=1}^K b_{kj}^p : e \in S_j\}$; let $\overline{e} \in [n]$ index this set and $C_{\overline{e}} = \sum_{k=1}^K b_{k\overline{e}}^p$. Observe that $C_{\overline{e}} \leq C^\star$ for all $e \in U$.

To bound the ℓ_p-norm of the cost, let $C_k = \sum_{j=1}^n b_{kj} \cdot X_j$ be the cost of the randomly rounded solution under the k^{th}

cost function, and let $C := \sum_{k=1}^{K} C_k^p$. Also, for each element $e \in U$ and $k \in [K]$, define:

$$D_{ek} = \begin{cases} b_{k\overline{e}} & \text{If } e \text{ is not covered} \\ 0 & \text{Otherwise.} \end{cases}$$

$$D_e = \begin{cases} C_{\overline{e}} & \text{If } e \text{ is not covered} \\ 0 & \text{Otherwise.} \end{cases}$$

Note that $D_e = \sum_{k=1}^{K} D_{ek}^p$. The p^{th} power of the objective function is:

$$\begin{aligned} \overline{C} &= \sum_{k=1}^{K} \left(C_k + \sum_{e \in U} D_{ek} \right)^p \\ &\leq 2^p \sum_{k=1}^{K} C_k^p + 2^p \sum_{k=1}^{K} \left(\sum_{e \in U} D_{ek} \right)^p \\ &\leq 2^p \cdot C + 2^p \sum_{k=1}^{K} r^p \sum_{e \in U} D_{ek}^p \\ &= 2^p \cdot C + (2r)^p \sum_{e \in U} D_e \qquad \text{(III.12)} \end{aligned}$$

We now bound $\mathbb{E}[\overline{C}]$ using (III.12). Observe that $\mathbb{E}[C_k] \leq 4p \log r \cdot \sum_{j=1}^{n} b_{kj} \cdot x_j$. Since each C_k is the sum of independent non-negative random variables, we can bound $\mathbb{E}[C_k^p]$ using a concentration inequality involving p^{th} moments [32]:

$$\begin{aligned} \mathbb{E}[C_k^p] &\leq K_p \cdot \left(\mathbb{E}[C_k]^p + \sum_{j=1}^{n} \mathbb{E}[b_{kj}^p \cdot X_j^p] \right) \\ &\leq K_p \cdot \left((4p \log r)^p \left(\sum_{j=1}^{n} b_{kj} \cdot x_j \right)^p \right. \\ &\qquad \left. + \; 4p \log r \sum_{j=1}^{n} b_{kj}^p \cdot x_j \right). \end{aligned}$$

Above $K_p = O(p/\log p)^p$. By linearity of expectation,

$$\begin{aligned} \mathbb{E}[C] &= \sum_{k=1}^{K} \mathbb{E}[C_k^p] \\ &\leq K_p (4p \log r)^p \sum_{k=1}^{K} \left(\left(\sum_{j=1}^{n} b_{kj} \cdot x_j \right)^p + \sum_{j=1}^{n} b_{kj}^p \cdot x_j \right) \\ &= K_p (4p \log r)^p \cdot f(x). \end{aligned}$$

Thus we have $\mathbb{E}[C] = O\left(\frac{p^3}{\log p} \cdot \log d \cdot \log r \right)^p \cdot C^\star$.

Observe that $\mathbb{E}\left[\sum_{e \in U} D_e \right] = \sum_{e \in U} \Pr[e \text{ uncovered}] \cdot C_{\overline{e}} \leq r^{-2p} \cdot \sum_{e \in U} C^\star = r^{1-2p} \cdot C^\star$. Using these bounds in (III.12), we have $\mathbb{E}[\overline{C}] \leq 2^p \cdot \mathbb{E}[C] + (2r)^p \sum_{e \in U} \mathbb{E}[D_e] = O\left(\frac{p^3}{\log p} \cdot \log d \cdot \log r \right)^p \cdot C^\star$. □

