Minimum-Cost Network Design with (Dis)economies of Scale

Matthew Andrews
Bell Labs, Alcatel-Lucent
Murray Hill, NJ, USA
Email: andrews@research.bell-labs.com

Spyridon Antonakopoulos
Bell Labs, Alcatel-Lucent
Murray Hill, NJ, USA
Email: spyros@research.bell-labs.com

Lisa Zhang
Bell Labs, Alcatel-Lucent
Murray Hill, NJ, USA
Email: ylz@research.bell-labs.com

***Abstract*—Given a network, a set of demands and a cost function $f(\cdot)$, the min-cost network design problem is to route all demands with the objective of minimizing $\sum_e f(\ell_e)$, where ℓ_e is the total traffic load under the routing. We focus on cost functions of the form $f(x) = \sigma + x^\alpha$ for $x > 0$, with $f(0) = 0$. For $\alpha \le 1$, $f(\cdot)$ is subadditive and exhibits behavior consistent with economies of scale. This problem corresponds to the well-studied Buy-at-Bulk network design problem and admits polylogarithmic approximation and hardness.**

In this paper, we focus on the less studied scenario of $\alpha > 1$ with a positive startup cost $\sigma > 0$. Now, the cost function $f(\cdot)$ is neither subadditive nor superadditive. This is motivated by minimizing network-wide energy consumption when supporting a set of traffic demands. It is commonly accepted that, for some computing and communication devices, doubling processing speed more than doubles the energy consumption. Hence, in Economics parlance, such a cost function reflects *diseconomies of scale*.

We begin by discussing why existing routing techniques such as randomized rounding and tree-metric embedding fail to generalize directly. We then present our main contribution, which is a polylogarithmic approximation algorithm. We obtain this result by first deriving a bicriteria approximation for a related capacitated min-cost flow problem that we believe is interesting in its own right. Our approach for this problem builds upon the well-linked decomposition due to Chekuri-Khanna-Shepherd [1], the construction of expanders via matchings due to Khandekar-Rao-Vazirani [2], and edge-disjoint routing in well-connected graphs due to Rao-Zhou [3]. However, we also develop new techniques that allow us to keep a handle on the total cost, which was not a concern in the aforementioned literature.

***Index Terms*—approximation algorithms; energy-efficient networks; diseconomies of scale.**

I. Introduction

We consider a minimum-cost network design problem with the following familiar formulation. We are given a traffic matrix that specifies demands to be transported over a network. We also have a set of network resources that incur cost for carrying traffic. Specifically, each resource e is associated with a cost function $f_e(\cdot)$ such that a cost of $f_e(\ell_e)$ is incurred if e carries a traffic load of ℓ_e. The objective of the design problem is to choose a route for each traffic demand so that the total cost $\sum_e f_e(\ell_e)$ is minimized.

Buy-at-Bulk network design is a well-studied problem that falls under this formulation. For Buy-at-Bulk, the cost functions $f_e(\cdot)$ exhibit *economies of scale*. That is, higher traffic load yields lower cost per unit traffic carried. More precisely, the functions $f_e(\cdot)$ for Buy-at-Bulk are *subadditive*, i.e.

$$f_e(x) + f_e(x') \ge f_e(x + x')$$

for any $x, x' \ge 0$. Buy-at-bulk has been extensively studied, because subadditive functions often model accurately the cost for purchasing link capacity in a variety of networks, whether it is a classic commodity network or a modern communication infrastructure. Polylogarithmic upper and lower bounds on the approximability of the Buy-at-Bulk problem are known, see e.g. [4], [5], [6], [7].

In this paper, we focus on a less studied case in which the cost function exhibits *(dis)economies of scale*. Our primary motivation for studying (dis)economies of scale is to take into account the *energy* cost of running a network. Energy conservation is attracting increasing attention in the fields of computing and networking, both because of the rapidly increasing monetary costs of powering large networks and server farms, as well as a desire to decrease the environmental impact of these operations [8], [9]. In this context, $f_e(\cdot)$ models an energy curve, reflecting how much energy e consumes as a function of its processing speed x. Here, e stands for a generic networking or computing device such as a CPU, communication link, edge router, etc. We assume that devices have the capability of *speed scaling*, which refers to adjusting the processing speed according to the traffic load. Speed scaling is a popular research topic, see e.g. [10], [11], [12], [13], [14], [15], [16], [17]. It is also a feature in some commercial products such as the Intel Pentium processors [18], standards like ADSL2 and ADSL2+ [19], and proposals to the IEEE 802.3az task forces [20], [21].

Extensive studies have suggested that the energy consumption of some devices may exhibit *diseconomies of scale* and thus be characterized by *superadditive* functions, i.e.

$$f_e(x) + f_e(x') \le f_e(x + x')$$

for any $x, x' \ge 0$. This implies, for instance, that doubling processing speed more than doubles the energy consumption, which is particularly true if increasing the speed of a device requires increasing both the clock speed and the supply voltage of a microprocessor. Many papers model the power requirements of a microprocessor as a polynomial function of the clock speed, such as $f_e(x) = \delta_e x^\alpha$, where δ_e and α are parameters associated with the device. While the exponent α

has been usually assumed to be around 3 [22], more recent estimates are markedly smaller. In particular, its value is 1.11, 1.66, and 1.62 for the Intel PXA 270, a TCP offload engine, and the Pentium M 770, respectively [23].

Min-cost network design under superadditive functions was recently studied in [24]. However, in general the problem can be inapproximable. For example, if the cost function is given by $f_e(x) = 0$ for $0 \le x \le 1$ and $f_e(x) = x-1$ for $x > 1$, then it was shown in [24] that finding a routing that incurs zero total cost is equivalent to solving the Edge-Disjoint-Paths (EDP) problem. Since EDP is NP-hard, it follows that achieving any finite approximation ratio for the above cost function is also NP-hard. Nevertheless, this is a rather unnatural function, especially in the context of energy curves, since the cost remains zero even for some non-zero speeds. Under cost functions $f_e(x) = \delta_e x^\alpha$, which are more natural for modeling energy consumption, [24] showed that a variant of randomized rounding achieves a constant approximation ratio, assuming that α is a constant.

