# Online Generalized Network Design Under (Dis)Economies of Scale

Viswanath Nagarajan* Lily Wang*

**Abstract**

We consider a general online network design problem where a sequence of $N$ *requests* arrive over time, each of which needs to use a subset of the available *resources* $E$. The cost incurred by a resource $e \in E$ is some function $f_e$ of its total load $\ell_e$. The objective is to minimize the total cost $\sum_{e \in E} f_e(\ell_e)$. We focus on cost functions that exhibit (dis)economies of scale, which are of the form $f_e(x) = \sigma_e + \xi_e \cdot x^{\alpha_e}$ if $x > 0$ (and zero if $x = 0$), where the exponent $\alpha_e \geq 1$.

Our main result is a deterministic online algorithm with tight competitive ratio $\Theta(1 + \max_{e \in E} (\sigma_e/\xi_e)^{1/\alpha_e})$ when $\alpha_e$ is constant. This framework is applicable to many network design problems, including multicommodity routing, Steiner tree/forest connectivity and set-connectivity. Even in special cases such as multicommodity routing in undirected graphs with edge-costs, this is the first online algorithm to handle non-uniform resource cost and with a competitive ratio independent of the network size and number of requests. Our online competitive ratio also matches the previous-best offline approximation ratio. Our approach is based on the online primal-dual method for convex programs.

## 1 Introduction

Network design problems (involving selecting a subgraph with certain connectivity properties) are of significant practical and theoretical interest. A classic setting in network design is as follows. There are several requests that need to be routed through a network, where each resource $e$ has a non-decreasing cost-function $f_e$ that determines the cost $f_e(\ell_e)$ incurred at $e$ as a function of its load $\ell_e$. The objective is to minimize the overall cost $\sum_e f_e(\ell_e)$.

Traditional network design models involve *concave* cost-functions. These are cost functions that exhibit "economies of scale", i.e., a larger load results in a smaller cost-per-unit-load. This is the setting in buy-at-bulk network design, that has been studied extensively in approximation and online algorithms [8, 19, 16]. The most basic problems in this setting are Steiner tree and forest [1, 27].

Recent applications in energy-efficient scheduling and routing have motivated the study of cost-functions with "diseconomies of scale" [5, 32]. Here, larger load results in a larger cost-per-unit-load. These functions capture the energy consumption of network resources that are *speed scalable* and adjust their speed in proportion to their load. The energy consumed at speed/load $x$ grows super-linearly as $x^\alpha$ where the exponent $\alpha > 1$. For most technologies, exponent $\alpha$ lies between 1 and 3 [5, 36].

As discussed in [5], a more accurate model for energy consumption involves a start-up cost in addition to the super-linear $x^\alpha$ term. This leads to the cost function:

$$f_e(x) = \begin{cases} 0 & \text{if } x = 0 \\ \sigma_e + \xi_e \cdot x^{\alpha_e} & \text{if } x > 0 \end{cases}, \tag{1.1}$$

where the parameters $\sigma_e, \xi_e \geq 0$ and $\alpha_e \geq 1$ depend on the particular device (resource). The first term $\sigma_e$ represents the cost incurred in simply keeping the device powered-on but idle and the second term $\xi_e \cdot x^{\alpha_e}$ represents the cost incurred due to speed-scaling. These cost functions exhibit *both* economies and diseconomies of scale. Indeed, they appear concave for small values of the load $x$ and convex for large values of the load. So these functions are said to exhibit *(dis)economies of scale*. A major challenge in designing algorithms for such cost-functions is that one needs the balance two opposing goals (1) aggregating demands in the concave regime and (2) separating demands in the convex regime. Prior work [6, 7, 31] has mainly focused on the special case of *uniform* (or related) cost functions where the $\alpha_e$s and $\frac{\sigma_e}{\xi_e}$s are uniform across all resources $e$.

Recently, [22] studied a large class of *generalized network design* problems under cost functions of the form (1.1), which included routing requests, Steiner tree/forest connectivity and set-connectivity in undirected and directed graphs. The main result in [22] was a unified approximation framework that provided an $O\left(1 + \max_e \left(\frac{\sigma_e}{\xi_e}\right)^{1/\alpha_e}\right)$ approximation algorithm assuming only a "minimum cost oracle" that can satisfy a *single* request at minimum cost.

*Industrial and Operations Engineering Department, University of Michigan. Email: {viswa,lilyxy}@umich.edu. Supported in part by NSF grants CCF-1750127 and CMMI-1940766.

In this paper, we consider the same class of generalized network design (GND) problems as [22], but in the *online setting*. Here, requests arrive over time and each request needs to be (irrevocably) assigned to some resources immediately upon arrival. Our main result is a deterministic online algorithm with competitive ratio $O\left(1 + \max_e \left(\frac{\sigma_e}{\xi_e}\right)^{1/\alpha_e}\right)$, which even matches the best approximation ratio known for GND. We also show that no deterministic online algorithm can do better (up to a constant factor).

**1.1 Problem Definition** In the generalized network design (GND) problem, we have a set $E$ of resources and $N$ requests that use these resources. Each request $i \in [N]$ is associated with:

- a collection $\mathcal{P}_i \subseteq 2^E$ of "replies" where an algorithm needs to choose some $p_i \in \mathcal{P}_i$ in order to satisfy request $i$. The reply collections may be specified implicitly.
- a weight vector $w_i \in R^E_{\geq 1}$ where request $i$ induces a load of $w_{i,e}$ on each resource $e$ that it uses. Note that the weights on different resources may be unrelated. (The requirement that weights/demands of requests are at least one is common to all prior work.)

Each resource $e \in E$ is associated with an individual cost function $f_e : R \to R$ of the form (1.1). We will refer to such functions as *(D)oS functions*. We emphasize that the parameters $\sigma_e$, $\xi_e$ and $\alpha_e$ may be different across resources. So we can handle networks with heterogenous resources (for example, routers running on different technologies).

A solution is just a choice of reply $p_i \in \mathcal{P}_i$ for each request $i \in [N]$. Then, the load on each resource $e \in E$ is $\ell_e = \sum_{i:e \in p_i} w_{i,e}$. The objective is to minimize the total cost $\sum_{e \in E} f_e(\ell_e)$.

In the online setting, the requests $i \in [N]$ arrive over time, and the algorithm should choose a reply $p_i \in \mathcal{P}_i$ for each request $i$ immediately upon arrival (which cannot be changed later). As usual, we use competitive analysis to measure the performance of an online algorithm, which is relative to the offline optimum that knows the entire request sequence upfront.

We use $m := |E|$ to denote the number of resources. For each resource $e \in E$, define $q_e := (\sigma_e/\xi_e)^{1/\alpha_e}$. Note that $q_e$ is the value of load $x$ at which the two terms $\sigma_e$ and $\xi_e \cdot x^{\alpha_e}$ in the (D)oS cost function $f_e(x)$ become equal. Let $q := \max_{e \in E} q_e$. Also, let $\alpha := \max_{e \in E} \alpha_e$ denote the maximum exponent in the (D)oS functions.

**Min-cost Oracle** We will assume that the reply-collections $\mathcal{P}_i$ are such that one can find an approximately min-cost reply efficiently. Formally, we assume that there is a $\tau$-approximation algorithm for the problem $\min_{p \in \mathcal{P}_i} \sum_{e \in p} d_e$ for any request $i \in [N]$ and any scalars $\{d_e \geq 0\}_{e \in E}$. If computational complexity is not a consideration (which is sometimes the case with online algorithms) then this assumption is satisfied trivially with $\tau = 1$.

**Example 1 (multicommodity routing).** The resources $E$ are edges in some directed graph $G = (V, E)$. Each request $i \in [N]$ consists of a source $s_i \in V$, destination $t_i \in V$ and demand $d_i \geq 1$. For each $i \in [N]$, the reply-collection $\mathcal{P}_i$ consists of all $s_i - t_i$ paths in $G$, and the weights $w_{i,e} = d_i$ for all $e \in E$. The resulting GND instance corresponds to selecting an $s_i - t_i$ routing path carrying $d_i$ units of flow (for each request $i$), so as to minimize the total energy cost of the routing. The min-cost oracle in this case corresponds to the shortest path problem in directed graphs, which admits an exact algorithm: so $\tau = 1$.