IV. Online Packing Framework

In this section, we consider the convex online packing problem. We consider the case that $f^\star$ is a convex multi-variate polynomial with non-negative coefficients, zero constant term and maximum degree q. We shall state exactly what properties of $f^\star$ are needed for our proofs. Although there are more general objective functions for which our packing algorithm can work (for instance $f^\star(z) = \sum_{i=1}^{n} z_i^{1.5}$), we think that for ease of exposition, it is simpler to restrict ourselves to the case of polynomials (with integral degrees).

```
Fractional Packing Algorithm: Initialize $x := z := 0$. When the $k^{th}$ constraint vector $a_k = (a_{k1}, \ldots, a_{kn})$ arrives in round $k$:
1) Set $y_k := 0$.
2) While $\sum_{i=1}^{n} a_{ki} x_i < 1$, do:
    • Continuously increase $y_k$.
    • Simultaneously for each $i \in [n]$, increase $z_i$ at rate $\frac{dz_i}{dy_k} = a_{ki}$.
    • Increase $x$ according to $x = \nabla f^\star(\rho z)$.
```

Fig. IV.2. Algorithm for super-linear $f^\star$

The reader can check that the special class of polynomials we describe above satisfy the first two conditions in the following Assumption 10.

Assumption 10 (Nice function $f^\star$). *1) The function $f^\star$ is convex, monotone, and $f^\star(\mathbf{0}) = 0$.*

2) The function $f^\star$ is differentiable, and the gradient $\nabla f^\star$ is monotone. Moreover, there is some $q > 0$ such that for all $z \in \mathbb{R}_+^n$, $\langle \nabla f^\star(z), z \rangle \leq q \cdot f^\star(z)$

3) The offline convex packing problem has bounded optimal objective. ‡‡

We first obtain an algorithm for the special case when $f^\star$ is "super-linear", which means, in addition to Assumption 10, suppose the packing cost function $f^\star$ satisfies the following condition. There exists some $\lambda > 1$ such that for all $\rho \geq 1$, for all $z \in \mathbb{R}_+^n$, $\nabla f^\star(\rho z) \geq \rho^{\lambda - 1} \cdot \nabla f^\star(z)$.

Here, the vector x plays an auxiliary role and is initialized to 0. Throughout the algorithm, we maintain the invariant $z = A^T y$ and $x = \nabla f^\star(\rho z)$ for some parameter $\rho > 1$ to be determined later. In round $k \in [m]$, the vector $a_k = (a_{k1}, a_{k2}, \ldots, a_{kn})$ is given. The variable y_k is initialized to 0, and is continuously increased while $\sum_{i \in [n]} a_{ki} x_i < 1$. To maintain $z = A^T y$, for each $i \in [n]$, z_i is increased at rate $\frac{dz}{dy_k} = a_{ki}$. As the coordinates of z are increased, the vector x is increased according to the invariant $x = \nabla f^\star(\rho z)$. Since $\nabla f^\star$ is monotone, as y_k increases, both z and z increase monotonically. We show in Lemma 11 that unless the offline packing problem is unbounded, eventually $\sum_{i \in [n]} a_{ki} x_i$ reaches 1, at which moment y_k stops increasing and round k finishes.

Observe that the coordinates of x are increased monotonically throughout the algorithm. Below we show that at the end of the process, the constraints $\sum_{i \in [n]} a_{ji} x_i \geq 1$ are satisfied

‡‡Note that there are instances for which the offline convex packing problem whose objective can be arbitrarily large. Consider, for example, a convex packing problem with a linear cost function $f^\star(z) = \frac{1}{2n} \sum_{i \in [n]} z_i$ and the first request is $a_1 = 1$. Then, by letting y_1 to be arbitrarily large, we can get a feasible packing assignment with arbitrarily large objective. Obviously, such instances are not interesting for practical purposes. Hence, we will assume that the offline optimal is bounded.

for all $j \in [m]$. Hence, the vector x is feasible for the covering problem.

In the rest of the section, for $k \in [m]$, we let $z^{(k)}$ denote the vector z at the end of round k, where $z^{(0)} := 0$.

