Mathematical Programming (2020) 184:155–182
https://doi.org/10.1007/s10107-019-01409-9

FULL LENGTH PAPER

Series A

Online covering with ℓ_q-norm objectives and applications to network design

Xiangkun Shen[1] · Viswanath Nagarajan[1]

Received: 27 June 2018 / Accepted: 5 June 2019 / Published online: 18 June 2019

Abstract

We consider fractional online covering problems with ℓ_q-norm objectives as well as its dual packing problems. The problem of interest is of the form $\min\{f(x) : Ax \geq 1, x \geq 0\}$ where $f(x) = \sum_e c_e \|x(S_e)\|_{q_e}$ is the weighted sum of ℓ_q-norms and A is a non-negative matrix. The rows of A (i.e. covering constraints) arrive online over time. We provide an online $O(\log d + \log \rho)$-competitive algorithm where $\rho = \frac{a_{\max}}{a_{\min}}$ and d is the maximum of the row sparsity of A and $\max |S_e|$. This is based on the online primal-dual framework where we use the dual of the above convex program. Our result is nearly tight (even in the linear special case), and it expands the class of convex programs that admit online algorithms. We also provide two applications where such convex programs arise as relaxations of discrete optimization problems, for which our result leads to good online algorithms. In particular, we obtain an improved online algorithm (by two logarithmic factors) for non-uniform buy-at-bulk network design and a poly-logarithmic competitive ratio for throughput maximization under ℓ_p-norm capacities.

Mathematics Subject Classification 90C25 · 68W27

1 Introduction

The online primal-dual method is a widely used approach for online problems. This involves solving a discrete optimization problem online as follows (i) formulate a linear

A preliminary version of this paper appeared in the proceedings of ICALP 2017 [33].
Part of V. Nagarajan's work was done while visiting the Simons institute for theoretical computer science (UC Berkeley) during the *Algorithms and Uncertainty* program. This research was supported in part by NSF CAREER Grant CCF-1750127.

✉ Viswanath Nagarajan
viswa@umich.edu

1 Department of Industrial & Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA

programming relaxation and obtain a primal-dual online algorithm for it; (ii) obtain an online rounding algorithm for the resulting fractional solution. While this is similar to a linear programming (LP) based approach for offline optimization problems, a key difference is that solving the LP relaxation in the online setting is highly non-trivial. (Recall that there are general polynomial time algorithms for solving LPs offline.) So there has been a lot of effort in obtaining good online algorithms for various classes of LPs: see [1,15,26] for pure covering LPs, [15] for pure packing LPs and [5] for certain mixed packing/covering LPs. Such online LP solvers have been useful in obtaining online algorithms for various problems, eg. set cover [2], facility location [1], machine scheduling [5], caching [8] and buy-at-bulk network design [24].

Recently, [6] initiated a systematic study of online fractional covering and packing with *convex* objectives; see also the full versions [7,13,17]. These papers obtained good online algorithms for a large class of fractional convex covering problems. They also demonstrated the utility of this approach via many applications that could not be solved using just online LPs. However these results were limited to convex objectives $f : \mathbb{R}^n_+ \to \mathbb{R}_+$ satisfying a monotone gradient property, i.e. $\nabla f(z) \geq \nabla f(y)$ pointwise for all $z, y \in \mathbb{R}^n$ with $z \geq y$. There are however many natural convex functions that do not satisfy such a gradient monotonicity condition. Note that this condition requires the Hessian $\nabla^2 f(x)$ to be pointwise non-negative in addition to convexity which only requires $\nabla^2 f(x)$ to be positive semidefinite.

One of the goals in this paper is to expand the class of convex programs with good online algorithms. To this end, we focus on convex functions f that are sums of different ℓ_q-norms. This is a canonical class of convex functions with non-monotone gradients and prior results are not applicable; see Sect. 1.1 for more details. Another goal in this paper is to obtain better online algorithms for discrete optimization using such convex relaxations. Here, we show that sum of ℓ_q-norm objectives arise naturally as relaxations of some network design/routing problems for which our result leads to better online algorithms.

We show that covering programs with sums of ℓ_q-norm objectives (and their dual packing programs) admit an online algorithm with a logarithmic competitive ratio. This result is nearly tight because there is a logarithmic lower bound even for online covering LPs (which corresponds to an ℓ_1 norm objective).

We also provide two applications of our fractional solver. The first is a covering application: we obtain improved competitive ratios (by two logarithmic factors) for online non-uniform buy-at-bulk problems. The second is a packing application: we obtain the first poly-logarithmic online algorithm for throughput maximization with "group" edge capacities where there is an ℓ_p-norm constraint on the flows through some subsets of edges.

Given that we achieve log-competitive online algorithms for sums of ℓ_q-norms, a natural question is whether such a result holds for all norms. Recall that any norm is a convex function. It turns out that a log-competitive algorithm is not possible for general norms. This follows from a result in [7] which shows an $\Omega(q \log d)$ lower bound for minimizing the objective $\|Bx\|_q$ under covering constraints (where B is a non-negative matrix and d is the row-sparsity of the constraint matrix A). It is still an interesting open question to identify the correct competitive ratio for general norm functions.

1.1 Our results and techniques

We consider the online covering problem

$$\begin{aligned} \min \quad & \sum_{e=1}^{r} c_e \|x(S_e)\|_{q_e} \\ \text{s.t.} \quad & Ax \geq \mathbf{1}, \\ & x \geq \mathbf{0}. \end{aligned} \tag{P}$$

Above, each $S_e \subseteq [n] := \{1, 2, \dots n\}$, $q_e \geq 1$, $c_e \geq 0$ and A is a non-negative $m \times n$ matrix. For any $x \in \mathbb{R}^n$ and $S \subseteq [n]$, we use $x(S) \in \mathbb{R}^{|S|}$ to denote the vector with coordinates $(x_i)_{i \in S}$; moreover, given any $q \geq 1$ we use $\|x(S)\|_q = \left(\sum_{i \in S} x_i^q\right)^{1/q}$. For any subset $S \subseteq [n]$ we use $\overline{S} := [n] \backslash S$. We also consider the dual of the above convex program, which is the following packing problem:

$$\begin{aligned} \max \quad & \sum_{k=1}^{m} y_k \\ \text{s.t.} \quad & A^T y = \mu, \\ & \sum_{e=1}^{r} \mu_e = \mu, \\ & \|\mu_e(S_e)\|_{p_e} \leq c_e, \quad \forall e \in [r], \\ & \mu_e(\overline{S}_e) = \mathbf{0}, \quad \forall e \in [r], \\ & y \geq \mathbf{0}. \end{aligned} \tag{D}$$

The values p_e above satisfy $\frac{1}{p_e} + \frac{1}{q_e} = 1$; so $\|\cdot\|_{p_e}$ is the dual norm of $\|\cdot\|_{q_e}$. This dual can be derived from (P) using Lagrangian duality; see Sect. 2.

Our framework captures the classic setting of packing/covering LPs when $r = n$ and for each $e \in [n]$ we have $S_e = \{e\}$ and $q_e = 1$. Our first main result is:

Theorem 1 *There is an $O(\log d + \log \rho)$-competitive online algorithm for (P) and (D) where the covering constraints in (P) and variables y in (D) arrive over time. Here $\rho = \frac{a_{\max}}{a_{\min}}$, $a_{\max} := \max\{a_{ij}\}$, $a_{\min} := \min\{a_{ij} : a_{ij} > 0\}$, and d is the maximum of the row-sparsity of A and $\max_{e=1}^{r} |S_e|$.*

We note that this bound is also the best possible, even in the linear case [15] when we require monotone primal and dual variables. For just the covering problem, a better $O(\log d)$ bound is known for linear objectives [26] and for monotone-gradient convex functions [6]: these results involve non-monotone dual variables. Obtaining a similar $O(\log d)$ bound for our covering program (P) remains an open question.

The algorithm in Theorem 1 is the natural extension of the primal-dual approach for online LPs [15]. We use the gradient $\nabla f(x)$ at the current primal solution x as the cost function, and use this to define a multiplicative update for the primal. Simultaneously, the dual solution y is increased additively. This algorithm is in fact identical to the one in [6] for convex functions with monotone gradients. Our contribution here is in the analysis of this algorithm, which requires new ideas to deal with non-monotone gradients. In Appendix B, we show that a naive approach using the previous proof ideas fails.

We also provide two applications of Theorem 1, one using the result for the covering problem (P) and another using the packing problem (D).

Non-uniform multicommodity buy-at-bulk. This is a well-studied network design problem in the offline setting [18,19]. The setting is as follows. We are given an undirected (or directed) graph $G = (V, E)$ with a monotone subadditive cost function $g_e : \mathbb{R}_+ \to \mathbb{R}_+$ on each edge $e \in E$ and a collection $\{(s_i, t_i)\}_{i=1}^m$ of m source/destination pairs. The goal is to find an $s_i - t_i$ path P_i for each $i \in [m]$ such that the objective $\sum_{e \in E} g_e(load_e)$ is minimized; here $load_e$ is the number of paths using e. In its online version, the source-destination pairs arrive incrementally over time and we need to select the path for each pair immediately upon arrival. The first poly-logarithmic competitive ratio for the online problem was obtained recently in [24]. A key step in this result was a fractional online algorithm for a specific mixed packing-covering LP. By utilizing Theorem 1 we improve the competitive ratio of this step from $O(\log^3 n)$ to $O(\log n)$ which is also the best possible. Combined with the other steps in [24], we obtain:

Theorem 2 *There is an $O(\alpha\beta\gamma \cdot \log^3 n)$-competitive ratio for non-uniform multicommodity buy-at-bulk, where α is the "junction tree" approximation ratio, β is the integrality gap of the natural LP relaxation for single-source instances, and γ is the competitive ratio for single-source instances.*

See Sect. 4 for more details on the parameters α, β and γ. The corresponding competitive ratio in [24] was $O(\alpha\beta\gamma \cdot \log^5 n)$. In particular, for undirected multicommodity buy-at-bulk we obtain an $O(\log^9 n)$ competitive ratio, improving over the $O(\log^{11} n)$ ratio in [24].

The main idea in Theorem 2 is a reformulation of the LP from [24] as a pure covering program where the objective is a sum of ℓ_1 and ℓ_∞ norms. This reformulation uses the equivalence of maximum-flow and minimum-cut. The resulting covering program has an exponential number of constraints: but we still obtain a polynomial-time online algorithm using a suitable separation oracle.

Throughput maximization with *ℓ_p-norm capacities.* The online problem of maximizing throughput subject to edge capacities is a classic online optimization problem [4,15]. Here we are given a directed graph with m edges and edge capacities $u(e)$. Source/destination requests (s_i, t_i) arrive in an online fashion. An algorithm needs to select a subset of requests to accept and assign a path to each accepted request so that the load on each edge e is at most u_e. The goal is to maximize the number of accepted requests. We consider a natural generalization where there are capacity constraints on *groups* of edges: each such constraint requires the ℓ_{p_j}-norm of the loads on some edge-subset S_j to be at most a given capacity c_j.

