This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Routing for Power Minimization in the Speed Scaling Model

Matthew Andrews, Antonio Fernández Anta, *Senior Member, IEEE, ACM*, Lisa Zhang, and Wenbo Zhao

***Abstract*—We study network optimization that considers power minimization as an objective. Studies have shown that mechanisms such as *speed scaling* can significantly reduce the power consumption of telecommunication networks by matching the consumption of each network element to the amount of processing required for its carried traffic. Most existing research on speed scaling focuses on a single network element in isolation. We aim for a network-wide optimization. Specifically, we study a routing problem with the objective of provisioning guaranteed speed/bandwidth for a given demand matrix while minimizing power consumption. Optimizing the routes critically relies on the characteristic of the speed–power curve $f(s)$, which is how power is consumed as a function of the processing speed s. If f is superadditive, we show that there is no bounded approximation in general for integral routing, i.e., each traffic demand follows a single path. This contrasts with the well-known logarithmic approximation for subadditive functions. However, for common speed–power curves such as polynomials $f(s) = \mu s^\alpha$, we are able to show a constant approximation via a simple scheme of randomized rounding. We also generalize this rounding approach to handle the case in which a nonzero startup cost σ appears in the speed–power curve, i.e., $f(s) = \begin{cases} 0, & \text{if } s = 0 \\ \sigma + \mu s^\alpha, & \text{if } s > 0. \end{cases}$ We present an $O((\sigma/\mu)^{1/\alpha})$-approximation, and we discuss why coming up with an approximation ratio independent of the startup cost may be hard. Finally, we provide simulation results to validate our algorithmic approaches.**

***Index Terms*—Power saving, routing, speed scaling, wireline networks.**

I. Introduction

ENERGY conservation is emerging as a key issue in computing and networking as the information and communications technologies (ICT) sector significantly steps up the energy efficiency of its products and services in response to growing energy bills, government mandates, as well as societal pressures to minimize the carbon emissions by the sector [26], [34]. Many methods for avoiding waste and improving energy efficiency are being developed. One opportunity to reap significant potential saving in data networking is achieving *proportionality* [9]. Proportionality refers to a goal in which the power consumption by a network element is in proportion to the carried traffic load. We use network element as a generic term that represents a computing and communication resource such as a router, switch, CPU, or a link connecting this equipment.

As indicated in a study conducted by the U.S. Department of Energy [3], the current network elements and telecommunication networks are not designed with energy optimization as an objective or a constraint. They are often designed for peak traffic for reasons such as accommodating future growth, planned maintenance or unexpected failures, or quality-of-service guarantees. At the same time, the power consumption of network elements is often defined by the peak profile and varies little for typical traffic, which can be a small fraction of the peak. By a conservative estimate in the same study, at least 40% of the total consumption by network elements such as switches and routers can be saved if proportionality is achieved. This translates to a saving of 24 billion kWh per year attributed to data networking [3].

Two popular methods for effectively matching power consumption to traffic load are via *speed scaling* or *powering down*. The former refers to setting the processing speed of a network element according to traffic load, and the latter refers to turning off the element. Both methods are the subject of active research, though most of the work focuses on optimizing an individual element in isolation [8], [13], [18], [21]–[24], [28], [36]. In addition, enabling sleep modes and sizing power to traffic are also features in some commercial products such as the Intel pentium processors [16], standards like ADSL2 and ADSL2+ [19], or proposals to the IEEE 802.3az task forces [17], [31]. Our goal is to examine the optimization problems that arise in a network consisting of multiple network elements.

We focus on the speed scaling model in this paper and study the power down model in a separate effort [5]. We assume that each network element e has the speed scaling capability, characterized by a speed–power curve $f_e(s)$, which is how e consumes power as a function of its processing speed s. We propose a routing problem with the objective of provisioning for a long-timescale traffic matrix and minimizing the total power consumption by the network elements over the entire network. Routing determines the traffic load on each network element,

Manuscript received September 18, 2010; revised April 06, 2011; accepted May 21, 2011; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor P. Van Mieghem. This work was supported in part by the National Natural Science Foundation of China under Grant 61020106002, Comunidad de Madrid under Grant S2009TIC-1692, and Spanish MICINN under Grant TIN2008–06735-C02-01. The work of A. Fernández Anta and W. Zhao was done partly while visiting Bell Labs, Murray Hill, NJ. The first version of this paper appeared in the IEEE Conference on Computer Communications (INFOCOM), San Diego, CA, March 15–19, 2010.

M. Andrews and L. Zhang are with Bell Labs, Murray Hill, NJ 07974 USA (e-mail: andrews@research.bell-labs.com; ylz@research.bell-labs.com).

A. Fernández Anta is with the Institute IMDEA Networks, Madrid 28918, Spain (e-mail: antonio.fernandez@imdea.org).

W. Zhao is with the Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92037 USA (e-mail: w3zhao@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2011.2159864

and this in turn determines the power consumption specified by the speed–power curve $f_e(\cdot)$.

The algorithmic aspect of this routing problem critically relies on the nature of the speed–power curve $f_e(s)$. For example, if the curve should be linear $f_e(s) = \mu s$, then shortest path routing is optimal. Unfortunately, situations in reality are far more complex. For example, several preliminary studies on Ethernet links, edge routers, e.g., [12] and [20], and the well-accepted understanding of optical links and equipment suggest that the power consumption for network elements as such may grow *subadditively* with the speed. That is, doubling speed less than doubles the power, or more formally $f_e(s_1)+f_e(s_2) \geq f_e(s_1+s_2)$. In this case, the routing problem corresponds to the well-studied problem of buy-at-bulk network design (BAB), e.g., [6], [7], [14], and [15]. BAB has a logarithmic approximation and almost matching logarithmic hardness.

On the other hand, the power consumption of a microprocessor grows *superadditively* with the speed. That is, doubling the speed more than doubles the power consumption, or more formally $f_e(s_1) + f_e(s_2) \leq f_e(s_1 + s_2)$. Furthermore, the speed–power curve is often modeled by a polynomial function $f_e(s) = \mu_e s^\alpha$ where μ_e and α are parameters associated with the device. (While α has been usually assumed to be around 3 [11], it has been recently estimated to be much 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 [35].) Note that if *fractional* routes are allowed, i.e., a demand may be carried on multiple paths between its source and destination, then the problem falls into the realm of convex optimization since μs^α is convex, and therefore is solvable in polynomial time [10]. However, for *integer routes* where each demand must be carried on one single path, the problem has, to the best of our knowledge, not received much attention before. Integral routing can be important for a number of reasons, e.g., if we wish to avoid problems associated with packet reordering.

In addition, a more accurate but more complex speed–power curve for a microprocessor may be $f_e(s) = \begin{cases} 0, & \text{if } s = 0 \\ \sigma_e + \mu_e s^\alpha, & \text{if } s > 0, \end{cases}$ where σ_e represents the nonnegligible power consumption by leakage currents; see, e.g., [16]. This speed–power function is neither superadditive nor subadditive, and little is known about routing optimization subject to such functions. For the rest of the paper, we explore these two well-motivated speed–power functions by showing how to approximate the optimal solution as well as the limit to which approximation can be accomplished.

