Energy-Efficient Network Routing with Discrete Cost Functions

Lin Wang[1,4], Antonio Fernández Anta[2],
Fa Zhang[1], Chenying Hou[1,4], and Zhiyong Liu[3] ⋆

[1]Center for Advanced Computing Research and Key Lab of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
{wanglin,zhangfa,houchenying}@ict.ac.cn
[2]Institute IMDEA Networks
antonio.fernandez@imdea.org
[3]China State Key Lab for Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences
zyliu@ict.ac.cn
[4]Graduate University of Chinese Academy of Sciences

Abstract. Energy consumption is an important issue in the design and use of networks. In this paper, we explore energy savings in networks via a rate adaptation model. This model can be represented by a cost-minimization network routing problem with discrete cost functions. We formulate this problem as an integer program, which is proved to be NP-hard. Then a constant approximation algorithm is developed. In our proposed method, we first transform the program into a continuous-cost network routing problem, and then we approximate the optimal solution by a two-step rounding process. We show by analysis that, for uniform demands, our method provides a constant approximation for the uniform network routing problem with discrete costs. A bicriteria network routing problem is also developed so that a trade-off can be made between energy consumption and network delay. Analytical results for this latter model are also presented.

Keywords: network optimization, network routing, approximation

1 Introduction

Energy-aware computing has recently become a hot research topic. The increasingly widespread use of Internet and the sprouting of data centers are having a dramatic impact on the global energy consumption. The energy consumed comes from the aggregate power used by many devices (CPUs, hubs, switches, routers). Recent studies ([13], [14]) show that there is significant room for energy saving in current networks in general. The main reason for this is that these networks

⋆ This research was supported in part by the Comunidad de Madrid grant S2009TIC-1692, Spanish MICINN grant TEC2011-29688-C02-01, and National Natural Science Foundation of China grant 61020106002, 61161160566.

are designed with a significant level of redundancy and over-provisioning, to guarantee QoS and to tolerate peak load and traffic variations. However, since networks usually carry only a small fraction of the peak, a significant portion of the energy consumed is wasted. Ideally, the energy consumed in a network should be proportional to the traffic load carried.

Prior work on energy efficiency have mostly focused on two techniques to save energy: speed scaling and powering down. Under speed scaling, it is assumed that the power consumed by a device working at speed s has the form $P = s^{\beta}$, where $\beta > 1$ is a constant. This comes from the well known cube-root rule, which states that the speed is approximately the cube root of the power consumed. Thus, the general energy saving model with speed scaling results in a network routing problem with a convex polynomial cost function $f_e(x_e) = \mu_e x_e^{\beta}$, where μ_e and β are constants, and x_e is the total traffic carried by device e (see [5], [7], [8], [11], [17]). Another approach to save energy is achieved by powering down the devices while they are idle. Andrews et al. [4] considered that network elements operate only in the full-rate active mode or the zero-rate sleeping mode. They demonstrated a trade-off between energy consumption and latency. Nedevschi et al. [15] explored both speed scaling and power down to reduce global energy consumption. Heller et al. [9] proposed a centralized method named *ElasticTree*, which powers down some of the routers or switches and then yields the energy-efficient routes in the data center network. At the same time, some other models, such as the adversary queueing model, were used to explore the energy saving in networks [3].

In this paper we consider an energy saving model called *rate adaptation.* In this model, network devices can operate in one of several speeds and each device chooses a proper state according to its current traffic load. Gunaratne et al. [12] first proposed a method which worked with the adaptive link rate (ALR). Also in [15], the authors studied the rate adaptive model combined with powering down devices. Here we will present a formal model that uses rate adaptation as power saving strategy and provides route assignment for message transmission from a global view of the network. Then an approximation method to solve the energy saving problem will be developed.

1.1 Related Work

The network routing problem is described as follows. We are given a set of traffic demands and want to inseparably route them over a transmission network. The total traffic x_e on link e incurs a cost which is defined by a cost function $f_e(x_e)$. Our objective is to find routes for all demands so that the total incurred cost $\sum_e f_e(x_e)$ is minimized.

There has been significant work on the general network routing problem. Note that the complexity of this problem depends on the cost function defined on each edge. For instance, if we choose $f_e(\cdot)$ as subadditive functions which have the property of *economies of scale*, the problem becomes the well-studied Buy-at-Bulk problem. Awerbuch and Azar [6] provided an $O(\log^2 n)$ randomized approximation algorithm for this problem. Andrews [1] showed that for any constant $\gamma > 0$, there is no $O(\log^{\frac{1}{2}-\gamma} N)$-approximation algorithm for non-uniform Buy-at-Bulk, and there is no $O(\log^{\frac{1}{4}-\gamma} N)$-approximation algorithm for the uniform version, unless $NP \in ZPTIME(n^{polylog\ n})$.