Theorem 3 *The throughput maximization problem with ℓ_p-norm capacities admits a randomized $O(\log m)$-competitive algorithm when:*

1. *the capacity $c_j = \Omega(\log m)\,|S_j|^{1/p_j}$ for each group j, or*
2. *each capacity may be violated by an $O(\log^{1+1/p} m)$ factor and $p = \min_j p_j$,*

where m is maximum of the number of edges and the number of constraint subsets.

The two algorithms above rely on different convex relaxations, both of which have the form of our dual program (D). So Theorem 1 can be used directly to solve these convex programs. In order to obtain integral solutions, the algorithms use a natural randomized rounding. The first algorithm runs in polynomial time, whereas the second algorithm takes exponential time. On the other hand, the second algorithm achieves a better capacity violation when $|S_j|$ is large.

We note that some "high capacity" assumption is required (regardless of running time) to obtain any sub-polynomial competitive ratio even in the usual throughput maximization problem where each $|S_j| = 1$ [4,10].

Our approach for Theorem 1. We first show that by duplicating variables and using an online separation oracle approach one can ensure that the sets $\{S_e\}_{e=1}^r$ are disjoint. The use of a separation oracle in the online context is similar to [1]. The disjoint structure of S_es allows for a simple expression for ∇f which is useful in the later analysis. Then we utilize the specific form of the primal-dual convex programs (P) and (D) and an explicit expression for ∇f to show that the dual y is approximately feasible. In particular we show that $\|y^T A(S_e)\|_{p_e} \le O(\log d\rho) \cdot c_e$ for each $e \in [r]$; here $A(S_e)$ denotes the submatrix of A with columns from S_e. Note that this is a weaker requirement than upper bounding $A^T y$ pointwise by $\nabla f(\bar{x})$ which was the approach in [6] for functions with monotone gradients.

In order to bound $\|y^T A(S_e)\|_{p_e}$, we analyze each $e \in [r]$ separately. We partition the steps of the algorithm into *phases* where phase j corresponds to steps where $\Phi_e = \sum_{i \in S_e} x_i^{q_e} \approx \theta^j$; here $\theta > 1$ is a parameter that depends on q_e. The number of phases can be bounded using the fact that Φ_e is monotonically increasing. By triangle inequality we upper bound $\|y^T A(S_e)\|_{p_e}$ by $\sum_j \|y_{(j)}^T A(S_e)\|_{p_e}$ where $y_{(j)}$ denotes the dual variables that arrive in phase j. And in each phase j, we can upper bound $\|y_{(j)}^T A(S_e)\|_{p_e}$ using the differential equations for the primal and dual updates.

1.2 Related work

The online primal-dual framework for linear programs [16] is fairly well understood. Tight results are known for the class of packing and covering LPs [15,26], with competitive ratio $O(\log d)$ for covering LPs and $O(\log d\rho)$ for packing LPs; here d is the row-sparsity and ρ is the ratio of the maximum to minimum entries in the constraint matrix. Such LPs are very useful because they correspond to the LP relaxations of many combinatorial optimization problems. Combining the online LP solver with suitable online rounding schemes, good online algorithms have been obtained for many problems, eg. set cover [2], group Steiner tree [1], caching [8] and ad-auctions [14]. Online algorithms for LPs with mixed packing and covering constraints were obtained in [5]; the competitive ratio was improved in [6]. Such mixed packing/covering LPs were also used to obtain an online algorithm for capacitated facility location [5]. A more complex mixed packing/covering LP was used recently in [24] to obtain online algorithms for non-uniform buy-at-bulk network design: as an application of our result, we obtain a simpler and better (by two log-factors) online algorithm for this problem.

There have also been a number of results utilizing the online primal-dual framework with *convex* objectives for specific problems, eg. matching [22], caching [30], energy-efficient scheduling [21,25] and welfare maximization [11,28]. All of these results involve separable convex/concave functions. Recently, [6] considered packing/covering problems with general (non-separable) convex objectives, but (as discussed previously) this result requires a monotone gradient assumption on the convex function. The sum of ℓ_q-norm objectives considered in this paper does not satisfy this condition.

All the results above (as well as ours) involve convex objectives and linear constraints. We note that [23] obtained online primal-dual algorithms for certain semidefinite programs (i.e. involving non-linear constraints). While both our result and [23] generalize packing/covering LPs, they are not directly comparable.

We also note that online algorithms with ℓ_q-norm objectives have been studied previously for many scheduling problems, eg. [3,9]. More recently [7] used ideas from the online primal-dual approach in an online algorithm for unrelated machine scheduling where the objective is the sum of ℓ_p-norm of loads and startup costs. These results use different techniques and are not directly comparable to ours.

2 Preliminaries

Recall the primal covering problem (P) and its dual packing problem (D). In the online setting, the constraints in the primal and variables in the dual arrive over time. We need to maintain monotonically increasing primal (x) and dual (y) solutions.

Deriving the dual packing problem. We first describe how (D) can be derived as the Lagrangian dual of (P). Let

$$f_e(x) = c_e \|x(S_e)\|_{q_e} \text{ for each } e \in [r] \qquad \text{and} \qquad f(x) = \sum_{e=1}^{r} f_e(x).$$

The Fenchel conjugate of f is $f^*(\mu) = \max_{x \in \mathbb{R}^n_+} \{\mu^T x - f(x)\}$ for $\mu \geq 0$ [12, §3.3.1]. Since $f_e(x) = c_e \|x(S_e)\|_{q_e}$, for $\mu_e \in \mathbb{R}^n_+$,

$$f_e^*(\mu_e) = \begin{cases} 0, & \text{if } \|\mu_e(S_e)\|_{p_e} \leq c_e \text{ and} \mu_e(\overline{S}_e) = \mathbf{0}, \\ \infty, & \text{otherwise.} \end{cases} \tag{1}$$

For completeness, the derivation of $f_e^*(\cdot)$ is shown in Appendix A.

The Lagrangian dual of problem (P) is given by

$$
\begin{aligned}
&\sup_{y\geq 0}\ \inf_{x\geq 0}\left(\sum_{e=1}^{r} c_e\|x(S_e)\|_{q_e} + y^T(\mathbf{1}-Ax)\right)\\
&\quad = \sup_{y\geq 0}\left(\sum_{k=1}^{m} y_k - \sup_{x\geq 0}\left((A^T y)^T x - \sum_{e=1}^{r} c_e\|x(S_e)\|_{q_e}\right)\right)\\
&\quad = \sup_{y\geq 0}\left(\sum_{k=1}^{m} y_k - f^*(A^T y)\right) \qquad (2)
\end{aligned}
$$

where $f^*(\cdot)$ is the conjugate function of $f(\cdot)$. Let $\mu = A^T y$. Note that $\mu \geq \mathbf{0}$ since $y \geq \mathbf{0}$ and A is a nonnegative matrix. We can apply the Moreau-Rockafellar formula [32, §6.8] [34, Thm 3.2] to calculate $f^*(\cdot)$ because $f(x) = \sum_{e=1}^r f_e(x)$ is closed, proper, continuous and convex. This yields:

$$
\begin{aligned}
f^*(\mu) &= f_1^*(\mu) \oplus \cdots \oplus f_r^*(\mu)\\
&= \inf_{\mu_1+\cdots+\mu_r=\mu}\left\{\sum_{e=1}^{r} f_e^*(\mu_e)\right\}, \qquad \forall \mu \in \mathbb{R}^n,
\end{aligned}
$$

where $\oplus$ is the infimal convolution.

Using the expression (1) for $f_e^*(\cdot)$, problem (2) can be reformulated as

$$
\begin{aligned}
\max\quad & \sum_{k=1}^{m} y_k\\
\text{s.t.}\quad & A^T y = \mu,\\
& \sum_{e=1}^{r} \mu_e = \mu,\\
& \|\mu_e(S_e)\|_{p_e} \leq c_e, \quad \forall e \in [r],\\
& \mu_e(\overline{S}_e) = \mathbf{0}, \quad \forall e \in [r],\\
& y \geq \mathbf{0}.
\end{aligned}
$$

which is exactly the packing problem (D).

Note that strong duality holds since the Slater's condition holds [12, §5.2.3], that is, there is $x \in \mathbb{R}^n$ such that $Ax > 1$ and $x > 0$.

Disjointness assumption on $S_e s$. We next show that one can assume that the sets $\{S_e\}_{e=1}^r$ are *disjoint* without loss of generality. This leads to a much simpler expression for ∇f that will be used in Sect. 3. An online algorithm for the covering problem (P) is said to be *primal-dual* if it also maintains dual variables in (D) and the primal objective is bounded in terms of the dual objective.

Lemma 1 *Suppose there is a polynomial time α-competitive algorithm $\mathcal{A}$ for the covering problem (P) with disjoint S_e. Then, there is a polynomial time $O(\alpha)$-competitive algorithm for (P) on general instances. Moreover, if algorithm $\mathcal{A}$ is primal-dual and maintains monotonically non-decreasing dual variables, then there is a polynomial time $O(\alpha)$-competitive algorithm for the packing problem (D) on general instances.*

Proof Let $\mathcal{A}$ denote an α-competitive algorithm for the covering problem (P) with disjoint S_e. We assume that it is a minimal algorithm, that is when constraint k arrives it stops increasing x when $\sum_{i=1}^{n} a_{ki} x_i = 1$. (Any online algorithm can be ensured to be of this form.)

Given an instance $\mathcal{P}_I$ of the covering problem (P) with general $\{S_e\}_{e=1}^{r}$, we define an instance $\mathcal{P}_J$ with disjoint S'_e as follows. For each variable x_i, we introduce r copies $x_i^{(1)}, \ldots, x_i^{(r)}$ where $x_i^{(e)}$ corresponds to the possible occurrence of x_i in S_e. So there are nr variables in $\mathcal{P}_J$. For each $e \in [r]$ we set S'_e to consist of the variables $x_i^{(e)}$ for $i \in S_e$. So $\{S'_e\}_{e=1}^{r}$ are disjoint. For each constraint $a_k^T x \geq 1$ in instance $\mathcal{P}_I$, we introduce a family of r^n constraints in instance $\mathcal{P}_J$ which corresponds to all combinations of the $x_i^{(e)}$ variables, namely

$$\sum_{i=1}^{n} a_{ki} \cdot x_i^{(e_i)} \geq 1, \quad \forall e_1 \in [r],\ e_1 \in [r],\ \ldots e_n \in [r]. \tag{3}$$

If $\bar{x}$ is a feasible solution of $\mathcal{P}_I$, then $x_i^{(e)} = \bar{x}_i$, for all $e \in [r]$ and $i \in [n]$ is a feasible solution to $\mathcal{P}_J$ with the same objective value. Conversely, if x is a feasible solution for $\mathcal{P}_J$ then $\bar{x}_i = \min_{e=1}^{r} x_i^{(e)}$ for all $i \in [n]$ is a feasible solution for $\mathcal{P}_I$ with at most the same objective value. Hence, instances $\mathcal{P}_I$ and $\mathcal{P}_J$ share the same optimal value. So an α-competitive algorithm for $\mathcal{P}_J$ also leads to one for $\mathcal{P}_I$. However, this is not a polynomial time reduction as there are exponentially many constraints in J. In order to deal with this, we use a separation oracle based algorithm, as in [1]. The separation oracle is described as Algorithm 1.