A. Previous Work

Speed scaling has been widely studied to save power consumption at the single-element level. Yao *et al.* [36] were the first to study speed scaling in processors, in the form of task scheduling problems. They assumed that the power to run a processor at a speed s grows superlinearly with s, and they explored the problem of scheduling a set of tasks with the least power. Speed scaling has also been combined with powering down in the same context of power-efficient task scheduling [23]. (The survey of Irani and Pruhs [22] reviews results and open problems under the speed scaling and powerdown models for processors.) In networks, most effort has been invested in reducing the consumption at the edge of the Internet (edge links and routers). For instance, Gunaratne *et al.* [20] have proposed a Markovian model to optimize single Ethernet link usage with speed scaling.

To the best of our knowledge, only a few papers study minimizing power consumption at a global wired network level. For instance, Nedevschi *et al.* [30] explore both speed scaling and power down as techniques to globally reduce power consumption. When using speed scaling, they consider two alternative models, one in which only the frequency of the transmission can be scaled and one in which also the operational voltage can be scaled. The authors propose heuristics for these models, which are evaluated empirically. While minimizing power consumption has always been a concern in wireless networks [25] (since mobile devices work on limited-energy batteries), they are intrinsically different from wireline networks and are not considered here.

II. Model and Results

We are given a network modeled by an undirected graph G and a set of demands. Each demand i requests d_i integer units of bandwidth between a source node s_i and a destination node t_i. We are also given a cost function $f_e(s)$ on each link[1] e that represents the power consumption for routing s units of demand through link e. As mentioned in the Introduction, our aim is to route all of the demands on integral routes with minimum cost. The routing problem is said to be NP-hard for most functions $f_e(\cdot)$. For example, even for a very restrictive case in which $f_e(\cdot)$ is binary, depending on whether link e carries nonzero traffic, the routing problem corresponds to the NP-hard problem of Steiner Forest in graphs [27]. We therefore consider approximation algorithms. A polynomial-time algorithm is a β-approximation if for all instances it returns a solution at most β times the optimal. A problem has no β-approximation if no polynomial-time algorithm can guarantee a β-approximation for all instances under complexity assumptions such as $\mathrm{P} \neq \mathrm{NP}$.

Formally, the *min-power routing problem* can be formulated as the following program. Let binary variable $y_{i,e}$ indicate whether demand i passes through link e, and x_e be the total load on e. Our route optimization problem can be formulated as follows:

$$(P_1) \quad \min \sum_e f_e(x_e)$$

subject to

$$x_e = \sum_i y_{i,e} d_i \ \forall e$$
$$y_{i,e} \in \{0, 1\} \ \forall i, e$$
$$y_{i,e} : \text{flow conservation.}$$

Let $I_i(v)$ and $O_i(v)$ be the amount of demand i entering and leaving node v, respectively, and $F_i(v) = O_i(v) - I_i(v)$. Flow conservation means that $F_i(s_i) = d_i$, $F_i(t_i) = -d_i$, and $F_i(v) = 0$ for any other node v. As mentioned in the Introduction, if the cost function $f_e(\cdot)$ is subadditive, then

[1]Cost functions for other network elements are left for future study.

this corresponds to the well-studied buy-at-bulk network design problem. The following summarizes the main results for buy-at-bulk.

Theorem 1 (BAB): For subadditive cost functions $f_e(\cdot)$, (P_1) has $O(\log N)^2$ approximation ratio [7] and $\Omega(\log^{1/4} N)$ hardness bound [6] if $f_e(\cdot)$ is uniform over all e; (P_1) has $O(\log^3 N)$ approximation ratio and $\Omega(\log^{1/2} N)$ hardness bound if $f_e(\cdot)$ can be different from edge to edge. Here, N is the size of the network.

In this paper, we are interested in less studied functions such as superadditive and mixed sub- and superadditive $f_e(\cdot)$. To provide a contrast with subadditive functions, we first show via a simple reduction that extremely simple superadditive functions, such as $f_e(x) = \max\{0, x-1\}$, lend to unbounded approximations.

Lemma 2: If a monotone function $f_e(\cdot)$ satisfies $f_e(1) = 0$ and $f_e(2) > 0$ for all e, then there is no polynomial-time algorithm to the min-power routing problem with any finite approximation ratio unless P = NP.

Proof: The reduction is from the edge-disjoint path (EDP) problem, which is known to be NP-hard. Given a network and a set of demands, EDP decides if all demands can be routed along edge-disjoint paths. If EDP has a solution, then the resulting load on each edge is at most 1, which implies a solution of cost 0 for the min-power problem. In contrast, if EDP has no solution, in any solution some link must have load at least 2, which implies an optimal min-power solution of cost at least $f(2) > 0$. Hence, a bound-approximation to the min-power problem would return a zero solution iff EDP has a solution. ■

We focus on two nonsubadditive functions in this paper both because they do not have the issue stated in Lemma 2 and because they closely model power consumption of certain network elements, as discussed earlier. We state the following main results.

- In Section III, we consider polynomial functions of the form $f_e(s) = \mu_e s^\alpha$. For uniform demands where d_i is the same for all i, we prove a γ-approximation where γ only depends on α. Since, as mentioned before, α is very small in practice (less than 2), we consider this to be a constant approximation. This result generalizes to an approximation that is logarithmic on $D = \max_i d_i$ for nonuniform demands.
- In Section IV, we consider polynomial functions with a startup cost, $f_e(s) = \begin{cases} 0, & \text{if } s = 0 \\ \sigma_e + \mu_e s^\alpha, & \text{if } s > 0. \end{cases}$ We present a simple $O(K)$-approximation for unit demands, where K is the number of demands. We also show an $O((\max_e\{\sigma_e/\mu_e\})^{1/\alpha} + 1)$-approximation, independent of K, for uniform demands. Again, for nonuniform demands, an additional factor logarithmic in D appears in the approximation ratios.
- In Section V, we evaluate our proposed approximation algorithms via simulations. For polynomials without a startup cost, randomized rounding performs superbly. When the startup is large, both approximations from Section IV are less than satisfactory. However, we present a heuristic that appears to rectify the situation.

[2]All logarithms are to the base 2.

III. APPROXIMATION FOR POLYNOMIAL COST FUNCTIONS

In this section, we use randomized rounding on the convex program (P_1) to approximate the optimal cost for polynomial cost functions $f_e(s) = \mu_e s^\alpha$.

We first relax the binary constraint on flow variables $y_{i,e} \in \{0,1\}$ to $y_{i,e} \in [0,1]$. As a result, for polynomial cost functions $f_e(\cdot)$, the routing problem is convex programming and is optimally solvable [10]. From the optimal fractional routing, we round the fractional flow in the Raghavan–Thompson manner [32] as follows. We first decompose the fractional solution defined by $y_{i,e}$ into weighted flow paths for each demand i via the following standard procedure. We repeatedly extract paths connecting the source and destination nodes of demand i from the subgraph defined by links e for which $y_{i,e} > 0$. If p is extracted, then the weight of p is $w_{i,p} = \min_{e \in p} y_{i,e}$ and the $y_{i,e}$ value of every link along p is reduced by $w_{i,p}$. The flow conservation constraint on $y_{i,e}$ guarantees that when the last path is extracted for demand i, every $y_{i,e}$ is zero. Following the flow decomposition, we randomly and independently choose one path from the potentially multiple paths for each demand i, using the path weights $w_{i,p}$ as probabilities. At the end of the rounding, every demand follows one single path. Observe that the event that an edge e is in the path chosen for demand i occurs with probability $\sum_{p:e \in p} w_{i,p} = y_{i,e}$. Additionally, this event is independent from the event of e being in the path chosen for any other demand $j \neq i$.