On the other hand, for an even more accurate energy curve, a non-negligible *startup cost* is unavoidable. More specifically, we are interested in cost functions of the following form:

$$f_e(x) = \begin{cases} 0 & \text{for } x = 0 \\ \sigma_e + \delta_e x^\alpha & \text{for } x > 0 \end{cases}, \tag{1}$$

where $\sigma_e > 0$ may represent e.g. the cost required to keep a device active at an almost-idle state, or a fixed amount of energy needed to turn on the device. For example, σ_e includes the significant energy consumption due to leakage currents [18]. We remark that if $\alpha = 1$, we obtain one of the classic Buy-at-Bulk cost functions, which involves a startup cost plus a linear function. However, for the case in which $\alpha > 1$, $f_e(x)$ is obviously no longer subadditive, since the superadditive term x^α dominates for large values of x. We refer to this problem as min-cost network design with *(dis)economies* of scale, putting parentheses around the prefix "dis-" to stress that $f_e(x)$ exhibits behavior consistent with both economies of scale (for sufficiently small x only) *and* with diseconomies of scale (for large x only).

The difficulty in devising approximation algorithms under (dis)economies of scale comes from the fact that it is hard to know whether our routing should be aiming for more aggregation of demands, which would lower the cost coming from the σ_e terms, or more separation of demands, which would lower the cost coming from the $\delta_e x^\alpha$ terms. As we explain later, these aspects of the cost function mean that standard techniques such as tree metric embedding or randomized rounding cannot yield a satisfactory approximation, at least not in a straightforward way.

A. Model and Results

More formally, an instance of min-cost network design with (dis)economies of scale consists of a network, represented by an undirected graph $G = (V, E)$, and a set $\mathcal{D} = \{1, 2, \ldots, k\}$ of traffic demands, where the i^{th} demand ($1 \le i \le k$) is associated with an unordered pair of terminals $(s_i, t_i) \in V \times V$ and an integer $\mathsf{dem}_i > 0$ indicating the requested bandwidth. We assume links represent the abstracted resources, and each link $e \in E$ is associated with a cost function $f_e(\cdot)$. Our goal is to route all demands in an unsplittable fashion with the objective of minimizing the total cost $\sum_{e \in E} f_e(\ell_e)$, where the *load* ℓ_e of link e equals the total amount of traffic routed through e.

We focus on the *uniform* version of the cost function (1); namely, $f_e(\cdot)$ differs only by a constant factor from link to link:

$$f_e(x) = \begin{cases} 0 & \text{for } x = 0 \\ c_e(\sigma + x^\alpha) & \text{for } x > 0 \end{cases}. \tag{2}$$

Our main result is a polylogarithmic approximation algorithm for $\alpha > 1$ and $\sigma > 0$, in which case $f_e(\cdot)$ is neither superadditive nor subadditive.

As a byproduct, we obtain a polylogarithmic bicriteria approximation for the *capacitated network design problem*. The precise relationship between the two problems is demonstrated later in Lemma 2. In the capacitated network design problem, we are given an undirected graph (or multigraph without self-loops) $G^\circ = (V^\circ, E^\circ)$, where each link $e \in E^\circ$ has cost κ_e and all links have capacity q, as well as a set $\mathcal{D}^\circ = \{1, 2, \ldots, k^\circ\}$ of traffic demands. Again, the i^{th} demand ($1 \le i \le k^\circ$) is associated with an unordered pair of terminals $(s_i^\circ, t_i^\circ) \in V^\circ \times V^\circ$ and an integer $\mathsf{dem}_i^\circ > 0$ indicating the requested bandwidth. The goal is to route all demands in an unsplittable fashion, as before, while ensuring that the sum of bandwidths of the demands routed through each link e does not exceed q. Here, our objective is to minimize the total cost of links used in the routing, i.e. those with non-zero load. Our main result for this problem is a polylogarithmic approximation when we allow the capacity on each edge to be exceeded by a polylogarithmic factor.

B. Related Work

Under uniform subadditive cost functions, the well-studied Buy-at-Bulk problem has an $O(\log n)$-approximation [4], where $n = |V|$, via tree metric embeddings [25], [26]; it is also hard to approximate to within an $\Omega\big(\log^{1/4-\varepsilon} n\big)$ factor [7]. Under non-uniform subadditive cost functions, i.e. when the $f_e(\cdot)$ associated with different links may be completely unrelated to each other, Buy-at-Bulk has polylogarithmic approximation [6], [27] and is hard to approximate to within an $\Omega\big(\log^{1/2-\varepsilon} n\big)$ factor [7].

For the superadditive function $f_e(x) = \delta_e x^\alpha$, randomized rounding can lead to a constant approximation for unit demands [24]. On the contrary, for any $\alpha > 1$, there is a uniform cost function of the form (2) such that no $\Omega\big(\log^{1/4-\varepsilon} n\big)$ approximation is possible, even for unit demands [24]. This hardness result follows easily from the hardness of Buy-at-Bulk. It also indicates that, when $\alpha \ge 1$, the introduction of a startup cost $\sigma > 0$ in the cost function makes the problem intrinsically harder to optimize.

We stress that the capacitated network design problem we consider in this paper has some important differences from the Generalized Steiner Network problem for which Jain's iterative rounding algorithm provides a 2-approximation. In the latter problem, a set $\{r_{ij}\}$ of demands is given, and the goal is to find a minimum cost network such that there are r_{ij} disjoint paths between each node pair (i, j). In another — much more difficult — version, these paths need not be disjoint, but up to q_e of them may use edge e. This variant was studied by Carr et al. [28], who showed how to obtain an approximation ratio dependent on the number of non-zero coefficients in each row of the integer programming formulation. Observe, however, that these problems are quite different from ours, since the connectivity requirements of each demand need to be satisfied *in isolation*. By contrast, in our problem we aim to route *all* the demands *simultaneously*, and so the capacity of each edge must be no less than the *total* amount of traffic that is routed through it.

