Online Primal-Dual Algorithms with Configuration Linear Programs

Nguyễn Kim Thắng
IBISC, Univ Evry, University Paris Saclay, France
kimthang.nguyen@univ-evry.fr

Abstract

In this paper, we present primal-dual algorithms for online problems with *non-convex* objectives. Problems with convex objectives have been extensively studied in recent years where the analyses rely crucially on the convexity and the Fenchel duality. However, problems with non-convex objectives resist against current approaches and non-convexity represents a strong barrier in optimization in general and in the design of online algorithms in particular. In our approach, we consider configuration linear programs with the multilinear extension of the objectives. We follow the multiplicative weight update framework in which a novel point is that the primal update is defined based on the gradient of the multilinear extension. We introduce new notions, namely *(local) smoothness*, in order to characterize the competitive ratios of our algorithms. The approach leads to competitive algorithms for several problems with convex/non-convex objectives.

2012 ACM Subject Classification Theory of computation → Online algorithms

Keywords and phrases Configuration Linear Programs, Primal-Dual, Smoothness

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2020.45

Related Version A full version of the paper is available at `https://www.ibisc.univ-evry.fr/~thang/Papers/conf-pd.pdf`.

Funding Research supported by the ANR project OATA n° ANR-15-CE40-0015-01.

1 Introduction

In the paper, we consider problems of minimizing the total cost of resources used to satisfy online requests. One phenomenon observed in various situations, known as the *economy of scale*, consists of sub-linear growth of cost in the amount of used resources. This happens in many scenarios in which one gets a discount when buying resources in bulk. A representative setting is the extensively-studied domain of sub-modular optimization. Another phenomenon, known as the *diseconomy of scale*, is that the cost grows super-linearly in the quantity of resources used. An illustrative example for this phenomenon is the energy cost of computation where the cost grows super-linearly as a function of speeds. The diseconomy of scale has been widely studied in the domain of convex optimization [5]. However, in many settings, the costs are the mix of both phenomena and the objective functions are indeed non-convex. Non-convex objective functions appear in various problems, ranging from scheduling, sensor energy management, to influence and revenue maximization, and facility location. For example, in scheduling of malleable jobs on parallel machines, the cost grows as a non-convex function [15] which is due to the parallelization and the synchronization. Besides, in the practical aspect of facility location, the facility costs to serve clients are rarely constant or simply a convex function of the number of clients. The costs would initially increase fast until some threshold on the number of clients, then become more stable before quickly increase again as the number of clients augments. This behaviour of cost functions widely happens in economy. Such situations call for the design of algorithms with performance guarantee for non-convex objective functions. In this paper, we consider problems in which the cost

increases *arbitrarily* with the amount of used resources. We measure the performance of an algorithm by the *competitive ratio*. Specifically, an algorithm is r-competitive if for any instance, the ratio between the cost of the algorithm and that of an optimal solution is at most r.

1.1 An Optimization Problem and Primal-Dual Approach

We first consider the following problem and illustrate our primal-dual approach based on configuration LPs.

Optimization Problem. There is a set of resources $\mathcal{E}$ and requests arrive online. At the arrival of request i, a set of feasible strategies (actions) $\mathcal{S}_i$ to satisfy request i is revealed. Each strategy $s_{ij} \in \mathcal{S}_i$ consists of a subset of resources in $\mathcal{E}$. Each resource e is associated to an *arbitrary* non-negative non-decreasing cost function $f_e : 2^{\mathcal{E}} \to \mathbb{R}^+$; the cost induced by resource e depends on the set of requests using e and $f_e(\emptyset) = 0$. The cost of a solution is the total cost of resources, i.e., $\sum_e f_e(A_e)$ where A_e is the set of requests using resource e. The goal is to design an algorithm that upon the arrival of each request, selects a feasible strategy for the request while maintaining the cost of the overall solution as small as possible.

Primal-Dual Approach. We consider an approach based on linear programming for the problem. The first crucial step for any LP-based approach is to derive a LP formulation with reasonable *integrality gap*, which is defined as the ratio between the optimal integer solution of the formulation and the optimal solution without the integer condition. As the cost functions are non-linear, it is not surprising that the natural relaxation suffers from large integrality gap. Consider the following simple (offline) setting. Given a graph consisting of only two nodes s, t and m parallel edges between s and t. There is a single request with the demand of routing from s to t. In this setting, an edge corresponds to a resource and a strategy is a path connecting s to t which is also an edge (among m given edges). For every edge e, the cost function $f_e(x_e) = x_e^2$ where x_e is the quantity of flow passing through edge e. The most natural formulation of the problem would be: $\min \sum_{e=1}^{m} x_e^2$ such that $\sum_{e=1}^{m} x_e = 1, x_e \in \{0, 1\}$. By relaxing the integrality constraint of x_e, the fractional optimum has value $1/m$ (by routing $x_e = 1/m$ through each of m edges) whereas the integer optimum is 1. So the integrality gap is arbitrarily large for a very simple offline setting. We resolve this issue by a new formulation in form of a configuration LP. We start with a natural formulation and systematically strengthen the natural formulations by introducing an exponential number of new variables and new constraints connecting new variables to original ones. Consequently, the new formulation, in form of a configuration LP, significantly reduces the integrality gap.

The configuration LPs have been used mostly in offline settings and the approach is to round an optimal fractional solution to an integer one and bound the approximation ratio. The first encountered difficulty of this approach is that a configuration LP has exponential size, so one has to look for a separation oracle in order to compute an optimal fractional solution. Finding separation oracles is in general far from trivial and it represents an obstacle in using configuration LP to design performant algorithms. Besides, many rounding schemes are *intrinsically offline* and it is not suitable in online setting where input is released in pieces.

To overcome these difficulties, we consider a primal-dual approach with configuration LPs. First, primal-dual is particularly appropriate since one does not have to compute an optimal fractional solution that needs the full information on the instance. Second, in our approach,

the dual variables of the configuration LP have intuitive meanings and the dual constraints indeed guide the decisions of the algorithm. The key step in the approach is to show that the constructed dual variables constitute a dual feasible solution. In order to prove the dual feasibility, we define a notion of *smoothness* of functions. This definition is inspired by the smoothness framework introduced by Roughgarden [22] in the context of algorithmic game theory to characterize the price of anarchy for large classes of games. The smoothness notion allows us not only to prove the dual feasibility but also to establish the competitiveness of algorithms in our approach. We characterize the performance of algorithms using the notion of smoothness in a similar way as the price of anarchy characterized by the smoothness argument [22]. Through this notion, we show an interesting connection between online algorithms and algorithmic game theory.

▶ **Definition 1.** *Let $\mathcal{N}$ be a set of requests. A set function $f : 2^{\mathcal{N}} \to \mathbb{R}^+$ is* (λ, μ)-smooth *if for any set $A = \{a_1, \ldots, a_n\} \subseteq \mathcal{N}$ and any collection $B_1 \subseteq B_2 \subseteq \ldots \subseteq B_n \subseteq B \subseteq \mathcal{N}$, the following inequality holds: $\sum_{i=1}^{n} \left[f\big(B_i \cup a_i\big) - f\big(B_i\big)\right] \leq \lambda f\big(A\big) + \mu f\big(B\big)$.*

Intuitively, given a (λ, μ)-smooth function, the quantity $\frac{\lambda}{1-\mu}$ measures how far the function is from being linear. If a function is linear then it is $(1, 0)$-smooth. Informally, the inequality in the definition of smoothness means the following. Imagine that B_i is the current solution of an algorithm at step i. Then, if the total marginal increase by following a strategy a_i at step i (the left-hand side) can be bounded by a combination of the algorithm cost (the second term of the right-hand side) and the cost of an adversary (potentially the set of all strategies a_i's), then the algorithm is competitive (again, the competitive ratio depends on λ and μ). We say that a set of cost functions $\{f_e : e \in \mathcal{E}\}$ is (λ, μ)-*smooth* if every function f_e is (λ, μ)-*smooth*.