Lemma 11 (Dual Feasibility)**.** *Recall our assumption that the offline optimal packing objective is bounded. Then, in each round $k \in [m]$, eventually we have $\sum_{i\in[n]} a_{ki}x_i \geq 1$, and y_k will stop increasing.*

Proof. During round $k \in [m]$, the algorithm increases y_k only when $\sum_{i=1}^{n} a_{ki}x_i < 1$. Therefore, recalling $z = A^T y$, when the algorithm increases y_k, it also increases each z_i at rate $\frac{dz_i}{dy_k} = a_{ki}$.

Hence, we have

$$\begin{aligned}\tfrac{\partial P(y)}{\partial y_k} &= 1 - \langle a_k, \nabla f^\star(z)\rangle \geq 1 - \tfrac{1}{\rho^{\lambda-1}} \cdot \langle a_k, \nabla f^\star(\rho z)\rangle \\ &= 1 - \tfrac{1}{\rho^{\lambda-1}} \cdot \langle a_k, x\rangle \geq 1 - \tfrac{1}{\rho^{\lambda-1}},\end{aligned}$$

where the first inequality follows from the assumption $\nabla f^\star(\rho z) \geq \rho^{\lambda-1} \cdot \nabla f^\star(z)$, and the last inequality follows because $\langle a_k, x\rangle < 1$ when y_k is increased.

Therefore, suppose for contrary that $\langle a_k, x\rangle$ never reaches 1, then the objective function $P(y)$ increases at least at some positive rate $1 - \frac{1}{\rho^{\lambda-1}}$ (recalling $\rho > 1$ and $\lambda \geq 2$) as y_k increases, which means the offline packing problem is unbounded, contradicting our assumption. □

Lemma 12 (Bounding Increase in y)**.** *For $k \in [m]$, let $z^{(k)}$ denote the vector z at the end of round k, where $z^{(0)} := 0$. Then, at the end of round k when y_k stops increasing (by Lemma 11)*

$$y_k \geq \tfrac{1}{\rho} \cdot \left(f^\star(\rho z^{(k)}) - f^\star(z^{(k-1)})\right) \ .$$

In particular, since $f^\star(0) = 0$, this implies that at the end of the algorithm,

$$\sum_{k\in[m]} y_k \geq \tfrac{1}{\rho} \cdot f^\star(\rho z^{(m)}) \ .$$

Proof. Recall again that y_k increases only when $\langle a_k, x\rangle < 1$, we have

$$1 \geq \sum_{i\in[n]} a_{ki}x_i = \sum_{i\in[n]} x_i \cdot \frac{dz_i}{dy_k} \ .$$

Hence, integrating this with respect to y_k throughout round k, and observing that $x = \nabla f^\star(\rho z)$, we have

$$y_k \geq \int_{z=z^{(k-1)}}^{z^{(k)}} \langle \nabla f^\star(\rho z), dz\rangle = \frac{1}{\rho}\cdot(f^\star(\rho z^{(k)}) - f^\star(z^{(k-1)})) \ ,$$

where the last equality comes from the fundamental theorem of calculus for path integrals of vector fields. □

Lemma 13 (Super-linear $f^\star$)**.** *For a super-linear function $f^\star$, by setting $\rho := \lambda^{\frac{1}{\lambda-1}}$ in the Fractional Packing Algorithm, we obtain an $O\big(\frac{q\lambda}{\lambda-1}\big)$-competitive online algorithm for the online convex packing problem.*

Proof. Suppose $z^{(m)} = A^T y$ is the vector at the end of the algorithm, and $x^{(m)} = \nabla f^\star(\rho z^{(m)})$. Then, we have

$$C(x^{(m)}) = f(\nabla f^\star(\rho z^{(m)})) \leq (q-1) \cdot f^\star(\rho z^{(m)}) \ ,$$

where the inequality follows from applying Lemma 4(c) with the roles of f and $f^\star$ reversed.