To the best of our knowledge, there are no previous polylogarithmic approximation algorithms for either our minimum cost network design problem with (dis)economies of scale or our formulation of capacitated network design. Nevertheless, we remark that the math programming community has studied cutting plane techniques for our capacitated problem, see e.g. [29]. Indeed, it is a special case of a problem known as Fixed Charge Network Flow [30].

C. A Hard Example

We now give a simple example for which neither randomized rounding nor routing based on tree metric embeddings yields a polylogarithmic approximation in the presence of (dis)economies of scale. This example was already discussed in [24], but we present it again here for completeness. Consider a pair of terminals (s, t), m parallel edges between them, and the cost function $f(x) = m + x^2$ for $x > 0$. We also have m unit demands that need to be routed between s and t. The optimal integral solution for this problem buys $\sqrt{m}$ edges and routes $\sqrt{m}$ of the demands on each edge. The total cost is $\Theta(m^{3/2})$.

With randomized rounding, we first solve the corresponding linear relaxation. The fractional optimal solution may buy a $1/m$ fraction of each edge and route a $1/m$ fraction of every demand through it. The total cost for this fractional routing is $m(\frac{1}{m})m + m = 2m$, which illustrates that the *integrality gap* of this relaxation is at least $\Omega(\sqrt{m})$. Moreover, if we apply randomized rounding by treating the fractional value assigned to each route as a probability distribution and choosing a route for every demand according to that distribution, then the probability that an edge is picked equals the probability that some demand picks it when all the demands choose a route uniformly at random. This probability is $1-(1-\frac{1}{m})^m \geq 1-\frac{1}{e}$. Therefore, the expected cost of the rounded solution is at least $m^2(1-\frac{1}{e})$, which is an $\Omega(\sqrt{m})$ factor more than the optimal integral cost.

In the tree metric embedding approach that was used for the uniform Buy-at-Bulk problem, we approximate the underlying network with a tree and then solve the problem on that tree, with the same cost function. However, since our example only has two nodes, the only possible tree on the two nodes consists of a single edge connecting them. Hence when we solve the problem on that tree, all m demands are routed on one edge. Under our cost function, this solution has cost $m+m^2$, which is again an $\Omega(\sqrt{m})$ factor more than the optimal integral cost.

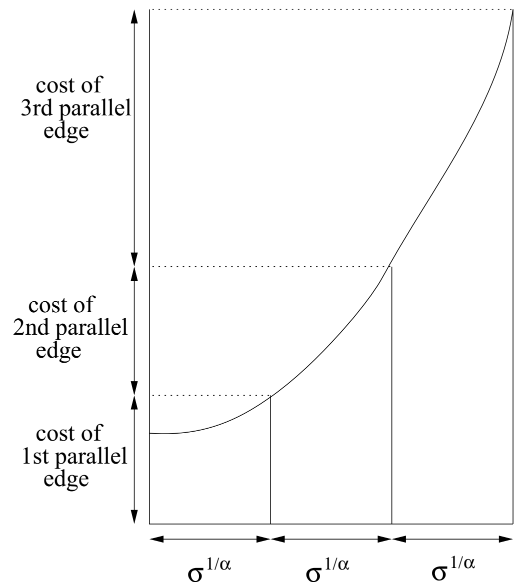

Fig. 1. Discretization of the cost function.

II. Overview

Given that demands can request different bandwidths, we first partition them into multiple buckets, each of which contains demands whose bandwidths are within a factor 2 of each other. Demands in the same bucket are therefore (almost) uniform in size, and we treat each bucket separately.

We set a parameter $\mu = \sqrt[\alpha]{\sigma}$. The significance of μ is that for loads $< \mu$ the startup cost σ becomes the dominant term in $f_e(\cdot)$, whereas the situation is reversed for loads $\geq \mu$. Hence, for a bucket of demands with bandwidth $\geq \mu$ each, we may simply use randomized rounding to route them and achieve a constant approximation, as proposed in [24].

By contrast, to route a bucket of small demands (with $\mathsf{dem}_i < \mu$), we begin by *aggregating* the demands and creating *superterminals*, each of which gathers a $\Theta(\mu)$ amount of traffic. In order to relate the costs of the aggregated and the original instances, we make use of a minimum Steiner forest defined on the original demands. Similar aggregation approaches have been explored before, e.g. in [31], [1]. We remark that the significance of aggregation becomes apparent during a filtering procedure later on.

Subsequently, we convert the aggregated instance into an instance of capacitated network design, with the same demand set and the same network, except that we replace every link e by a set of parallel links with capacity μ each. The cost of the i^{th} such parallel link is given by $f_e(i\mu)-f_e((i-1)\mu)$. This may be viewed as a discretization of the cost function, as depicted in Figure 1. Lemma 2 implies that an approximation for the capacitated instance can be transformed into an approximation for the aggregated instance. Therefore, most of the technical exposition in this paper focuses on capacitated network design.

For a capacitated instance where demand sizes are comparable to link capacities, we begin by obtaining a fractional optimal solution. Using that, we then decompose the network graph into a number of components, each of which is *well-cut-linked*. We say that a graph is well-cut-linked if it has no small cuts with a large number of terminals on both sides of the cut. Chekuri et al. [1] introduced this notion of well-linkedness, and showed that any fractional flow can always be decomposed into a set of disjoint and well-cut-linked components, without losing too much of the demand. Henceforth, we consider one component at a time.

One of the key steps in our analysis is manipulating the fractional flow from the well-linked decomposition. We ensure, via a filtering procedure, that each of the terminals has at least λ flow emanating from it, where λ has a carefully chosen inverse polylogarithmic value. The demand aggregation in the prior step ensures that a polylogarithmic fraction of the demands survive this filtering process. We then use the integrality theorem of minimum-cost flow to argue that there is a flow *of no greater cost* than the aforementioned fractional flow, such that the fraction of each link that is used is a multiple of λ. This in turn implies that there is an *integral* solution of cost no greater than $1/\lambda$ times the minimum cost flow. As a result, we have a technique for obtaining cheap building blocks which, combined with a method due to Khandekar-Rao-Vazirani [2], allows us to construct — at low cost — an expander that can be embedded in the existing graph.