Closely related to our paper is the work of Andrews et al. [2]. The authors studied a new kind of minimum-cost network design with (dis)economic of scale and presented a polylogarithmic approximation algorithm to solve this problem. In [5], randomized rounding was used to achieve a constant approximation for uniform demands. Bansal et al. [7] studied the speed scaling model with arbitrary cost functions. They gave a $(3+\epsilon)$-competitive algorithm for this problem. Unlike the works mentioned above, we focus on the network routing problem with discrete cost functions rather than continuous ones.

1.2 Our Results

We aim to solve the minimum-energy routing in this paper. In Section 2, we give the formalized expression of the model and then prove it is hard to approximate. In Section 3, we introduce our proposed method. It first transforms the model into a general network routing problem with continuous cost functions. This is done by transforming the discrete function $f(\cdot)$ into a continuous function $g(\cdot)$ introducing a bounded error. Then uses a two-step rounding process to approximate the optimal set of routes. An analysis is given to show that our method obtains a constant approximation for this problem. In Section 4, we extend our model to a bicriteria network routing problem which considers not only the energy cost but also the latency so that trade-off can be made between the performance and energy consumption. Last, in Section 5, we draw conclusions.

2 The Model

We are given a directed graph $G = (V, E)$ and a set of traffic demands $D = (d_1, d_2, ..., d_k)$ where the i^{th} demand, $1 \leq i \leq k$, requests d_i units of bandwidth provisioned between a source node s_i and a sink node t_i. Unless otherwise said, in the following we assume unit demands, i.e., $d_i = 1$. We assume that links represent the abstracted resources, and each link can operate at one of a constant number of different rates $R_1 < R_2 < ... < R_m$. Note that for energy conservation consideration, it is reasonable to set numbers of different rates for newly designed network devices. Each rate R_i, $1 \leq i \leq m$, has an cost of $f(R_i)$. Our goal is to route all demands in a unsplittable fashion with the objective of minimizing the total cost. Note that unsplittable routing is important in many cases in order to avoid packets reordering.

2.1 Hardness

Not surprisingly, the minimum cost routing problem with discrete functions is NP-hard. Furthermore, we show here that, in general, it cannot even be approximated. This is shown in the following theorem.

Theorem 1. *There is no polynomial time approximation algorithm for the minimum cost routing problem with any finite approximation ratio, unless P=NP. This holds even if all links have the same cost function $f(\cdot)$, and the function is discrete and takes only 2 values.*

Proof. We prove the theorem by using reduction from the edge-disjoint paths (EDP) problem. This problem decides whether a given collection of pairs (a source and a sink in each pair) of nodes can be connected via edge-disjoint paths in a given network. It is known that EDP is NP-hard. We show now that any algorithm A that ρ-approximates ($1 \leq \rho < \infty$) the minimum cost network routing problem for uniform discrete cost functions of 2 values can be used to solve the EDP problem. This will prove the theorem.

Consider an instance of the EDP problem on a network G. The instance of the network routing problem has one unit demand for each pair of nodes. The cost function is as follows.

$$f(x) = \begin{cases} 0 & x \leq 1, \\ 1 & 1 < x \ . \end{cases} \tag{1}$$

Observe that if there are disjoint paths for the pairs of the EDP problem, then the network routing problem has a solution of zero cost. Then, algorithm A must return a solution that also has zero cost. On the other hand, if there are no disjoint paths, the optimal solution of the network routing problem has cost at least 1, and A will return a solution whose cost is in the interval $[1, \rho]$. Hence, the algorithms A can be used to solve the EDP problem.

From the above reduction, we conclude that the problem is hard to be approximated because we have not given any restrictions on $f(R_i)/f(R_{i-1})$ which may be unbounded. If we bound the ratio between any two adjacent steps of the cost function, the reduction in the proof of Theorem 1 can not be built, and the inapproximability result may not hold any more. In particular, the problem with restricted step ratio can be approximated by a constant approximation ratio. We will give the details in the following sections and will discuss the step cost function with step ratio restriction. From now on, we will regard the above ratio as a constant.