When the k^{th} covering constraint $\sum_{i=1}^{n} a_{ki} x_i \geq 1$ arrives in $\mathcal{P}_I$
while $\sum_{i=1}^{n} a_{ki} \cdot \min_{e=1}^{r} x_i^{(e)} < \frac{1}{2}$ **do**
  let $e_i = \arg\min_{e=1}^{r} x_i^{(e)}$ for all $i \in [n]$;
  add constraint $\sum_{i=1}^{n} a_{ki} \cdot x_i^{(e_i)} \geq 1$ to instance $\mathcal{P}_J$ and run algorithm $\mathcal{A}$;
end
Output current solution $\bar{x}_i = 2 \cdot \min_{e=1}^{r} x_i^{(e)}$ for all $i \in [n]$;

Algorithm 1: Separation Oracle Based Algorithm for General S_e

It is obvious that the output solution is feasible for instance $\mathcal{P}_I$. As x is an α-competitive solution to $\mathcal{P}_J$, the output solution is 2α-competitive for $\mathcal{P}_I$. It remains to show that Algorithm 1 runs in polynomial time upon arrival of any constraint k. For this, define potential function $\psi = \sum_{i=1}^{n} \sum_{e=1}^{r} a_{ki} \cdot x_i^{(e)}$ which is monotone non-decreasing. We know that $\max_{i,e} a_{ki} x_i^{(e)} \leq 1$ since algorithm $\mathcal{A}$ is minimal. So $\psi \leq rn$. In each iteration of Algorithm 1, $\sum_{i=1}^{n} a_{ki} \cdot x_i^{(e_i)}$ increases by at least $\frac{1}{2}$, i.e. ψ also increases by at least $\frac{1}{2}$. So the number of iterations is bounded by $2rn$ which is polynomial. This completes the first part of the proof.

Let $\mathcal{D}_I$ and $\mathcal{D}_J$ be the dual programs for $\mathcal{P}_I$ and $\mathcal{P}_J$ respectively. By strong duality and the fact that $\mathcal{P}_I$ and $\mathcal{P}_J$ share the same optimal value, $\mathcal{D}_I$ and $\mathcal{D}_J$ also have the same optimal value. Let $\mu' \in \mathbb{R}^{nr}$ denote the μ-variables in $\mathcal{D}_J$. Recall from (3) that each constraint k in $\mathcal{P}_I$ corresponds to r^n constraints in $\mathcal{P}_J$: let ${y'}_{k,\ell}$ for $\ell \in [r^n]$ denote the dual variables in $\mathcal{D}_J$ for these constraints. Given a feasible dual solution μ', y' for $\mathcal{D}_J$, we can obtain a feasible solution for $\mathcal{D}_I$ by setting $y_k = \sum_{\ell} {y'}_{k,\ell}$, for all k,

$$\mu_e(i) = \begin{cases} \mu'(e,i) & \text{if } i \in S_e \\ 0 & \text{if } i \notin S_e \end{cases}, \text{for} \quad i \in [n] \quad \text{and} \quad e \in [r],$$

and $\mu = \sum_{e=1}^{r} \mu_e$. Note also that the objective value of (y, μ) in $\mathcal{D}_I$ equals that of (y', μ') in $\mathcal{D}_J$, Furthermore, since the online algorithm maintains monotone variables y', the corresponding y-variables are also monotone. The running time is polynomial (same as for the primal instance). Finally, as the algorithm is primal-dual, we obtain an $O(\beta)$-competitive ratio for the dual problem $\mathcal{D}_I$ as well. This completes the second part of the proof. $\square$

Henceforth we will assume that the sets $\{S_e\}_{e=1}^{r}$ are disjoint. Our algorithm in this case (Sect. 3) is primal-dual and maintains monotone duals. Using Lemma 1 we would then obtain online algorithms for both covering and packing on general instances.

When $\{S_e\}_{e=1}^{r}$ are disjoint, the constraints $\sum_{e=1}^{r} \mu_e = \mu$, $\|\mu_e(S_e)\|_{p_e} \leq c_e$ and $\mu_e(\overline{S}_e) = \mathbf{0}$ for $e \in [r]$ are equivalent to $\|\mu(S_e)\|_{p_e} \leq c_e$ for $e \in [r]$ and $\mu(\cap_e \overline{S}_e) = \mathbf{0}$. Then the dual packing problem (D) simplifies to:

$$\max \left\{ \sum_{k=1}^{m} y_k \ : \ A^T y = \mu, \ \|\mu(S_e)\|_{p_e} \leq c_e \ \forall e \in [r], \ \mu(\cap_e \overline{S}_e) = \mathbf{0}, \ y \geq \mathbf{0} \right\}. \tag{DD}$$

This is the dual program that will be used in Sect. 3. We show below (for completeness) that weak duality holds for the primal program (P) and its dual (DD). We note that strong duality also holds because (P) satisfies Slater's condition; however we do not use this fact in in Sect. 3.

Lemma 2 *For any pair of feasible solutions x to* (P) *and (y, μ) to* (DD)*, we have*

$$\sum_{e=1}^{r} c_e \|x(S_e)\|_{q_e} \geq \sum_{k=1}^{m} y_k.$$

Proof This follows from the following inequalities:

$$\sum_{k=1}^{m} y_k = y^T \mathbf{1} \le y^T Ax = \mu^T x \le \sum_{e=1}^{r} \sum_{i \in S_e} \mu_i \cdot x_i$$
$$\le \sum_{e=1}^{r} \|\mu(S_e)\|_{p_e} \cdot \|x(S_e)\|_{q_e}$$
$$\le \sum_{e=1}^{r} c_e \cdot \|x(S_e)\|_{q_e}.$$

The first inequality is by primal feasibility; the second inequality is by $x \ge 0$, $\mu_i \ge 0$ and $\mu_i = 0$ if $i \in \cap_e \overline{S_e}$. The third inequality is by Hölder's inequality. The last inequality is by dual feasibility. □

3 Algorithm and analysis

Let $f(x) = \sum_{e=1}^{r} c_e \|x(S_e)\|_{q_e}$ denote the primal objective in (P).

When the k^{th} request $\sum_{i=1}^{n} a_{ki} x_i \ge 1$ arrives;
Let τ be a continuous variable denoting the current time;
while *the constraint is unsatisfied, i.e.,* $\sum_{i=1}^{n} a_{ki} x_i < 1$ **do**
  For each i with $a_{ki} > 0$, increase x_i at rate $\frac{\partial x_i}{\partial \tau} = \frac{a_{ki}x_i + \frac{1}{d}}{\nabla_i f(x)} = \frac{a_{ki}x_i + \frac{1}{d}}{c_e x_i^{q_e - 1}} \|x(S_e)\|_{q_e}^{q_e - 1}$;
  ; // If $\nabla_i f(x) = 0$, increase x_i at rate $\frac{\partial x_i}{\partial \tau} = \infty$;
1 Increase y_k at rate $\frac{\partial y_k}{\partial \tau} = 1$;
2 Set $\mu = A^T y$;
end

Algorithm 2: Algorithm for ℓ_q-norm packing/covering

In order to ensure that the gradient ∇f is defined, the primal solution x starts off as $\delta \cdot \mathbf{1}$ where $\delta > 0$ is arbitrarily small. So the initial primal value is at most $n\delta \cdot \max_{e=1}^{r} c_e$, which can be made arbitrarily small.

It is clear that the algorithm maintains a feasible and monotonically non-decreasing primal solution x. The dual solution (y, μ) is also monotonically non-decreasing, but not necessarily feasible. We will show that (y, μ) is $O(\log \rho d)$-approximately feasible, i.e. the packing constraints in (DD) are violated by at most an $O(\log \rho d)$ factor.

Lemma 3 *The primal objective $f(x)$ is at most twice the dual objective $\sum_{k=1}^{m} y_k$.*

Proof We will show that the rate of increase of the primal is at most twice that of the dual. Consider the algorithm upon the arrival of some constraint k. Then

$$\frac{df(x)}{d\tau} = \sum_{i:a_{ki}>0} \nabla_i f(x) \cdot \frac{\partial x_i}{\partial \tau} = \sum_{i:a_{ki}>0} \left(a_{ki}x_i + \frac{1}{d}\right) \le 2.$$

The inequality comes from the fact that (i) the process for the kth constraint is terminated when $\sum_i a_{ki} x_i = 1$ and (ii) the number of non-zeroes in constraint k is at most d. Also it is clear that the dual objective increases at rate one, which finishes the proof. $\square$

Lemma 4 *The dual solution* (y, μ) *is* $O(\log \rho d)$*-approximately feasible, i.e.*

$$\mu(\cap_e \overline{S}_e) = 0, \tag{4}$$

and

$$\|\mu(S_e)\|_{p_e} \;\leq\; O(\log \rho d) \cdot c_e, \qquad \forall e \in [r]. \tag{5}$$

Proof First we prove (4). For any $i \in \cap_e \overline{S}_e$ we always have $\nabla_i f(x) = 0$: we will show that $\mu_i = 0$ always. Consider the arrival of any constraint $\sum_{i=1}^n a_{ki} x_i \geq 1$. If $a_{ki} = 0$ then $\frac{\partial \mu_i}{\partial \tau} = 0$. If $a_{ki} > 0$ then x_i increases at ∞ rate: so the constraint will be satisfied immediately without increasing y_k, so μ_i also stays 0.

In order to prove (5), fix any $e \in [r]$. When $q_e = 1$, the corresponding part of the objective function is reduced to the linear case $c_e \sum_{i \in S_e} x_i$ and we want to prove $\|\mu(S_e)\|_\infty \leq O(\log \rho d) \cdot c_e$ for all $e \in [r]$. It is equivalent to prove that $\mu_i \leq O(\log \rho d) \cdot c_e$ for all $i \in S_e$. In this case, we have

$$\frac{\partial x_i}{\partial \tau} = \frac{a_{ki}\, x_i + \frac{1}{d}}{c_e}, \quad \frac{\partial y_k}{\partial \tau} = 1, \quad \frac{\partial \mu_i}{\partial \tau} = a_{ki}$$
$$\Rightarrow d\mu_i = \frac{c_e\, a_{ki}}{a_{ki}\, x_i + \frac{1}{d}} dx_i$$

This means that the increase in μ_i over the entire algorithm is:

$$\Delta\mu_i \leq \int_0^{\frac{1}{a_{\min}}} \frac{c_e\, a_{ki}}{a_{ki}\, x_i + \frac{1}{d}} dx_i = c_e \cdot \ln\left(\frac{a_{ki} \cdot d}{a_{\min}} + 1\right) = O(\log \rho d) \cdot c_e.$$

Recall that $a_{\min} = \min\{a_{ij} : a_{ij} > 0\}$. The first inequality above follows from the fact that when $x_i = 1/a_{\min}$ all constraints involving i are satisfied (so x_i will not be increased further).