Using this expander, we would like to argue that we can route a polylogarithmic fraction of the demands in a disjoint manner, as proposed in Rao-Zhou [3]. The catch here is that each superterminal collects $\Theta(\mu)$ traffic, which could prevent enough demands from simultaneously being routed disjointly. We therefore resort to a decomposition implied by König's Theorem [32] and route in $\Theta(\mu)$ rounds, where each round handles demands from distinct superterminals.

Not all demands are yet routed, as the filtering process and the disjoint routing within the expander only routes a polylogarithmic fraction of those. However, since we have an upper bound on the unrouted demands, it is straightforward to show that by recursively repeating the procedure a polylogarithmic number of times (and hence incurring a polylogarithmic factor in edge capacity violation and a polylogarithmic increase of the cost), we can in fact route all of them.

III. The algorithm

Before proceeding, we provide some additional definitions. In the following, $\mathcal{D}'$ stands for any given subset of the demands in $\mathcal{D}$. Denote the total demand $\sum_{i=1}^{k} \mathsf{dem}_i$ by D, the cost of the optimal solution by opt, and the minimum cost of a partial solution that routes the demands in $\mathcal{D}' \subseteq \mathcal{D}$ by $\mathsf{opt}_{\mathcal{D}'}$. Furthermore, let $\mu = \sqrt[\alpha]{\sigma}$ and $\mathsf{dmax} = \max_{i \in \mathcal{D}} \mathsf{dem}_i$.

Without loss of generality, we assume that each node is a terminal for at most one demand in $\mathcal{D}$. If that does not hold for some node $v \in V$, we simply create sufficiently many copies of v, each connected to v by an edge with zero cost coefficient, and replace v by a distinct copy of itself in whichever demand pair it appears. Clearly, this transformation does not affect the optimal solution cost, and the size of the transformed graph is only polynomially larger than that of the original one.

A. Preprocessing

a) Bucketing demands: To begin with, partition $\mathcal{D}$ into $\zeta = \lfloor \log \mathsf{dmax} \rfloor + 1 = O(\log D)$ subsets. In particular, for $1 \le j \le \zeta$ define

$$\mathcal{D}_j = \left\{ i \in \mathcal{D} \;\middle|\; 2^{j-1} \le \mathsf{dem}_i < 2^j \right\} .$$

Moreover, for every j and $i \in \mathcal{D}_j$, round dem_i up to 2^j; this adjustment entails only a constant factor loss in the approximation. Subsequently, we shall construct a partial solution for each $\mathcal{D}_j$.

b) Routing large demands: Note that if $2^j \ge \mu$, then in any partial solution that routes the demands in $\mathcal{D}_j$ only, for every network link $e \in E$ we have either $\ell_e = 0$ or $\ell_e \ge 2^j \ge \mu$. Consequently, we may approximate each function f_e by $f'_e(\ell_e) = 2c_e\ell_e^\alpha$, since $\frac{1}{2}f'_e(\ell_e) \le f_e(\ell_e) \le f'_e(\ell_e)$ for the aforementioned range of values of ℓ_e. Therefore, using the algorithm of Andrews et al. [24], we produce a partial solution that routes the demands in $\mathcal{D}_j$ with cost $\eta^\alpha \mathsf{opt}_{\mathcal{D}_j}$, where η is a constant.

c) Aggregating small demands: On the other hand, suppose that $2^j < \mu$. Let us construct an instance of the well-known Steiner forest problem on the graph G, with edge weights $c_e\sigma$ and terminal pairs (s_i, t_i), $i \in \mathcal{D}_j$. It is easy to see that the minimum weight of a Steiner forest is at most $\mathsf{opt}_{\mathcal{D}_j}$, hence by applying the 2-approximation algorithm in [33] we can find a Steiner forest H of weight $\le 2\mathsf{opt}_{\mathcal{D}_j}$ efficiently.

Naturally, the connected components of H, say $H_1, \dots, H_\theta$, are trees. Take each component $H_p = (V_p, E_p)$, $1 \le p \le \theta$, root it at an arbitrary leaf node $v_0 \in V_p$, and denote by T_p a depth-first-search traversal of H_p. Then, apply Procedure 1 on H_p to designate certain nodes of V_p as *superterminals* and assign each $v \in V_p$ to a superterminal.

For any $i \in \mathcal{D}_j$, let $\tilde{s}_i$ and $\tilde{t}_i$ be the superterminals to which s_i and t_i are assigned, respectively. For each demand $i \in \mathcal{D}_j$, create a so-called *aggregated demand*, simply by replacing the pair (s_i, t_i) with $(\tilde{s}_i, \tilde{t}_i)$. Then, consider an *aggregated instance* of the problem on the graph G with those aggregated demands only. We now derive the lemma below.

Lemma 1. *Any solution to the aggregated instance specified above may be converted into a partial solution that routes the demands in $\mathcal{D}_j$, and vice versa. If C_{aggr} and C_{orig} are the respective costs of these solutions, then in one direction we can guarantee that*

$$C_{\text{orig}} \le 2^{\alpha-1} \left(C_{\text{aggr}} + 2(1+4^\alpha)\mathsf{opt}_{\mathcal{D}_j} \right) ,$$

whereas in the other that

$$C_{\text{aggr}} \le 2^{\alpha-1} \left(C_{\text{orig}} + 2(1+4^\alpha)\mathsf{opt}_{\mathcal{D}_j} \right) .$$

Procedure 1 Aggregation of small demands

```
initially all nodes in V_p are unassigned
for v ∈ V_p do
    d(v) ← dem_i if ∃i ∈ D_j s.t. v = s_i or v = t_i, else 0
w ← 0; W ← ∅
v_curr ← v_0
v_prev ← v_last ← node visited last in T_p
while Σ_v d(v) ≥ 3μ, summed over unassigned nodes do
    while w < μ do
        w ← w + d(v_curr)
        W ← W ∪ {v_curr}
        v_prev ← v_curr
        v_curr ← node visited after v_curr in T_p
    if W ≠ ∅ then
        declare v_prev a superterminal
        assign all nodes in W to v_prev
        w ← 0; W ← ∅
declare v_last a superterminal
assign all currently unassinged nodes to v_last
```