▶ **Theorem 2.** *Assume that all resource cost functions are (λ, μ)-smooth for some parameters $\lambda > 0$, $\mu < 1$. Then there exists a greedy $\frac{\lambda}{1-\mu}$-competitive algorithm for the general problem.*

Applications. We show the applicability of the theorem by deriving competitive algorithms for several problems in online setting, such as Minimum Power Survival Network Routing, Vector Scheduling, Energy-Efficient Scheduling, Prize Collecting Energy-Efficient Scheduling, Non-Convex Facility Location. We mention in the following the energy-efficient scheduling problem and refer the reader to the full paper for other applications.

In Online Energy-Efficient Scheduling, one has to process jobs on unrelated machines without migration with the objective of minimizing the total energy. No result has been known for this problem for parallel machine environments. Among others, a difficulty is the construction of a formulation with bounded integrality gap. We notice that for this problem, Gupta et al. [13] gave a primal-dual competitive algorithm for a single machine. However, their approach cannot be used for unrelated machines due to the large integrality gap of their formulation. For this problem, we present competitive algorithms for *arbitrary* cost functions beyond the convexity property. Note that the convexity of the cost functions is a crucial property employed in previous works. If the cost functions have typical form $f(x) = x^\alpha$ then the competitive ratio of our algorithm is $O(\alpha^\alpha)$ and this is optimal up to a constant factor for all the problems above.

1.2 Primal-Dual Approach for Covering Problems

Covering Problems. Let $\mathcal{E}$ be a set of n resources and let $f : \{0, 1\}^n \to \mathbb{R}^+$ be an *arbitrary* monotone cost function. Let $x_e \in \{0, 1\}$ be a variable indicating whether resource e is selected. The covering constraints $\sum_e a_{i,e} x_e \geq 1$ for every i are revealed one-by-one and

at any step, one needs to maintain a feasible integer solution $\mathbf{x}$. The goal is to design an algorithm that minimizes $f(\mathbf{x})$ subject to the online covering constraints and $x_e \in \{0,1\}$ for every e.

Approach and Contribution. We extend our primal-dual approach to study the covering problems. Very recently, Azar et al. [2] have presented a general primal-dual framework when function f is convex and its gradient is monotone on every coordinate (so a subclass of convex functions). The framework is indeed inspired by the Buchbinder-Naor framework [7] for linear objectives and is along the line of current results for convex objectives [9, 3, 19, 13, 14, 10, 2]. A common point of those works is that they rely crucially on the convexity of cost functions and Fenchel duality.

We overcome the obstacle of non-convexity (and also for general convexity without the property that the gradient is monotone on every coordinate) by a considering configuration LP and the multilinear extension of function f. Given $f : \{0,1\}^n \to \mathbb{R}^+$, its *multilinear extension* $F : [0,1]^n \to \mathbb{R}^+$ is defined as $F(\mathbf{x}) := \sum_S \prod_{e\in S} x_e \prod_{e\notin S}(1-x_e) \cdot f(\mathbf{1}_S)$ where $\mathbf{1}_S$ is the characteristic vector of S (i.e., the e^{th}-component of $\mathbf{1}_S$ equals 1 if $e \in S$ and equals 0 otherwise). An alternative way to define F is to set $F(\mathbf{x}) = \mathbb{E}\big[f(\mathbf{1}_T)\big]$ where T is a random set such that a resource e appears independently in T with probability x_e. Note that $F(\mathbf{1}_S) = f(\mathbf{1}_S)$.

Having inspired by the approach for the optimization problem in the previous section, we introduce the notion of *locally-smooth* for minimization problems and characterize the competitive ratio using the local smoothness' parameters. Given two vectors $\mathbf{x}$ and $\mathbf{y}$ in $[0,1]^n$, let $\mathbf{x} \vee \mathbf{y}$ be the vector such that its component at coordinate e' is $\max\{x_{e'}, y_{e'}\}$.

▶ **Definition 3.** *Let $\mathcal{E}$ be a set of n resources. A differentiable function $F : [0,1]^n \to \mathbb{R}^+$ is* (λ,μ)-min-locally-smooth *if for any set $S \subseteq \mathcal{E}$, and for all vectors $\mathbf{x}^e \in [0,1]^n$ where $e \in \mathcal{E}$, the following inequality holds:*

$$\sum_{e\in S} \nabla_e F(\mathbf{x}^e) \le \lambda F\big(\mathbf{1}_S\big) + \mu F\big(\mathbf{x}\big) \tag{1}$$

where $\mathbf{x} := \bigvee_{e\in S} \mathbf{x}^e$, meaning that $x_{e'} = \max_e\{x^e_{e'}\}$ for every coordinate e'.

If the gradient $\nabla F(\mathbf{x})$ is non-decreasing on every coordinate, we only need a simpler version. We say that a differentiable function $F : [0,1]^n \to \mathbb{R}^+$ with monotone gradient is (λ,μ)-*min-locally-smooth* if for any set $S \subseteq \mathcal{E}$, and for any vector $\mathbf{x} \in [0,1]^n$, the following inequality holds.

$$\sum_{e\in S} \nabla_e F(\mathbf{x}) \le \lambda F\big(\mathbf{1}_S\big) + \mu F\big(\mathbf{x}\big) \tag{2}$$

Let us explain intuitively the local smoothness by considering Inequality (2). Assume that $\mathbf{x}$ is the current solution of an algorithm. Then, if the *local* increase of the objective function F (the left-hand side) at the current solution in any direction (including the direction chosen by an adversary) can be bounded by a combination of the current cost (the second term of the right-hand side) and the cost of an adversary (the first term of the right-hand side), then the algorithm is competitive. The competitive ratio will be determined as a function of λ and μ.

Building upon the primal-dual framework in [2, 7], we present a competitive algorithm for the fractional covering problem.

▶ **Theorem 4.** *Let F be the multilinear extension of the objective cost f and d be the maximal row sparsity of the constraint matrix, i.e., $d = \max_i |\{a_{ie} : a_{ie} > 0\}|$. Assume that F is $\big(\lambda, \frac{\mu}{\ln(1+2d^2)}\big)$-min-locally-smooth for some parameters $\lambda > 0$ and $\mu < 1$. Then there exists a $O\big(\frac{\lambda}{1-\mu} \cdot \ln d\big)$-competitive algorithm for the fractional covering problem.*

Our algorithm, as well as the one in [2] for convex with monotone gradients and the recent algorithm for ℓ_k-norms [20], are extensions of the Buchbinder-Naor primal-dual framework [7]. A distinguishing point of our algorithm compared to the ones in [2, 20] relies on the multiplicative update, which is crucial in online primal-dual methods. The approaches in [2, 20] use the gradient $\nabla f(\mathbf{x})$ at the current primal solution $\mathbf{x}$ to define a multiplicative update for the primal. In our approach, we multiplicatively update the primal by some parameter related to the gradient of the multilinear extension $\nabla F(\mathbf{x})$. This parameter is always maintained to be at least $\nabla F(\mathbf{x})$ and in case $\nabla F(\mathbf{x})$ is non-decreasing, the parameter is indeed equal to $\nabla F(\mathbf{x})$. This multiplicative update, together with the configuration LPs and the notion of local smoothness, enable us to derive a competitive algorithm for convex objective functions whose gradients are not necessarily monotone and more generally, for non-convex objectives. Moreover, other advantage of our approach are: (i) it avoids the cumbersome technical details in the analysis as well as in the assumptions of objective functions; (ii) it reduces the analysis of bounding the competitive ratios to determining the local-smoothness parameters.