On the other hand,

$$P(y) = \textstyle\sum_{k\in[m]} y_k - f^\star(z^{(m)}) \geq \frac{1}{\rho} \cdot f^\star(\rho z^{(m)}) - f^\star(z^{(m)}) \ . \tag{IV.13}$$

Hence, it follows that

$$\begin{aligned}\frac{C(x)}{P(y)} &\leq \frac{(p-1) \cdot f^*(\rho z^{(m)})}{\frac{1}{\rho} \cdot f^*(\rho z^{(m)}) - f^*(z^{(m)})} && \text{(IV.14)}\\ &\leq \frac{(p-1) \cdot \rho^\lambda \cdot f^*(z^{(m)})}{\frac{1}{\rho} \cdot \rho^\lambda \cdot f^*(z^{(m)}) - f^*(z^{(m)})} && \text{(IV.15)}\\ &= (p-1) \cdot \frac{\rho^\lambda}{\rho^{\lambda-1} - 1} && \text{(IV.16)}\end{aligned}$$

where the penultimate inequality follows because the assumption $\nabla f^\star(\rho z) \geq \rho^{\lambda-1} \cdot \nabla f^\star(z)$ implies that $f^\star(\rho z^{(m)}) = \rho \int_{z=0}^{z^{(m)}} \langle \nabla f^\star(\rho z), dz\rangle \geq \rho^\lambda \int_{z=0}^{z^{(m)}} \langle \nabla f^\star(z), dz\rangle = \rho^\lambda \cdot f^\star(z^{(m)}))$, and the function $t \mapsto \frac{t}{\frac{t}{\rho} - f^\star(z^{(m)})}$ is decreasing.

Choosing $\rho := \lambda^{\frac{1}{\lambda-1}}$ and observing that $x^{(m)}$ is feasible for the covering problem, we have

$$\frac{P^{\text{opt}}}{P(y)} \leq \frac{C^{\text{opt}}}{P(y)} \leq \frac{C(x^{(m)})}{P(y)} \leq (q-1) \cdot \frac{\lambda^{\frac{\lambda}{\lambda-1}}}{\lambda - 1} \ .$$

The result then follows because $\lambda^{\frac{1}{\lambda-1}} = \big(1 + (\lambda-1)\big)^{\frac{1}{\lambda-1}} \leq e$. □

Then we discuss the a general function $f^\star$ that satisfies Assumption 10. Recall that we consider a polynomial $f^\star$ with non-negative coefficients and zero constant term. Hence, we apply the following transformation $f^\star(z) = \langle c, z\rangle + \widehat{f}^\star(z)$, where $c = \nabla f^\star(\mathbf{0})$. Observe that $\widehat{f}^\star(z)$ no longer has linear terms, thus the second condition of Lemma 13 always holds with $\lambda = 2$ (and hence $\rho = 2$).

Lemma 14 ($\widehat{f^*}$ has bounded growth and is superlinear.)**.** *We have the following.*

- *For all $z \in \mathbb{R}_+^n$, $\langle \nabla \widehat{f}^*(z), z\rangle \leq q \cdot \widehat{f}^*(z)$.*
- *For all $\rho \geq 1$ and $z \geq \mathbf{0}$, $\nabla \widehat{f}^\star(\rho z) \geq \rho \cdot \nabla \widehat{f}^\star(z)$.*

Proof. To prove the first statement, we can pick any non-zero term M in $\widehat{f}^*$. Since the coefficient of M is positive and its degree is at most q, it follows that $\langle \nabla M(z), z\rangle \leq q \cdot M(z)$. Therefore, summing up all the terms, we get $\langle \nabla \widehat{f}^*(z), z\rangle \leq q \cdot \widehat{f}^*(z)$.

For the second statement, if $\widehat{f}^*$ is the zero function, then the statement trivially holds. Otherwise, observe that in $\widehat{f}^*(x)$, only terms with degree at least 2 can have non-zero coefficients. Therefore, for $z \geq \mathbf{0}$ and $\rho \geq 1$, we have $\nabla \widehat{f}^\star(\rho z) \geq \rho \cdot \nabla \widehat{f}^\star(z)$, i.e., the condition in Lemma 13 is satisfied with $\lambda = 2$. □

In terms of $\widehat{f}^*$, the objective function becomes

$$P(y) = \langle 1, y\rangle - f^\star(A^T y) = \langle 1 - Ac, y\rangle - \widehat{f}^\star(A^T y) \ .$$

Moreover, the corresponding covering problem becomes $\min_{x \geq 0} \widehat{f}(x)$ subject to $Ax \geq 1 - Ac$. In other words, in round $k \in [m]$, as the vector a_k arrives, the covering constraint becomes $\langle a_k, x\rangle \geq b_k$, where $b_k := 1 - \langle a_k, c\rangle$. If $b_k \leq 0$, then the constraint is automatically satisfied, and we set $y_k = 0$ such that round k finishes immediately. Otherwise, we can run the Fractional Packing Algorithm using the function $\widehat{f}^\star$ and the constraint vector $\frac{a_k}{b_k}$ in round k.