**Example 2 (set connectivity and set-strong-connectivity).** The resources $E$ are edges in some undirected (resp. directed) graph $G = (V, E)$. Each request $i \in [N]$ consists of a subset $T_i \subseteq V$ of nodes and demand $d_i \geq 1$. The reply-collection $\mathcal{P}_i$ consists of all edge-subsets that induce a connected (resp. strongly connected) subgraph containing $T_i$. The weights $w_{i,e} = d_i$ for all $e \in E$. The resulting GND instance corresponds to selecting an overlay network for each terminal-set $T_i$ that can support $d_i$ units of flow. The min-cost oracle for the undirected case corresponds to the Steiner tree problem: so we have $\tau = 1.39$ [14]. In the directed case, the oracle is the strongly connected Steiner subgraph problem, for which we have (i) $\tau = k^\epsilon$ for any constant $\epsilon > 0$ in polynomial time [17] or (ii) $\tau = O(\frac{\log^2 k}{\log \log k})$ in *quasi-polynomial* time [28, 26]. Here $k = \max_i |T_i|$ is the maximum number of terminals in any request.

**1.2 Our Results and Techniques** Our main result is the following:

THEOREM 1.1. *There is a polynomial time $O(q\tau + (\mathsf{e}\alpha\tau)^\alpha)$-competitive deterministic online algorithm for* GND *assuming a $\tau$-approximation algorithm for the min-cost oracle.*

Above, $\mathsf{e} \approx 2.718$ is the base of the natural logarithm. The running time of this algorithm is $O(Nm + N \cdot \Phi(m))$ where $\Phi(m)$ is the time taken by the min-cost oracle. Note that when $\tau = 1$, we obtain a competitive ratio of $O(q + (\mathsf{e}\alpha)^\alpha)$.

To the best of our knowledge, previous online algorithms for GND were restricted to the case of multicommodity routing in undirected graphs with uniform

edge-cost functions [7]. Our result provides a unified framework to address various types of requests (including Steiner and set-connectivity) in both undirected and directed graphs. Moreover, this is the first competitive ratio (even in the previously-studied setting [7]) that does not grow with the network size or the number of requests. Finally, our result also applies to *non-uniform* cost functions: in this setting, no online algorithm was known even for single-commodity routing with edge costs.

As noted earlier, our competitive ratio matches the $O_\alpha(q\tau + \tau^\alpha)$ approximation algorithm for GND obtained in [22].[1] Even when used in the offline setting, our algorithm has several advantages. First, the dependence on $\alpha$ in the approximation ratio is better: we obtain a factor of $(\mathsf{e}\alpha)^\alpha = \mathsf{e}^{\alpha(1+\ln\alpha)}$ whereas the previous algorithm had a $3^{\alpha^2}$ factor [23]. Second, our algorithm is deterministic whereas the previous algorithm was randomized. Third, our running time is better. Fourth, our algorithm itself is very simple and (arguably) simpler to analyze.

To prove Theorem 1.1, we first show that any (D)oS function $f_e(x)$ of the form (1.1) can be well-approximated by a weighted sum of power functions of form $h_e(x) = \eta_e \cdot x + \xi_e \cdot x^{\alpha_e}$. This reduction loses a factor of $2(\sigma_e/\xi_e)^{1/\alpha_e}$ in the objective. This allows us to then focus on the GND problem under (non-uniform) power cost functions, which is a convex objective.

For GND under power cost functions, if we were only interested in an offline approximation algorithm, we could use the approach in [32] that was based on a convex relaxation and rounding to obtain an $A_\alpha$-approximation algorithm for GND (assuming $\tau = 1$). Here, $A_\alpha \approx (\frac{\alpha}{\ln(1+\alpha)})^\alpha$ is the fractional Bell number. This approach however does not work in the online setting. Instead, we use a more direct approach motivated by work on online load balancing with $\ell_p$-norms [9]. For each request $i$, our algorithm basically selects the reply in $\mathcal{P}_i$ that results in the smallest increase in the objective. (The actual algorithm involves tracking a modified objective function.) We analyze our algorithm using the online primal-dual method for convex programs. The idea is to (1) write a convex relaxation for GND and its dual, and (2) upper bound the (integral) primal objective by some factor $\rho$ times the dual objective. By weak duality, we then obtain a competitive ratio of $\rho$.

There have been a number of recent papers using the online primal-dual approach for convex programs (see §1.3 for more details). The work closest to ours is [29], where an $O(\alpha)^\alpha$-competitive algorithm was obtained for the special case of GND with uniform $\alpha$ power cost functions and multicommodity routing requests. Our approach is more general as it can handle a much wider class of requests and non-uniform $\alpha_e$ powers. From a technical perspective, while our primal convex program is the natural extension of that in [29] (for multicommodity routing), we use a different (re)formulation of the dual program and also set dual variables differently. Our dual formulation is easier to reason about, and hence allows for a clean analysis even in more general settings.

Implementing the above approach directly leads to an $O(q(\mathsf{e}\alpha\tau)^\alpha)$-competitive algorithm for GND using a $\tau$-approximate min-cost oracle. To obtain the more refined guarantee in Theorem 1.1, we improve both steps above. In the reduction from (D)oS functions $f_e(x)$ to power functions $h_e(x)$, we show that the factor $q_e$ loss only affects the linear term in $h_e(x)$. Then, in the online algorithm for GND under power functions, we show that the greedy objective can be further modified to ensure a stronger $O(\tau)$ competitive ratio for the linear terms, while the non-linear terms incur an $O((\mathsf{e}\alpha\tau)^\alpha)$ competitive ratio.

We also provide a nearly matching lower bound for online GND:

THEOREM 1.2. *Every deterministic online algorithm for* GND *has competitive ratio* $\Omega\left(q + (1.44\alpha)^\alpha\right)$.

As usual with online lower bounds, this is information-theoretic and independent of computational requirements. So this nearly matches the $O(q + (\mathsf{e}\alpha)^\alpha)$ competitive ratio from Theorem 1.1 when $\tau = 1$. The lower bound instance involves single-commodity routing requests in directed graphs. The $\Omega(q)$ part of the lower bound relies on a construction similar to the online directed Steiner tree lower bound [24]. The $\Omega((1.44\alpha)^\alpha)$ part of the lower bound follows from the corresponding result for online load balancing with $\alpha^{th}$ power of loads [15].

Finally, we can also extend our main result to a larger class of functions called *real exponent polynomials* (REP) that were studied in [22]. These have the form

$$\bar{f}_e(x) = \begin{cases} 0 & \text{if } x = 0 \\ \sigma_e + \sum_{j=1}^{q} \xi_{e,j} \cdot x^{\alpha_{e,j}} & \text{if } x > 0 \end{cases}, \tag{1.2}$$

where the parameters $\sigma_e, \xi_{e,1}, \cdots \xi_{e,q} \geq 0$ and the exponents $\alpha_{e,j} \geq 1$.

THEOREM 1.3. *There is a polynomial time $O(Q\tau + (\mathsf{e}\alpha\tau)^\alpha)$-competitive deterministic online algorithm for* GND *under REP cost functions assuming a $\tau$-approximation algorithm for the min-cost oracle. Here $Q = \max_{e\in E} \min_{j\in[q]} (\sigma_e/\xi_{e,j})^{1/\alpha_{e,j}}$.*

[1] The $O_\alpha$ notation treats $\alpha$ as constant and suppresses factors that depend on $\alpha$.

The idea here is to reduce any GND problem with REP costs into another instance with (D)oS cost functions of form (1.1) but with more resources.