Applications. Specifically, we apply our algorithm to the following classes of functions. First, for the class of non-negative polynomials of degree k, the algorithm yields a $O\big((k \log d)^k\big)$-competitive fractional solution that matches a result in [2]. Second, beyond convexity, we consider a natural class of non-convex cost functions which represent a typical behaviour of resources in serving demand requests. Non-convexity represents a strong barrier in optimization in general and for the design of algorithms in particular. We show that our algorithm is competitive for this class of functions.

1.3 Primal-Dual Algorithm for Packing Problems

Packing Problems. Let $\mathcal{E}$ be a set of n resources and let $f : \{0,1\}^n \to \mathbb{R}^+$ be an *arbitrary* monotone function. Let $x_e \in \{0,1\}$ be a variable indicating whether resource e is selected. The packings constraints $\sum_e b_{i,e} x_e \leq 1$ for every i are given in advance and resources e are revealed online one-by-one. At any time, one needs to maintain a feasible integer solution $\mathbf{x}$. The goal is to design an algorithm that maximizes $f(\mathbf{x})$ subject to the online packing constraints and $x_e \in \{0,1\}$ for every e.

We follow the primal-dual approach to design competitive algorithms for fractional packing problems. We introduce an appropriate notion of smoothness for maximization problems. We notice that this notion is different to that for minimization problems. On one hand, it is due to different natures of minimization and maximization problems. On the other hand, in non-convex problems only weak duality holds while strong duality does not. So informally, there is no symmetry between primal and dual. Specifically, in linear programming, the dual of the dual is the primal while this property does not hold in non-convex settings.

▶ **Definition 5.** *A differentiable function $F : [0,1]^n \to \mathbb{R}^+$ is* (λ, μ)-max-locally-smooth *if for any set $S \subset \mathcal{E}$, and for any vectors $\mathbf{x}^e \in [0,1]^n$, the following inequality holds:*

$$\sum_{e \in S} \nabla_e F(\mathbf{x}^e) \geq \lambda F\big(\mathbf{1}_S\big) - \mu F\big(\mathbf{x}\big).$$

where $\mathbf{x} := \bigvee_{e \in S} \mathbf{x}^e$, meaning that $x_{e'} = \max_e \{x^e_{e'}\}$ for any coordinate e'.

Similar to the approach for covering constraints, the max-local-smoothness notion allows us to prove the dual feasibility and also to establish the competitiveness of algorithms, which is determined in terms of the max-locally-smoothness parameters.

▶ **Theorem 6.** *Let F be the multilinear extension of the objective cost f. Denote the row sparsity $d := \max_i |\{b_{ie} : b_{ie} > 0\}|$ and $\rho := \max_i \max_{e,e':b_{ie'}>0} b_{ie}/b_{ie'}$. Assume that F is (λ, μ)-max-locally-smooth for some parameters $\lambda > 0$ and μ. Then there exists a $O\big(\frac{2\ln(1+d\rho)+\mu}{\lambda}\big)$-competitive algorithm for the fractional packing problem.*

Note that when f is a linear function, the smooth parameters are $\lambda = 1$ and $\mu = 0$. In this case, the performance guarantee is the same (up to a constant factor) as that of maximizing a linear function under packing constraints [7] and therefore asymptotically optimal.

Applications. We consider applications to online submodular maximization problems. Submodular functions are interesting since they are neither convex nor concave. Besides, submodular maximization constitutes a major research agenda in optimization, machine learning and has been widely studied. However, in the *online adversarial* setting, not much has been known especially for submodular maximization with constraints. Designing competitive algorithms for online submodular maximization has been identified as an important direction in the recent survey [17]. Buchbinder and Naor [8] have studied the online problem of maximizing the sum of weighted rank functions subject to matroid constraints. The objective here is a particular submodular function. The authors give an algorithm with competitive ratio depending logarithmically on the numbers of elements and on weighted rank functions. In another approach, Buchbinder et al. [6] have considered submodular optimization with preemption, where one can reject previously accepted elements, and have given constant competitive algorithms for unconstrained and knapsack-constraint problems.

We show that there exists an algorithm that yields competitive fractional solutions for online submodular maximization under packing constraints. The competitive ratio is $O\big(\log(1 + d\rho)\big)$ which is independent of the submodular objective. Note that using the online contention resolution rounding schemes [12], one can obtain randomized algorithms for several specific constraint polytopes, for example, knapsack polytopes, matching polytopes and matroid polytopes.

1.4 Related work

In this section we summarize related work to our approach. The related work of each specific problem/application will be given in the corresponding section.

In our approach, a crucial element to characterize the performance of an algorithm is the smoothness property of functions. The smooth argument is introduced by Roughgarden [22] in the context of algorithmic game theory and it has successfully characterized the performance of equilibria (price of anarchy) in many classes of games such as congestion games, etc [22]. This notion inspires the definition of smoothness in our paper.

Primal-dual methods have been shown to be powerful tools in online computation. Buchbinder and Naor [7] presented a primal-dual method for linear programs with packing/covering constraints. Their method unifies several previous potential-function-based analyses and give a principled approach to design and analyze algorithms for problems with linear relaxations. Convex objective functions have been extensively studied in online settings in recent years, in areas such as energy-efficient scheduling [1, 21, 9, 16, 3], paging [19], network routing [13], combinatorial auctions [4, 14], matching [10]. Recently, Azar et al. [2] gave an unified framework for covering/packing problems with convex objectives whose gradients

are monotone. Consequently, improved algorithms have been derived for several problems. The above approaches rely crucially on the convexity of cost functions. Specifically, the construction of dual programs is based on convex conjugates and Fenchel duality for primal convex programs. Very recently, Nagarajan and Shen [20] have considered objective functions as the sum of ℓ_k-norms. This class of functions does not fall into the framework developped in [2] since the gradients are not necessarily monotone. Nagarajan and Shen [20] proved that the algorithm presented in [2] yields a nearly tight $O\big(\log d + \log \frac{\max a_{ij}}{\min a_{ij}}\big)$-competitive ratio where a_{ij}'s are entries in the covering matrix. Using these approaches, it is not clear how to design competitive algorithms for *non-convex* functions or even for other convex functions whose gradient is not necessarily monotone on every coordinate. A distinguishing point of our approach is that it gives a framework to study non-convex cost functions.

2 Primal-Dual Algorithm for the optimization problem

Recall that the problem consists of a set of resources $\mathcal{E}$ and requests which arrive online. At the arrival of request i, a set of feasible strategies (actions) $\mathcal{S}_i$ to satisfy request i is revealed. Each strategy $s_{ij} \in \mathcal{S}_i$ consists of a subset of resources in $\mathcal{E}$. Each resource e is associated to a non-negative non-decreasing *arbitrary* cost function $f_e : 2^{\mathcal{E}} \to \mathbb{R}^+$ and the cost induced by resource e depending on the set of requests using e. The cost of a solution is the total cost of resources, i.e., $\sum_e f_e(A_e)$ where A_e is the set of requests using resource e. The goal is to design an algorithm that upon the arrival of each request, selects a feasible strategy while maintaining the cost of the overall solution as small as possible.