2.2 Integer Program Formulation

Formally, we can formulate the described routing problem with integer program (P_1). The binary variable $y_{i,e}$ indicates whether demand i uses link e, while x_e is the total load on e. Flow conservation means that for each demand i the source s_i generates a flow of d_i, the sink absorbs a flow d_i, and for the other vertices the incoming and outgoing flows of demand i are the same. Observe that for $x_e \leq z_e$, $f(x_e) = f(z_e)$. This results in the discrete property of the cost function $f(\cdot)$. More precisely, $f(x)$ is a non-decreasing step function of x, where x is the speed of each link. In practice, cost functions for network resources can be different. Here we just take a uniform cost function for convenience. There is no doubt that solving (P_1) is NP-hard for the $0-1$ constraint on variable $y_{i,e}$. Since solving our network routing problem is NP-hard (as implicitly shown in Theorem 1), so we have no hope on finding the optimal solution.

$$(P_1) \quad \min \quad \sum_e f(z_e)$$

subject to

$$\begin{aligned}
& x_e = \sum_i y_{i,e} && \forall e \\
& x_e \le z_e && \forall e \\
& z_e \in \{R_0, R_1, ..., R_m\} && \forall e \\
& y_{i,e} \in \{0, 1\} && \forall i, e \\
& y_{i,e} : \ \text{flow conservation} &&
\end{aligned}$$

3 The Approximation Algorithm

In this section, it is shown how to approximate a solution of (P_1). First we use a particular interpolation method to transform the cost function of the original program into a continuous one, which is indeed to relax the discretion. It makes the program to be solvable while introducing a bounded error. Then, we approximately solve the transformed program by a two-step rounding process. This process assigns routes to the demands and determines the rates of links. We assign a path for each demand by randomized rounding and then round the link rates based on the determined routes. At last, we analyze the performance of the proposed method.

3.1 Transforming the Program

We use a special interpolation method to simplify our optimization program by replacing the step function $f(\cdot)$ with a continuous function $g(\cdot)$. Before applying interpolation, we have to decide the form of the function $g(x)$ we want to get. It has suggested that most network devices consume energy in a superadditive manner [5]. That is, doubling the speed more than doubles the energy consumption. Hence the energy curve is often modeled by a polynomial function $g(x) = \mu x^\beta$ where μ and β are constants associated with network elements. More precisely, the parameter β in the ordinary form of energy consumption has been usually assumed to be in the interval $(1, 3)$ [10]. The objective here is to transform a step cost function into a function in the form of $g(x) = \mu x^\beta$. Although, as mentioned, typically β will be larger than 1, and hence $g(\cdot)$ will be a convex function, the proposed interpolation method does not impose such restriction.

Now we discuss how to apply the transformation from a step function to a continuous one. A common approach has been using midpoints of the steps as discrete values and fitting by mean squares. This approach is not appropriate if the step of the function have unequal length. Another popular method is to do interpolation on a set of points which is obtained by sampling the original function. Unfortunately, using this technique the error of the interpolation depends on the sampling method we choose, and is hard to be estimated. Here we use an alternative [16] based on integral minimization, where each point on the original function has to be considered as an observation. Without depending on some other parameters, the method works well for the fitting of step functions.

Definition Consider the original function $f(x)$, and the one to be fitted $g(x)$, as described before. $f(x)$ is defined as follow.

$$f(x) = \begin{cases} y_1, & x_0 < x \leq x_1, \\ y_2, & x_1 < x \leq x_2, \\ \ldots \\ y_m, & x_{m-1} < x \leq x_m, \end{cases} \tag{2}$$

where in our case $y_i = f(R_i)$ $(1 \leq i \leq m)$ is the energy consumption value of each state and $x_i = R_i,\ x_{i+1} = R_{i+1}$ $(0 \leq i < m)$ represents the lower and upper boundaries of the speed for each state. We aim to fit $g(x)$ to $f(x)$.

Integral Minimization The integral to minimize can be represented as

$$\begin{aligned} G(\mu, \beta) &= \int [f(x) - g(x|\mu, \beta)]^2 dx \\ &= \sum_{i=1}^{m} \int_{x_{i-1}}^{x_i} [y_i - (\mu x^{\beta})]^2 dx \ . \end{aligned} \tag{3}$$

Since $g(x)$ is not a linear function, this minimization problem is hard to solve. But it is linear in a logarithmic transformation. Observe that

$$\log(g(x)) = \log \mu + \beta \log x \ . \tag{4}$$

Let us define $v_i = \log y_i$, $w = \log x$, and $\mu' = \log \mu$. Then, the alternative integral that we will in fact use can be obtained as

$$H(\mu, \beta) = \sum_{i=1}^{m} \int_{w_{i-1}}^{w_i} [v_i - (\mu' + \beta w)]^2 dw. \tag{5}$$