Sketch of proof: The first direction is established as follows. Using the edges of H, for $i \in \mathcal{D}_j$ we route dem_i flow between s_i and $\tilde{s}_i$, as well as between t_i and $\tilde{t}_i$. By the construction of superterminals, the load on any edge of H is at most $\mu+3\mu = 4\mu$ in this routing. Hence, its cost is bounded by

$$\sum_{e\in H}(1+4^\alpha)c_e\sigma \le 2(1+4^\alpha)\mathsf{opt}_{\mathcal{D}_j}\ ,$$

and when combined with the aforementioned solution to the aggregated instance, it produces a partial solution for the demands in $\mathcal{D}_j$ with cost not exceeding

$$2^{\alpha-1}\left(C_{\text{aggr}} + 2(1+4^\alpha)\mathsf{opt}_{\mathcal{D}_j}\right)\ .$$

Finally, the argument for the opposite direction is entirely similar. ■

d) Reduction to capacitated network design: At this point, let us create an instance of the capacitated min-cost network design problem. The multigraph $G^\circ = (V^\circ, E^\circ)$ has the same node set $V^\circ = V$ as G. Moreover, for every link $e \in E$, we add at most $\omega = |\mathcal{D}_j| \le k$ parallel edges $e_1, e_2, \ldots, e_\omega$ to E°, where e_z has cost $c_e\sigma(z^\alpha - (z-1)^\alpha)$, $1 \le z \le \omega$. The edge capacity q is set to μ, and the demand set $\mathcal{D}^\circ$ consists of exactly the same (aggregated) demands as in the aggregated instance discussed earlier. Owing to our choice of edge costs, the next lemma is readily verified.

Lemma 2. *The ratio of the optimal cost of the capacitated min-cost network design instance defined above to the optimal cost of the aggregated instance lies between* 1 *and* 2*.*

B. Solving capacitated network design

A fractional relaxation of the capacitated network design problem can be formulated as the linear program $\mathsf{LP3}$, where the variable $x_{e,i}$ indicates the fraction of demand $i \in \mathcal{D}^\circ$ that is routed along link $e \in E^\circ$. Provided that $\mathsf{LP3}$ is feasible, let

$\mathsf{LP3}$: Fractional capacitated network design

$$\text{minimize} \quad \sum_{e\in E^\circ}\kappa_e\sum_{i=1}^{k^\circ}x_{e,i} \tag{3a}$$

$$\text{subject to} \quad \sum_{i=1}^{k^\circ}x_{e,i} \le q \qquad \forall e \in E^\circ \tag{3b}$$

$$\left\langle \begin{matrix}\text{flow conservation}\\ \text{constraints on } x_{e,i}\end{matrix}\right\rangle \tag{3c}$$

$$0 \le x_{e,i} \le 1 \qquad \begin{matrix}\forall e \in E^\circ,\\ \forall i = 1,2,\ldots,k^\circ\end{matrix} \tag{3d}$$

$\mathsf{opt}_{\mathsf{LP3}}$ be the optimal (fractional) solution cost. Our objective is to find an integral routing, i.e. one in which all $x_{e,i} \in \{0,1\}$, such that the total cost of edges used is at most $\beta\mathsf{opt}_{\mathsf{LP3}}$ and the load on each edge is at most γq — where β and γ are specified later.

1) The case $q = 1$: In this section we describe a solution to the special case of unit capacity ($q = 1$) and generalize it to $q > 1$ in Section III-B2. For $q = 1$, dem_i° should be 1 for all $i \in \mathcal{D}^\circ$. Demand aggregation in the preprocessing step is therefore not necessary. This allows us to continue to assume without loss of generality that the demand terminals are distinct. As we shall see in Step 4 of the algorithm, distinct terminals make edge-disjoint routing in expander graphs more manageable.

a) Step 1: We obtain a fractional optimal solution $\mathbf{x}_{i,e}$ for the linear program $\mathsf{LP3}$. Let $\tau_e = \sum_i \mathbf{x}_{i,e}$ be the load on edge e in the fractional solution. Take G^τ to be the same graph as G°, but with each edge e having capacity τ_e instead of q. Clearly, the fractional routing implied by $\mathbf{x}_{i,e}$ is still feasible in G^τ, because $\tau_e \le q$ for all e. Moreover, $\mathsf{opt}_{\mathsf{LP3}} = \sum_e \kappa_e\tau_e$ is a lower bound on the cost of any integral routing in G° that respects edge capacities.

b) Step 2: We apply the following theorem of Chekuri, Khanna and Shepherd [1].

Theorem 3 ([1])**.** *We can decompose G^τ into node-disjoint subgraphs $G^\tau_1, G^\tau_2, \ldots, G^\tau_\phi$ and produce a weight function π on the terminals with the properties listed below. Denote by $\mathcal{D}^\circ_r \subseteq \mathcal{D}^\circ$ the set of induced terminal pairs in G^τ_r, and by X_r the set of terminals in $\mathcal{D}^\circ_r$.*

- $0 \le \pi(s_i) = \pi(t_i) \le 1$ *for all* $i \in \mathcal{D}^\circ$.
- *Each* G^τ_r *is* π-cut-linked. *That is, for any* $A \subseteq V(G^\tau_r)$*, the capacity on the cut defined by* A *and* $\bar{A} = V(G^\tau_r)\setminus A$ *in* G^τ_r *is at least* $\min\{\pi(A), \pi(\bar{A})\}$.
- *If* n° *is the number of nodes in* G°*, then*

$$\sum_{r=1}^{\phi}\pi(X_r) = \Omega\big(|\mathcal{D}^\circ|\log^{-2}n^\circ\big)\ .$$

In effect, the function π indicates how much of each terminal's original demand remains after the decomposition. Subsequently, we apply a filter to eliminate terminals with too little demand and ensure that all remaining terminals have exactly $\lambda = 1 \big/ \lceil \log^3 n^\circ \rceil$ demand. This is achieved by defining a new function ρ on the terminals, which is closely related to π. For a terminal $u \in X_r$, let

$$\rho(u) = \begin{cases} 0 & \text{if } \pi(u) < \lambda; \\ \lambda & \text{otherwise.} \end{cases}$$

Lemma 4. *The function ρ has properties similar to those of π. More specifically,*

1) $0 \le \rho(s_i) = \rho(t_i) \le 1$ *for all* $i \in \mathcal{D}^\circ$*;*
2) *each* G_r^τ *is* ρ*-cut-linked; and*
3) $\sum_{r=1}^{\phi} \rho(X_r) = \Omega\big(|\mathcal{D}^\circ| \lambda \log^{-2} n^\circ\big)$.