The case $q_e > 1$ is the main part of the analysis. In order to prove the desired upper bound on $\|\mu(S_e)\|_{p_e}$ we use a potential function $\Phi = \sum_{i \in S_e} (x_i^{q_e})$. Let phase zero denote the period where $\Phi \leq \zeta := (\frac{1}{a_{\max} \cdot d^2})^{q_e}$; recall that $a_{\max} = \max\{a_{ij}\}$. For each $\ell \geq 1$, phase ℓ is the period where $\theta^{\ell-1} \cdot \zeta \leq \Phi < \theta^\ell \cdot \zeta$. Here $\theta > 1$ is a parameter depending on q_e that will be determined later. Note that $\Phi \leq d(\frac{1}{a_{\min}})^{q_e}$ as variable x_i will never be increased beyond $1/a_{\min}$. Hence if L denotes the number of phases then $(\frac{1}{a_{\max} \cdot d^2})^{q_e} \theta^L \leq d(\frac{1}{a_{\min}})^{q_e}$, which implies $L \leq 3q_e \cdot \log(d\rho)/\log\theta$. Next, we bound the increase in $\|\mu(S_e)\|_{p_e}$ for each phase separately.

For any phase, we have the following equalities

$$\frac{\partial x_i}{\partial \tau} = \frac{a_{ki}x_i + \frac{1}{d}}{c_e x_i^{q_e-1}} \|x(S_e)\|_{q_e}^{q_e-1}, \qquad \frac{\partial y_k}{\partial \tau} = 1, \qquad \frac{\partial \mu_i}{\partial \tau} = a_{ki}$$

$$\Rightarrow d\mu_i = \frac{c_e\, a_{ki}\, x_i^{q_e-1}}{\left(\sum_{j\in S_e} x_j^{q_e}\right)^{1-\frac{1}{q_e}} \left(a_{ki}x_i + \frac{1}{d}\right)} dx_i. \tag{6}$$

Phase zero. Suppose that each x_i increases to α_i in phase zero. From (6) we have

$$d\mu_i \le \frac{d\, c_e\, a_{ki}\, x_i^{q_e-1}}{\left(\sum_{j\in S_e} x_j^{q_e}\right)^{1-\frac{1}{q_e}}} dx_i \qquad \Rightarrow \qquad \frac{1}{d\, c_e\, a_{ki}} d\mu_i \le \frac{x_i^{q_e-1}}{\left(\sum_{j\in S_e} x_j^{q_e}\right)^{1-\frac{1}{q_e}}} dx_i.$$

This means that the increase $\Delta\mu_i$ in μ_i (during phase zero) can be bounded as:

$$\frac{1}{d\, c_e\, a_{ki}} \Delta\mu_i \le \int_{\delta}^{\alpha_i} \frac{x_i^{q_e-1}}{\left(\sum_{j\in S_e} x_j^{q_e}\right)^{1-\frac{1}{q_e}}} dx_i \le \int_0^{\alpha_i} 1 dx_i \le \alpha_i.$$

Since in phase zero, $\Phi \le (\frac{1}{a_{\max}\cdot d^2})^{q_e}$, we know that each $\alpha_i \le \frac{1}{a_{\max}\cdot d^2}$. So $\Delta\mu_i \le \frac{c_e}{d}$ and at the end of phase zero, we have $\|\mu(S_e)\|_{p_e} \le \|\mu(S_e)\|_1 \le c_e$. The last inequality is because $d \ge \max_e |S_e|$.

Phase $\ell \ge 1$. Let Φ_0 and Φ_1 be the value of Φ at the beginning and end of this phase respectively. In phase ℓ, suppose that each x_i increases from s_i to t_i. Then,

$$d\mu_i = \frac{c_e\, a_{ki}\, x_i^{q_e-1}}{\left(\sum_{j\in S_e} x_j^{q_e}\right)^{1-\frac{1}{q_e}} \left(a_{ki}x_i + \frac{1}{d}\right)} dx_i \le \frac{c_e x_i^{q_e-2}}{\left(\sum_{j\in S_e} x_j^{q_e}\right)^{1-\frac{1}{q_e}}} dx_i$$

So the increase $\Delta\mu_i$ in μ_i during this phase is:

$$\Delta\mu_i \le \int_{s_i}^{t_i} \frac{c_e x_i^{q_e-2}}{\left(\sum_{j\in S_e} x_j^{q_e}\right)^{1-\frac{1}{q_e}}} dx_i.$$

Note that variables $x_{i'}$ for $i' \ne i$ can also increase in this phase: so we cannot directly bound the above integral. This is precisely where the potential Φ is useful. We know that throughout this phase, $\sum_{j\in S_e} x_j^{q_e} \ge \Phi_0$. So the increase in μ_i during this phase is:

$$\Delta\mu_i \le c_e \int_{s_i}^{t_i} \frac{x_i^{q_e-2}}{\Phi_0^{1-\frac{1}{q_e}}} dx_i = c_e \frac{t_i^{q_e-1} - s_i^{q_e-1}}{(q_e-1)\Phi_0^{1-\frac{1}{q_e}}} = c_e \frac{t_i^{q_e-1} - s_i^{q_e-1}}{(q_e-1)\Phi_0^{\frac{1}{p_e}}}.$$

Note that here is where we used the assumption that $q_e > 1$: it is needed in evaluating the integral. Now,

$$\begin{aligned}(\Delta\mu_i)^{p_e} &\le \frac{c_e^{p_e}}{(q_e-1)^{p_e}\,\Phi_0}\cdot\left(t_i^{q_e-1}-s_i^{q_e-1}\right)^{p_e}\\ &\le \frac{c_e^{p_e}}{(q_e-1)^{p_e}\,\Phi_0}\cdot\left(t_i^{(q_e-1)p_e}-s_i^{(q_e-1)p_e}\right)\\ &= \frac{c_e^{p_e}}{(q_e-1)^{p_e}\,\Phi_0}\cdot\left(t_i^{q_e}-s_i^{q_e}\right)\end{aligned}$$

The second inequality above uses the fact that $(z_1+z_2)^{p_e} \ge z_1^{p_e}+z_2^{p_e}$ for any $p_e \ge 1$ and $z_1, z_2 \ge 0$, with $z_1 = s_i^{q_e-1}$ and $z_2 = t_i^{q_e-1} - s_i^{q_e-1}$. The last equality uses $\frac{1}{p_e}+\frac{1}{q_e}=1$.

We can now bound

$$\sum_{i\in S_e}(\Delta\mu_i)^{p_e} \le \frac{c_e^{p_e}}{(q_e-1)^{p_e}\,\Phi_0}\cdot\sum_{i\in S_e}\left(t_i^{q_e}-s_i^{q_e}\right) = \frac{c_e^{p_e}\,(\Phi_1-\Phi_0)}{(q_e-1)^{p_e}\,\Phi_0} \le \frac{c_e^{p_e}(\theta-1)}{(q_e-1)^{p_e}}$$

Let vector $\mu^{(\ell)} \in \mathbb{R}^{S_e}$ denote the increase in variables $\{\mu_i : i \in S_e\}$ during phase ℓ. It follows from the above that $\|\mu^{(\ell)}\|_{p_e} \le \frac{c_e}{q_e-1}(\theta-1)^{1/p_e}$.

Combining across phases. Note that the final vector $\mu = \sum_{\ell\ge 0}\mu^{(\ell)}$. By triangle inequality, we have

$$\|\mu\|_{p_e} \le \sum_{\ell\ge 0}\|\mu^{(\ell)}\|_{p_e} \le c_e + \sum_{\ell\ge 1}\|\mu^{(\ell)}\|_{p_e} \le c_e\left(1+\frac{3q_e(\theta-1)^{1/p_e}}{(q_e-1)\log\theta}\cdot\log(d\rho)\right) \tag{7}$$

The last inequality uses $\|\mu^{(\ell)}\|_{p_e} \le \frac{c_e}{q_e-1}(\theta-1)^{1/p_e}$ and that the number of phases is at most $3q_e\cdot\log(d\rho)/\log\theta$.

To complete the proof we show next that for any $q_e > 1$, there is some choice of $\theta > 1$ such that the right-hand-side above is $O(\log(d\rho))\cdot c_e$.

Case 1: $q_e \ge 2$. In this case, setting $\theta = 2$, we have $\frac{3q_e}{(q_e-1)}(\theta-1)^{1/p_e}/\log\theta \le 6$.

Case 2: $1 < q_e < 2$. Here we set $\theta = 1+(q_e-1)^{-\epsilon p_e}$, where $\epsilon = \frac{1}{-\log(q_e-1)} > 0$.

We have

$$\begin{aligned}\frac{(\theta-1)^{\frac{1}{p_e}}}{\log\theta} &\le \frac{(\theta-1)^{\frac{1}{p_e}}}{\log(q_e-1)^{-\epsilon p_e}} = \frac{(q_e-1)^{-\epsilon}}{\log(q_e-1)^{-\epsilon p_e}}\\ &= \frac{(q_e-1)^{-\epsilon}}{-\epsilon p_e\log(q_e-1)} = \frac{(q_e-1)^{-\epsilon}}{p_e} = \frac{2}{p_e}.\end{aligned}$$

The first inequality above uses that $\theta - 1 = (q_e - 1)^{-\epsilon p_e} > 1$. Thus we have

$$\frac{3q_e(\theta-1)^{1/p_e}}{(q_e-1)\log\theta} \le \frac{6q_e}{(q_e-1)p_e} = 6,$$

where the last equality uses $\frac{1}{p_e} + \frac{1}{q_e} = 1$.

So in either case we have that the right-hand-side of (7) is at most $(1+6\log(d\rho))\cdot c_e$. □

Combining Lemmas 2, 3 and 4, we obtain Theorem 1.

4 Application to online buy-at-bulk network design

In the non-uniform multicommodity buy-at-bulk problem, we are given a directed or undirected graph $G = (V, E)$ with a monotone subadditive cost function $g_e : \mathbb{R}_+ \to \mathbb{R}_+$ on each edge $e \in E$ and a collection $\{(s_i, t_i)\}_{i=1}^m$ of m source/destination pairs. The goal is to find an $s_i - t_i$ path P_i for each $i \in [m]$ such that the objective $\sum_{e\in E} g_e(load_e)$ is minimized; here $load_e$ is the number of paths using e. An equivalent view of this problem involves two costs c_e and ℓ_e for each edge $e \in E$ and the objective $\sum_{e\in\cup P_i} c_e + \sum_{e\in E} \ell_e \cdot load_e$. In the online setting, the pairs (s_i, t_i) arrive over time and we need to decide on the path P_i immediately after the i^{th} pair arrives.

Recently, [24] gave a modular online algorithm for non-uniform buy-at-bulk problems with competitive ratio $O(\alpha\beta\gamma \cdot \log^5 n)$ where:

- β is the integrality gap of the natural LP relaxation for single-source instances, where all s_is correspond to the same node.
- γ is the competitive ratio of an online algorithm for single-source instances.
- α is the "junction tree" approximation ratio. A junction-tree is a specific solution structure (introduced in [19]) that enables a reduction from multicommodity to single-source instances. In such a solution, the m pairs are partitioned into groups and each group $S \subseteq [m]$ corresponds to a root vertex $r \in V$ such that the path for each pair in S goes through r. There is no sharing of costs across groups: in particular, we view the solution for each group as using a distinct copy of the graph. The value α is the worst-case ratio of the cost of a junction-tree solution to the optimum.