Obviously, the fractional optimal solution is a lower bound of the integral optimal solution. If we could bound the difference between the rounded solution and the fractional optimal, we would have bounded the difference between the rounded solution and the integral optimal. Unfortunately, the direct application of randomized rounding as described above does not guarantee a good approximation. For example, consider a network with two nodes u, v and m parallel links connecting them, one unit-demand with source u and destination v, and a uniform cost function $f_e(x) = x^\alpha$. The optimal fractional solution to (P_1) distributes the demand evenly among the m links, resulting in a cost of $m \cdot f_e(1/m) = 1/m^{\alpha-1}$. The optimal integral solution has to send the demand along one of the edges, resulting in cost $f_e(1) = 1$. Hence, the integrality gap is $m^{\alpha-1}$. However, we now show how to adapt this procedure in order to overcome this difficulty.

A. Uniform Demands

The essence of the previous example stems from the behavior of $f_e(\cdot)$ in the interval $[0,1]$. We observe that for unit demands in fact, for $x \in [0,1]$ we can use the cost function $f_e(x) = \mu x$ since μx and μx^α agree on $x = 0$ and $x = 1$. More importantly, if we do this, the integrality gap in the aforementioned example disappears. Formally, for unit demands, i.e., $d_i = 1$, we define the cost function

$$g_e(x) = \mu_e \max\{x, x^\alpha\}. \quad (1)$$

Note that minimizing $\sum_e g_e(x_e)$ has the same integral optimal as the original program (P_1) since $f_e(\cdot)$ and $g_e(\cdot)$ agree on all integral values. In addition, the optimal fractional solution with respect to $g_e(\cdot)$ can still be obtained by convex programming as $g_e(\cdot)$ is still convex after linearizing $f_e(\cdot)$ in the interval $[0,1]$. We use this observation to show that randomized rounding gives a constant factor approximation for unit demands. Let $x_e^* = \sum_i y_{i,e}$ be the flow on link e under the optimal fractional routing. Let $\hat{y}_{i,e}$ be an indicator variable for the event that e is in the path chosen for demand i after rounding (recall that $\hat{y}_{i,e} = 1$ with probability $y_{i,e}$). Finally, let $\hat{x}_e = \sum_i \hat{y}_{i,e}$ be the resulting rounded flow. We show the following.

Lemma 3: For unit demands, randomized rounding the optimal fractional solution x_e^* with respect to the cost function $g_e(x)$ guarantees that $E[g_e(\hat{x}_e)] = E[f_e(\hat{x}_e)] \le \gamma g_e(x_e^*)$, for some constant γ and all links e.

Proof: Observe that $E[\hat{x}_e] = x_e^*$. We consider two cases $x_e^* \le 1$ and $x_e^* > 1$.

Case 1) $x_e^* \le 1$. We show that $E[f_e(\hat{x}_e)] \le \gamma_1 g_e(x_e^*)$ for some constant γ_1. We partition the possible values of $\hat{x}_e$ into the ranges $[0,1), [1,2), [2,4), \ldots$. We have

$$\begin{aligned} E[f_e(\hat{x}_e)] &\le f_e(\hat{x}_e = 0) Pr[\hat{x}_e < 1] \\ &\quad + \sum_{j\ge 0} f_e(\hat{x}_e = 2^{j+1}) Pr[\hat{x}_e \ge 2^j] \\ &\le 0 + \sum_{j\ge 0} \mu_e (2^{j+1})^\alpha \left(\frac{e^{\frac{2^j}{x_e^*}-1}}{\left(\frac{2^j}{x_e^*}\right)^{\frac{2^j}{x_e^*}}} \right)^{x_e^*} \\ &= \mu_e x_e^* \sum_{j\ge 0} \frac{2^{\alpha(j+1)}}{x_e^*} \frac{x_e^{*2^j}}{e^{x_e^*}} \left(\frac{e}{2^j}\right)^{2^j} \\ &\le g_e(x_e^*) \sum_{j\ge 0} 2^{\alpha(j+1)-(j-\lg e)2^j}. \end{aligned} \tag{2}$$

The first inequality follows from the definition of expectation. The second follows from a Chernoff bound [33, Theorem 3.35], [29, Theorem 4.4(1)]. We obtain the third inequality via algebraic manipulation and the fact that $0 \le x_e^* \le 1$. Let $j_0 = \lceil 2\lg(\alpha+4) \rceil$. Via further algebraic manipulation, we can show that all the terms in (2) for which $j \ge j_0$ add up to at most 1. For $j \in [0, j_0]$, the term $-(j - \lg e)2^j \le \lg e$, while the term $\alpha(j+1) \le \alpha(j_0+1)$. Hence, there is a constant $\gamma_1 \le 1 + j_0 2^{\alpha(j_0+1)+\lg e}$ such that $E[f_e(\hat{x}_e)] \le \gamma_1 g_e(x_e^*)$.

Case 2) $x_e^* > 1$. We partition the possible values of x_e into the ranges $[0, x_e^*), [x_e^*, 2x_e^*), [2x_e^*, 4x_e^*), \ldots$. By the definition of expectation, we have

$$\begin{aligned} E[f_e(\hat{x}_e)] &\le f_e(\hat{x}_e = x_e^*) Pr[\hat{x}_e \ge 0] \\ &\quad + \sum_{j\ge 0} f_e(\hat{x}_e = 2^{j+1} x_e^*) Pr[\hat{x}_e \ge 2^j x_e^*] \\ &\le g_e(x_e^*) + \sum_{j\ge 0} 2^{\alpha(j+1)} g_e(x_e^*) \left(\frac{e^{2^j - 1}}{(2^j)^{2^j}} \right)^{x_e^*} \\ &\le g_e(x_e^*) \left(1 + \sum_{j\ge 0} 2^{\alpha(j+1)-(j-\lg e)2^j - \lg e} \right) \end{aligned}$$

where the second inequality follows from a Chernoff bound [33, Theorem 3.35], [29, Theorem 4.4] (1), and the third inequality follows from $x_e^* > 1$. The summation for $j \ge 2$ can be bounded similar to the one in (2). Hence, there is a constant $\gamma_2 \le 2 + j_0 2^{\alpha(j_0+1)}$ such that $E[f_e(\hat{x}_e)] \le \gamma_2 g_e(x_e^*)$. Combining both cases, we have that, for $\gamma = \max\{\gamma_1, \gamma_2\} > 0$, every link e satisfies that $E[f_e(\hat{x}_e)] \le \gamma g_e(x_e^*)$. ■

It is easy to see that Lemma 3 for unit demands also applies to uniform demands in which $d_i = d$ for all demands i. By linearizing $f_e(\cdot)$ in the range of $[0,d]$ instead of $[0,1]$, we define

$$g_e(x) = \mu_e \max\{d^{\alpha-1}x, x^\alpha\}. \tag{3}$$

By randomized rounding the fractional optimal solution with respect to $g_e(\cdot)$, we can easily derive the following parallel to Lemma 3.

Corollary 4: For uniform demands, $E[f_e(\hat{x}_e)] \le \gamma g_e(x_e^*)$ for all e where $g_e(\cdot)$ is defined in (3).

The previous results only examine the expected value of the solution. We now show how to convert this into a result that holds with high probability.

Theorem 5: For uniform demands in which all d_i are equal, randomized rounding guarantees a γ-approximation in the expected value of the total power cost, where γ is the constant in Lemma 3. Furthermore, for any constant c, randomized rounding guarantees a $c\gamma$-approximation with probability at least $1 - 1/c$.