**1.3 Related Work** Most of the prior work in network design under (D)oS cost functions has focused on multicommodity routing requests with uniform weights (i.e., $w_{i,e} = d_i$ for all resources $e$ and requests $i$). [5] were the first to study this model and obtained an $O(q \cdot \log^{\alpha-1} D)$-approximation algorithm where $D = \max_{i=1}^{N} d_i$ is the maximum weight. When $\sigma_e = 0$ for all resources $e$ (in which case the objective is a weighted sum of power functions), [32] obtained an improved $A_\alpha$-approximation algorithm. These results apply to undirected as well as directed graphs.

Further results are known for multicommodity routing in *undirected graphs* in the special case of *uniform* cost functions, where $f_e(x) = c_e \cdot f(x)$ for a common (D)oS function $f(x)$. When costs are incurred on edges, [6] obtained a poly-logarithmic $O(\log^{O(\alpha)} N)$-approximation algorithm, and [7] later improved the approximation ratio to $O(\log^{\alpha} N)$. When costs are incurred on nodes (which is harder than the edge-version), [31] obtained an $O(\log^{O(\alpha)} N)$-approximation algorithm. All these results rely crucially on the uniformity of the cost function. In particular, they use the fact that it is best to aggregate $q = (\sigma/\xi)^{1/\alpha}$ units of demand, after which the aggregated demands can be routed in a "well separated" manner. It is unclear if these techniques can be used for non-uniform costs as the "aggregate demand" quantity for different resources is different (it is $q_e = (\sigma_e/\xi_e)^{1/\alpha_e}$ for each resource $e$). Furthermore, these results relied on cut-sparsification and small flow-cut gaps, which do not extend to directed graphs. In fact, the directed Steiner forest problem (which is a special case of GND) is hard to approximate better than $\Omega(2^{\log^{1-\epsilon} N})$ for any constant $\epsilon > 0$ [21]. We note that the parameter $q \approx N$ for GND instances corresponding to Steiner forest: so we cannot expect an approximation ratio much better than $poly(q)$ for GND. In fact, any $o(\sqrt{q})$-approximation algorithm for GND would improve on the best approximation ratio known for directed Steiner forest [18, 25].

As mentioned earlier, [22] considered the much wider class of GND problems, and obtained an $O(q)$-approximation algorithm. As discussed in [22], their result extends prior work involving (D)oS cost functions in several ways: unrelated weights, non-uniform cost functions, strongly polynomial runtime etc. Our result inherits all these advantages even in the online setting. The technique in [22] was based on the "smoothness" toolbox from [35]. Our approach (discussed above) is completely different, and leads to a much simpler algorithm.

In the online setting, [7] obtained an $\tilde{O}(\log^{3\alpha+1} N)$-competitive randomized algorithm for multicommodity routing in undirected graphs with uniform cost functions on edges and uniform weights. This ratio is incomparable to the $O(q + (\mathsf{e}\alpha)^{\alpha})$ deterministic online ratio that we obtain (even in more general settings). When $\sigma_e = 0$ for all resources $e$ and all $\alpha_e$ are uniform, $O(\alpha)^{\alpha}$-competitive online algorithms were known for load balancing [9] and multicommodity routing [29]. Our algorithm can be seen as a natural extension of these results to the setting of GND. [9] used a potential-function analysis that appears hard to extend to non-uniform $\alpha_e$s. As discussed in §1.2, though our approach as well as [29] are based on the online primal-dual method, there are important differences as well.

The online primal-dual method (see the survey [13]) is a very general technique that has led to several strong results in online algorithms. Typically, this approach is applied with covering/packing linear-program relaxations, e.g. [3, 2, 12]. However, a number of recent papers, e.g. [29, 4, 20, 10, 33, 30], have extended this to the setting of covering programs with convex objectives. Our result adds to this line of work. Although our fractional relaxation is a "convex covering program" as studied in [10], we cannot use the general-purpose algorithm presented there because the number of variables in our relaxation for GND is exponential: the competitive ratio in [10] is logarithmic in the number of variables. We note however that our idea of setting dual variables based on the gradient of the primal objective (at the final solution) was partly motivated from [10].

**1.4 Paper Outline** We start with the reduction from (D)oS cost functions to weighted power functions in §2. In §3 we provide a fractional online algorithm for the natural convex relaxation of GND under power cost functions. Then, in §4 we extend this to an integral online algorithm. §5 puts things together and finishes the proofs of Theorems 1.1 and 1.3. Finally, §6 provides the online lower bounds (Theorem 1.2).

## 2 Reducing (D)oS Functions to Weighted Power Functions

We first make the simple but useful observation that any cost-function $f_e$ of the form (1.1) can be approximated by a *convex power function*, at the loss of a multiplicative factor $2q_e$, where $q_e := (\sigma_e/\xi_e)^{1/\alpha_e}$. To this end, define for each $e \in E$, a new function

$$h_e(x) := \xi_e q_e^{\alpha_e - 1} \cdot x + \xi_e \cdot x^{\alpha_e}, \quad \text{for all } x \geq 0. \tag{2.3}$$

LEMMA 2.1. *For each $e \in E$ and $x \in \{0\} \cup R_{\geq 1}$, we*

*have*

$$\frac{1}{2}\cdot h_e(x) \le f_e(x) \le \max\{q_e,1\}\cdot \xi_e q_e^{\alpha_e-1}\cdot x + \xi_e\cdot x^{\alpha_e} \le \max\{q_e,1\}\cdot h_e(x).$$

*Proof.* At $x=0$ the inequalities trivially hold. So we assume $x \ge 1$ in the rest of the proof. For the first inequality, we divide it into two cases. If $x < q_e$, then

$$h_e(x) = \xi_e q_e^{\alpha_e-1}\cdot x + \xi_e\cdot x^{\alpha_e} \le \xi_e q_e^{\alpha_e} + \xi_e\cdot x^{\alpha_e} = \sigma_e + \xi_e\cdot x^{\alpha_e} = f_e(x)$$

If $x \ge q_e$, then

$$h_e(x) = \xi_e q_e^{\alpha_e-1}\cdot x + \xi_e\cdot x^{\alpha_e} \le 2\xi_e x^{\alpha_e} \le 2(\xi_e x^{\alpha_e}+\sigma_e) = 2f_e(x)$$

For the second inequality, we have

$$\max\{1,q_e\}\cdot \xi_e q_e^{\alpha_e-1}\cdot x + \xi_e\cdot x^{\alpha_e} \ge \xi_e q_e^{\alpha_e}\cdot x + \xi_e\cdot x^{\alpha_e} = \sigma_e x + \xi_e\cdot x^{\alpha_e} \ge \sigma_e + \xi_e\cdot x^{\alpha_e} = f_e(x),$$

where the second inequality uses $x \ge 1$. ∎

Recall that $q := \max_{e\in E} q_e$. By Lemma 2.1, at the loss of factor $2\max\{q,1\}$, it essentially suffices to solve the GND problem under power cost functions, where each resource $e \in E$ has a cost function of the form $g_e(x) = c_e\cdot x^{\alpha_e}$. To be precise, there is also a linear term in the cost-function for each $e$: see §5 for details. In the next two sections, we provide online algorithms for GND under *weighted power functions*.

## 3 Fractional Online Algorithm

We consider the following convex program relaxation for GND, denoted $(P)$.

$$\min \quad \sum_{e\in E} c_e\cdot\left(\sum_{i=1}^{N} w_{i,e}\sum_{p\in\mathcal{P}_i: e\in p} x_{i,p}\right)^{\alpha_e}$$

$$\text{s.t.} \quad \sum_{p\in\mathcal{P}_i} x_{i,p} \ge 1, \qquad \forall i\in[N] \tag{3.4}$$

$$\mathbf{x}\ge\mathbf{0}.$$

Note that all constraints are of "covering type" and the objective is convex. However, there are an exponential number of variables as the replies $\mathcal{P}_i$ are implicitly specified. We will solve this program approximately using the online primal-dual method. We provide a continuous time online algorithm, that is easier to describe and analyze (Theorem 3.1). In [34], we explain how to obtain a polynomial time implementation at a small loss in the competitive ratio.