And now (5) is to be minimized with respect to the parameters of the general quadratic equation. Necessary conditions obtained by setting the first partial derivatives equal to zero are

$$\begin{cases} \dfrac{\partial H}{\partial \mu'} = \displaystyle\sum_{i=1}^{m} \int_{w_{i-1}}^{w_i} -2[v_i - \mu' - \beta w]dw = 0, \\ \dfrac{\partial H}{\partial \beta} = \displaystyle\sum_{i=1}^{m} \int_{w_{i-1}}^{w_i} -2w[v_i - \mu' - \beta w]dw = 0 \ . \end{cases} \tag{6}$$

It is obvious that the second derivatives are all positive. By solving equation (6), we can get the values of parameters μ' and β, and from μ' it is obtained μ. From these, the objective function $g(x)$ of the interpolation can be determined.

Bound on the Interpolation Error As our method is proposed to approximate the optimal solution, it is important to bound the error introduced. During the interpolation process, the error comes from the gap between the original function $f(x)$ and the fitted function $g(x)$. We define this gap as follow.

$$Gap = \max_x \left\{ \frac{f(x)}{g(x)}, \frac{g(x)}{f(x)} \right\} \ . \tag{7}$$

While using this gap definition as *interpolation error*, we can show the following theorem.

Theorem 2. *Given a $f(x)$ such that $y_i/y_{i-1} \le \sigma$ ($\sigma > 1$), the interpolation error satisfies $Gap \in [\frac{2\sigma}{\sigma+1}, \sigma]$, when $y_0 \neq 0$.*

Proof. (Sketch) The proof is conducted as follows. It can be shown that functions $f(\cdot)$ and $g(\cdot)$ intersect in each interval $[R_{i-1}, R_i]$. This is the key of the proof. Then, consider two cases $f(x) \ge g(x)$ and $f(x) \le g(x)$. In both cases we assume there is a bound δ for the interpolation error, and then we derive that δ satisfies some conditions in order to maintain the bound. Thus we obtain the results.

As a result, the error is not so big because our interpolation method aims to minimize the error. Another observation is that the cost is decreased when $f(x) > g(x)$ but increased when $f(x) < g(x)$. This brings a two side effects on the error.

New Integer Program Once the function $g(x)$ is obtained, the optimization can be rewritten as follows.

$$(P_2) \quad \min \ \sum_e g(x_e)$$

subject to

$$x_e = \sum_i y_{i,e} \qquad \forall e$$

$$y_{i,e} \in \{0, 1\} \qquad \forall i, e$$

$$y_{i,e} : \ \text{flow conservation}$$

The problem now turns into an integer program with a convex [1] objective function. Of course we can conclude that the problem is still NP-hard for the convex objective and the 0-1 constraint on $y_{i,e}$.

3.2 Two-step Rounding

In this section, we introduce a two-step rounding method to complete the routing and rates determination. Our routing problem has been transformed into integer programming (P_2) with a convex objective function. After solving (P_2), we also need to choose a proper transmission rate for each link.

First we use randomized routing in (P_2) to approximate the optimal cost and extract routing paths for all demands. The basic idea of randomized routing is to use random choices to convert an optimal solution of a relaxation of the problem into a probabilistically provable approximation to the optimal solution of the original problem. To apply it to (P_2), first the binary constraint $y_{i,e} \in \{0, 1\}$ is relaxed to $y_{i,e} \in [0, 1]$. This transforms the integer program into a linear program (with convex objective function), which is optimally solvable in polynomial time. Then, we get the optimal fractional solution by solving the relaxed convex programming. Finally, randomized decisions are used to round the fractional flow.

We use the Raghavan-Thompson randomized rounding. The algorithm runs as follows. Once the optimal fractional solution has been found, the flow assigned to links is mapped to flows in paths as follows. For each demand i, first we generate a sub-graph G_i defined by links e where $y^*_{i,e} > 0$. (The flows, or weights, $y^*_{i,e}$ are the optimal fractional solution of the relaxed program.) Then, we extract a path p connecting the source and destination nodes and select the weight $y^*_{i,e}$ of the bottleneck link $e \in p$ to be the weight of this path, which is denoted as w_p. Hereafter the weight $y^*_{i,e}$ of each link e in path p is decreased by w_p. Run the above procedure repeatedly until all weights $y^*_{i,e}$ on the G_i become zero. Because of the flow conservation constraint, this can always be achieved. At last, we randomly select one path for each demand i using the path weights as probabilities. After this rounding, there is one path for each demand.