Proof: The expected total cost after randomized rounding is $E[\sum_e f_e(\hat{x}_e)] = \sum_e E[f_e(\hat{x}_e)] \le \gamma \sum_e g_e(x_e^*) \le \gamma \mathrm{Opt}$, where Opt is the integral solution to (P_1). By Markov's inequality, the probability that a rounded solution is more than $c\gamma \cdot \mathrm{Opt}$ is upper-bounded by $1/c$. ■

B. Nonuniform Demands

We now prove an $O(\log^{\alpha-1} D)$-approximation for nonuniform demands with cost function $f_e(x) = \mu_e x^\alpha$, where $D = \max_i d_i$ is the maximum demand bandwidth. Note that in the application of interest, $f_e(\cdot)$ represents the speed–power curve for which the typical value of α is under 3. Hence, the logarithmic approximation ratio has a small exponent.

Theorem 6: For nonuniform demands, randomized rounding can be used to achieve a $O(\log^{\alpha-1} D)$-approximation, where $D = \max_i d_i$.

Proof: We partition the demands into $\log D$ groups, where group $j \ge 0$ consists of demands whose d_i is in the range of $[2^j, 2^{j+1})$. We treat each group separately. For group j, we assume each demand requests bandwidth of exactly 2^{j+1} and invoke the randomized rounding algorithm for those demands. Let $x_e^{(j)*}$ be the load on link e due to the optimal fractional solution, and let $\hat{x}_e^{(j)}$ be the load after the rounding. Both $x_e^{(j)*}$ and $\hat{x}_e^{(j)}$ are calculated with respect to demand bandwidth rounded up to 2^{j+1}. Let $\mathrm{Opt}^{(j)}$ be the optimal solution with respect to demands in group j, and Opt be the optimal solution with respect to demands in all groups. Note that both $\mathrm{Opt}^{(j)}$ and Opt are with respect to actual demand bandwidth.

We have

$$\begin{aligned}\sum_e E\left[f_e(\sum_j \hat{x}_e^{(j)})\right] &\leq \sum_e E\left[(\log D)^{\alpha-1}\sum_j f_e(\hat{x}_e^{(j)})\right]\\ &\leq \sum_e (\log D)^{\alpha-1}\gamma\sum_j g_e^{(j)}(x_e^{(j)*})\\ &\leq (\log D)^{\alpha-1}\gamma\sum_j 2^\alpha \mathrm{Opt}^{(j)}\\ &\leq (\log D)^{\alpha-1}\gamma 2^\alpha \mathrm{Opt}.\end{aligned}$$

The first inequality is due to the convexity of $f_e(x) = \mu_e x^\alpha$, namely $f(\sum_{j=0}^{\log D} x_e^{(j)}) \leq (\log D)^{\alpha-1}\sum_j f(x_e^{(j)})$. In the second inequality $g_e^{(j)}(\cdot)$ refers to (3) for $d = 2^{j+1}$, the linearization of the function $f_e(\cdot)$ for demand bandwidth 2^{j+1}. The second inequality is due to Corollary 4. The third inequality holds since, as before, $\sum_e g_e(x_e^*)$ is a lower bound on the optimal integral solution and each demand bandwidth has been rounded up by a factor of at most 2. The last inequality holds due to the superadditive nature of $f_e(\cdot)$.

Finally, we note that $\sum_e E\left[f(\sum_j \hat{x}_e^{(j)})\right]$ upper-bounds the expected total cost since $\hat{x}_e^{(j)}$ is calculated based on bandwidth that is rounded up. This completes the proof. ■

IV. Polynomial Functions With Startup Cost

We now turn our attention to speed–power curves that are polynomials with a startup cost. These functions have the form $f_e(x) = \begin{cases} 0, & \text{if } x = 0 \\ \sigma_e + \mu_e x^\alpha, & \text{if } x > 0. \end{cases}$ Note that for $\alpha \leq 1$, such a function is concave, and Theorem 1 summarizes its approximability. When $\alpha > 1$, the function is neither convex nor concave, and therefore convex programming cannot obtain an optimal fractional solution to (P_1).

In the following, we provide two approximations. The first one is based on rounding a newly formed convex program defined in (P_2). The resulting approximation ratio depends on the number of demands. The second one replaces the neither convex nor concave function $f_e(\cdot)$ with a convex function $h_e(\cdot)$ that "resembles" $f_e(\cdot)$, and then uses randomized rounding on the problem (P_1) with objective function $h_e(\cdot)$. The resulting ratio depends on the parameters σ_e and μ_e.

A. Approximation With Respect to the Number of Demands

The following is a natural formulation that handles polynomial functions with a startup cost:

$$(P_2)\ \min \quad \sum_e \sigma_e z_e + g_e(x_e)$$

subject to

$$\begin{aligned} x_e &= \sum_i y_{i,e}\cdot d_i\\ y_{i,e} &\leq z_e\\ y_{i,e}, z_e &\in \{0,1\}\\ y_{i,e} &: \text{flow conservation}\end{aligned}$$

where in the integer formulation z_e represents whether or not the startup cost on link e is paid for, i.e., whether or not we route any demand on it. The second constraint enforces the condition that we cannot route any demand on a link unless its startup cost is paid for. In the objective function, $g_e(x_e)$ is a linearization of μx^α as in Section III. For example, $g_e(x_e)$ is defined as in (3) for uniform demands. Again, the optimal integral solution to (P_2) is the same as to the objective of minimizing $\sum_e \sigma_e z_e + f_e(x_e)$, and its continuous relaxation is convex.

Theorem 7: For uniform demands, randomized rounding of the optimal fractional solution to (P_2) guarantees a $(K+\gamma)$-approximation to the optimal integral solution in expectation, where K is the number of demands and γ is the constant in Corollary 4.

Proof: Let z^*, y^*, and x^* be the optimal fractional solution, and let $\hat{z}, \hat{y}$, and $\hat{x}$ be the solutions that we get from the rounding. From Lemma 3, we have that $E[g_e(\hat{x}_e)] \leq \gamma g_e(x_e^*)$ for some constant γ. It remains to relate $\sum_e \sigma_e z_e^*$ and $\sum_e \sigma_e \hat{z}_e$. We have

$$\begin{aligned} E[\hat{z}_e] &= Pr(\hat{z}_e = 1) = 1 - Pr(\hat{y}_{i,e} = 0 \text{ for all } i)\\ &= 1 - \Pi_i(1 - y_{i,e}^*) \leq \sum_i y_{i,e}^* \leq K z_e^*.\end{aligned}$$

The last inequality comes from the fact that in the fractional solution, each $y_{i,e}^*$ is constrained to be at most z_e^*. Putting everything together, we have that the rounded solution has expected value at most $(K+\gamma)$ times higher than the optimal solution. ■

We remark that this ratio can be better than the naive ratio that would be obtained by simply routing each demand along the minimum-hop path since that solution could route all demands along a single edge whereas the optimum solution might route all demands along separate edges. Using this fact, it is easy to construct examples where the cost of minimum hop is a factor $O(K^{\alpha-1})$ away from optimal.

The theorem above can be generalized to nonuniform demands. Combining the analysis for Theorems 6 and 7, we have the following.

Theorem 8: For nonuniform demands, randomized rounding can be used to achieve a $O(K + \log^{\alpha-1} D)$-approximation.