Proof: The first property follows directly from the fact that $\pi(s_i) = \pi(t_i)$ and the definition of ρ.

To verify the second property, consider any $A \subseteq V(G_r^\tau)$ and $\bar{A} = V(G_r^\tau) \setminus A$. Since the component G_r^τ is π-cut-linked, the capacity of the cut defined by A and $\bar{A}$ is at least $\min\{\pi(A), \pi(\bar{A})\}$. By the definition of ρ, $\pi(A) \ge \rho(A)$ and $\pi(\bar{A}) \ge \rho(\bar{A})$. Consequently,

$$\min\{\rho(A), \rho(\bar{A})\} \le \min\{\pi(A), \pi(\bar{A})\} \ ,$$

and hence the cut is at least $\min\{\rho(A), \rho(\bar{A})\}$, so every G_r^τ is ρ-cut-linked as well.

For the last property, observe that

$$\sum_r \rho(X_r) \ge \lambda \left(\sum_r \pi(X_r) - \lambda |\mathcal{D}^\circ| \right) \ ,$$

since the sum of π values that were reduced to zero in ρ is at most $\lambda |\mathcal{D}^\circ|$ and the remaining π values were reduced by at most a factor of λ. ■

c) Step 3: Henceforth, we concentrate on one subgraph G_r^τ at a time. Recall that X_r is the set of terminals in G_r^τ, and let $X'_r = \{u \in X_r \mid \rho(u) = \lambda\}$. Since G_r^τ is ρ-cut-linked, we deduce that for *any* partition of X'_r into two equal halves (A, B), we can support a flow $\xi_{G_r^\tau}$ such that the amount of flow emanating from each node in A and the amount of flow absorbed by each node in B are both exactly λ. The existence of $\xi_{G_r^\tau}$ follows from the max-flow/min-cut theorem and the fact that G_r^τ is ρ-cut-linked. Moreover, if $\xi_{G_r^\tau}(e)$ indicates the amount of flow on $e \in E(G_r^\tau)$, we have $\xi_{G_r^\tau}(e) \le \tau_e$.

Now, let G_r^χ have the same sets of nodes and edges as G_r^τ, but with capacity 1 for each edge. In G_r^χ we determine a min-cost flow $\xi_{G_r^\chi}$ such that, again, the amount of flow emanating from each node in A and the amount of flow absorbed by each node in B are both exactly λ. Since 1 is an exact multiple of λ, the integrality theorem for min-cost flow guarantees that $\xi_{G_r^\chi}$ is λ-integral, meaning that $\xi_{G_r^\chi}(e)$ is an exact multiple of λ, for every $e \in E(G_r^\chi)$. Furthermore,

$$\sum_{e \in E(G_r^\chi)} \kappa_e \xi_{G_r^\chi}(e) \le \sum_{e \in E(G_r^\tau)} \kappa_e \xi_{G_r^\tau}(e) \ ,$$

because capacity constraints in G_r^χ are more relaxed than those in G_r^τ.

Scaling up $\xi_{G_r^\chi}$ by a factor $1/\lambda$, we obtain an integral flow $\xi'_{G_r^\chi}$, under which the the amount of flow emanating from each node in A and the amount of flow absorbed by each node in B are both exactly 1. Therefore, $\xi'_{G_r^\chi}$ may be decomposed into $|A|$ paths, each carrying one unit of flow from a (distinct) node in A to a (also distinct) node in B. We call this a *path-matching* between the sets A and B. Note that any edge $e \in E(G_r^\chi) = E(G_r^\tau)$ belongs to at most $1/\lambda$ paths, and that the total cost of edges used by these paths is bounded by

$$\begin{aligned} \sum_{e \in E(G_r^\chi)} \kappa_e \xi'_{G_r^\chi}(e) &= \sum_{e \in E(G_r^\chi)} \kappa_e \xi_{G_r^\chi}(e)/\lambda \\ &\le \sum_{e \in E(G_r^\tau)} \kappa_e \xi_{G_r^\tau}(e)/\lambda \\ &\le \sum_{e \in E(G_r^\tau)} \kappa_e \tau_e/\lambda \ . \end{aligned}$$

Lemma 5. *Consider any equal partition (A, B) of X'_r. We can compute a path-matching between A and B within G_r^τ such that each edge in $G_r^\tau \subseteq G^\tau$ is used by no more than $1/\lambda$ such paths, and the total cost of links used is at most $\sum_{e \in E(G_r^\tau)} \kappa_e \tau_e/\lambda$.*

d) Step 4: At this point, we will use our above method for constructing path-matchings as a building block for our routing, in conjunction with two results stated below, due to Khandekar-Rao-Vazirani [2] and Rao-Zhou [3].