Let $E_1 = \{e\in E : \alpha_e = 1\}$. The dual of convex program $(P)$ is below, denoted $(D)$.

$$\max \quad \sum_{i=1}^{N} y_i - \sum_{e\in E\setminus E_1}\frac{c_e\alpha_e}{\beta_e}\cdot z_e^{\beta_e}$$

$$\text{s.t.} \quad \sum_{e\in p} w_{i,e}c_e\alpha_e\cdot z_e \ge y_i, \qquad \forall p\in\mathcal{P}_i,\ \forall i\in[N] \tag{3.5}$$

$$z_e \le 1, \qquad \forall e\in E_1 \tag{3.6}$$

$$\mathbf{y},\mathbf{z}\ge\mathbf{0}.$$

Above, for each $e\in E\setminus E_1$, value $\beta_e > 1$ is the conjugate of $\alpha_e$, i.e. $\frac{1}{\alpha_e}+\frac{1}{\beta_e}=1$. Note that there are no terms in the dual objective corresponding to $e\in E_1$. We derive this dual in the full version [34]. It turns out that strong duality holds for this primal-dual pair. However, we will only use weak duality, which is proved below.

LEMMA 3.1. *For any primal $x\in(P)$ and dual $(y,z)\in(D)$ solutions,*

$$\sum_{e\in E} c_e\cdot\left(\sum_{i=1}^{N} w_{i,e}\sum_{p\in\mathcal{P}_i:e\in p} x_{i,p}\right)^{\alpha_e} \ge \sum_{i=1}^{N} y_i - \sum_{e\in E\setminus E_1}\frac{c_e\alpha_e}{\beta_e}\cdot z_e^{\beta_e}.$$

*Proof.* For easier notation, let $\ell_e := \sum_i\sum_{p\in\mathcal{P}_i:e\in p} w_{i,e}\cdot x_{i,p}$ be the fractional load on each $e\in E$. For each $e\in E_1$, let $\beta_e=\infty$: note that $\frac{1}{\beta_e}z_e^{\beta_e}=0$ as $z_e\le 1$. We will show that

$$\sum_i y_i \le \sum_{e\in E}\alpha_e c_e\cdot\left(\frac{1}{\alpha_e}\cdot\ell_e^{\alpha_e}+\frac{1}{\beta_e}\cdot z_e^{\beta_e}\right),$$

which would prove the lemma. Indeed, we have:

$$\sum_i y_i \le \sum_i\left(\sum_{p\in\mathcal{P}_i} x_{i,p}\right)\cdot y_i \tag{3.7}$$

$$\le \sum_i\sum_{p\in\mathcal{P}_i} x_{i,p}\cdot\left(\sum_{e\in p} w_{i,e}c_e\alpha_e\cdot z_e\right) \tag{3.8}$$

$$= \sum_e c_e\alpha_e\cdot z_e\left(\sum_i\sum_{p\in\mathcal{P}_i:e\in p} w_{i,e}\cdot x_{i,p}\right) \tag{3.9}$$

$$= \sum_e c_e\alpha_e\cdot z_e\cdot\ell_e \tag{3.10}$$

$$\le \sum_{e\in E_1} c_e\cdot\ell_e + \sum_{e\in E\setminus E_1} c_e\alpha_e\cdot z_e\cdot\ell_e \tag{3.11}$$

$$\le \sum_e c_e\alpha_e\left(\frac{1}{\alpha_e}\cdot\ell_e^{\alpha_e}+\frac{1}{\beta_e}\cdot z_e^{\beta_e}\right). \tag{3.12}$$

Above, the inequality in (3.7) is by constraint (3.4) and non-negativity, and the inequality in (3.8) is by constraint (3.5). The equality in (3.9) is by interchanging summation. The inequality in (3.11) is by constraint (3.6) and the last inequality in (3.12) is by Young's inequality, which says $A \cdot B \leq \frac{1}{\alpha} \cdot A^\alpha + \frac{1}{\beta} \cdot B^\beta$ for any $A, B \geq 0$ and $\alpha, \beta > 1$ with $\frac{1}{\alpha} + \frac{1}{\beta} = 1$. This completes the proof. ∎

**Fractional online algorithm for** $(P)$ Upon arrival of request $i$, for each continuous time $t \in [0,1]$, do the following:

1. Choose reply $p^* \in \mathcal{P}_i$ using the min-cost oracle under costs $d_e = \alpha_e c_e \cdot \ell_e^{\alpha_e - 1} \cdot w_{i,e}$ for each $e \in E$, where $\ell_e = \sum_i \sum_{p \in \mathcal{P}_i : e \in p} w_{i,e} \cdot x_{i,p}$ is the current fractional load on $e$.

2. Raise primal variable $x_{i,p^*}$ at rate one, i.e. $\frac{\partial}{\partial t} x_{i,p^*} = 1$.

THEOREM 3.1. *The fractional online algorithm has competitive ratio at most $\alpha^\alpha$ where $\alpha = \max_{e \in E} \alpha_e$.*

*Proof.* The proof is by dual fitting: we will provide a feasible dual solution $(y, z)$ and show that the online primal solution $\bar{x}$ has objective at most $\alpha^\alpha$ times the dual objective. Combined with Lemma 3.1, this would imply the theorem.

Let $\bar{\ell}_e = \sum_i \sum_{p \in \mathcal{P}_i : e \in p} w_{i,e} \cdot \bar{x}_{i,p}$ be the final load on each $e \in E$. Let $\delta \in (0,1]$ be some parameter, and define the dual solution:

$$z_e = \delta \cdot \bar{\ell}_e^{\alpha_e - 1}, \qquad \forall e \in E.$$

$$y_i = \min_{p \in \mathcal{P}_i} \sum_{e \in p} w_{i,e} c_e \alpha_e \cdot z_e, \qquad \forall i \in [N].$$

Note that dual-constraint (3.6) is satisfied as $z_e = \delta \leq 1$ for all $e \in E_1$. Moreover, (3.5) is satisfied by definition of $y$. So $(y, z)$ is a feasible dual solution. For each request $i$, let $q_i \in \mathcal{P}_i$ denote the reply that achieves the minimum cost in the definition of $y_i$ above.

We now relate the primal objective $\bar{P} = \sum_e c_e \cdot \bar{\ell}_e^{\alpha_e}$ with the dual objective $D$, by showing:

$$D \;\geq\; \left(\delta - (\alpha - 1) \cdot \delta^{\frac{\alpha}{\alpha - 1}}\right) \cdot \bar{P} \tag{3.13}$$

Consider the algorithm when some request $i$ arrives. For each time $t \in [0,1]$, if $p^* \in \mathcal{P}_i$ is the current reply and $\{\ell_e\}_{e \in E}$ denotes the current loads, then by the primal update:

$$\frac{\partial}{\partial t} \bar{P} = \sum_{e \in p^*} c_e \alpha_e \left( \sum_{i=1}^N w_{i,e} \sum_{p \in \mathcal{P}_i : e \in p} x_{i,p} \right)^{\alpha_e - 1}$$

$$\text{with } w_{i,e} = \sum_{e \in p^*} w_{i,e} c_e \alpha_e \cdot \ell_e^{\alpha_e - 1} \leq \sum_{e \in q_i} w_{i,e} c_e \alpha_e \cdot \ell_e^{\alpha_e - 1}$$

$$\leq \sum_{e \in q_i} w_{i,e} c_e \alpha_e \cdot \bar{\ell}_e^{\alpha_e - 1} = \frac{1}{\delta} \cdot y_i.$$