B. Approximation With Respect to σ_e and μ_e

For the second approximation, we use a convex function $h_e(\cdot)$ in place of $f_e(\cdot)$, and randomized rounding on the optimal fractional solution with respect to $h_e(\cdot)$. To obtain an approximation ratio, we need to bound the difference between fractional and integral solutions, and the difference between $h_e(\cdot)$ and $f_e(\cdot)$. It is intuitive that we would like $h_e(\cdot)$ to be close to $f_e(\cdot)$.

The function $h_e(\cdot)$ starts with a line through the origin and switches to be $f_e(\cdot)$ at some point. Let $(s_e, f_e(s_e))$ be the point at which the line tangent to the curve $f_e(\cdot)$ goes through the origin. If $s_e \leq 1$, then $h_e(\cdot)$ begins with the straight line through origin to $(1, f_e(1))$ and continues on $f_e(\cdot)$, as shown in Fig. 1 (left). Otherwise, $h_e(\cdot)$ begins with the tangent line up to

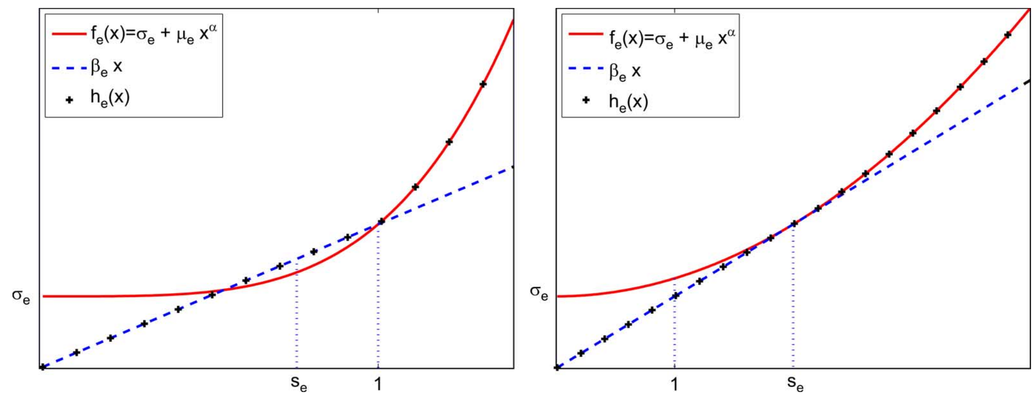


Fig. 1. $f_e(\cdot)$ and its approximation $h_e(\cdot)$. *(left)* $s_e \le 1$. *(right)* $s_e > 1$.

the tangent point $(s_e, f_e(s_e))$ and continues on $f_e(\cdot)$, as shown in Fig. 1 (right).

More formally, let $s_e = (\sigma_e/((\alpha-1)\mu_e))^{1/\alpha}$. We define the parameter β_e and the function $h_e(x)$, for each edge e, as follows:

$$\beta_e = \begin{cases} \sigma_e + \mu_e, & \text{if } s_e < 1 \\ \alpha\mu_e(\sigma_e/((\alpha-1)\mu_e))^{1-1/\alpha}, & \text{if } s_e \ge 1 \end{cases}$$
$$h_e(x) = \begin{cases} \beta_e x, & \text{if } x \in [0, \max(1, s_e)) \\ \sigma_e + \mu_e x^\alpha, & \text{if } x \ge \max(1, s_e). \end{cases}$$

It can be observed that the function $h_e(x)$ is continuous, convex, and satisfies $h_e(x) \le f_e(x)$, for all integral $x \ge 0$.

Theorem 9: For unit demands, applying randomized rounding to the fractional solution obtained from the convex program (P_1) minimizing $\sum_e h_e(x_e)$ guarantees $E[f_e(\hat{x}_e)] \le O(1 + (\sigma_e/\mu_e)^{1/\alpha}) \cdot h_e(x_e^*)$, for each link e.

Proof: Let us fix an edge e. We break the proof in three cases, $x_e^* \le 1$, $x_e^* \ge \max(1, s_e)$, and $x_e^* \in (1, s_e)$.

Case 1) $x_e^* \le 1$. We partition the possible values of $\hat{x}_e$ into the ranges $[0,1), [1,2), [2,4), \ldots$. By the definition of expectation, we have

$$\begin{aligned} E[f_e(\hat{x}_e)] &\le f_e(\hat{x}_e = 0)\Pr[\hat{x}_e < 1] \\ &\quad + \sum_{j\ge 0} f_e(\hat{x}_e = 2^{j+1})\Pr[\hat{x}_e \ge 2^j] \\ &\le 0 + \sum_{j\ge 0}(\sigma_e + \mu_e(2^{j+1})^\alpha)\Pr[\hat{x}_e \ge 2^j] \\ &\le \beta_e x_e^* \frac{\sigma_e + \mu_e}{\beta_e} \sum_{j \ge 0} \frac{2^{\alpha(j+1)}}{x_e^*} \frac{x_e^{*2^j}}{e^{x_e^*}} \left(\frac{e}{2^j}\right)^{2^j} \end{aligned}$$

where the last inequality follows from a Chernoff bound. The sum was shown in the proof of Lemma 3 to be bounded by a constant γ_1. If $s_e < 1$, then $\beta_e = \sigma_e + \mu_e$ and hence $\frac{\sigma_e+\mu_e}{\beta_e} = 1$. Otherwise, $s_e \ge 1$, and then $\beta_e = \Theta(\mu_e^{1/\alpha}\sigma_e^{1-1/\alpha})$. Since $s_e = \Theta((\sigma_e/\mu_e)^{1/\alpha})$, then $\mu_e/\sigma_e = O(1)$. In either case, we get

$$\frac{\sigma_e + \mu_e}{\beta_e} = O(1 + (\sigma_e/\mu_e)^{1/\alpha}) \tag{4}$$

and then $E[f_e(\hat{x}_e)] \le O(1 + (\sigma_e/\mu_e)^{1/\alpha}) \cdot h_e(x_e^*)$.

Case 2) $x_e^* \ge \max(1, s_e)$. In this case, we have $E[f_e(\hat{x}_e)] \le \gamma_2 \cdot h_e(x_e^*)$, from a proof identical to case 2 in Lemma 3.

Case 3) $x_e^* \in (1, s_e)$. Note that this case can only occur if $s_e \ge 1$. We partition the possible values of $\hat{x}_e$ into the ranges $[0, x_e^*), [x_e^*, 2x_e^*), [2x_e^*, 4x_e^*), \ldots$. By the definition of expectation, we have

$$\begin{aligned} &E[f_e(\hat{x}_e)] \\ &\le f_e(\hat{x}_e = x_e^*)\Pr[\hat{x}_e \ge 0] \\ &\quad + \sum_{j\ge 0} f_e(\hat{x}_e = 2^{j+1}x_e^*)\Pr[\hat{x}_e \ge 2^j x_e^*] \\ &\le \beta_e x_e^* \Bigg(\frac{\sigma_e + \mu_e (x_e^*)^\alpha}{\beta_e x_e^*} \\ &\qquad + \sum_{j \ge 0} \frac{\sigma_e + \mu_e(2^{j+1}x_e^*)^\alpha}{\beta_e x_e^*} \Pr[\hat{x}_e \ge 2^j x_e^*] \Bigg) \\ &\le h_e(x_e^*) \frac{\sigma_e + \mu_e}{\beta_e} \cdot \left(1 + \sum_{j\ge 0} 2^{\alpha(j+1)}\Pr[\hat{x}_e \ge 2^j x_e^*]\right) \end{aligned}$$

where the third inequality follows from the fact that $\frac{\sigma_e + \mu_e x^\alpha}{\beta_e x}$ is nonincreasing for $x \in (1, s_e)$. From (4), $\frac{\sigma_e+\mu_e}{\beta_e} = O(1 + (\sigma_e/\mu_e)^{1/\alpha})$, and the other factor of $h_e(x_e^*)$ was shown in the proof of Lemma 3 to be bounded by a constant γ_2. Therefore, $E[f_e(\hat{x}_e)] \le O(1 + (\sigma_e/\mu_e)^{1/\alpha}) \cdot h_e(x_e^*)$.