Theorem 6 ([2])**.** *Given a set $\mathcal{V}$ of N nodes and a procedure that finds a matching for any specified equal partition (A, B) of $\mathcal{V}$, we may efficiently determine $\psi = \Theta\big(\log^2 N\big)$ such partitions $(A_1, B_1), (A_2, B_2), \ldots, (A_\psi, B_\psi)$, so that the union of their corresponding matches results in a graph with $\Theta(1)$ expansion.*

Theorem 7 ([3])**.** *Consider an expander graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, as in Theorem 6, and a number of pairs $(\hat{s}_i, \hat{t}_i) \in \binom{\mathcal{V}}{2}$. If each node belongs to at most one pair, then we may connect at least a $\Omega\big(\log^{-2} N\big)$ fraction of these pairs using edge-disjoint paths in $\mathcal{G}$.*

Since a path-matching between two equal-size node sets can be viewed as a (conceptual) matching, we may use the procedure implied by Lemma 5 in Theorem 6 to produce an "expander", each of whose edges represents in effect a path in G_r^τ. The union of these paths forms a subgraph of G_r^τ with total edge cost not exceeding

$$O\big(\log^2 n^\circ/\lambda\big) \cdot \sum_{e \in E(G_r^\tau)} \kappa_e \tau_e \ ,$$

and every edge is contained in no more than $O\big(\log^2 n^\circ/\lambda\big)$ paths. Theorem 7 suggests that we can route at least an $\Omega\big(\log^{-2} n^\circ\big)$ fraction of the $|X'_r|/2$ demands with terminals in X'_r, using each of the aforementioned paths at most once. Across all node-disjoint subgraphs $G_1^\tau, \ldots, G_\phi^\tau$, $\Omega\big(|\mathcal{D}^\circ| \log^{-4} n^\circ\big)$ demands may thus be routed. Hence, we can route *all* the demands in $\mathcal{D}^\circ$ by recursively applying this entire process $O(\log^5 n^\circ)$ times, because

$$|\mathcal{D}^\circ|\left(1-\Omega(\log^{-4} n^\circ)\right)^{O(\log^5 n^\circ)} < |\mathcal{D}^\circ|/n^\circ \le 1 \ .$$

The total cost of this solution is bounded by $O(\log^7 n^\circ/\lambda) \cdot \mathsf{opt}_{\mathsf{LP3}}$, and the load on each link is at most $O(\log^7 n^\circ/\lambda)$.

Theorem 8. *We have found an integral routing such that the total cost of edges used is at most $\beta\mathsf{opt}_{\mathsf{LP3}}$ and the load on each edge is at most γ, with $\beta = \gamma = \operatorname{polylog}(n^\circ)$.*

2) The case $q = \mu$: We now generalize our algorithm for the case $q = 1$ to the case $q = \mu$. Recall that demand aggregation in the preprocessing step in Section III-A creates superterminals, each of which terminates demands of size between μ and 3μ. This creates an additional complication that not all of the demand at one superterminal X is necessarily destined to the same superterminal Y. We highlight the necessary changes in our algorithm.

For the aggregated instance on the superterminals, we first obtain an optimal fractional solution as before. We again perform the Chekuri et al. decomposition to obtain a set of well-linked instances. After this step some of the superterminals may only have a small amount of aggregate demand left. We therefore filter out all of the superterminals whose aggregate demand is less than λ, where λ is redefined to be $\mu/\lceil\log^3 n^\circ\rceil$. The new λ value and the fact that each superterminal initially had $\Theta(\mu)$ aggregated demands allow us to show that at least a $\Omega(\lambda \log^{-2} n^\circ)$ fraction of aggregated demands are not filtered out. For those remaining demands, we again use the integrality theorem of min-cost flow and well-linkedness to show that there is a low-cost integral matching between any equal-sized partition of the superterminals.

However, due to aggregation each superterminal is not unique to some aggregated demand. In order to apply Theorem 7, we note that demands within a single bucket have size equal to 2^j for some fixed j. König's lemma [32] states that we can decompose any bipartite graph with edge degree at most $3\mu/2^j$ into $3\mu/2^j$ disjoint matchings. Hence, we can decompose the problem on the superterminals into $3\mu/2^j$ separate problems such that in each separate problem each superterminal represents at most one demand. Then, whenever we construct an expander, we look at the separate problems one-by-one and disjointly route the demands for each. Whenever a demand is routed, it consumes 2^j of the capacity on each edge along its route. Consequently, in order to route all of the $3\mu/2^j$ separate problems, we need at most 3 copies of each edge and the cost at most triples.

Therefore, Theorem 8 extends to $q = \mu$.

3) Wrapping up: Going back to the original network design problem with (dis)economies of scale, recall that we process one bucket of demands at a time. For the j^{th} bucket, $1 \le j \le \zeta$, we have a partial solution of cost at most $\operatorname{polylog}(n)\,\mathsf{opt}_{\mathcal{D}_j}$. Combining these partial solutions yields a routing with cost $\zeta^\alpha \operatorname{polylog}(n)\,\mathsf{opt} = \operatorname{polylog}(n, D)\,\mathsf{opt}$, assuming that α is a constant.

Theorem 9. *Uniform network design with (dis)economies of scale has a $\operatorname{polylog}(n, D)$ approximation.*

IV. Conclusion

In this paper, we have derived a (β,γ)-bicriteria approximation for the uniform capacitated problem, where β, polylogarithmic in value, gives the cost guarantee and γ, also polylogarithmic in value, bounds the blowup in link capacity. We have also shown that this approximation implies a $\operatorname{poly}(\beta,\gamma)$ approximation for the uniform min-cost network design problem under the cost function (2). We have focused on this particular cost function as it provides a natural model when considering the energy cost of a network.

However, other cost functions can directly benefit from the (β,γ)-bicriteria approximation as well. For example, if

$$\begin{aligned} f(\lceil x/\mu\rceil\mu) &\le \nu_1 f(x) \ , \\ f(\gamma x) &\le \nu_2 \gamma f(x) \ , \\ \text{and} \quad f((i+1)\mu) - f(i\mu) &\ge f(i\mu) - f((i-1)\mu) \end{aligned}$$

for some parameter μ and every positive integer i, then min-cost network design admits an $O(\beta\nu_1\nu_2)$ approximation under the cost function $f(\cdot)$. If ν_1 and ν_2 are polylogarithmic in size, then the resulting approximation is again polylogarithmic.

Of course, the main open question is how to handle the non-uniform versions of the capacitated problem and the min-cost network design problem with (dis)economies of scale, respectively. We leave them both as challenging future work.

Acknowledgments

The authors wish to thank Chandra Chekuri for insightful discussions. This work was completed with the support of NSF contract CCF-0728980 and the U.S. Department of Energy (DOE), award no. DE-EE0002887. However, any opinions, findings, conclusions and recommendations expressed herein are those of the authors and do not necessarily reflect the views of the DOE.