Next, we present a proof of Theorem 2 based on the above discussion.

Proof of Theorem 2: As discussed above, we write $c := \nabla f^\star(\mathbf{0})$ and $\widehat{f}^\star(x) := f^\star(x) - \langle c, x\rangle$ as the convex function. (Observe that if $f^\star$ is a polynomial function containing only linear terms, then the problem is trivial because the objective is unbounded when there is some round $k \in [m]$ such that $b_k > 0$.)

For ease of exposition, we can assume that for each $k \in [m]$, $b_k := 1 - \langle a_k, c\rangle > 0$. Otherwise, we can essentially ignore the variable y_k by setting it to 0. We denote B as the diagonal matrix whose (k, k)-th entry is b_k. By writing $w := By$, the objective function can be expressed in terms of w as $\widehat{P}(w) := \langle 1, w\rangle - \widehat{f}^\star((B^{-1}A)^T w)$.

Hence, we can run the Fractional Packing Algorithm using function $\widehat{f}^\star$ such that in round $k \in [m]$, when the vector a_k arrives, we can transform it by dividing each coordinate by b_k before passing it to the algorithm.

By Lemmas 13 and 14, using $\lambda = 2$, it follows that the algorithm has competitive ratio $O(q)$. ■

As an application of the above fractional solver (with some additional ideas to ensure integral allocation), we can obtain a competitive algorithm for a generalized version of combinatorial auction where the seller may produce any number of copies of each item subject to a production cost function $f^\star$.

Theorem 15 (Combinatorial Auction with Production Cost)**.** *Suppose the production cost function $f^\star$ satisfies the conditions in Theorem 2. Then, there is an $O(q)$-competitive online algorithm for the online combinatorial auction problem with production cost $f^\star$, with an additive error that depends on $f^\star$ (but does not depends on the number of buyers). Specifically, the algorithm returns a solution with value at least $\Omega(\frac{\mathsf{OPT}}{q}) - V(f^\star)$, where $V(f^\star) := \frac{1}{\rho} \cdot (f^\star(\rho(q+1)\mathbf{1}) - f^\star(\rho\mathbf{1}))$ and $\rho = 2$.*

References

[1] Y. Azar, I. R. Cohen, and D. Panigrahi, "Online covering with convex objectives and applications," *arXiv preprint arXiv:1412.3507*, 2014.

[2] N. Buchbinder, S. Chen, A. Gupta, V. Nagarajan, and J. S. Naor, "Online packing and covering framework with convex objectives," *arXiv preprint arXiv:1412.8347*, 2014.

[3] T. H. Chan, Z. Huang, and N. Kang, "Online convex covering and packing problems," *CoRR*, vol. abs/1502.01802, 2015.

[4] N. Buchbinder and J. S. Naor, "The design of competitive online algorithms via a primal-dual approach," *Found. Trends Theor. Comput. Sci.*, vol. 3, no. 2-3, pp. 93–263, 2007. [Online]. Available: http://dx.doi.org/10.1561/0400000024

[5] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor, "The online set cover problem," *SIAM J. Comput.*, vol. 39, no. 2, pp. 361–370, 2009.

[6] N. Bansal, N. Buchbinder, and J. Naor, "A primal-dual randomized algorithm for weighted paging," *J. ACM*, vol. 59, no. 4, p. 19, 2012.

[7] ——, "Randomized competitive algorithms for generalized caching," *SIAM J. Comput.*, vol. 41, no. 2, pp. 391–414, 2012.

[8] ——, "Towards the randomized k-server conjecture: A primal-dual approach," in *SODA*, 2010, pp. 40–55.