■

The approximation above also applies to uniform demands. Similar to Theorem 6, we also have the following for nonuniform demands.

Theorem 10: For polynomial functions with startup costs, randomized rounding can be used to achieve a $O(1 + (\max_e\{\sigma_e/\mu_e\})^{1/\alpha})(\log D)^{\alpha-1}$-approximation, where $D = \max_i d_i$.

Proof: The proof is verbatim to that of Theorem 6 with a small difference in the last step, as follows:

$$\begin{aligned} E\left[\sum_e f(\hat{x}_e)\right] &\le O(1 + (\frac{\sigma_e}{\mu_e})^{1/\alpha})(\log D)^{\alpha-1}\sum_j 2^\alpha \mathrm{Opt}^{(j)} \\ &\le (1 + (\frac{\sigma_e}{\mu_e})^{1/\alpha})(2\log D)^\alpha \mathrm{Opt}. \end{aligned}$$

The last inequality follows from the fact that $\sum_{j=1}^{\log D} \mathrm{Opt}^{(j)} \leq \mathrm{Opt} \cdot \log D$. ■

Remarks: The results of the previous section work well when σ_e or K are small, but give less good bounds when these parameters are large. Recall from Theorem 1 that we have a range of techniques that guarantee a polylogarithmic approximation for the buy-at-bulk problem (i.e., the problem where the cost functions are subadditive, e.g., when $\alpha \leq 1$). Unfortunately, these techniques cannot produce approximation ratios better than polynomial in the network size when $\alpha > 1$. A detailed discussion of why these techniques fail can be found in [4]. In addition, the same paper also shows an approximation polylogarithmic in the size of the input. That is, the approximation ratio is independent of the number of inputs and the startup cost. We make two comments on this polylogarithmic approximation. First, polylogarithmic approximation is the best one can hope for. The hardness result of Theorem 1 generalizes to the case in which $\alpha > 0$. More specifically, we have the following.

Theorem 11: For any $\alpha > 1$, there is a uniform polynomial cost function with startup cost such that no algorithm can guarantee a $\log^{1/4} N$-approximation unless $\mathrm{NP} \subseteq \mathrm{ZPTIME}(n^{\mathrm{polylog} n})$.

The proof of this result is complexity-theory specific and is beyond the scope of this paper. Second, the algorithm of [4] is quite sophisticated, using techniques such as graph expansions. Therefore, the randomized rounding approach in this paper has the advantage of simplicity.

V. Experimental Results

In this section, we provide the detailed experimental findings. We associate cost functions like those previously presented to the links of real networks, implement the approximation algorithms presented in Sections III and IV, and compare the approximate solutions against both the optimal and the straightforward shortest-paths solutions. The reason for comparing against the latter is to show that routing without energy in mind can be wasteful. As we shall see, we observe a consistent savings of 10% or more over shortest-paths. This gives initial evidence that a nonnegligible percentage of power saving could come from global network planning such as routing.

We obtain the optimal integral solutions by solving the relevant integer programs using CPLEX solver [1]. For our approximation algorithms, we use the CVX solver [2] to obtain the optimal fractional solutions before applying randomized rounding. Most of our experiments are conducted on the Abilene Research network, which consists of 10 nodes and 13 links, and the NSF Network, which consists of 14 nodes and 20 links. See Fig. 2. We also test scalability on larger networks.

A. Polynomial Cost Function Without Startup: $f_e(x) = \mu_e X^\alpha$

We use a quadratic function $f_e(x) = x^2$ for our experiments. For each network, we perform the routing algorithm with different number of demands, where the number ranges from twice the number of nodes to six times the number of nodes. The source and sink nodes of each demand are chosen uniformly at random. We concentrate on unit demands. For each routing instance, we compare four values of interest: the optimal integral solution from CPLEX, the optimal fractional solution from CVX, the rounded integral solution, and the short-paths solution. The four curves in Fig. 3(a) and (b) correspond to the ratio of these four values all normalized by the optimal integral. We observe the following.

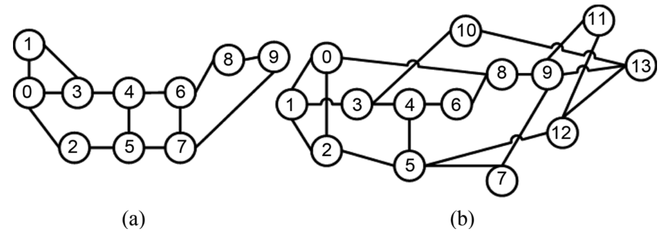


Fig. 2. (a) Abilene Research network. (b) NSF network.

1) The optimal fractional values are very close to the integral optimal. The difference is at most 0.84% in the Abilene Research network and at most 1.2% in the NSF network. This suggests that the optimal fractional solution (which is polynomially obtainable) can be a good lower bound in the absence of the optimal integral solution (which is NP-hard).
2) The randomized rounding solutions are within 4% of the integral optimal in the Abilene Research network and within 0.5% in the NSF network. This suggests that randomized rounding performs even better in practice than the approximation ratio analyzed in Lemma 3.
3) The randomized rounding solutions are consistently at least 10% better than the shortest-path solutions.

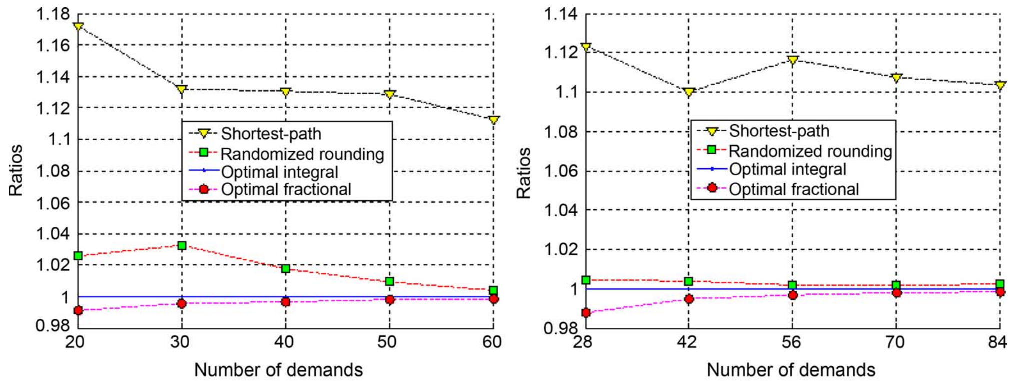
To explain these findings, we examine the *link load*, which is the total demand flow going through e. We observe that the maximum link load as a result of the integral optimal, fractional optimal, and randomized rounding are quite close to one another. However, the maximum load of the shortest-path solutions is often significantly higher, as shortest-path routing does not intend to balance the link load and therefore incurs high cost.