One of the main components in the result in [24] was an $O(\log^3 n)$-competitive fractional online algorithm for a certain mixed packing/covering LP. Here we show that Theorem 1 can be used to provide a better (and tight) $O(\log n)$-competitive ratio for the same LP. This leads to the improved $O(\alpha\beta\gamma \cdot \log^3 n)$-competitive ratio stated in Theorem 2.

The LP relaxation. We now describe the LP relaxation used in [24]. Let $\mathcal{T} = \{s_i, t_i : i \in [m]\}$ denote the set of all sources/destinations. For each $i \in [m]$ and root $r \in V$ variable z_{ir} denotes the extent to which both s_i and t_i route to/from r: this corresponds to assigning pair i to the group (in the junction-tree solution) with root r. For each $r \in V$ and $e \in E$, variable x_{er} denotes the extent to which edge e is used in the routing to root r: this corresponds to whether/not edge e is used in the junction-tree solution for root r. For each $r \in V$ and $u \in \mathcal{T}$, variables $\{f_{r,u,e} : e \in E\}$ represent a flow between r and u.

$$\begin{aligned}
\min \quad & \sum_{r\in V}\sum_{e\in E} c_e \cdot x_{e,r} \;+\; \sum_{r\in V}\sum_{e\in E} \ell_e \cdot \sum_{u\in\mathcal{T}} f_{r,u,e} \\
\text{s.t.} \quad & \sum_{r\in V} z_{ir} \geq 1, \qquad \forall i \in [m] \\
& \{f_{r,s_i,e} : e \in E\} \text{ is a flow from } s_i \text{ to } r \text{ of } z_{ir} \text{ units}, \qquad \forall r \in V,\ \forall i \in [m] \\
& \{f_{r,t_i,e} : e \in E\} \text{ is a flow from } r \text{ to } t_i \text{ of } z_{ir} \text{ units}, \qquad \forall r \in V,\ \forall i \in [m] \\
& f_{r,u,e} \leq x_{e,r}, \qquad \forall u \in \mathcal{T},\ \forall e \in E,\ \forall r \in V \\
& x, f, z \geq 0
\end{aligned}$$

The above LP is not of packing or covering type due to the flow constraints: there are both positive and negative signs on variables. The online algorithm in [24] for this LP uses various ideas and has competitive ratio $O(D \cdot \log n)$ w.r.t. the optimal *integral* solution; here D is an upper bound on the length of any $s_i - t_i$ path (note that D can be as large as n). Using a height reduction operation, they could ensure that $D = O(\log n)$ while incurring an additional $O(\log n)$-factor loss in the objective. This lead to the $O(\log^3 n)$ factor for the fractional online algorithm. Here we provide an improved $O(\log n)$-competitive algorithm that does not require any bound on the path-lengths and that also guarantees the approximation relative to the optimal *fractional* solution.

For any $r \in V$ and $u \in \mathcal{T}$, let $\mathsf{MC}(r, u)$ denote the $u - r$ (resp. $r - u$) minimum cut in the graph with edge capacities $\{f_{r,u,e} : e \in E\}$ if u is a source (resp. destination). By the max-flow min-cut theorem, it follows that $z_{ir} \leq \min\{\mathsf{MC}(r, s_i), \mathsf{MC}(r, t_i)\}$. Using this, we can combine the first three constraints of the above LP into the following:

$$\sum_{r\in V} \min\{\mathsf{MC}(r, s_i), \mathsf{MC}(r, t_i)\} \;\geq\; 1, \qquad \forall i \in [m].$$

For a fixed $i \in [m]$, this constraint is equivalent to the following. For each $r \in V$, pick either an $s_i - r$ cut (under capacities $f_{r,s_i,\star}$) or an $r - t_i$ cut (under capacities $f_{r,t_i,\star}$), and check if the total cost of these cuts is at least one. Moreover, given values for the f-variables, it is optimal for the LP to set $x_{er} = \max_{u\in\mathcal{T}} f_{r,u,e}$ for all $e \in E$ and $r \in V$.

This leads to the following equivalent reformulation that eliminates the x and z variables. Below, we use the notation $f_{r,u}(S) = \sum_{e\in S} f_{r,u,e}$ for any subset $S \subseteq E$ and $r \in V$, $u \in \mathcal{T}$.

$$\begin{aligned}
\min \quad & \sum_{r\in V}\sum_{e\in E} c_e \cdot \left(\max_{u\in\mathcal{T}} f_{r,u,e}\right) \;+\; \sum_{r\in V}\sum_{e\in E} \ell_e \cdot \sum_{u\in\mathcal{T}} f_{r,u,e} \\
\text{s.t.} \quad & \sum_{r\in R_s} f_{r,s_i}(S_r) + \sum_{r\in R_t} f_{r,t_i}(T_r) \geq 1, \qquad \forall i \in [m],\ \forall (R_s, R_t) \text{ partition of } V, \\
& \qquad \forall S_r : s_i - r \text{ cut},\ \forall r \in R_s,\ \forall T_r : r - t_i \text{ cut},\ \forall r \in R_t, \\
& f \geq 0.
\end{aligned}$$

Note that $\ell_{\log(n)}$-norm is a constant approximation for ℓ_∞-norm. Therefore we can reformulate the above objective function (at the loss of a constant factor) as the sum of $\ell_{\log(n)}$ and ℓ_1 norms. Our fractional solver applies to this convex covering problem, and yields an $O(\log n)$-competitive ratio; note that $\rho = 1$ for this instance.

In order to get a polynomial running time, we can use the natural "separation oracle" approach (as in Sect. 2) to produce violated covering constraints. This is described in Algorithm 3. Each iteration of Algorithm 3 runs in polynomial time since the minimum cuts can be computed in polynomial time. In order to bound the number of iterations, consider the potential $\psi = \sum_{e\in E}(f_{r,s_i,e} + f_{r,t_i,e})$. Note that $0 \le \psi \le 2|E|$ and each iteration increases ψ by at least $\frac{1}{2}$. So the number of iterations is at most $4|E|$.

```
When the i^th request (s_i, t_i) arrives
while ∑_{r∈V} min {MC(r, s_i), MC(r, t_i)} < 1/2 do
    For each r ∈ V, compute MC(r, s_i) and MC(r, t_i) and the respective cuts S_r and T_r;
    Let R_s = {r ∈ V : MC(r, s_i) ≤ MC(r, t_i)} and R_t = V\R_s;
    Run Algorithm 2 with constraint ∑_{r∈R_s} f_{r,s_i}(S_r) + ∑_{r∈R_t} f_{r,t_i}(T_r) ≥ 1;
end
```

Algorithm 3: Separation Oracle Based Algorithm for Buy-at-Bulk

Results for specific buy-at-bulk problems. Using existing results from offline and single-source versions of these problems, Theorem 2 implies the following:

– For undirected edge-weighted buy-at-bulk we obtain an $O(\log^9 n)$-competitive ratio in polynomial time using $\alpha = O(\log n)$ [19], $\beta = O(\log n)$ [20] and $\gamma = O(\log^4 n)$ [31]. This improves upon the $O(\log^{11} n)$-competitive ratio that follows from [24].
– For undirected node-weighted buy-at-bulk we obtain an $O(\log^9 n)$-competitive ratio in quasipolynomial time using $\alpha = O(\log n)$ [19], $\beta = O(\log n)$ [19] and $\gamma = O(\log^4 n)$ [1,24]. This again improves upon the $O(\log^{11} n)$-competitive ratio that follows from [24]. The quasipolynomial runtime is due to the online single-source algorithm that relies on the height-reduction technique for directed Steiner problems [27].
– As discussed in [24], we can also obtain the same competitive ratios for the prize-collecting variants of these problems, where pairs may be left disconnected by paying a penalty in the objective. So our result implies an $O(\log^9 n)$-competitive ratio here as well.

5 Application to throughput maximization with ℓ_p-norm capacities

The online problem of maximizing multicommodity flow was studied in [4,15]. In this problem, we are given a directed graph $G = (V, E)$ with edge capacities $u(e)$. Requests (s_i, t_i) arrive in an online fashion. The algorithm needs to accept a subset of these requests and choose an $s_i - t_i$ path for each accepted request i. The number of paths using any edge e (referred to as the *load* of edge e) is not allowed to exceed its capacity $u(e)$. The goal is to maximize the number of accepted requests.

Here we consider an extension with capacity constraints on subsets of edges. In particular, we are also given a number of groups where the j^{th} group consists of a subset $S_j \subseteq E$ and requires the ℓ_{p_j}-norm of the loads of these edges to be at most c_j, i.e. $\sum_{e \in S_j} L_e^{p_j} \leq c_j^{p_j}$ where L_e denotes the load of edge e. The objective is again to maximize the number of accepted requests. Note that if each $|S_j| = 1$ then we recover the classic setting of individual edge capacities.

In this section we assume (without loss of generality) that the subsets S_j form a partition of E. By subdividing edges if necessary, we can ensure that the subsets S_j are disjoint. If $\cup_j S_j \subsetneq E$ then we can just add a dummy group consisting of edges $E \setminus \cup_j S_j$ and assign a very high capacity to the dummy group. We denote the number of edges by m; as the groups are disjoint, the number of groups is at most m. We also use i to index requests, j to index groups and e to index edges.

We provide two online algorithms for this problem. The first algorithm (in Sect. 5.1) runs in polynomial time and achieves an $O(\log m)$-competitive ratio when each $c_j = \Omega(\log m) \cdot |S_j|^{1/p_j}$. Without the high-capacity assumption, this implies an $O(\log m)$-competitive ratio while violating capacities by an $O(m^{1/p} \cdot \log m)$ factor, where $p = \min_j p_j$. The second algorithm (in Sect. 5.2) allows for any capacities and achieves an $O(\log m)$-competitive ratio while violating capacities by an $O(\log^{1+1/p} m)$ factor, where $p = \min_j p_j$. The second algorithm provides a better capacity violation than the first (for arbitrary capacities). However, the second algorithm does not run in polynomial time. The two algorithms rely on different convex relaxations, both of which correspond to our dual problem (D). We note that in the absence of a high-capacity assumption (or some capacity violation), there is no sub-polynomial randomized competitive ratio even in the special case where $|S_j| = 1$ [10].

A randomized (α, β)-bicriteria competitive algorithm finds a solution that (i) has expected objective value at least $\frac{1}{\alpha}$ times the offline optimum, and (ii) violates each capacity constraint by at most factor β with probability one.

5.1 Polynomial-time $(O(\log m), O(m^{1/p} \log m))$ bicriteria algorithm

Here we prove the first part of Theorem 3, which is restated below:

Theorem 4 *Assume that $c_j = \Omega(\log m) \cdot |S_j|^{1/p_j}$ for each j. Then there is a polynomial-time randomized $O(\log m)$-competitive online algorithm for throughput maximization with ℓ_p-norm capacities, where m is the maximum of the number of edges in the graph and the number of constraint subsets.*

In particular, we will show that (i) the algorithm's solution satisfies all capacities with probability one and (ii) has expected objective at least an $O(\log m)$ fraction of the optimum.