Above, the first inequality is by the choice of the current reply $p^*$ at time $t$, the second inequality is by monotonicity of the primal solution $x$ over time, and the last equality is by the choice of the dual value $y_i$. It follows that the increase in $\bar{P}$ due to request $i$ is at most $\frac{y_i}{\delta}$. Adding over all $i$,

$$\bar{P} \leq \frac{1}{\delta} \sum_{i=1}^N y_i.$$

Now, consider the contribution of the $z$-variables to the dual objective:

$$\sum_{e \in E \setminus E_1} \frac{c_e \alpha_e}{\beta_e} \cdot z_e^{\beta_e} = \sum_{e \in E \setminus E_1} \delta^{\beta_e} \frac{c_e \alpha_e}{\beta_e} \left(\bar{\ell}_e^{\alpha_e - 1}\right)^{\beta_e}$$

$$= \sum_{e \in E \setminus E_1} \delta^{\beta_e} c_e (\alpha_e - 1) \bar{\ell}_e^{\alpha_e}$$

$$\leq \delta^{\frac{\alpha}{\alpha - 1}} (\alpha - 1) \sum_{e \in E \setminus E_1} c_e \bar{\ell}_e^{\alpha_e}.$$

The equalities use the fact that $\frac{1}{\beta_e} = 1 - \frac{1}{\alpha_e}$. The inequality above uses that $\delta \leq 1$ and $\beta_e = 1 + \frac{1}{\alpha_e - 1} \geq 1 + \frac{1}{\alpha - 1}$ for all $e$. Finally, the right-hand-side above is at most $\delta^{\frac{\alpha}{\alpha - 1}} (\alpha - 1) \cdot \bar{P}$. Therefore, the dual objective is:

$$D = \sum_{i=1}^N y_i - \sum_{e \in E \setminus E_1} \frac{c_e \alpha_e}{\beta_e} \cdot z_e^{\beta_e} \geq \delta \cdot \bar{P} - \delta^{\frac{\alpha}{\alpha - 1}} (\alpha - 1) \cdot \bar{P},$$

which proves (3.13). Finally, choosing $\delta = 1/\alpha^{\alpha - 1}$, we obtain $\bar{P} \leq \alpha^\alpha \cdot D$. ∎

In the full version [34], we show how this algorithm can be implemented in polynomial time (with a small loss in the competitive ratio).

## 4 Integer Online Algorithm

We now provide an integral online algorithm for GND. It is well-known (see e.g. [5]) that the convex relaxation $(P)$ used in §3 has a polynomially large integrality gap even for single-commodity routing on undirected

graphs. To get around this, we use an idea from [11] for load balancing, by adding additional *linear terms* corresponding to the $\alpha_e^{th}$ power of loads from individual requests. Let $\rho \geq 1$ be a parameter to be set later. Upon the arrival of request $i$, we do the following:

- Choose reply $p_i \in \mathcal{P}_i$ using the min-cost oracle under the costs
(4.14)
$$\psi_e = \alpha_e c_e \cdot \ell_e^{\alpha_e - 1} \cdot w_{i,e} + \frac{\rho}{\mathsf{e}^\alpha} \cdot c_e \alpha_e w_{i,e}^{\alpha_e}, \text{ for each } e \in E,$$
where $\ell_e := \sum_{j<i: e \in p_j} w_{j,e}$ is the current load on $e$.

THEOREM 4.1. *The online* GND *algorithm has competitive ratio at most* $2(\mathsf{e}\alpha)^\alpha$ *where* $\alpha = \max_{e \in E} \alpha_e$.

We prove this result in the rest of this section. Let $A_e$ denote the final load on each resource $e \in E$. The online algorithm's objective is then $A := \sum_e c_e \cdot A_e^{\alpha_e}$.

We will use a different (stronger) convex relaxation for GND and relate $A$ to the new relaxation. The new relaxation has the same constraints in $(P)$ but the objective is now:

$$\sum_{e \in E} c_e \cdot \left( \sum_{i=1}^N w_{i,e} \sum_{p \in \mathcal{P}_i : e \in p} x_{i,p} \right)^{\alpha_e} + \sum_{e \in E} \frac{c_e \alpha_e}{\mathsf{e}^\alpha} \cdot \sum_{i=1}^N w_{i,e}^{\alpha_e} \sum_{p \in \mathcal{P}_i : e \in p} x_{i,p} \tag{4.15}$$

LEMMA 4.1. *The optimal value of the new convex program with objective* (4.15) *is at most* $(1 + \alpha \mathsf{e}^{-\alpha}) \cdot \mathsf{OPT}$, *where* OPT *is the optimal value of the (integral)* GND *instance.*

*Proof.* Consider an optimal solution to GND with objective OPT. We set a corresponding solution for $(P)$ by setting $x_{i,p}$ to 1 if $p$ is the reply used to satisfy request $i$ and 0 otherwise. Using the fact that each $x_{i,p}$ is either 0 or 1, we have for each $e$,

$$\sum_{i=1}^N w_{i,e}^{\alpha_e} \sum_{p \in \mathcal{P}_i : e \in p} x_{i,p} \leq \left( \sum_{i=1}^N w_{i,e} \sum_{p \in \mathcal{P}_i : e \in p} x_{i,p} \right)^{\alpha_e}$$

So, the objective of the new relaxation is at most

$$(1 + \frac{\alpha}{\mathsf{e}^\alpha}) \sum_{e \in E} c_e \Big( \sum_{i=1}^N w_{i,e} \sum_{\substack{p \in \mathcal{P}_i : \\ e \in p}} x_{i,p} \Big)^{\alpha_e} = (1 + \frac{\alpha}{\mathsf{e}^\alpha}) \mathsf{OPT},$$

which proves the lemma. ∎

To make notation simpler, for the analysis we imagine adding dummy resources $E' = \{e' : e \in E\}$ corresponding to the second term in the new objective. We set $\alpha_{e'} := 1$, $c_{e'} := 1$ and $w_{i,e'} := \frac{c_e \alpha_e}{\mathsf{e}^\alpha} w_{i,e}^{\alpha_e}$ for all $i \in [N]$ and $e \in E$. Moreover, we extend each reply $p \in \mathcal{P}_i$ so that it contains *both* copies $e, e'$ of each resource $e \in p$. The new reply collections are referred to as $\{\mathcal{P}'_i\}_{i=1}^N$. The dual of the new convex program, denoted $(D')$, is given below.

$$\max \quad \sum_{i=1}^N y_i - \sum_{e \in E \setminus E_1} \frac{c_e \alpha_e}{\beta_e} \cdot z_e^{\beta_e} \tag{4.16}$$

$$\text{s.t.} \quad \sum_{e \in p} w_{i,e} c_e \alpha_e \cdot z_e \geq y_i, \qquad \forall p \in \mathcal{P}'_i, \forall i \in [N]$$

$$z_e \leq 1, \qquad \forall e \in E_1 \cup E' \tag{4.17}$$

$$\mathbf{y}, \mathbf{z} \geq \mathbf{0}.$$

Above, $E_1 = \{e \in E : \alpha_e = 1\}$. Note that all the dummy resources $E'$ have the exponent $\alpha_e = 1$: so they do not appear in the second term of the dual objective.

Define the dual solution:

$$z_e := \frac{1}{\rho} \cdot A_e^{\alpha_e - 1}, \qquad \forall e \in E.$$

$$z_{e'} := 1, \qquad \forall e' \in E'.$$

$$y_i := \min_{p' \in \mathcal{P}'_i} \sum_{e \in p'} w_{i,e} c_e \alpha_e \cdot z_e$$

$$= \min_{p \in \mathcal{P}_i} \sum_{e \in p} \Big( w_{i,e} c_e \alpha_e \cdot z_e + \frac{c_e \alpha_e}{\mathsf{e}^\alpha} w_{i,e}^{\alpha_e} \Big), \quad \forall i \in [N].$$

The second equality above (for $y_i$) follows from the definitions of the new reply-collection $\mathcal{P}'_i$ and weights $w_{i,e'}$, and the setting $z_{e'} = 1$ for $e' \in E'$. Note that dual-constraint (4.17) is satisfied as $z_e = \delta \leq 1$ for all $e \in E_1$ and $z_{e'} = 1$ for all $e' \in E'$. Moreover, (4.16) is satisfied by definition of $y$. So $(y, z)$ is a feasible dual solution. For each request $i$, let $q_i \in \mathcal{P}_i$ denote the reply that achieves the minimum cost in the definition of $y_i$ above. We now relate $A$ with the dual objective $D$.