B. Polynomial Function With Startup: $f_e(x) = \sigma_e + \mu_e X^\alpha$

We use $f_e(x) = \sigma + x^2$ as the cost function, where $\sigma \in \{4, 16, 64, 256, 1024\}$.[3] Again, the number of demands varies from twice the number of nodes to six times the number of nodes. We compare a number of routing strategies here. Two of the strategies correspond to the $O(K)$-approximation and $O(\sqrt{\sigma})$-approximation, as shown in Theorems 7 and 9. Not surprisingly the $O(\sqrt{\sigma})$-approximation performs poorly for large σ, as the approximate function $h_e(\cdot)$ deviates significantly from $f_e(\cdot)$ for large σ. The $O(K)$-approximation is also less than satisfactory for large σ. The difficulty for the large startup cost is that a large σ encourages aggregating traffic to minimize the number of *active* links, namely those carrying nonzero traffic. On the other hand, the convex nature of x^2 encourages load-balancing traffic to avoid paying quadratic cost on high loads. The balance between these contradicting objectives is challenging.

We offer a heuristic Greedy_ActiveLinks (see Fig. 4) that helps to shrink the set of active links. Initially, we assume every

[3]Exponentially increasing values of σ are used to evaluate the impact of this parameter.



(a) (b)

Fig. 3. From top to bottom, values due to shortest-paths, randomized rounding, integral optimal, and fractional optimal, all normalized by the integral optimal. (a) Abilene Research network. (b) NSF network.

Greedy_ActiveLinks($G = (V, E)$):
Let $S \leftarrow \min \sum_{e \in E} x_e^2$
  via randomized rounding on $G(V, E)$
Let $E' \leftarrow E$, $E'' \leftarrow \emptyset$, and $C \leftarrow S + \sigma|E|$
while (true):
  For $e \in E'$ begin:
    1. $E'' \leftarrow E' \setminus \{e\}$
    2. Let $S \leftarrow \min \sum_{e \in E''} x_e^2$
      via randomized rounding on $G(V, E'')$
    3. If $S + \sigma|E''| < C$,
      then $E' \leftarrow E'', C \leftarrow S + \sigma|E''|$ and break
  end
  If no improvement for all $e \in E'$, break
end

Fig. 4. Pseudocode for the Greedy_ActiveLinks heuristic.

link is active, namely the active link set $E' = E$. We minimize $\sum_{e \in E'} x_e^2$ to a value, say S. (We know from the previous findings that randomized rounding performs extremely well for $\min \sum_{e \in E'} x_e^2$.) The total cost $\sum_{e \in E} f_e(x_e)$ is therefore $S + \sigma|E'|$. Note that the routes may not use every link in E, but we nevertheless pay σ for all. During each subsequent iteration, we aim to remove one link from the active set so that the total cost is reduced. Again, for the current active set E', we minimize $\sum_{e \in E'} x_e^2$. The process stops when we can no longer shrink the active set, either due to disconnectivity or increasing cost.

We refer to the solution from the above Greedy_ActiveLinks heuristic as *Greedy_RR*, where RR refers to randomized rounding. From Table I, we observe that Greedy_RR improves a great deal over the $O(K)$- and $O(\sqrt{\sigma})$-approximations for large σ, whereas for small σ, the $O(K)$- and $O(\sqrt{\sigma})$-approximations continue to have an advantage. We therefore have a *Combined* strategy, which is to run all three strategies for each instance and keep the best.

Again, we compare against the shortest-paths solution (*SP*). Like the case in which $\sigma = 0$, shortest-paths is worse off than the $O(K)$- and $O(\sqrt{\sigma})$-approximation for small σ. For large σ, all three strategies perform poorly. If we combine the greedy heuristic combined with shortest-paths, specifically by running shortest paths in line 2 of Greedy_ActiveLinks, the improvement of the resulting solution *Greedy_SP* is less than satisfactory.

TABLE I
NSF NETWORK

NSF Network		Startup cost σ				
Solutions	dem.	4	16	64	256	1024
SP	28	1.107	1.053	1.137	1.314	1.389
$Greedy\_SP$	28	1.107	1.053	1.123	1.259	1.247
K-approx	28	1.017	1.120	1.270	1.434	1.516
$\sqrt{\sigma}$-approx	28	1.005	1.022	1.142	1.292	1.382
$Greedy\_RR$	28	1.129	1.071	1.071	1.133	1.099
$Combined$	28	1.005	1.022	1.071	1.133	1.099
SP	56	1.091	1.074	1.046	1.152	1.380
$Greedy\_SP$	56	1.091	1.074	1.046	1.130	1.252
K-approx	56	1.001	1.004	1.143	1.271	1.479
$\sqrt{\sigma}$-approx	56	1.001	1.005	1.020	1.146	1.371
$Greedy\_RR$	56	1.167	1.124	1.056	1.064	1.162
$Combined$	56	1.001	1.004	1.020	1.064	1.162
SP	84	1.077	1.065	1.038	1.103	1.249
$Greedy\_SP$	84	1.077	1.064	1.038	1.103	1.212
K-approx	84	1.001	1.001	1.012	1.212	1.374
$\sqrt{\sigma}$-approx	84	1.002	1.002	1.003	1.084	1.260
$Greedy\_RR$	84	1.175	1.141	1.074	1.068	1.091
$Combined$	84	1.001	1.001	1.003	1.068	1.091

Due to space consideration, we only present numbers for the NSF network in Table I. Again, all the values are normalized by the optimal integral values.

C. Running Time and Larger Networks

The average running time for the CVX solver is around 2–3 s for obtaining optimal fractional solutions to all the instances presented so far. The CPLEX solver is also fast for obtaining the integral optimal to all the instances with small startup values, namely $\sigma = 0, 4, \ldots, 64$. The running times vary from 30 s to 3 min. However, for larger startup cost $\sigma = 256, 1024$, CPLEX takes significantly longer. For example when $\sigma = 1024$ and the number of demand pairs is six times the number of nodes, it took CPLEX longer than 17 h to get a solution with relative error within 2.1% on the NSF network. For larger networks with at least 25 nodes, CPLEX has trouble even for $\sigma = 0$.

We repeated our experiments on random sparse networks with 100 nodes and expected node degree of 4. Although we cannot obtain optimal integral solutions, our findings of the performance of other algorithms and heuristics are consistent with our findings on the Abilene Research network and the NSF network.

VI. Conclusion

In this paper, we consider a min-cost integer routing problem where the cost function represents the speed–power curve of a network element. Subadditive cost functions are well studied. We focus on the less-studied polynomial functions and polynomials with a startup cost. The problem is interesting for two reasons. First, the cost function closely models the power consumption of some network elements, and network-wide optimization is a well-motivated but underexplored direction for power minimization. Second, it brings light to a challenging combinatorial optimization problem. We have presented approximation for polynomial functions and polynomial functions with startup cost, although the approximation ratios depend on the startup cost or the size of demands.

Since the publication of the first version of this paper in IEEE INFOCOM 2010, a new approximation algorithm that guarantees polylogarithmic approximation for the case of nonzero startup cost has been developed [4]. This answers one open question left by this paper. However, several interesting open questions are left open by this work. For instance, our algorithms only attempt to optimize the power consumed in the network, while it would be natural to explore algorithms that attempt to optimize performance as a function of several additional parameters, like latency or throughput. In this paper, we have focused on two nonsubadditive functions. A very interesting problem is considering functions that have both subadditive and superadditive components. This appears to be a difficult problem that we leave for future work. Another question that is left open is the impact of limiting the links capacity in the presented results.