References

[1] C. Chekuri, S. Khanna, and F. B. Shepherd, "Multicommodity flow, well-linked terminals, and routing problems," in *Proceedings of ACM STOC*, 2005, pp. 183–192.

[2] R. Khandekar, S. Rao, and U. Vazirani, "Graph partitioning using single commodity flows," in *Proceedings of ACM STOC*, 2006, pp. 385–390.

[3] S. Rao and S. Zhou, "Edge disjoint paths in moderately connected graphs," *SIAM Journal on Computing*, vol. 39, no. 5, pp. 1856–1887, 2010.

[4] B. Awerbuch and Y. Azar, "Buy-at-bulk network design," in *Proceedings of IEEE FOCS*, 1997, pp. 542–547.

[5] M. Charikar and A. Karagiozova, "On non-uniform multicommodity buy-at-bulk network design," in *Proceedings of ACM STOC*, 2005, pp. 176–182.

[6] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour, "Approximation algorithms for nonuniform buy-at-bulk network design," *SIAM Journal on Computing*, vol. 39, no. 5, pp. 1772–1798, 2010. [Online]. Available: http://link.aip.org/link/?SMJ/39/1772/1

[7] M. Andrews, "Hardness of buy-at-bulk network design," in *Proceedings of IEEE FOCS*, 2004, pp. 115–124.

[8] "SMART2020: Enabling the low carbon economy in the information age," 2008. [Online]. Available: http://www.smart2020.org

[9] P. Kurp, "Green computing," *Commun. ACM*, vol. 51, no. 10, pp. 11–13, 2008.

[10] F. F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in *Proceedings of IEEE FOCS*, 1995, pp. 374–382.

[11] M. Li, B. J. Liu, and F. F. Yao, "Min-energy voltage allocation for tree-structured tasks," in *Proceedings of COCOON*, 2005, pp. 283–296.

[12] N. Bansal, T. Kimbrel, and K. Pruhs, "Speed scaling to manage energy and temperature," *Journal of the ACM*, vol. 54, no. 1, 2007.

[13] H.-L. Chan, W.-T. Chan, T. W. Lam, L.-K. Lee, K.-S. Mak, and P. W. H. Wong, "Energy efficient online deadline scheduling," in *Proceedings of ACM-SIAM SODA*, 2007, pp. 795–804.

[14] N. Bansal, H.-L. Chan, and K. Pruhs, "Speed scaling with an arbitrary power function," in *Proceedings of ACM-SIAM SODA*, 2009, pp. 693–701.

[15] S. Irani and K. Pruhs, "Algorithmic problems in power management," *SIGACT News*, vol. 36, no. 2, pp. 63–76, 2005.

[16] C. Gunaratne, K. J. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with adaptive link rate (ALR)," *IEEE Trans. Computers*, vol. 57, no. 4, pp. 448–461, 2008.

[17] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *NSDI*, J. Crowcroft and M. Dahlin, Eds. USENIX Association, 2008, pp. 323–336.

[18] "Enhanced Intel SpeedStep technology for the Intel Pentium M processor," Intel White Paper 301170-001, 2004.

[19] G. Ginis, "Low-power modes for ADSL2 and ADSL2+," White Paper, Broadband Communications Group, Texas Instruments, Jan. 2005. [Online]. Available: http://focus.ti.com/pdfs/bcg/adsl2_low_power_wp.pdf

[20] IEEE P802.3az Energy Efficient Ethernet Task Force, "Task Force meeting materials," 2008. [Online]. Available: http://grouper.ieee.org/groups/802/3/az/public/index.html

[21] P. Patel-Predd, "Energy-efficient ethernet," *IEEE Spectrum*, p. 13, May 2008.

[22] D. Brooks, P. Bose, S. Schuster, H. M. Jacobson, P. Kudva, A. Buyuktosunoglu, J.-D. Wellman, V. V. Zyuban, M. Gupta, and P. W. Cook, "Power-aware microarchitecture: Design and modeling challenges for next-generation microprocessors," *IEEE Micro*, vol. 20, no. 6, pp. 26–44, 2000.

[23] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems," in *Proceedings of IEEE INFOCOM*, 2009, pp. 2007–2015.

[24] M. Andrews, A. Fernandez Anta, L. Zhang, and W. Zhao, "Routing for energy minimization in the speed scaling model," in *Proceedings of IEEE INFOCOM*, 2010.

[25] Y. Bartal, "Probabilistic approximation of metric spaces and its algorithmic applications," in *Proceedings of IEEE FOCS*, 1996, pp. 184–193.

[26] J. Fakcharoenphol, S. Rao, and K. Talwar, "A tight bound on approximating arbitrary metrics by tree metrics," *Journal of Computer and System Sciences*, vol. 69, pp. 485–497, 2004.

[27] G. Kortsarz and Z. Nutov, "Approximating some network design problems with node costs," in *Proceedings of APPROX-RANDOM*, 2009, pp. 231–243.

[28] R. D. Carr, L. Fleischer, V. J. Leung, and C. A. Phillips, "Strengthening integrality gaps for capacitated network design and covering problems," in *Proceedings of ACM-SIAM SODA*, 2000, pp. 106–115.

[29] B. Gendron, T. G. Crainic, and A. Frangioni, "Multicommodity capacitated network design," in *Telecommunications Network Planning*, B. Sansò and P. Soriano, Eds., 1999, pp. 1–19.

[30] M. Hewitt, G. L. Nemhauser, and M. W. P. Savelsbergh, "Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem," *INFORMS Journal on Computing*, vol. 22, no. 2, pp. 314–325, 2010.

[31] A. Gupta, A. Kumar, and T. Roughgarden, "Simpler and better approximation algorithms for network design," in *Proceedings of ACM STOC*, 2003, pp. 365–372.

[32] D. Kőnig, "Graphok és matrixok," *Matematikai és Fizikai Lapok*, vol. 38, pp. 116–119, 1931.

[33] M. X. Goemans and D. P. Williamson, "A general approximation technique for constrained forest problems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 296–317, 1995.