Consider the algorithm when some request $i$ arrives. Let $\ell_e$ denote the load on each $e \in E$ before request $i$ is assigned. Recall that $p_i \in \mathcal{P}_i$ is the selected reply.

Then, the increase in the algorithm's objective, $(\Delta A)_i$

$$= \sum_{e \in p_i} c_e \left( (\ell_e + w_{i,e})^{\alpha_e} - \ell_e^{\alpha_e} \right)$$

$$\leq \sum_{e \in p_i} c_e \alpha_e (\ell_e + w_{i,e})^{\alpha_e - 1} w_{i,e} \tag{4.18}$$

$$\leq \sum_{e \in p_i} c_e \alpha_e w_{i,e} \left( \mathsf{e} \cdot \ell_e^{\alpha_e - 1} + \alpha_e^{\alpha_e - 1} \cdot w_{i,e}^{\alpha_e - 1} \right) \tag{4.19}$$

$$= \mathsf{e} \cdot \sum_{e \in p_i} \left( c_e \alpha_e w_{i,e} \ell_e^{\alpha_e - 1} + \frac{1}{\mathsf{e}} c_e \alpha_e^{\alpha_e} w_{i,e}^{\alpha_e} \right)$$

$$\leq \mathsf{e} \cdot \sum_{e \in p_i} \left( c_e \alpha_e w_{i,e} \ell_e^{\alpha_e - 1} + \frac{\rho}{\mathsf{e}^\alpha} \cdot c_e \alpha_e w_{i,e}^{\alpha_e} \right) \tag{4.20}$$

$$\leq \mathsf{e} \cdot \sum_{e \in q_i} \left( c_e \alpha_e w_{i,e} \ell_e^{\alpha_e - 1} + \frac{\rho}{\mathsf{e}^\alpha} \cdot c_e \alpha_e w_{i,e}^{\alpha_e} \right) \tag{4.21}$$

$$= \mathsf{e}\rho \cdot \sum_{e \in q_i} \left( \frac{1}{\rho} \cdot c_e \alpha_e w_{i,e} \ell_e^{\alpha_e - 1} + \frac{c_e \alpha_e}{\mathsf{e}^\alpha} w_{i,e}^{\alpha_e} \right)$$

$$\leq \mathsf{e}\rho \cdot \sum_{e \in q_i} \left( \frac{1}{\rho} \cdot c_e \alpha_e w_{i,e} A_e^{\alpha_e - 1} + \frac{c_e \alpha_e}{\mathsf{e}^\alpha} w_{i,e}^{\alpha_e} \right) \tag{4.22}$$

$$= \mathsf{e}\rho \cdot \sum_{e \in q_i} \left( w_{i,e} c_e \alpha_e \cdot z_e + \frac{c_e \alpha_e}{\mathsf{e}^\alpha} w_{i,e}^{\alpha_e} \right) \tag{4.23}$$

$$= \mathsf{e}\rho \cdot y_i. \tag{4.24}$$

The inequality in (4.18) uses convexity of the $x^{\alpha_e}$ function. The inequality in (4.19) uses the inequality $(X+Y)^{\alpha-1} \leq \mathsf{e} \cdot X^{\alpha-1} + \alpha^{\alpha-1} \cdot Y^{\alpha-1}$ for $\alpha \geq 1$ and $X, Y \geq 0$, which follows from Lemma 4.1 in [9] (by setting $c = \mathsf{e}$). The inequality in (4.20) uses $\rho \geq (\mathsf{e}\alpha)^{\alpha-1}$ which we will ensure. The inequality in (4.21) uses the choice of $p_i$ under the costs (4.14). The inequality in (4.22) uses the fact that loads are monotonically non-decreasing. The equalities in (4.23) and (4.24) use the definition of reply $q_i$ and choice of dual variables $y_i$ and $z_e$. Adding over all $i$,

$$A \leq \mathsf{e}\rho \cdot \sum_{i=1}^{N} y_i.$$

Now, consider the contribution of the $z$-variables to the dual objective:

$$\begin{aligned} \sum_{e \in E \setminus E_1} \frac{c_e \alpha_e}{\beta_e} \cdot z_e^{\beta_e} &= \sum_{e \in E \setminus E_1} \rho^{-\beta_e} \frac{c_e \alpha_e}{\beta_e} \left( A_e^{\alpha_e - 1} \right)^{\beta_e} \\ &= \sum_{e \in E \setminus E_1} \rho^{-\beta_e} c_e (\alpha_e - 1) A_e^{\alpha_e} \\ &\leq \rho^{-\frac{\alpha}{\alpha-1}} (\alpha - 1) \sum_{e \in E \setminus E_1} c_e A_e^{\alpha_e} \end{aligned}$$

which follows the same way as for the fractional online algorithm. Therefore, the dual objective is:

$$\begin{aligned} D &= \sum_{i=1}^{N} y_i - \sum_{e \in E \setminus E_1} \frac{c_e \alpha_e}{\beta_e} \cdot z_e^{\beta_e} \\ &\geq \left( \frac{1}{\mathsf{e}\rho} - \rho^{-\frac{\alpha}{\alpha-1}} (\alpha - 1) \right) \cdot A. \end{aligned}$$

Finally, choosing $\rho = (\mathsf{e}\alpha)^{\alpha-1}$, we obtain $A \leq (\mathsf{e}\alpha)^\alpha \cdot D$. Combined with the observation that $D \leq (1 + \alpha \mathsf{e}^{-\alpha})\mathsf{OPT}$ (by Lemma 4.1), we obtain Theorem 4.1.

### 4.1 Using Approximate Min-Cost Replies

Here we consider the situation where an exact min-cost reply cannot be computed efficiently. This is indeed the case in some applications. We extend our online algorithm so that it also works with approximately min-cost replies. Moreover, we obtain a stronger guarantee for the linear terms in the objective, which will be used in proving our main result (see §5). Recall that $E_1 \subseteq E$ denotes the resources with exponent $\alpha_e = 1$. We obtain the following result, which is proved in the full version [34].

THEOREM 4.2. *Assume that there is a $\tau$-approximation algorithm for the min-cost oracle in* GND. *Then, there is a polynomial time $2(\mathsf{e}\alpha\tau)^\alpha$-competitive online algorithm for* GND. *In fact, if $L$ and $H$ denote the costs incurred by the algorithm on resources in $E_1$ and $E \setminus E_1$ respectively, then $L \leq 2\tau \cdot \mathsf{OPT}$ and $H \leq 2(\mathsf{e}\alpha\tau)^\alpha \cdot \mathsf{OPT}$.*