Formulation. We consider the formulation for the resource cost minimization problem following the configuration LP construction in [18]. We say that A is a *configuration* associated to resource e if A is a subset of requests using e. Let x_{ij} be a variable indicating whether request i selects strategy (action) $s_{ij} \in \mathcal{S}_i$. For configuration A and resource e, let z_{eA} be a variable such that $z_{eA} = 1$ if and only if for every request $i \in A$, $x_{ij} = 1$ for some strategy $s_{ij} \in \mathcal{S}_i$ such that $e \in s_{ij}$ and for every request $i \notin A$, $x_{ij} = 0$ for any strategy $s_{ij} \in \mathcal{S}_i$ such that $e \in s_{ij}$. In other words, $z_{eA} = 1$ iff the set of requests using e is exactly A. We consider the following formulation and the dual of its relaxation.

$$\min \sum_{e,A} f_e(A) z_{e,A}$$

$$\sum_{j:s_{ij}\in\mathcal{S}_i} x_{ij} = 1 \qquad \forall i$$

$$\sum_{A:i\in A} z_{eA} = \sum_{j:e\in s_{ij}} x_{ij} \qquad \forall i, e$$

$$\sum_{A} z_{eA} = 1 \qquad \forall e$$

$$x_{ij}, z_{eA} \in \{0,1\} \qquad \forall i, j, e, A$$

$$\max \sum_i \alpha_i + \sum_e \gamma_e$$

$$\alpha_i \le \sum_{e:e\in s_{ij}} \beta_{ie} \qquad \forall i, j$$

$$\gamma_e + \sum_{i\in A} \beta_{ie} \le f_e(A) \qquad \forall e, A$$

In the primal, the first constraint guarantees that request i selects some strategy $s_{ij} \in \mathcal{S}_i$. The second constraint ensures that if request i selects strategy s_{ij} that contains resource e then in the solution, the set of requests using e must contain i. The third constraint says that in the solution, there is always a configuration associated to resource e.

Algorithm. We first interpret intuitively the dual variables, dual constraints and derive useful observations for a competitive algorithm. Variable α_i represents the increase of the total cost due to the arrival of request i. Variable $\beta_{i,e}$ stands for the marginal cost on resource e if request i uses e. By this interpretation, the first dual constraint clearly indicates the behaviour of an algorithm. That is, if a new request i is released, select a strategy $s_{ij} \in \mathcal{S}_i$ that minimizes the marginal increase of the total cost. Therefore, we deduce the following greedy algorithm.

Let A_e^* be the set of current requests using resource e. Initially, $A_e^* \leftarrow \emptyset$ for every e. At the arrival of request i, select strategy s_{ij}^* that is an optimal solution of

$$\min \sum_{e \in s_{ij}} \left[f_e(A_e^* \cup i) - f_e(A_e^*) \right] \qquad \text{over} \qquad s_{ij} \in \mathcal{S}_i. \tag{3}$$

Although computational complexity is not a main issue for online problems, we notice that in many applications, the optimal solution for this mathematical program can be efficiently computed (for example when f_e's are convex and $\mathcal{S}_i$ can be represented succinctly in form of a polynomial-size polytope).

Dual variables. Assume that all resource cost f_e are (λ, μ)-smooth for some fixed parameters $\lambda > 0$ and $\mu < 1$. We are now constructing a dual feasible solution. Define α_i as $1/\lambda$ times the optimal value of the mathematical program (3). Informally, α_i is proportional to the increase of the total cost due to the arrival of request i. Note that this increase is also called *marginal cost* due to request i. For each resource e and request i, define

$$\beta_{i,e} := \frac{1}{\lambda} \left[f_e(A_{e,\prec i}^* \cup i) - f_e(A_{e,\prec i}^*) \right]$$

where $A_{e,\prec i}^*$ is the set of requests using resource e (due to the algorithm) prior to the arrival of i. In other words, β_{ij} equals $1/\lambda$ times the marginal cost of resource e if i uses e. Finally, for every resource e define the dual variable $\gamma_e := -\frac{\mu}{\lambda} f_e(A_e^*)$ where A_e^* is the set of all requests using e (at the end of the instance).

▶ **Lemma 7.** *The dual variables defined as above are feasible.*

Proof. The first dual constraint follows immediately from the definitions of $\alpha_i, \beta_{i,e}$ and the decisions by the algorithm. Specifically, the right-hand side of the constraint represents $1/\lambda$ times the increase cost if the request selects a strategy s_{ij}. This is larger than $1/\lambda$ times the minimum increase cost optimized over all strategies in $\mathcal{S}_i$, which is α_i.

We now show that the second constraint holds. Fix a resource e and a configuration A. The corresponding constraint reads

$$\begin{aligned} & -\frac{\mu}{\lambda} f_e(A_e^*) + \frac{1}{\lambda} \sum_{i \in A} \left[f_e(A_{e,\prec i}^* \cup i) - f_e(A_{e,\prec i}^*) \right] \le f_e(A) \\ \Leftrightarrow \qquad & \sum_{i \in A} \left[f_e(A_{e,\prec i}^* \cup i) - f_e(A_{e,\prec i}^*) \right] \le \lambda f_e(A) + \mu f_e(A_e^*). \end{aligned}$$

This inequality is due to the definition of (λ, μ)-smoothness for resource e. Hence, the second dual constraint follows. ◀

▶ **Theorem 2.** *Assume that all resource cost functions are (λ, μ)-smooth for some parameters $\lambda > 0$, $\mu < 1$. Then there exists a greedy $\frac{\lambda}{1-\mu}$-competitive algorithm for the general problem.*

Proof. By the definitions of dual variables, the dual objective is

$$\sum_i \alpha_i + \sum_e \gamma_e = \sum_e \frac{1}{\lambda} f_e(A_e^*) - \sum_e \frac{\mu}{\lambda} f_e(A_e^*) = \frac{1-\mu}{\lambda} \sum_e f_e(A_e^*)$$

Besides, the cost of the solution due to the algorithm is $\sum_e f_e(A_e^*)$. Hence, the competitive ratio is at most $\lambda/(1-\mu)$. ◀

Applications. Despite the simplicity of the algorithm, Theorem 2 yields *optimal* competitive ratios for several problems. Among others, we give optimal algorithms for energy efficient scheduling problems (in unrelated machine environment) and the facility location with client-dependent cost problem. Prior to our work, no competitive algorithm has been known for the problems. The proofs are now reduced to computing smooth parameters λ, μ that subsequently imply the competitive ratios. We refer the reader to the full paper for the details about those applications.

3 Primal-Dual Framework for Covering Problems

Formulation. We say that $S \subset \mathcal{E}$ is a *configuration* if $\mathbf{1}_S$ corresponds to a feasible solution. Let x_e be a variable indicating whether the resource e is used. For configuration S, let z_S be a variable such that $z_S = 1$ if and only if $x_e = 1$ for every resource $e \in S$, and $x_e = 0$ for $e \notin S$. In other words, $z_S = 1$ iff $\mathbf{1}_S$ is the selected solution to the problem. For any subset $A \subset \mathcal{E}$, define $c_{i,A} = \max\{1 - \sum_{e' \in A} a_{i,e'}; 0\}$ and $a_{i,e,A} := \min\{a_{i,e}; c_{i,A}\}$. Denote $b_{i,e,A} = \frac{a_{i,e,A}}{c_{i,A}}$ where $c_{i,A} > 0$. We consider the following formulation and the dual of its relaxation.

$$\begin{aligned}
\min \sum_S f(\mathbf{1}_S) z_S & \\
\sum_{e \notin A} b_{i,e,A} \cdot x_e \geq 1 & \quad \forall i, A \subset \mathcal{E} \\
\sum_{S: e \in S} z_S = x_e & \quad \forall e \\
\sum_S z_S = 1 & \\
x_e, z_S \in \{0,1\} & \quad \forall e, S
\end{aligned}
\qquad\qquad
\begin{aligned}
\max \sum_{i,A} \alpha_{i,A} + \gamma & \\
\sum_i \sum_{A: e \notin A} b_{i,e,A} \cdot \alpha_{i,A} \leq \beta_e & \quad \forall e \\
\gamma + \sum_{e \in S} \beta_e \leq f(\mathbf{1}_S) & \quad \forall S \\
\alpha_i \geq 0 & \quad \forall i
\end{aligned}$$

In the primal, the first constraints are knapsack-constraints of the form $\sum_{e \notin A} a_{i,e,A} \cdot x_e \geq c_{i,A}$ corresponding to the given polytope. Intuitively, this constraint imposes the quantities of x_e's to be chosen, assuming if a set of resources A is selected, in order to satisfy the original constraint. Note that it is sufficient to consider only constraints with $c_{i,A} > 0$. The second constraint ensures that if a resource e is chosen then the selected solution must contain e. The third constraint says that one solution (configuration) must be selected.