[9] N. Bansal, N. Buchbinder, A. Madry, and J. Naor, "A polylogarithmic-competitive algorithm for the *k*-server problem," *J. ACM*, vol. 62, no. 5, p. 40, 2015.

[10] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. S. Naor, "A general approach to online network optimization problems," *ACM Trans. Algorithms*, vol. 2, no. 4, pp. 640–660, 2006.

[11] A. Ene, D. Chakrabarty, R. Krishnaswamy, and D. Panigrahi, "Online buy-at-bulk network design," in *FOCS*, 2015, pp. 545–562.

[12] N. Buchbinder and J. Naor, "Improved bounds for online routing and packing via a primal-dual approach," in *FOCS*, 2006, pp. 293–304.

[13] Y. Azar, U. Bhaskar, L. Fleischer, and D. Panigrahi, "Online mixed packing and covering," in *SODA*, 2013, pp. 85–100.

[14] Z. Huang and A. Kim, "Welfare maximization with production costs: A primal dual approach," in *SODA*. SIAM, 2015, pp. 59–72.

[15] N. R. Devanur, K. Jain, and R. D. Kleinberg, "Randomized primal-dual analysis of ranking for online bipartite matching," in *SODA*, 2013, pp. 101–107.

[16] N. Buchbinder, K. Jain, and J. Naor, "Online primal-dual algorithms for maximizing ad-auctions revenue," in *ESA*, 2007, pp. 253–264.

[17] N. Buchbinder and J. S. Naor, "Online primal-dual algorithms for covering and packing," *Math. Oper. Res.*, vol. 34, no. 2, pp. 270–286, 2009. [Online]. Available: http://dx.doi.org/10.1287/moor.1080.0363

[18] A. Gupta and V. Nagarajan, "Approximating sparse covering integer programs online," *Math. Oper. Res.*, vol. 39, no. 4, pp. 998–1011, 2014.

[19] N. Bansal, K. Pruhs, and C. Stein, "Speed scaling for weighted flow time," *SIAM J. Comput.*, vol. 39, no. 4, pp. 1294–1308, 2009.

[20] N. R. Devanur and Z. Huang, "Primal dual gives almost optimal energy efficient online algorithms." in *SODA*. SIAM, 2014, pp. 1123–1140.

[21] Y. Azar, N. R. Devanur, Z. Huang, and D. Panigrahi, "Speed scaling in the non-clairvoyant model," in *SPAA*, 2015, pp. 133–142.

[22] I. Menache and M. Singh, "Online caching with convex costs: Extended abstract," in *SPAA*, 2015, pp. 46–54.

[23] A. Gupta, R. Krishnaswamy, and K. Pruhs, "Online primal-dual for non-linear optimization with applications to speed scaling," in *WAOA*, 2012, pp. 173–186.

[24] A. Blum, A. Gupta, Y. Mansour, and A. Sharma, "Welfare and profit maximization with production costs," in *FOCS*, Nov 2011, pp. 77–86.

[25] N. R. Devanur and K. Jain, "Online matching with concave returns," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012, pp. 137–144.

[26] S. Agrawal and N. R. Devanur, "Fast algorithms for online stochastic convex programming," in *SODA*, 2015, pp. 1405–1424.

[27] K. Bhawalkar, S. Gollapudi, and D. Panigrahi, "Online set cover with set requests," in *APPROX/RANDOM*, 2014, pp. 64–79.

[28] J. Borwein and A. Lewis, *Convex analysis and and Nonlinear Optimization*. Springer, 2006.

[29] S. Khuller, J. Li, and B. Saha, "Energy efficient scheduling via partial shutdown," in *SODA*, 2010, pp. 1360–1372.

[30] L. Fleischer, "Data center scheduling, generalized flows, and submodularity," in *ANALCO*, 2010, pp. 56–65.

[31] J. Li and S. Khuller, "Generalized machine activation problems," in *SODA*, 2011, pp. 80–94.

[32] R. Latała, "Estimation of moments of sums of independent real random variables," *Ann. Probab.*, vol. 25, no. 3, pp. 1502–1513, 1997. [Online]. Available: http://dx.doi.org/10.1214/aop/1024404522