## 5 Application to GND with (D)oS Costs

We now complete the proof of our main result (Theorem 1.1). Given an instance $\mathcal{I}$ of GND with (D)oS costs as in (1.1), we use Lemma 2.1 to define a new instance $\mathcal{J}$ of GND with power cost functions, as follows. For each original resource $e \in E$, we have two copies $e_1$ and $e_a$. Let $E_1 := \{e_1 : e \in E\}$ and $E_a := \{e_a : e \in E\}$: so the resources in $\mathcal{J}$ are $E' = E_1 \cup E_a$. Define scalars $c_{e_1} := \xi_e q_e^{\alpha_e - 1}$ and $c_{e_a} := \xi_e$ for all $e \in E$. Also, define exponents $\alpha_{e_1} := 1$ and $\alpha_{e_a} := \alpha_e$ for all $e \in E$. The weighted power functions in instance $\mathcal{J}$ are $g_r(x) := c_r \cdot x^{\alpha_r}$ for all resources $r \in E'$. The reply-collections are extended so that for each reply $p \in \mathcal{P}_i$ in $\mathcal{I}$, there is a corresponding reply in $\mathcal{J}$ that contains *both* copies of resources $e \in p$. For each $e \in E$, note that function $h_e(x)$ used in Lemma 2.1 is $h_e(x) = g_{e_1}(x) + g_{e_a}(x)$. Using the first inequality in Lemma 2.1, the optimal value of instance $\mathcal{J}$ is $\mathsf{OPT}_\mathcal{J} \leq 2 \cdot \mathsf{OPT}_\mathcal{I}$. Now, using Theorem 4.2 on instance $\mathcal{J}$, we obtain:

$$L = \sum_{r \in E_1} c_r A_r \leq 2\tau \cdot \mathsf{OPT}_\mathcal{J} \text{ and}$$

$$H = \sum_{r \in E_a} c_r A_r^{\alpha_r} \leq 2(\mathsf{e}\tau\alpha)^\alpha \cdot \mathsf{OPT}_\mathcal{J},$$

where $\{A_r\}_{r\in E'}$ denote the final loads in the algorithm. For each $e \in E$, note that $A_{e_1} = A_{e_a}$; we use $A_e$ to denote this load. As every weight is at least one, we have each $A_e \in \{0\} \cup R_{\geq 1}$. The objective value in the original instance $\mathcal{I}$ is

$$\sum_{e\in E} f_e(A_e) \leq \sum_{e\in E} \left(\max\{q_e, 1\} \cdot \xi_e q_e^{\alpha_e - 1} \cdot A_e + \xi_e \cdot A_e^{\alpha_e}\right) \tag{5.25}$$

$$= \sum_{e\in E} \left(\max\{q_e, 1\} \cdot c_{e_1} \cdot A_e + c_{e_a} \cdot A_e^{\alpha_e}\right) \tag{5.26}$$

$$\leq \max\{q, 1\} \cdot L + H \tag{5.27}$$

$$\leq 2\left(\max\{q, 1\}\tau + (\mathsf{e}\tau\alpha)^\alpha\right) \cdot \mathsf{OPT}_{\mathcal{J}} \tag{5.28}$$

$$\leq 4\left(\max\{q, 1\}\tau + (\mathsf{e}\tau\alpha)^\alpha\right) \cdot \mathsf{OPT}_{\mathcal{I}}. \tag{5.29}$$

Inequality (5.25) is by Lemma 2.1 (2nd inequality) and $A_e \in \{0\} \cup R_{\geq 1}$. The equality in (5.26) is by definition of the scalars $c_r$ and the inequality in (5.27) is by definition of $L$ and $H$. In (5.28), the inequality is by the above bounds on $L$ and $H$, and the inequality in (5.29) uses $\mathsf{OPT}_{\mathcal{J}} \leq 2 \cdot \mathsf{OPT}_{\mathcal{I}}$.

**Remark:** The requirement that every weight is at least one is crucial in obtaining our result. As noted earlier, this requirement also appears in all prior work, e.g. [5, 6, 7, 32, 22]. In fact, any $r(q)$ competitive ratio for GND under arbitrary weights (possibly less than one) leads to an $O(1)$-competitive online algorithm, which is not possible even for the simplest setting of single-commodity flow in edge-weighted undirected graphs. To see this, consider a new instance of GND with weights $w'_{i,e} = w_{i,e}/q$ and parameters $\sigma'_e = \sigma_e/q^\alpha$ and $\xi'_e = \xi_e$. Note that the new GND instance is equivalent to the old one (the objective value of each solution is scaled down by $q^\alpha$). Moreover, the new value $q' = 1$, which means that we have an $r(q') = O(1)$ competitive algorithm.

**REP cost functions** We now consider the GND problem under more general costs of the form (1.2) and prove Theorem 1.3. The main idea is to replace each resource $e \in E$ with $q$ *copies* $e_1, \cdots e_q$ each with (D)oS cost function of the usual form (1.1). Then, we will directly apply Theorem 1.1.

For each $e \in E$, (by renumbering if needed) let

$$\left(\frac{\sigma_e}{\xi_{e,1}}\right)^{1/\alpha_{e,1}} = \min_{j=1}^{q} \left(\frac{\sigma_e}{\xi_{e,j}}\right)^{1/\alpha_{e,j}}.$$

The new GND instance has resources $\bar{E} := \{e_j : j \in [q], e \in E\}$. For each $e \in E$, set $\alpha_{e_j} := \alpha_{e,j}$, $\xi_{e_j} := \xi_{e,j}$, and

$$\sigma_{e_j} := \begin{cases} \sigma_e & \text{if } j = 1 \\ 0 & \text{if } j = 2, \cdots q \end{cases}$$

Let $f_{e_j}(x)$ denote the (D)oS cost function for each $e_j \in \bar{E}$. Clearly, $\bar{f}_e(x) = \sum_{j=1}^q f_{e_j}(x)$ for all $x \geq 0$ and $e \in E$. Moreover,

$$q := \max_{f\in\bar{E}} \left(\frac{\sigma_f}{\xi_f}\right)^{1/\alpha_f} = \max_{e\in E} \min_{j\in[q]} \left(\frac{\sigma_e}{\xi_{e,j}}\right)^{1/\alpha_{e,j}} = Q.$$

Recall the definition of $Q$ in Theorem 1.3. For each request $i$, the new reply-collection is

$$\bar{\mathcal{P}}_i := \{\cup_{e\in p}\{e_1, \cdots e_q\} \,:\, p \in \mathcal{P}_i\},$$

i.e., each new reply corresponds to selecting all copies of the resources in some original reply $p$. Assuming a $\tau$-approximation algorithm for the min-cost oracle under $\mathcal{P}_i$, it is easy to obtain a $\tau$-approximation algorithm for the new min-cost oracle under $\bar{\mathcal{P}}_i$. Indeed, given costs $d : \bar{E} \to R_+$ we define costs $d' : E \to R_+$ as $d'_e := \sum_{j=1}^q d_{e_j}$ for each $e \in E$ and apply the oracle for $\mathcal{P}_i$.

Using Theorem 1.1 on the new GND instance, we obtain an $O(q\tau + (\mathsf{e}\alpha\tau)^\alpha) = O(Q\tau + (\mathsf{e}\alpha\tau)^\alpha)$ competitive online algorithm under REP cost functions. This proves Theorem 1.3. The runtime of this algorithm is $O(Nmq + N\Phi(mq))$ where $\Phi(\cdot)$ denotes the time taken by the min-cost oracle.

## 6 Lower Bounds

We now show that our competitive ratio is tight up to a constant factor and prove Theorem 1.2.

We consider the single commodity routing problem (SSR) in directed graphs, which is a special case of GND. We are given a directed graph $(V, E)$ with weight $c_e \geq 0$ associated with each edge $e \in E$. There is a common source $s \in V$ and each online request $i$ corresponds to routing unit flow from $s$ to a sink node $t_i \in V$. The edge cost function of each edge is $f_e(x) = c_e \cdot f(x)$ where

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ \sigma + x^\alpha & \text{if } x > 0 \end{cases}.$$

Note that $q = \sigma^{1/\alpha}$. The min-cost reply oracle corresponds to shortest path: so we also have a polynomial time exact oracle in this case. We provide two different instances of SSR that show lower bounds of (i) $\Omega(q)$ for every choice of $\alpha \geq 1$ and $\sigma \geq 0$, and (ii) $\Omega((1.44\alpha)^\alpha)$ even when $q = 0$.