Algorithm. Assume that function $F(\cdot)$ is $\left(\lambda, \frac{\mu}{4\ln(1+2d^2)}\right)$–min-locally smooth. Let d be the maximal number of positive entries in a row, i.e., $d = \max_i |\{a_{ie} : a_{ie} > 0\}|$. Denote $\nabla_e F(\mathbf{x}) = \partial F(\mathbf{x})/\partial x_e$. Consider Algorithm 1 which follows the scheme in [2] but is more subtle due to the fact that the gradient is not necessarily monotone on every coordinate. In the algorithm, the current dual variable α increases at constant rate (Step 7) and the update of dual variables β's is shown in Step 9. We note an subtle point here: if $\beta_e < \frac{1}{\lambda} \nabla_e F(\mathbf{x})$

then set $\beta_e = \frac{1}{\lambda}\nabla_e F(\mathbf{x})$; otherwise if $\beta_e > \frac{1}{\lambda}\nabla_e F(\mathbf{x})$ then we do not set β_e as $\frac{1}{\lambda}\nabla_e F(\mathbf{x})$. So two invariants during the execution of the algorithm are that $\beta_e \geq \frac{1}{\lambda}\nabla_e F(\mathbf{x})$ and β_e is non-decreasing. The primal update rule follows a multiplicative increase where the increasing rate of x_e is inversely proportional to β_e (Step 10). Finally, using the same idea as in [2], some dual variables α will be decreased in order to maintain the feasibility of our dual solution.

Algorithm 1 Algorithm for Covering Constraints.

1: Initially, set $A^* \leftarrow \emptyset$. Intuitively, A^* consists of all resources e such that $x_e = 1$.
2: All primal and dual variables are initially set to 0.
3: At every step, always maintain $z_S = \prod_{e\in S} x_e \prod_{e\notin S}(1 - x_e)$.
4: Upon the arrival of primal constraint $\sum_e a_{k,e}x_e \geq 1$ do the following.
5: **while** $\sum_{e\notin A^*} b_{k,e,A^*}x_e < 1$ **simultaneously do** # `Increase primal, dual variables`
6: Let τ be the current time in the execution of the algorithm.
7: Increase τ at rate 1 and increase α_{k,A^*} at rate $\frac{1}{\lambda\cdot\ln(1+2d^2)}$.
8: **for** $e \notin A^*$ such that $b_{k,e,A^*} > 0$ **simultaneously do**
9: **if** $\beta_e < \frac{1}{\lambda}\nabla_e F(\mathbf{x})$ **then** $\beta_e \leftarrow \frac{1}{\lambda}\nabla_e F(\mathbf{x})$.
10: Increase x_e according to the following function $\frac{\partial x_e}{\partial \tau} \leftarrow \frac{b_{k,e,A^*}\cdot x_e + 1/d}{\lambda\cdot\beta_e}$.
11: **end for**
12: **if** $x_e = 1$ **then** update $A^* \leftarrow A^* \cup \{e\}$.
13: **while** $\sum_{i=1}^{k}\sum_{A:e\notin A} b_{i,e,A}\cdot\alpha_{i,A} > \beta_e$ for some $e \notin A^*$ **do** # `Decrease dual variables`
14: **for** (m_e^*, A) such that $b_{m_e^*,e,A} = \max_i\{b_{i,e,A} | \forall A : e \notin A, \forall 1 \leq i \leq k : \alpha_{i,A} > 0\}$ **do**
15: Increase $\alpha_{m_e^*,A}$ continuously at rate $-\frac{b_{k,e,A^*}}{b_{m_e^*,e,A}} \cdot \frac{1}{\lambda\cdot\ln(1+2d^2)}$.
16: **end for**
17: **end while**
18: **end while**

Dual variables. Variables $\alpha_{i,A}$ and β_e have been constructed in the algorithm. Let $\mathbf{x}$ be the current solution of the algorithm. Define $\gamma = -\frac{\mu}{4\lambda\cdot\ln(1+2d^2)}F(\mathbf{x})$. Note that due to the algorithm, $\beta_e \geq \frac{1}{\lambda}\cdot\nabla_e F(\mathbf{x})$.

The following lemma gives a lower bound on x-variables. Remark that the monotonicity of the gradient on every coordinate is crucial in the analysis of [2], in particular to prove the bounds on x-variables. However, in our approach that property is not needed.

▶ **Lemma 8.** *Let e be an arbitrary resource. At any moment during the execution of the algorithm where the k^{th} request has been released , it always holds that*

$$x_e \geq \frac{1}{\max b_{i,e,A}\cdot d}\left[\exp\left(\frac{\ln(1+2d^2)}{\beta_e}\cdot\sum_{A:e\notin A}\sum_i b_{i,e,A}\cdot\alpha_{i,A}\right) - 1\right]$$

where $\max b_{i,e,A} := \max\{b_{i,e,A} > 0 | \forall A : e \notin A, \forall 1 \leq i \leq k : \alpha_{i,A} > 0\}$.

Proof. Fix a resource e. We prove the lemma by induction. At the beginning of the instance, while no request has been released yet, both sides of the lemma are 0. Assume that the lemma holds until the arrival of the k^{th} request. Consider a moment τ during the execution of the algorithm and let A^* be the current set of resources e' such that $x_{e'} = 1$. If at time τ,

$x_e = 1$ then by the algorithm, the set A^* has been updated so that $e \in A^*$. The increasing rates of both sides in the lemma inequality are 0. In the remaining, assume that $x_e < 1$. Recall that by the algorithm, $\beta_e \geq \frac{1}{\lambda}\nabla_e F(\mathbf{x})$. We consider two cases $\beta_e > \frac{1}{\lambda}\nabla_e F(\mathbf{x})$ and $\beta_e = \frac{1}{\lambda}\nabla_e F(\mathbf{x})$.

Case 1: $\beta_e > \frac{1}{\lambda}\nabla_e F(\mathbf{x})$. In this case, by the algorithm, the value of β_e remains unchanged at time τ (Step 9), i.e., $\frac{\partial \beta_e}{\partial \tau} = 0$. Hence, the derivative of the right hand side of the lemma inequality according to τ is

$$\sum_i \frac{\partial \alpha_{i,A^*}}{\partial \tau} \cdot \frac{b_{i,e,A^*}}{\max b_{i,e,A} \cdot d} \cdot \frac{\ln(1+2d^2)}{\beta_e} \cdot \exp\left(\frac{\ln(1+2d^2)}{\beta_e} \cdot \sum_{A: e \notin A} \sum_i b_{i,e,A}\alpha_{i,A}\right)$$
$$\leq \frac{b_{k,e,A^*} \cdot x_e + 1/d}{\lambda \cdot \beta_e} = \frac{\partial x_e}{\partial \tau}$$

In the inequality, we use the induction hypothesis; $\frac{\partial \alpha_{k,A^*}}{\partial \tau} > 0$ and $\frac{\partial \alpha_{i,A^*}}{\partial \tau} \leq 0$ for $i \neq k$ and $\frac{\partial \beta_e}{\partial \tau} = 0$; and the increasing rate of α_{k,A^*} according to the algorithm. So the rate in the left-hand side is always larger than that in the right-hand side. Moreover, at some steps in the algorithm, α-variables might be decreased while the x-variables are maintained monotone. Hence, the lemma inequality holds.