References

[1] "IBM ILOG CPLEX optimizer," IBM, Armonk, NY [Online]. Available: http://www.ilog.com/products/cplex

[2] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming," 2011 [Online]. Available: http://www.stanford.edu/~boyd/cvx/

[3] *Proc. Vision Roadmap Workshop Routing Telecom Data Centers Toward Efficient Energy Use* Sunnyvale, CA, Oct. 2008 [Online]. Available: http://www1.eere.energy.gov/industry/datacenters/pdfs/vision_and_roadmap.pdf

[4] M. Andrews, S. Antonakopoulos, and L. Zhang, "Minimum-cost network design with (dis)economies of scale," in *Proc. 51st Annu. Symp. Found. Comput. Sci.*, Las Vegas, NV, Oct. 2010, pp. 585–592.

[5] M. Andrews, A. F. Anta, L. Zhang, and W. Zhao, "Routing and scheduling for energy and delay minimization in the powerdown model," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.

[6] M. Andrews, "Hardness of buy-at-bulk network design," in *Proc. 45th Annu. Symp. Found. Comput. Sci.*, Rome, Italy, Oct. 2004, pp. 115–124.

[7] B. Awerbuch and Y. Azar, "Buy-at-bulk network design," in *Proc. 38th Annu. Symp. Found. Comput. Sci.*, Miami Beach, FL, Oct. 1997, pp. 542–547.

[8] N. Bansal, T. Kimbrel, and K. Pruhs, "Speed scaling to manage energy and temperature," *J. ACM*, vol. 54, no. 1, 2007, Article no. 3.

[9] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[10] S. Boyd and L. Vandenberghe*, Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[11] 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, Nov.–Dec. 2000.

[12] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proc. 27th IEEE INFOCOM*, Apr. 2008, pp. 457–465.

[13] C. Ho-Leung, C. Wun-Tat, W. L. Tak, L. Lap-Kei, M. Kin-Sum, and W. H. W. Prudence, "Energy efficient online deadline scheduling," in *Proc. ACM–SIAM SODA*, 2007, pp. 795–804.

[14] C. Moses and K. Adriana, "On non-uniform multicommodity buy-at-bulk network design," in *Proc. ACM STOC*, 2005, pp. 176–182.

[15] C. Chandra, T. H. Mohammad, K. Guy, and R. S. Mohammad, "Approximation algorithms for non-uniform buy-at-bulk network design," in *Proc. IEEE FOCS*, 2006, pp. 677–686.

[16] "Enhanced Intel SpeedStep technology for the Intel Pentium M processor," Intel, Mountain View, CA, Intel White Paper 301170–001, 2004.

[17] "IEEE P802.3az energy efficient ethernet. Task force public area," 2008 [Online]. Available: http://grouper.ieee.org/groups/802/3/az/public/index.html

[18] M. Garrett, "Powering down," *Commun. ACM*, vol. 51, no. 9, pp. 42–46, 2008.

[19] G. George, "Low-power modes for adsl2 and adsl2+," Broadband Communications Group, Texas Instruments, Dallas, TX, White Paper, 2005.

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

[21] I. Sandy, K. G. Rajesh, and K. S. Sandeep, "Competitive analysis of dynamic power management strategies for systems with multiple power savings states," in *Proc. IEEE DATE*, 2002, pp. 117–123.

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

[23] S. Irani, S. K. Shukla, and R. Gupta, "Algorithms for power savings," *Trans. Algor.*, vol. 3, no. 4, 2007, Article no. 41.

[24] S. Irani, S. K. Shukla, and R. K. Gupta, "Online strategies for dynamic power management in systems with multiple power-saving states," *Trans. Embedded Comput. Syst.*, vol. 2, no. 3, pp. 325–346, 2003.

[25] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J.-C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Netw.*, vol. 7, no. 4, pp. 343–358, 2001.

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

[27] C.-L. Li, S. T. McCormick, and D. Simchi-Levi, "The point-to-point delivery and connection problems: Complexity and algorithms," *Discr. Appl. Math.*, vol. 36, no. 3, pp. 267–292, 1992.

[28] M. Li, J. L. Becky, and F. Y. Frances, "Min-energy voltage allocation for tree-structured tasks," in *Proc. COCOON*, L. Wang, Ed., 2005, vol. 3595, Lecture Notes in Computer Science, pp. 283–296.

[29] M. Michael and U. Eli*, Probability and Computing : Randomized Algorithms and Probabilistic Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[30] N. Sergiu, P. Lucian, I. Gianluca, R. Sylvia, and W. David, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proc. NSDI*, 2008, pp. 323–336.

[31] P. Patel-Predd, "Energy-efficient ethernet," *IEEE Spectrum*, vol. 45, no. 5, p. 13, May 2008.

[32] P. Raghavan and C. D. Thompson, "Randomized rounding: A technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, pp. 365–374, 1991.

[33] C. Scheideler, "Probabilistic methods for coordination problems," Habilitation thesis, Paderborn University, Paderborn, Germany, 2000.

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

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

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

Matthew Andrews received the B.A. degree (First Class Honours) in mathematics from Oxford University, Oxford, U.K., in 1993, and the Ph.D. degree in theoretical computer science from the Massachusetts Institute of Technology, Cambridge, in 1997.

He is a Distinguished Member of Technical Staff with the Mathematics of Networks and Systems Department, Bell Labs, Murray Hill, NJ. He holds a range of patents in the area of telecommunications. His research interests include wireless resource allocation, packet scheduling, and approximation algorithms. His recent work includes algorithms for joint scheduling and

congestion control in ad hoc networks and complexity theoretic results on the hardness of network design.

Antonio Fernández Anta (SM'02) graduated in computer science from the Universidad Politécnica de Madrid, Madrid, Spain, in 1991, and received the Ph.D. degree in computer science from the University of Louisiana at Lafayette in 1994.

He is a Senior Researcher with the Institute IMDEA Networks, Leganés, Spain, on leave from his position of Professor with the Universidad Rey Juan Carlos, Móstoles, Spain. Previously, he was a member of the faculty with the Universidad Politécnica de Madrid, Madrid, Spain. He was a Postdoctoral Researcher with the Massachusetts Institute of Technology, Cambridge, from 1995 to 1997. His research interests include data communications, computer networks, distributed processing, algorithms, and discrete and applied mathematics.

Dr. Fernández Anta is a Senior Member of the Association for Computing Machinery (ACM).

Lisa Zhang received the B.A. degree (*summa cum laude*) in mathematics from Wellesley College, Wellesley, MA, in 1993, and the Ph.D. degree in theory of computing from the Massachusetts Institute of Technology, Cambridge, in 1997.

She is a Member of Technical Staff with the Algorithms Research Group, Bell Labs, Murray Hill, NJ. She holds a range of patents in the area of telecommunications. Her research area is algorithm design and analysis. Her research broadly concerns algorithmic and complexity issues of networking, with a focus on design and optimization, routing and scheduling protocols, and stability and quality-of-service analyses.

Dr. Zhang twice won the Bell Labs President's Gold Award and the Lucent Chairman's Award.

Wenbo Zhao received the Bachelor of Science degree in mathematics from Nanjing University, Nanjing, China, in 2004, and the Master of Science degree in mathematics from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2007, and is currently pursuing the Ph.D. degree in computer science and engineering at the University of California, San Diego.

His research areas include computational complexity, spectral graph theory, and combinatorial optimization.