In a fractional version of the problem, a request can be satisfied by several paths and the allocation of bandwidth can be in the range [0, 1] instead of being restricted to {0, 1}. For request (s_i, t_i), let $\mathcal{P}_i$ be the set of simple paths from s_i to t_i. Variable $f_{i,P}$ is defined to be the amount of flow on the path P for request (s_i, t_i). The total profit is the (fractional) number of requests served. The complete fractional relaxation is given below:

$$
\begin{aligned}
\max \quad & \sum_i \sum_{P \in \mathcal{P}_i} f_{i,P} && && (8)\\
\text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_{i,P} \le 1, && \forall i && (9)\\
& \sum_i \sum_{P \in \mathcal{P}_i : e \in P} f_{i,P} = \mu_e, && \forall e && (10)\\
& \|\mu(S_j)\|_{p_j} \le c_j, && \forall j && (11)\\
& f \ge 0 && && (12)
\end{aligned}
$$

Constraint (9) ensures that at most one path is selected for each request, (10) assigns to each variable μ_e the load on edge e and (11) is the capacity constraint on each group. It is clear that when each $f_{i,P} \in \{0, 1\}$ we obtain an exact formulation of the routing problem. Rewriting constraint (9) as $\sum_{P \in \mathcal{P}_i} f_{i,P} = \nu_i$ and $\nu_i \le 1$, we have the following equivalent relaxation:

$$
\begin{aligned}
\max \quad & \sum_i \sum_{P \in \mathcal{P}_i} f_{i,P} && && (13)\\
\text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_{i,P} = \nu_i, && \forall i && (14)\\
& \sum_i \sum_{P \in \mathcal{P}_i : e \in P} f_{i,P} = \mu_e, && \forall e && (15)\\
& \|\nu\|_\infty \le 1, && && (16)\\
& \|\mu(S_j)\|_{p_j} \le c_j, && \forall j && (17)\\
& f \ge 0 && && (18)
\end{aligned}
$$

Note that this corresponds to the (dual) packing program (D). In particular, if z_i and x_e are the primal variables corresponding to constraints (14) and (15) respectively, the primal problem is:

$$
\begin{aligned}
\min \quad & \sum_j c_j \|x(S_j)\|_{q_j} + \sum_i z_i \\
\text{s.t.} \quad & z_i + \sum_{e \in P} x_e \ge 1, && \forall i,\ \forall P \in \mathcal{P}_i \\
& x, z \ge 0
\end{aligned}
$$

This is in the form of (P), so Theorem 1 can be applied. However, each request is associated with an exponential number of constraints and to obtain a polynomial-time algorithm we again need to apply a separation oracle. This is based on computing shortest paths in a modified graph: we add a vertex s_i' and edge (s_i', s_i) to graph G. Let z_i be the length of edge (s_i', s_i) and x_e be the length of each edge $e \in E$, and let H denote this edge-weighted graph.

When the i^{th} request (s_i, t_i) arrives
while *shortest* $s_i' - t_i$ *path in* H *has length less than* $\frac{1}{2}$ **do**
  Let $P \in \mathcal{P}_i$ be the path corresponding to the shortest $s_i' - t_i$ path in H;
  Run Algorithm 2 with request $z_i + \sum_{e \in P} x_e \geq 1$;
end

Algorithm 4: Online Algorithm for Throughput Maximization

The shortest path algorithm runs in polynomial time and it finds a constraint with $z_i + \sum_{e \in P} x_e < \frac{1}{2}$ (if any). To see that the number of iterations of Algorithm 4 is polynomial, define potential function $\psi = z_i + \sum_{e \in E} x_e$. We know that $\psi \leq m+1$ the number of edges in H since our algorithm is minimal, that is, each iteration terminates with $z_i + \sum_{e \in P} x_e = 1$. In each iteration, ψ increases by at least $\frac{1}{2}$. So the total number of iteration is at most $2m$. Finally, by doubling the variables z, x we have a feasible solution and the objective increases by factor two.

Using Algorithm 2, we obtain an $O(\log m)$-competitive online algorithm for the fractional relaxation (13)–(18). To get an integer solution, we use a simple randomized rounding algorithm. For each request i, choose a path $P \in \mathcal{P}_i$ with probability $\frac{f_{i,P}}{8}$, and choose no path with the remaining probability $1 - \frac{1}{8} \sum_{P \in \mathcal{P}_i} f_{i,P}$. For each request i and edge e, let $X_{i,e} = 1$ if the path chosen for request i contains edge e and $X_{i,e} = 0$ otherwise. Let $X_e = \sum_i X_{i,e}$ denote the load on each edge $e \in E$; note that X_e is the sum of independent $0 - 1$ random variables. Also, by the rounding algorithm and constraint (15), $\mathbb{E}(X_e) = \frac{\mu_e}{8}$ for each edge e. We use the following standard concentration inequality.

Theorem 5 (Chernoff Bound 1) *Let* $X = \sum_i X_i$ *where* $X_i = 1$ *with probability* p_i *and* $X_i = 0$ *with probability* $1 - p_i$, *and all* X_i *are independent. Then*

$$\Pr[X \geq (1+\epsilon)\mathbb{E}[X]] \leq e^{-\frac{\epsilon^2}{2+\epsilon}\mathbb{E}[X]} \quad \textit{for all } \epsilon > 0.$$

Let $\delta = 36 \log m$. Using this result on $X_e = \sum_i X_{i,e}$ with $\epsilon = 1 + 2\delta/\mu_e$,

$$\Pr\left[X_e > \frac{\mu_e}{4} + \frac{\delta}{4}\right] = \Pr\left[X_e > (1+\epsilon)\mathbb{E}[X_e]\right]$$
$$\leq e^{-\frac{(1+\frac{2\delta}{\mu_e})^2}{2+(1+\frac{2\delta}{\mu_e})}\frac{\mu_e}{8}} \leq e^{-\frac{\mu_e + 2\delta}{24}} \leq e^{-\frac{\delta}{12}} = \frac{1}{m^3}.$$

Then by union bound over all m edges, we obtain that $X_e \leq \frac{\mu_e}{4} + \frac{\delta}{4}$ for all $e \in E$, with probability at least $1 - 1/m^2$. Conditioned on this "good event" we have

$$\sum_{e\in S_j} X_e^{p_j} \le \sum_{e\in S_j}\left(\frac{\mu_e}{4}+\frac{\delta}{4}\right)^{p_j} \le \sum_{e\in S_j} 2^{p_j}\left(\frac{\mu_e^{p_j}}{4^{p_j}}+\frac{\delta^{p_j}}{4^{p_j}}\right)$$
$$= \frac{1}{2^{p_j}}(\|\mu(S_j)\|_{p_j}^{p_j} + |S_j|\delta^{p_j}) < c_j^{p_j},\ \forall j. \tag{19}$$

where the last inequality is by constraint (17) and $c_j = \Omega(\log m)\cdot |S_j|^{\frac{1}{p_j}}$.

In order to guarantee that we always find a feasible solution (i.e. satisfy all the capacities) we simply terminate the algorithm when *any* capacity constraint is about to be violated. Below ALG denotes the number of paths selected in the randomized rounding and $\overline{ALG}$ is the number of paths selected before the algorithm is terminated.

To prove the $O(\log m)$-competitive ratio, let OPT be the offline optimal value of the throughput maximization instance. We first assume $OPT = \Omega(\log m)$ and handle the case $OPT = O(\log m)$ later. Define the following random variables:

- For each request i, $A_i = 1$ if the rounding satisfies request i and $A_i = 0$ otherwise.
- $I = 0$ if the rounding satisfies all capacities and $I = 1$ otherwise.
- $ALG = \sum_i A_i$ the number of requests satisfied by the rounding.
- $\overline{ALG} = ALG$ if $I = 0$ and $\overline{ALG} = 0$ otherwise.
- $G = \min(ALG, OPT) - OPT \cdot I$.

Note that $\mathbb{E}[ALG] = \sum_i \mathbb{E}[A_i] = \sum_i \sum_{P\in\mathcal{P}_i} \frac{f_{i,P}}{8} \ge \frac{OPT}{O(\log m)}$ because our fractional online algorithm is $O(\log m)$-competitive. Moreover, assuming $OPT = \Omega(\log m)$ we have $\mathbb{E}[ALG] = \Omega(1)$.

By definition of G, we have $\overline{ALG} \ge G$ because:

1. if the rounded solution is feasible ($I = 0$) then $\overline{ALG} = ALG \ge \min(ALG, OPT)$, and
2. if the rounded solution is infeasible ($I = 1$) then $\overline{ALG} \ge 0 \ge G$.

Now we have

$$\mathbb{E}[\overline{ALG}] \ge \mathbb{E}[G] = \mathbb{E}[\min(ALG, OPT)] - OPT \cdot \mathbb{E}[I]$$
$$\ge \mathbb{E}[\min(ALG, OPT)] - \frac{OPT}{m^2}. \tag{20}$$

The last inequality is by (19) which implies $\mathbb{E}[I] \le \frac{1}{m^2}$.

We now use the Chernoff bound on the lower tail.

Theorem 6 (Chernoff Bound 2) *Let* $X = \sum_i X_i$ *where* $X_i = 1$ *with probability* p_i *and* $X_i = 0$ *with probability* $1 - p_i$*, and all* X_i *are independent. Then*

$$\Pr[X \le (1-\epsilon)\mathbb{E}[X]] \le e^{-\frac{\epsilon^2}{2}\mathbb{E}[X]} \quad \textit{for all } 0 < \epsilon < 1.$$

Recall $\mathbb{E}[ALG] = \Omega(1)$ (using our assumption on OPT) and $A_i \in \{0, 1\}$ for all i. By choosing $\epsilon = \frac{1}{2}$ in Theorem 6, with constant probability we have $ALG \ge \frac{1}{2}\mathbb{E}[ALG] \ge \frac{OPT}{O(\log m)}$. Finally, using $ALG \ge 0$ we have $\mathbb{E}[\min(ALG, OPT)] = \frac{OPT}{O(\log m)}$. Combined with (20) we obtain $\mathbb{E}[\overline{ALG}] = \frac{OPT}{O(\log m)}$.

We now handle the case $OPT = O(\log m)$. Note that in this case, just selecting a single path is an $O(\log m)$-competitive solution. The overall algorithm runs with probability half either the above rounding or the greedy choice of selecting any path for the first request. Note that selecting any path $P \in \mathcal{P}_i$ leads to a feasible solution because of our capacity assumption. Finally, the expected objective is $OPT/O(\log m)$, which completes the proof of Theorem 4.

5.2 An $(O(\log m), O(\log^{1+1/p} m))$ bicriteria algorithm

We now prove the second part of Theorem 3 (restated below).