Case 2: $\beta_e = \frac{1}{\lambda}\nabla_e F(\mathbf{x})$. In this case, by the algorithm, $\frac{1}{\lambda}\nabla_e F(\mathbf{x})$ is locally non-decreasing at τ (since otherwise, by Step 9, β_e is not maintained to be equal to $\frac{1}{\lambda}\nabla_e F(\mathbf{x})$). Therefore, $\frac{\partial \beta_e}{\partial \tau} \geq 0$ and so $\partial\left(\frac{1}{\beta_e}\right)/\partial\tau \leq 0$. Hence, the derivative of the right hand side of the lemma inequality according to τ is upper bounded by

$$\sum_i \frac{\partial \alpha_{i,A^*}}{\partial \tau} \cdot \frac{b_{i,e,A^*}}{\max b_{i,e,A} \cdot d} \cdot \frac{\ln(1+2d^2)}{\beta_e} \cdot \exp\left(\frac{\ln(1+2d^2)}{\beta_e} \cdot \sum_{A: e \notin A} \sum_i b_{i,e,A}\alpha_{i,A}\right)$$

which is bounded by $\frac{\partial x_e}{\partial \tau}$ by the same argument as the previous case. The lemma follows. ◀

▶ **Lemma 9.** *The dual variables defined as above are feasible.*

Proof. As long as a primal covering constraint is unsatisfied, the x-variables are always increased. Therefore, at the end of an iteration, the primal constraint is satisfied. Consider the first dual constraint. The algorithm always maintains that $\sum_i \sum_{A: e \notin A} b_{i,e,A}\alpha_{i,A} \leq \beta_e$ (strict inequality happens only if $x_e = 1$). Whenever this inequality is violated then by the algorithm, some α-variables are decreased in such a way that the increasing rate of $\sum_i \sum_{A: e \notin A} b_{i,e,A}\alpha_{i,A}$ is at most 0. Hence, by the definition of β-variables, the first dual constraint holds.

Consider the second dual constraint. Let $\mathbf{x}$ be the current solution of the algorithm. By the algorithm, for each fixed resource e, $\beta_e = \frac{1}{\lambda}\nabla_e F(\mathbf{y}^e)$ for some $\mathbf{y}^e$ where $y^e_{e'} \leq x_{e'}$ for every resource e'. (Since at some moment, the algorithm increases x_e without increasing β_e for some e.) Moreover, $\mathbf{y} := \bigvee_e \mathbf{y}^e \leq \mathbf{x}$ (meaning that $y_{e'} \leq x_{e'}$ for every e'). By definitions of dual variables, the second dual constraint (after rearranging terms) reads: $\frac{1}{\lambda}\sum_{e \in S} \nabla_e F(\mathbf{y}^e) \leq F(\mathbf{1}_S) + \frac{\mu}{4\lambda \cdot \ln(1+2d^2)} F(\mathbf{x})$. Besides, as F is monotone, $F(\mathbf{x}) \geq F(\mathbf{y})$. So in order to prove the second dual constraint, it is sufficient to prove that: $\frac{1}{\lambda}\sum_{e \in S} \nabla_e F(\mathbf{y}^e) \leq F(\mathbf{1}_S) + \frac{\mu}{4\lambda \cdot \ln(1+2d^2)} F(\mathbf{y})$. This inequality is exactly the $\left(\lambda, \frac{\mu}{4\ln(1+2d^2)}\right)$-min-local smoothness of F. Hence, the lemma follows. ◀

▶ **Theorem 4.** *Let F be the multilinear extension of the objective cost f and d be the maximal row sparsity of the constraint matrix, i.e., $d = \max_i |\{a_{ie} : a_{ie} > 0\}|$. Assume that F is $\left(\lambda, \frac{\mu}{\ln(1+2d^2)}\right)$-min-locally-smooth for some parameters $\lambda > 0$ and $\mu < 1$. Then there exists a $O\left(\frac{\lambda}{1-\mu} \cdot \ln d\right)$-competitive algorithm for the fractional covering problem.*

Proof. We will bound the increases of the cost and the dual objective at any time τ in the execution of Algorithm 1. Let A^* be the current set of resources e such that $x_e = 1$. The derivative of the objective with respect to τ is:

$$\sum_e \nabla_e F(\mathbf{x}) \cdot \frac{\partial x_e}{\partial \tau} = \sum_{\substack{e: b_{k,e,A^*} > 0 \\ x_e < 1}} \nabla_e F(\mathbf{x}) \cdot \frac{b_{k,e,A^*} \cdot x_e + 1/d}{\lambda \cdot \beta_e}$$
$$\leq \sum_{e: b_{k,e,A^*} > 0} \left(b_{k,e,A^*} \cdot x_e + \frac{1}{d} \right) \leq 2. \tag{4}$$

The first inequality follows $\nabla_e F(\mathbf{x}) \leq \lambda \cdot \beta_e$. The second inequality is due to the definition of d and the fact that $\sum_{e \notin A^*} b_{k,e,A^*} \cdot x_e \leq 1$ always holds during the algorithm.

For a time τ, let $U(\tau)$ be the set of resources e such that $\sum_i \sum_{A: e \notin A} b_{i,e,A} \alpha_{i,A} = \beta_e$ and $b_{k,e,A^*} > 0$. Note that $|U(\tau)| \leq d$ by definition of d. As long as $\sum_{e \notin A^*} b_{k,e,A^*} x_e < 1$, by Lemma 8, we have for every $e \in U(\tau)$,

$$\frac{1}{b_{k,e,A^*}} > x_e \geq \frac{1}{\max b_{i,e,A} \cdot d} \left[\exp\left(\ln(1 + 2d^2) \right) - 1 \right].$$

Therefore, $\frac{b_{k,e,A^*}}{\max_i b_{i,e,A}} \leq \frac{1}{2d}$.

We are now bounding the increase of the dual at time τ. The derivative of the dual with respect to τ is:

$$\frac{\partial D}{\partial \tau} = \sum_i \sum_A \frac{\partial \alpha_{i,A}}{\partial \tau} + \frac{\partial \gamma}{\partial \tau} = \sum_i c_{i,A^*} \cdot \frac{\partial \alpha_{i,A^*}}{\partial \tau} + \frac{\partial \gamma}{\partial \tau}$$
$$= \frac{1}{\lambda \cdot \ln(1 + 2d^2)} \left(1 - \sum_{e \in U(\tau)} \frac{b_{k,e,A^*}}{b_{m_e^*,e,A}} \right) - \frac{\mu}{4\lambda \cdot \ln(1 + 2d^2)} \sum_e \nabla_e F(\mathbf{x}) \cdot \frac{\partial x_e}{\partial \tau}$$
$$\geq \frac{1}{\lambda \cdot \ln(1 + 2d^2)} \left(1 - \sum_{e \in U(\tau)} \frac{1}{2d} \right) - \frac{\mu}{2\lambda \cdot \ln(1 + 2d^2)} \geq \frac{1 - \mu}{2\lambda \cdot \ln(1 + 2d^2)}.$$

The third equality holds since α_{k,A^*} is increased and other α-variables in $U(\tau)$ are decreased. The first inequality uses the fact that $\frac{b_{k,e,A^*}}{\max_i b_{i,e,A^*}} \leq \frac{1}{2d}$ and Inequality (4). The last inequality holds since $|U(\tau)| \leq d$. Hence, the competitive ratio is $O\big(\frac{\lambda}{1-\mu} \cdot \ln d\big)$. ◀

Applications

In this section, we are interested in deriving fractional solutions for different classes of objective function. Rounding schemes (in order to obtain integral solution) for concrete problems are problem-specific and are not the emphasized points here. (Note that several rounding techniques have been shown for different problems, for example in [2] for polynomials with non-negative coefficients, or using online contention resolution schemes for submodular functions [12].)