The $\Omega((1.44\alpha)^\alpha)$ lower bound follows from the restricted-assignment scheduling problem with $\ell_p$-norm of loads [15]. Recall, in that problem there are $m$ machines and $N$ jobs arrive over time. Each job $i$ specifies a subset $M_i$ of machines and needs to be assigned to one of them. The objective is to minimize the sum of $p^{th}$ powers of the machine loads. This corresponds to the directed graph on nodes $\{s\} \cup \{u_e\}_{e=1}^m \cup \{v_i\}_{i=1}^N$ where $s$ is the source, $u$-nodes correspond to machines and $v$-nodes correspond to jobs. There is an edge from $s$ to

each $u_e$ with weight 1. For each job $i \in [N]$ and machine $e \in M_i$, there is an edge from $u_e$ to $v_i$ of weight zero. We also set $\alpha = p$ and $\sigma = 0$. The resulting SSR instance is clearly equivalent to the scheduling problem.

The $\Omega(q)$ lower bound uses a construction similar to the lower bound for online directed Steiner tree [24]. Fix any value of $\sigma > 0$ and $\alpha \geq 1$ (which also fixes $q$). We will show an $\Omega(q)$ lower bound for SSR instances with this value of $\sigma$ and $\alpha$. The graph $G$ consists of a complete binary tree $B$ of depth $q$ rooted at node $t$ with all edges directed towards $t$, and source $s$ with edges to all nodes of the tree. Let $S$ denote all the edges out of $s$. All edges of the binary tree have weight zero and all edges in $S$ have weight one. The input sequence consists of $q$ requests as follows. At any point in the algorithm, let $A$ denote all edges that carry flow at least one: so the current cost is at least $|A \cap S| \cdot \sigma$. The first sink $t_1 = t$. For $i = 2, \cdots q$, sink $t_i$ is chosen to be the child of $t_{i-1}$ in $B$ such that $A$ does not contain an $s - t_i$ path. It is clear that $|A \cap S| \geq q$ at the end of this request sequence. So the online cost is at least $q\sigma$. Note that the sinks $t_1, \cdots t_q$ lie on a single directed path in the tree $B$: so an offline solution can just select the edges $(s, t_q)$ followed by $(t_i, t_{i-1})$ for $i = q, \cdots 2$. The cost of this solution is at most $\sigma + q^\alpha = 2\sigma$ as it uses only one edge in $S$ (which carries flow of $q$). Thus, the competitive ratio is at least $q/2$.

## References


[1] Ajit Agrawal, Philip N. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

[2] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. A general approach to online network optimization problems. *ACM Trans. Algorithms*, 2(4):640–660, 2006.

[3] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. The online set cover problem. *SIAM J. Comput.*, 39(2):361–370, 2009.

[4] S. Anand, Naveen Garg, and Amit Kumar. Resource augmentation for weighted flow-time explained by dual fitting. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1228–1241, 2012.

[5] Matthew Andrews, Antonio Fernández Anta, Lisa Zhang, and Wenbo Zhao. Routing for power minimization in the speed scaling model. *IEEE/ACM Trans. Netw.*, 20(1):285–294, 2012.

[6] Matthew Andrews, Spyridon Antonakopoulos, and Lisa Zhang. Minimum-cost network design with (dis)economies of scale. *SIAM J. Comput.*, 45(1):49–66, 2016.

[7] Antonios Antoniadis, Sungjin Im, Ravishankar Krishnaswamy, Benjamin Moseley, Viswanath Nagarajan, Kirk Pruhs, and Clifford Stein. Hallucination helps: Energy efficient virtual circuit routing. *SIAM J. Comput.*, 49(1):37–66, 2020.

[8] Baruch Awerbuch and Yossi Azar. Buy-at-bulk network design. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 542–547, 1997.

[9] Baruch Awerbuch, Yossi Azar, Edward F. Grove, Ming-Yang Kao, P. Krishnan, and Jeffrey Scott Vitter. Load balancing in the $l_p$ norm. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 383–391, 1995.

[10] Yossi Azar, Niv Buchbinder, T.-H. Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph Naor, and Debmalya Panigrahi. Online algorithms for covering and packing problems with convex objectives. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, New Brunswick, New Jersey, USA*, pages 148–157, 2016.

[11] Yossi Azar and Amir Epstein. Convex programming for scheduling unrelated parallel machines. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 331–337, 2005.

[12] Nikhil Bansal, Niv Buchbinder, and Joseph Naor. A primal-dual randomized algorithm for weighted paging. *J. ACM*, 59(4):19:1–19:24, 2012.

[13] 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.

[14] Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *J. ACM*, 60(1):6:1–6:33, 2013.

[15] Ioannis Caragiannis. Better bounds for online load balancing on unrelated machines. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 972–981, 2008.

[16] Deeparnab Chakrabarty, Alina Ene, Ravishankar Krishnaswamy, and Debmalya Panigrahi. Online buy-at-bulk network design. *SIAM J. Comput.*, 47(4):1505–1528, 2018.

[17] Moses Charikar, Chandra Chekuri, To-Yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed steiner problems. *J. Algorithms*, 33(1):73–91, 1999.

[18] Chandra Chekuri, Guy Even, Anupam Gupta, and Danny Segev. Set connectivity problems in undirected graphs and the directed steiner network problem. *ACM Trans. Algorithms*, 7(2):18:1–18:17, 2011.

[19] Chandra Chekuri, Mohammad Taghi Hajiaghayi, Guy Kortsarz, and Mohammad R. Salavatipour. Approximation algorithms for nonuniform buy-at-bulk network design. *SIAM J. Comput.*, 39(5):1772–1798, 2010.

[20] Nikhil R. Devanur and Zhiyi Huang. Primal dual gives almost optimal energy-efficient online algorithms. *ACM Trans. Algorithms*, 14(1):5:1–5:30, 2018.

[21] Yevgeniy Dodis and Sanjeev Khanna. Design networks with bounded pairwise distance. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 750–759, 1999.

[22] Yuval Emek, Shay Kutten, Ron Lavi, and Yangguang Shi. Approximating generalized network design under (dis)economies of scale with applications to energy efficiency. *J. ACM*, 67(1):7:1–7:33, 2020.

[23] Yuval Emek, Shay Kutten, Ron Lavi, and Yangguang Shi. Personal communication, 2020.

[24] Michalis Faloutsos, Rajesh Pankaj, and Kenneth C. Sevcik. The effect of asymmetry on the on-line multicast routing problem. *Int. J. Found. Comput. Sci.*, 13(6):889–910, 2002.

[25] Moran Feldman, Guy Kortsarz, and Zeev Nutov. Improved approximation algorithms for directed steiner forest. *J. Comput. Syst. Sci.*, 78(1):279–292, 2012.

[26] Rohan Ghuge and Viswanath Nagarajan. Quasi-polynomial algorithms for submodular tree orienteering and other directed network design problems. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1039–1048, 2020.

[27] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995.

[28] Fabrizio Grandoni, Bundit Laekhanukit, and Shi Li. $O(\log^2 k/\log\log k)$-approximation algorithm for directed steiner tree: a tight quasi-polynomial-time algorithm. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 253–264, 2019.

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

[30] Zhiyi Huang and Anthony Kim. Welfare maximization with production costs: A primal dual approach. *Games Econ. Behav.*, 118:648–667, 2019.

[31] Ravishankar Krishnaswamy, Viswanath Nagarajan, Kirk Pruhs, and Cliff Stein. Cluster before you hallucinate: approximating node-capacitated network design and energy efficient routing. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 734–743, 2014.

[32] Konstantin Makarychev and Maxim Sviridenko. Solving optimization problems with diseconomies of scale via decoupling. *J. ACM*, 65(6):42:1–42:27, 2018.

[33] Viswanath Nagarajan and Xiangkun Shen. 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*, pages 12:1–12:12, 2017.

[34] Viswanath Nagarajan and Lily Wang. Online generalized network design under (dis)economies of scale. *CoRR*, abs/2007.07721, 2020.

[35] Tim Roughgarden. Intrinsic robustness of the price of anarchy. *J. ACM*, 62(5):32:1–32:42, 2015.

[36] Adam Wierman, Lachlan L. H. Andrew, and Ao Tang. Power-aware speed scaling in processor sharing systems: Optimality and robustness. *Perform. Evaluation*, 69(12):601–622, 2012.