Theorem 7 *There is a randomized $(O(\log m), O(\log^{1+1/p} m))$-bicriteria competitive online algorithm for throughput maximization with ℓ_p-norm capacities, where m is the maximum of the number of edges in the graph and the number of constraint subsets and $p = \min_j p_j$.*

For this result we further strengthen the dual continuous relaxation to

$$
\begin{aligned}
\max \quad & \sum_i \sum_{P \in \mathcal{Q}_i} f_{i,P} && && (21)\\
\text{s.t.} \quad & \sum_{P \in \mathcal{Q}_i} f_{i,P} \le 1, && \forall i && (22)\\
& \sum_i \sum_{P \in \mathcal{Q}_i} |S_j \cap P| f_{i,P} \le c_j^{p_j}, && \forall j && (23)\\
& \sum_i \sum_{P \in \mathcal{Q}_i : e \in P} f_{i,P} = \mu_e, && \forall e && (24)\\
& \|\mu(S_j)\|_{p_j} \le c_j, && \forall j && (25)\\
& f \ge 0. && && (26)
\end{aligned}
$$

Here $\mathcal{Q}_i$ is the set of simple paths P between s_i and t_i such that $|S_j \cap P| \le c_j^{p_j}$ for all groups j.

Lemma 5 *Convex program (21)–(26) is a relaxation of the throughput maximization problem.*

Proof We first observe that only paths in $\cup_i \mathcal{Q}_i$ may be used in any feasible solution for the throughput maximization problem. Suppose (for a contradiction) that some $s_i - t_i$ path $P \in \mathcal{P}_i \backslash \mathcal{Q}_i$ is used. Then it is clear that the load induced by path P alone violates some capacity c_j, which contradicts the feasibility. This justifies using only $f_{i,P}$ variables corresponding to $P \in \mathcal{Q}_i$.

Now, note that any feasible solution to the throughput maximization problem corresponds to a solution (f, μ) with each $f_{i,P} \in \{0, 1\}$ that satisfies constraints (22), (24) and (25). The objective value (number of accepted requests) is clearly (21).

We only need to show that the new constraints (23) are also satisfied. We know that the ℓ_{p_j} load on each edge subset S_j is at most c_j. That is,

$$\sum_{e \in S_j} \left(\sum_i \sum_{P \in \mathcal{Q}_i : e \in P} f_{i,P} \right)^{p_j} \leq c_j^{p_j}, \qquad \forall j.$$

Then, using the fact that each $f_{i,P} \in \{0, 1\}$ we have for any group j,

$$\sum_{e \in S_j} \left(\sum_i \sum_{P \in \mathcal{Q}_i : e \in P} f_{i,P} \right)^{p_j} \geq \sum_{e \in S_j} \sum_i \sum_{P \in \mathcal{Q}_i : e \in P} f_{i,P} = \sum_i \sum_{e \in S_j} \sum_{P \in \mathcal{Q}_i : e \in P} f_{i,P}$$
$$= \sum_i \sum_{P \in \mathcal{Q}_i} |S_j \cap P| f_{i,P}.$$

Hence we obtain $\sum_i \sum_{P \in \mathcal{Q}_i} |S_j \cap P| f_{i,P} \leq c_j^{p_j}$ for all j, as desired. □

Now, the relaxation (21)–(26) can be recast as

$$\max \quad \sum_i \sum_{P \in \mathcal{Q}_i} f_{i,P} \tag{27}$$
$$\text{s.t.} \quad \sum_{P \in \mathcal{Q}_i} f_{i,P} = \nu_i, \qquad \forall i \tag{28}$$
$$\sum_i \sum_{P \in \mathcal{Q}_i} |S_j \cap P| f_{i,P} = \lambda_j, \qquad \forall j \tag{29}$$
$$\sum_i \sum_{P \in \mathcal{Q}_i : e \in P} f_{i,P} = \mu_e, \qquad \forall e \tag{30}$$
$$\|\nu\|_\infty \leq 1, \tag{31}$$
$$\|\mu(S_j)\|_{p_j} \leq c_j, \qquad \forall j \tag{32}$$
$$\|\lambda_j\|_\infty \leq c_j^{p_j}, \qquad \forall j \tag{33}$$
$$f \geq 0. \tag{34}$$

This is exactly in the form of our dual program (D). If z_i, y_j and x_e are primal variables corresponding to (28), (29) and (30) respectively, the primal program is

$$\min \quad \sum_j c_j \|x(S_j)\|_{q_j} + \sum_j c_j^{p_j} y_j + \sum_i z_i$$
$$\text{s.t.} \quad z_i + \sum_j |S_j \cap P| y_j + \sum_{e \in P} x_e \geq 1, \qquad \forall i, \forall P \in \mathcal{Q}_i$$
$$x, y, z \geq 0.$$

We can apply Theorem 1 to this primal formulation to obtain an $O(\log m)$ -competitive online algorithm. However, there are an exponential number of constraints and in this case we are not aware of an efficient separation oracle. The separation problem corresponds to resource constrained shortest path [29] and to the best of our knowledge, no efficient (exact or approximation) algorithms are known. So the running time of our fractional online algorithm is exponential.

The online randomized rounding algorithm is the same as Sect. 5.1. We again use Chernoff bound to prove that with high probability the load on each edge e is small. Recall that for each request i and edge e, random variable $X_{i,e}$ the the indicator if the path chosen for request i contains edge e, and $X_e = \sum_i X_{i,e}$ is the load on edge e. We also use $Y_{i,P}$ as the indicator that path $P \in \mathcal{Q}_i$ is selected for request i. We analyze the following two cases separately.

Case 1: $\mu_e \geq 1$. Let $\delta = 36 \log m$. By Chernoff bound (Theorem 5), we have

$$\Pr\left[X_e > \frac{\mu_e}{4} + \frac{\delta}{4}\right] \leq e^{-\frac{(1+\frac{2\delta}{\mu_e})^2}{2+(1+\frac{2\delta}{\mu_e})}\frac{\mu}{8}} \leq e^{-\frac{\mu_e+2\delta}{24}} \leq e^{-\frac{\delta}{12}} = \frac{1}{m^3}.$$

So, with probability $1 - \frac{1}{m^3}$ we have $X_e \leq (b \log m)\mu_e$, where $b \leq 10$, which implies $X_e^{p_j} \leq (3 \log m)^{p_j} \mu_e^{p_j}$ for all groups j.

Case 2: $\mu_e \leq 1$. Let random variable $R_e = 1$ if some path using e is selected, and $R_e = 0$ otherwise. Note that $R_e \leq \sum_i \sum_{P \in \mathcal{Q}_i : e \in P} Y_{i,P}$. Again by Chernoff bound (Theorem 5),

$$\Pr[X_e > b \log m] \leq \frac{1}{m^3}.$$

Conditioned on $X_e \leq b \log m$, we have $X_e \leq b \log m \cdot R_e$ because $X_e = 0$ if and only if $R_e = 0$. Then

$$X_e^{p_j} \leq (b \log m)^{p_j} \cdot R_e \leq (b \log m)^{p_j} \sum_i \sum_{P \in \mathcal{Q}_i : e \in P} Y_{i,P}, \qquad \forall j.$$

Let $\mathcal{E}$ denote the event that $X_e \leq (b \log m) \cdot \max\{\mu_e, 1\}$ for all edges e. It follows from above that $\Pr[\mathcal{E}] \geq 1 - \frac{1}{m^2}$. Moreover, conditioned on $\mathcal{E}$, we have for each group j:

$$\begin{aligned}\sum_{e \in S_j} X_e^{p_j} &\leq \sum_{e \in S_j} (b \log m)^{p_j} \mu_e^{p_j} + \sum_{e \in S_j} (b \log m)^{p_j} \sum_i \sum_{P \in \mathcal{Q}_i : e \in P} Y_{i,P} \\ &= (b \log m)^{p_j} \sum_{e \in S_j} \mu_e^{p_j} + (b \log m)^{p_j} \sum_i \sum_{e \in S_j} \sum_{P \in \mathcal{Q}_i : e \in P} Y_{i,P}\end{aligned}$$

$$
\begin{aligned}
&= (b\log m)^{p_j}\sum_{e\in S_j}\mu_e^{p_j} + (b\log m)^{p_j}\sum_i\sum_{P\in\mathcal{Q}_i}|S_j\cap P|\cdot Y_{i,P}\\
&\le (b\log m)^{p_j}\cdot c_j^{p_j} + (b\log m)^{p_j}\sum_i\sum_{P\in\mathcal{Q}_i}|S_j\cap P|\cdot Y_{i,P}. \qquad (35)
\end{aligned}
$$

where the last inequality is by constraint (25).

By the constraints (23) and definition of $\mathcal{Q}_i$, we have:

$$
\mathbb{E}\left[\sum_i\sum_{P\in\mathcal{Q}_i}|S_j\cap P|\cdot Y_{i,P}\right]\le c_j^{p_j} \quad \text{and} \quad |S_j\cap P|\le c_j^{p_j} \text{ for all } P\in\cup_i\mathcal{Q}_i.
$$

Note that the random variables $\sum_{P\in\mathcal{Q}_i}|S_j\cap P|\cdot Y_{i,P}$ are independent across requests i and bounded between 0 and $c_j^{p_j}$. By Chernoff bound (Theorem 5) and union bound over groups j, it follows that with probability at least $1-\frac{1}{m^2}$,

$$
\sum_i\sum_{P\in\mathcal{Q}_i}|S_j\cap P|\cdot Y_{i,P}\le(4\log m)c_j^{p_j}, \qquad \forall j.
$$

Let $\mathcal{F}$ denote the above event. Combined with (35), we obtain that conditioned on $\mathcal{E}$ and $\mathcal{F}$,

$$
\begin{aligned}
\sum_{e\in S_j}X_e^{p_j} &\le (b\log m)^{p_j}\cdot c_j^{p_j} + (b\log m)^{p_j}\cdot(4\log m)c_j^{p_j}\\
&\le 2(b\log m)^{p_j+1}\cdot c_j^{p_j},\ \forall j.
\end{aligned}
$$

As $\Pr[\mathcal{E}\wedge\mathcal{F}]\ge 1-\frac{2}{m^2}$ and $b\le 10$, it follows that the capacities are violated by at most an $O(\log^{1+1/p} m)$ factor with high probability. Here $p=\min_j p_j$.

The proof of the $O(\log m)$ competitive ratio and ensuring that the capacity bounds hold with probability one, are identical to that in Sect. 5.1. This completes the proof of Theorem 7.

6 Conclusion

In this paper we obtained a nearly tight $O(\log d+\log\rho)$-competitive algorithm for fractional online covering problems with ℓ_q-norm objectives and its dual packing problem. We also demonstrated the applicability of this result in two settings: non-uniform buy-at-bulk network design and throughput maximization under ℓ_p-norm capacities. Identifying online algorithms for other classes of convex programs is an interesting direction. Another open question is to design online algorithms for more combinatorial optimization problems using convex program relaxations.