Polynomials with non-negative coefficients. Let $g_\ell : \mathbb{R} \to \mathbb{R}$ for $1 \leq \ell \leq L$ be degree-k polynomials with non-negative coefficients and the cost function $f : \{0,1\}^n \to \mathbb{R}^+$ defined as $f(\mathbf{1}_S) = \sum_\ell b_\ell g_\ell\big(\sum_{e \in S} a_e\big)$ where $a_e \geq 0$ for every e and $b_\ell \geq 0$ for every $1 \leq \ell \leq L$. The multilinear extension F (of function f) is $((k \ln k)^{k-1}, \frac{k-1}{k \ln d})$-min-locally-smooth. So our algorithm yields the competitive ratio $O\big((k \ln d)^k\big)$. This result recovers the one proved in [2] for this class of cost functions. Note that Azar et al. [2] gave randomized algorithms for

several problems by rounding their fractional solutions. As one can approach a multilinear extension of any function up to a high precision [23], applying the same rounding schemes in [2] for the corresponding problems based on our fractional solutions, one can obtain randomized algorithms with similar bounds as in [2].

Beyond convex functions. Consider the following natural cost functions which represent more practical costs when serving clients as mentioned in the introduction (the cost initially increases fast then becomes more stable before growing quickly again). Let $g : \mathbb{R} \to \mathbb{R}$ be a non-convex function defined as $g(y) = y^k$ if $y \le M_1$ or $y \ge M_2$ and $g(y) = g(M_1)$ if $M_1 \le y \le M_2$ where $M_1 < M_2$ are some constant parameters. The cost function $f : \{0,1\}^n \to \mathbb{R}^+$ defined as $f(\mathbf{1}_S) = g\big(\sum_{e\in S} a_e\big)$ where $a_e \ge 0$ for every e. For this function f, the multilinear extension F is also $((k \ln k)^{k-1}, \frac{k-1}{k \ln d})$-min-locally-smooth. Hence, our algorithm is $O\big((k \ln d)^k\big)$-competitive for minimizing the non-convex objective function defined above under covering constraints.

4 Primal-Dual Framework for Packing Problems

Formulation. We say that $S \subset \mathcal{E}$ is a *configuration* if $\mathbf{1}_S$ corresponds to a feasible solution. Let x_e be a variable indicating whether the resource e is used. For configuration S, let z_S be a variable such that $z_S = 1$ if and only if $x_e = 1$ for every resource $e \in S$, and $x_e = 0$ for $e \notin S$. In other words, $z_S = 1$ iff $\mathbf{1}_S$ is the selected solution of the problem. We consider the following formulation and the dual of its relaxation.

$$\max \sum_S f(\mathbf{1}_S) z_S$$

$$\sum_e b_{i,e} \cdot x_e \le 1 \qquad \forall i$$

$$\sum_{S: e\in S} z_S = x_e \qquad \forall e$$

$$\sum_S z_S = 1$$

$$x_e, z_S \in \{0,1\} \qquad \forall e, S$$

$$\min \sum_i \alpha_i + \gamma$$

$$\sum_i b_{i,e} \cdot \alpha_i \ge \beta_e \qquad \forall e$$

$$\gamma + \sum_{e\in S} \beta_e \ge f(\mathbf{1}_S) \qquad \forall S$$

$$\alpha_i \ge 0 \qquad \forall i$$

In the primal, the first constraints represent the given polytope. Note that the box constraint $x_e \le 1$ is included among these constraints. The second constraint ensures that if a resource e is chosen then the selected solution must contain e. The third constraint says that one solution (configuration) must be selected.

Algorithm. Assume that function $F(\cdot)$ is (λ, μ)-max-locally smooth. Let d be the maximal number of positive entries in a row, i.e., $d = \max_i |\{b_{ie} : b_{ie} > 0\}|$. Define $\rho = \max_i \max_{e,e': b_{ie'}>0} b_{ie}/b_{ie'}$. Denote $\nabla_e F(\mathbf{x}) = \partial F(\mathbf{x})/\partial x_e$. In the algorithm, at the arrival of a new resource e, while $\nabla_e F(\mathbf{x}) > 0$ (i.e., one can still improve the cost by increasing x_e) and $\sum_i b_{i,e}\alpha_e \le \frac{1}{\lambda}\nabla_e F(\mathbf{x})$, the primal variable x_e and dual variables α_i's are increased by appropriate rates. We will argue in the analysis that the primal/dual solutions returned by the algorithm are feasible. Recall that by definition of the multilinear extension, $\nabla_e F(\mathbf{x}) = \mathbb{E}\big[f\big(\mathbf{1}_{R\cup\{e\}}\big) - f\big(\mathbf{1}_R\big)\big]$ where R is a random subset of resources $N \setminus \{e\}$ such

that e' is included with probability $x_{e'}$. Therefore, during the iteration of the while loop with respect to resource e, only x_e is modified and $x_{e'}$ remain fixed for $e' \neq e$, so $\nabla_e F(\mathbf{x})$ is constant during the iteration.

Algorithm 2 Algorithm for Packing Constraints.

All primal and dual variables are initially set to 0.
At every step, always maintain $z_S = \prod_{e \in S} x_e \prod_{e \notin S}(1 - x_e)$.
Upon the arrival of new resource e.
while $\sum_i b_{i,e}\alpha_i \leq \frac{1}{\lambda}\nabla_e F(\mathbf{x})$ and $\nabla_e F(\mathbf{x}) > 0$ **do**
  Increase τ at rate 1 and increase x_e at rate $\frac{1}{\nabla_e F(\mathbf{x}) \cdot \ln(1+d\rho)}$.
  for i such that $b_{i,e} > 0$ **simultaneously do**
    Increase α_i according to the following function $\frac{\partial \alpha_i}{\partial \tau} \leftarrow \frac{b_{i,e} \cdot \alpha_i}{\nabla_e F(\mathbf{x})} + \frac{1}{d\lambda}$
  end for
end while

Dual variables. Variables α_i's are constructed in the algorithm. Let $\mathbf{x}$ be the current solution of the algorithm and let $\mathbf{x}^e$ be the solution after the while loop with respect to resource e. Define $\gamma = \frac{\mu}{\lambda} F(\mathbf{x})$ where $\mathbf{x}$ and $\beta_e = \frac{1}{\lambda} \cdot \nabla_e F(\mathbf{x}^e)$. Note that by the observation above, during the while loop with respect to resource e, $\beta_e = \frac{1}{\lambda} \cdot \nabla_e F(\mathbf{x})$.

The following lemma gives a lower bound on α-variables.

▶ **Lemma 10.** *At any moment during the execution of the algorithm, it always holds that for every i*

$$\alpha_i \geq \frac{\nabla_e F(\mathbf{x})}{\max_{e'} b_{i,e'} \cdot d\lambda} \left[\exp\left(\ln\big(1 + d\rho\big) \cdot \sum_{e'} b_{i,e'} \cdot x_{e'} \right) - 1 \right].$$

▶ **Lemma 11.** *The dual variables defined as above are feasible.*

▶ **Theorem 6.** *Let F be the multilinear extension of the objective cost f. Denote the row sparsity $d := \max_i |\{b_{ie} : b_{ie} > 0\}|$ and $\rho := \max_i \max_{e,e' : b_{ie'} > 0} b_{ie}/b_{ie'}$. Assume that F is (λ, μ)-max-locally-smooth for some parameters $\lambda > 0$ and μ. Then there exists a $O\big(\frac{2\ln(1+d\rho)+\mu}{\lambda}\big)$-competitive algorithm for the fractional packing problem.*

Applications to online submodular maximization

Consider a online submodular maximization subject to packing constraints. We incorporate additional constraints $x_e \leq 2/3$ (instead of box constraint $x_e \leq 1$) for every e. Note that given a feasible solution $\mathbf{x}$ to the original problem, the solution $\frac{2}{3}\mathbf{x}$ also satisfies the original contraints since they are packing constraints. The advantage of adding these new constraints, as shown below, is that we can bound the smooth parameters while loosing only a constant factor in the competitive ratio. We are now determining smooth parameters of the multilinear extension F.

▶ **Lemma 12.** *Let f be an arbitrary submodular function. Then, the multilinear extension F is (1,1)-smooth if f is monotone and is $(1/3, 1)$-smooth if f is non-monotone.*

Proof. For arbitrary submodular function f, it holds that [11, Lemma III.5] for any vector $\mathbf{y}$ and any subset S, $F\big(\mathbf{1}_S \vee \mathbf{y}\big) \geq (1 - \max_e y_e) F(\mathbf{1}_S)$. Moreover, if f is monotone, $F\big(\mathbf{1}_S \vee \mathbf{y}\big) \geq F(\mathbf{1}_S)$.

Consider arbitrary vectors $\mathbf{x}^e$ and let $\mathbf{x} = \bigvee_e \mathbf{x}^e$. As F is the linear extension of a submodular function, $\nabla_e F(\mathbf{x}^e) \geq \nabla_e F(\mathbf{x}) = \mathbb{E}\big[f\big(\mathbf{1}_{R\cup\{e\}}\big) - f\big(\mathbf{1}_R\big)\big]$ where R is a random subset of resources $N \setminus \{e\}$ such that e' is included with probability $x_{e'}$. Therefore, for any subset S,

$$\begin{aligned} F(\mathbf{x}) + \sum_{e\in S} \nabla_e F(\mathbf{x}^e) &\geq F(\mathbf{x}) + \sum_{e\in S} \mathbb{E}\big[f\big(\mathbf{1}_{R\cup\{e\}}\big) - f\big(\mathbf{1}_R\big)\big] \\ &= \mathbb{E}\left[f(\mathbf{1}_R) + \sum_{e\in S}\big[f\big(\mathbf{1}_{R\cup\{e\}}\big) - f\big(\mathbf{1}_R\big)\big]\right] \geq \mathbb{E}\big[f(\mathbf{1}_{R\cup S})\big] = F\big(\mathbf{1}_S \vee \mathbf{x}\big) \\ &\geq \begin{cases} F(\mathbf{1}_S) & \text{if } f \text{ monotone,} \\ (1-\max_e x_e)F(\mathbf{1}_S) & \text{otherwise} \end{cases} \geq \begin{cases} F(\mathbf{1}_S) & \text{if } f \text{ monotone,} \\ 1/3\cdot F(\mathbf{1}_S) & \text{otherwise} \end{cases} \end{aligned}$$

where the second inequality is due to the submodularity of f, and the last inequality holds since $x_e \leq 2/3$ for every e. The lemma follows. ◀

The previous lemma and Theorem 6 lead to the following result.

▶ **Proposition 13.** *Algorithm 2 yields a $O\big(\ln(1+d\rho)\big)$-competitive fractional solution for maximizing (arbitrary) submodular functions under packing constraints.*

One can derive online randomized algorithms for specific problems by rounding the fractional solutions. For example, using the online contention resolution rounding schemes [12], one can obtain randomized algorithms for several specific constraint polytopes, for example, knapsack polytopes, matching polytopes and matroid polytopes.

5 Conclusion

In the paper, we have presented and systematically used a primal-dual framework to design competitive algorithms for problems with non-convex objective. Determining competitive ratios for several problems are now reduced to computing parameters of corresponding smoothness notions. We hope that the primal-dual approach based on configuration LPs, as well as smoothness notions, would provide useful tools to study non-linear/non-convex problems in different contexts. One interesting open question is to prove lower bounds for the considered problems in terms of smooth parameters.

References

1. S. Anand, Naveen Garg, and Amit Kumar. Resource augmentation for weighted flow-time explained by dual fitting. In *Proc. 23rd ACM-SIAM Symposium on Discrete Algorithms*, pages 1228–1241, 2012.
2. Yossi Azar, Niv Buchbinder, TH Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph Seffi Naor, and Debmalya Panigrahi. Online algorithms for covering and packing problems with convex objectives. In *Proc. Symposium on Foundations of Computer Science (FOCS)*, 2016.
3. Yossi Azar, Nikhil R. Devanur, Zhiyi Huang, and Debmalya Panigrahi. Speed scaling in the non-clairvoyant model. In *Proc. 27th ACM on Symposium on Parallelism in Algorithms and Architectures*, pages 133–142, 2015.
4. Avrim Blum, Anupam Gupta, Yishay Mansour, and Ankit Sharma. Welfare and profit maximization with production costs. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 77–86. IEEE, 2011.

5 Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

6 Niv Buchbinder, Moran Feldman, and Roy Schwartz. Online submodular maximization with preemption. In *Proc. 26th Symposium on Discrete Algorithms*, pages 1202–1216, 2015.

7 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.

8 Niv Buchbinder, Joseph Seffi Naor, R Ravi, and Mohit Singh. Approximation algorithms for online weighted rank function maximization under matroid constraints. In *International Colloquium on Automata, Languages, and Programming*, pages 145–156, 2012.

9 Nikhil R. Devanur and Zhiyi Huang. Primal dual gives almost optimal energy efficient online algorithms. In *Proc. 25th ACM-SIAM Symposium on Discrete Algorithms*, 2014.

10 Nikhil R. Devanur and Kamal Jain. Online matching with concave returns. In *Proc. 44th ACM Symposium on Theory of Computing*, pages 137–144, 2012.

11 Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *Symposium on Foundations of Computer Science (FOCS)*, pages 570–579, 2011.

12 Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *Proc. 27th Symposium on Discrete Algorithms*, pages 1014–1033, 2016.

13 Anupam Gupta, Ravishankar Krishnaswamy, and Kirk Pruhs. Online primal-dual for non-linear optimization with applications to speed scaling. In *Proc. 10th Workshop on Approximation and Online Algorithms*, pages 173–186, 2012.

14 Zhiyi Huang and Anthony Kim. Welfare maximization with production costs: a primal dual approach. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 59–72. SIAM, 2015.

15 Sascha Hunold. One step toward bridging the gap between theory and practice in moldable task scheduling with precedence constraints. *Concurrency and Computation: Practice and Experience*, 27(4):1010–1026, 2015.

16 Sungjin Im, Janardhan Kulkarni, Kamesh Munagala, and Kirk Pruhs. Selfishmigrate: A scalable algorithm for non-clairvoyantly scheduling heterogeneous processors. In *Proc. 55th IEEE Symposium on Foundations of Computer Science*, 2014.

17 Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pages 71–104. Cambridge University Press, 2014.

18 Konstantin Makarychev and Maxim Sviridenko. Solving optimization problems with diseconomies of scale via decoupling. In *Proc. 55th Symposium on Foundations of Computer Science*, pages 571–580, 2014.

19 Ishai Menache and Mohit Singh. Online caching with convex costs. In *Proceedings of the 27th ACM symposium on Parallelism in Algorithms and Architectures*, pages 46–54. ACM, 2015.

20 Viswanath Nagarajan and Xiangkun Shen. Online covering with sum of ℓ_q-norm objectives. In *Proc. 44th Colloquium on Automata, Languages and Programming (ICALP)*, 2017.

21 Kim Thang Nguyen. Lagrangian duality in online scheduling with resource augmentation and speed scaling. In *Proc. 21st European Symposium on Algorithms*, pages 755–766, 2013.

22 Tim Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM*, 62(5):32, 2015.

23 Jan Vondrák. Polyhedral techniques in combinatorial optimization. Lecture notes, November 18 2010.