A Deriving $f_e^*(\cdot)$

Recall that $f_e(x) = c_e\|x(S_e)\|_{q_e}$ where $x \in \mathbb{R}_+^n$ and $S_e \subseteq [n]$. For any $\mu \in \mathbb{R}_+^n$ we have

$$
\begin{aligned}
f_e^*(\mu) &= \sup_{x\in\mathbb{R}_+^n} \mu^T x - c_e\|x(S_e)\|_{q_e} \\
&= \sup_{x\in\mathbb{R}_+^n} \left\{\mu(\overline{S}_e)^T x(\overline{S}_e) + \mu_e(S_e)^T x(S_e) - c_e\|x(S_e)\|_{q_e}\right\}.
\end{aligned}
$$

If $\mu(\overline{S}_e) \neq \mathbf{0}$ then it is clear that $f_e^*(\mu) = \infty$. So we assume $\mu(\overline{S}_e) = \mathbf{0}$, in which case

$$
f_e^*(\mu) = \sup_{y\in\mathbb{R}_+^{|S_e|}} \left\{\mu(S_e)^T y - c_e\|y\|_{q_e}\right\}.
$$

Let $\|\cdot\|_{p_e}$ be the dual norm of $\|\cdot\|_{q_e}$. By the definition of the dual norm, if $\|\mu(S_e)\|_{p_e} > c_e$, there exists $z \in \mathbb{R}^{|S_e|}$ with $\|z\|_{q_e} \le 1$ such that $\mu(S_e)^T z > c_e$. As $\mu \ge 0$, we can ensure $z \ge 0$. Then taking $y = tz$ as $t \to \infty$, we have

$$
f_e^*(\mu) \ge \mu(S_e)^T y - c_e\|y\|_{q_e} = t(\mu(S_e)^T z - c_e\|z\|_{q_e}) \to \infty.
$$

On the other hand, if $\|\mu(S_e)\|_{p_e} \le c_e$, then by Hölder's inequality, for any $y \in \mathbb{R}_+^{|S_e|}$,

$$
\mu(S_e)^T y \le \|\mu(S_e)\|_{p_e}\|y\|_{q_e} \le c_e\|y\|_{p_e},
$$

which implies that $f_e^*(\mu) = 0$.

Summarizing the above cases, we have for any $\mu \in \mathbb{R}_+^n$:

$$
f_e^*(\mu) = \begin{cases} 0, & \text{if } \|\mu(S_e)\|_{p_e} \le c_e \text{ and } \mu(\overline{S}_e) = \mathbf{0}, \\ \infty, & \text{otherwise.} \end{cases}
$$

B Limitations of previous approaches in handling ℓ_q-norm objectives.

The general convex covering problem is

$$
\min\left\{f(x) : Ax \ge \mathbf{1},\ x \ge \mathbf{0}\right\},
$$

where $f : \mathbb{R}_+^n \to \mathbb{R}_+$ is a convex function and $A \in \mathbb{R}_+^{m\times n}$. Its dual is:

$$
\max\left\{\sum_{k=1}^m y_k - f^*(\mu) : A^T y = \mu,\ y \ge \mathbf{0}\right\},
$$

where $f^*(\mu) = \max_{x \in \mathbb{R}^n_+} \{\mu^T x - f(x)\}$ is the Fenchel conjugate of f. When f is the sum of ℓ_q-norms, these primal-dual convex programs reduce to (P) and (D).

We restrict the discussion of prior techniques to functions f with $\max_{x \in \mathbb{R}^n_+} \frac{x^T \nabla f(x)}{f(x)} \leq 1$ because this condition is satisfied by sums of ℓ_q norms.[1] At a high level, the analysis in [6] uses the gradient monotonicity to prove a *pointwise upper bound* $A^T y \leq \nabla f(\bar{x})$ where $\bar{x}$ is the final primal solution. This allows them to lower bound the dual objective by $\sum_{k=1}^m y_k$ because $f^*(\nabla f(\bar{x})) \leq 0$ for any $\bar{x}$ (see Lemma 4(d) in [6]). Moreover, proving the pointwise upper bound $A^T y \leq \nabla f(\bar{x})$ is similar to the task of showing dual feasibility in the *linear* case [15,26] where $\nabla f(\bar{x})$ corresponds to the (fixed) primal cost coefficients.

Below we give a simple example with an ℓ_q-norm objective where the pointwise upper bound $A^T y \leq \nabla f(\bar{x})$ is not satisfied by the online primal-dual algorithm unless the dual solution y is scaled down by a large (i.e. polynomial) factor. This means that one cannot obtain a sub-polynomial competitive ratio for (P) using this approach directly.

Consider an instance with objective function $f(x) = \|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$. So the gradient $\nabla f(x) = x/\|x\|_2$ which is not monotone. There are $m = \sqrt{n}$ covering constraints, where the k^{th} constraint is $\sum_{i=m(k-1)+1}^{km} x_i \geq 1$. Note that each variable appears in only one constraint. Let P be the value of the primal objective and D be the value of the dual objective at any time. Suppose that the rate of increase of the primal objective is at most α times that of the dual; α corresponds to the competitive ratio in the online primal-dual algorithm. Upon arrival of any constraint k, it follows from the primal updates that all the variables $\{x_i\}_{i=m(k-1)+1}^{km}$ increase from 0 to $\frac{1}{m}$. So the increase in P due to constraint k is $(\sqrt{k} - \sqrt{k-1})\frac{1}{\sqrt{m}}$ for iteration k. This means that the increase in D is at least $\frac{1}{\alpha}(\sqrt{k} - \sqrt{k-1})\frac{1}{\sqrt{m}}$, and so $y_k \geq \frac{1}{\alpha}(\sqrt{k} - \sqrt{k-1})\frac{1}{\sqrt{m}}$. Finally, since $\bar{x} = \frac{1}{m}\mathbf{1}$, we know that $\nabla f(\bar{x}) = \frac{1}{m}\mathbf{1}$ (recall $n = m^2$). On the other hand, $(A^T y)_1 = y_1 \geq \frac{1}{\alpha\sqrt{m}}$. Therefore, in order to guarantee $A^T y \leq \nabla f(\bar{x})$ we must have $\alpha \geq \sqrt{m} = n^{1/4}$.

References

1. Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., Naor, J.: A general approach to online network optimization problems. ACM Trans. Algorithms **2**(4), 640–660 (2006)
2. Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., Naor, J.: The online set cover problem. SIAM J. Comput. **39**(2), 361–370 (2009)
3. Awerbuch, B., Azar, Y., Grove, E.F., Kao, M., Krishnan, P., Vitter, J.S.: Load balancing in the l_p norm. In: FOCS, pp. 383–391 (1995)
4. Awerbuch, B., Azar, Y., Plotkin, S.: Throughput-competitive on-line routing. In: FOCS, pp. 32–40. IEEE (1993)
5. Azar, Y., Bhaskar, U., Fleischer, L.K., Panigrahi, D.: Online mixed packing and covering. In: SODA (2013)

[1] The result in [6] also applies to other convex functions with monotone gradients, but the competitive ratio depends exponentially on $\max_{x \in \mathbb{R}^n_+} \frac{x^T \nabla f(x)}{f(x)}$.

6. Azar, Y., Buchbinder, N., Chan, T.H., Chen, S., Cohen, I.R., Gupta, A., Huang, Z., Kang, N., Nagarajan, V., Naor, J., Panigrahi, D.: Online algorithms for covering and packing problems with convex objectives. In: FOCS, pp. 148–157 (2016)
7. Azar, Y., Cohen, I.R., Panigrahi, D.: Online covering with convex objectives and applications. CoRR abs/1412.3507 (2014)
8. Bansal, N., Buchbinder, N., Naor, J.: Randomized competitive algorithms for generalized caching. SIAM J. Comput. **41**(2), 391–414 (2012)
9. Bansal, N., Pruhs, K.: Server scheduling to balance priorities, fairness, and average quality of service. SIAM J. Comput. **39**(7), 3311–3335 (2010)
10. Bartal, Y., Fiat, A., Leonardi, S.: Lower bounds for on-line graph problems with application to on-line circuit and optical routing. SIAM J. Comput. **36**(2), 354–393 (2006)
11. Blum, A., Gupta, A., Mansour, Y., Sharma, A.: Welfare and profit maximization with production costs. In: FOCS, pp. 77–86 (2011)
12. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
13. Buchbinder, N., Chen, S., Gupta, A., Nagarajan, V., Naor, J.: Online packing and covering framework with convex objectives. CoRR abs/1412.8347 (2014)
14. Buchbinder, N., Jain, K., Naor, J.: Online primal-dual algorithms for maximizing ad-auctions revenue. In: ESA, pp. 253–264 (2007)
15. Buchbinder, N., Naor, J.: Online primal-dual algorithms for covering and packing. Math. Oper. Res. **34**(2), 270–286 (2009)
16. Buchbinder, N., Naor, J.S.: The design of competitive online algorithms via a primal-dual approach. Found. Trends Theor. Comput. Sci. **3**(2–3), 93–263 (2007)
17. Chan, T.H., Huang, Z., Kang, N.: Online convex covering and packing problems. CoRR abs/1502.01802 (2015)
18. Charikar, M., Karagiozova, A.: On non-uniform multicommodity buy-at-bulk network design. In: STOC, pp. 176–182. ACM (2005)
19. Chekuri, C., Hajiaghayi, M.T., Kortsarz, G., Salavatipour, M.R.: Approximation algorithms for nonuniform buy-at-bulk network design. SIAM J. Comput. **39**(5), 1772–1798 (2010)
20. Chekuri, C., Khanna, S., Naor, J.: A deterministic algorithm for the cost-distance problem. In: Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7–9, 2001, Washington, DC, USA, pp. 232–233 (2001)
21. Devanur, N.R., Huang, Z.: Primal dual gives almost optimal energy efficient online algorithms. In: SODA, pp. 1123–1140 (2014)
22. Devanur, N.R., Jain, K.: Online matching with concave returns. In: STOC, pp. 137–144. ACM (2012)
23. Elad, N., Kale, S., Naor, J.S.: Online semidefinite programming. In: ICALP, pp. 40:1–40:13 (2016)
24. Ene, A., Chakrabarty, D., Krishnaswamy, R., Panigrahi, D.: Online buy-at-bulk network design. In: FOCS, pp. 545–562 (2015)
25. Gupta, A., Krishnaswamy, R., Pruhs, K.: Online primal-dual for non-linear optimization with applications to speed scaling. In: WAOA, pp. 173–186 (2012)
26. Gupta, A., Nagarajan, V.: Approximating sparse covering integer programs online. Math. Oper. Res. **39**(4), 998–1011 (2014)
27. Helvig, C.S., Robins, G., Zelikovsky, A.: An improved approximation scheme for the group steiner problem. Networks **37**(1), 8–20 (2001)
28. Huang, Z., Kim, A.: Welfare maximization with production costs: a primal dual approach. In: SODA, pp. 59–72 (2015)
29. Mehlhorn, K., Ziegelmann, M.: Resource constrained shortest paths. In: Algorithms—ESA 2000, 8th Annual European Symposium, Saarbrücken, Germany, September 5-8, 2000, Proceedings, pp. 326–337 (2000)
30. Menache, I., Singh, M.: Online caching with convex costs: Extended abstract. In: SPAA, pp. 46–54 (2015)
31. Meyerson, A.: Online algorithms for network design. In: SPAA 2004: Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, June 27–30, 2004, Barcelona, Spain, pp. 275–280 (2004)
32. Moreau, J.J.: Fonctionnelles convexes. Séminaire Jean Leray **2**, 1–108 (1967)
33. Nagarajan, V., Shen, X.: Online covering with sum of ℓ_q-norm objectives. In: 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10–14, 2017, Warsaw, Poland, pp. 12:1–12:12 (2017)

34. Strömberg, T.: The operation of infimal convolution. Instytut Matematyczny Polskiej Akademi Nauk (1996). http://eudml.org/doc/271746