Secondly, the state of each link should be determined after the demand routes have been chosen. We select the speed of each link via the following rounding procedure. First, the carried traffic $\hat{x_e}$ of each link e is calculated as $\hat{x_e} = \sum_i \hat{y_{i,e}}$, where $\hat{y_{i,e}}$ is the amount of demand i that traverses link e after the rounding. Then for each link, we search the collection of possible operational speeds and choose the minimal s_e that can support the carried traffic. More formally,

$$s_e = \min\{R_i | (i \in [1, m]) \land (\hat{x_e} \leq R_i)\}. \tag{8}$$

With this the minimum cost routing problem with discrete cost functions has been solved as we have determined the link states and routed all the demands.

[1] Assuming $\beta \geq 1$. If $\beta < 1$, then $g(\cdot)$ is a concave function, and hence we have an instance of the Buy-at-Bulk problem. As mentioned, there is no constant ratio approximation in this case.

3.3 Performance Evaluation

Now we analyze the approximation ratio of the proposed approximation algorithm. Let x_e^* be the flow on link e under the optimal fractional routing, $\hat{x_e}$ be the rounded flow, and s_e be the selected operating state for link e by our methods. We show,

Theorem 3. *Let the ratio between any two adjacent steps of cost function $f(\cdot)$ be bounded by σ. For unit demands, the expected cost obtained with our routing method, $E[\sum_e f(s_e)]$, is a γ-approximation of the optimal solution with respect to the discrete cost function $f(x)$, where γ is a constant.*

The proof of this theorem proceeds by two steps. First we give the relation between solution by our two-step rounding and the one by Raghavan-Thompson randomized rounding. And then we bound the latter to optimal. Using these two results, we obtain the approximation ratio of the two-step rounding.

For the optimal fractional solution, the cost can be represented as $\sum_e g(x_e^*)$, and for the solution by Raghavan-Thompson randomized rounding, it is $\sum_e g(\hat{x_e})$, while after the two-step rounding, it is $\sum_e f(s_e)$. As we have discussed before, the gap between the original function $f(x)$ and the fitted function $g(x)$ has two sides effect on the total cost. Assume $s_e = R_i$, consider the following case.

Lemma 1. *If the ratio between any two adjacent steps of cost function $f(\cdot)$ is bounded by σ, then $f(s_e) \leq \sigma^2 g(\hat{x_e})$.*

Proof. The result follows since, from Theorem 2, the largest gap between $g(\hat{x_e})$ and $f(\hat{x_e})$ is σ. Then, from the relation between s_e and $\hat{x_e}$ (see Eq. 8), also $f(s_e) \leq \sigma f(\hat{x_e})$. And by Theorem 2, we have $f(s_e) \leq \sigma^2 f(\hat{x_e})$, which completes the proof.

Now we can give the proof of Theorem 3.

Proof. The expected cost of the solution found is $E[\sum_e f(s_e)]$. From Lemma 1, we have that $f(s_e) \leq \sigma^2 g(\hat{x_e})$, and hence $E[\sum_e f(s_e)] \leq \sigma^2 E[\sum_e g(\hat{x_e})]$. As it was shown in [5], there is a constant δ such that $E[\sum_e g(\hat{x_e})] \leq \delta \sum_e g(x_e^*)$.

To complete the proof, we observe from Theorem 2 that, for all x, $g(x)/\sigma \leq f(x)$. Then, if C^* is the cost the optimal solution of the routing problem with the step function $f(\cdot)$, the optimal fractional solution of the relaxation of P_2 satisfies that $C* \geq \sum_e g(x_e^*)/\sigma$. Putting it all together, we have that $E[\sum_e f(s_e)] \leq \sigma^2 \delta \sum_e g(x_e^*) \leq \sigma^3 \delta C^*$.

This result can be applied to uniform demands easily. For uniform demands where each traffic demand requests a bandwidth $d_i = d$, the total flow on each edge is d times of that in the case with unit demands. So we have,

Corollary 1. *For uniform demands, our routing method can also obtain a γ-approximation to the optimal integral solution in expectation, where γ is a constant.*

4 Model Extension: Bicriteria Network Routing

We give an extension to model (P_2) in this section. For practical applications, we should consider the network performance as well as the energy consumption. There are many issues related to the network performance, like queueing delay, transmission delay etc. For convenience consideration, here we just take the transmission delay from s_i to t_i for demand i as an example, but other assumptions can also work in our extended model. In order to express this new added metric, we assume on each edge e of original graph $G = (V, E)$, we have given a scale l_e to describe the latency. Thus our routing problem has two objectives, which are energy saving and network latency minimization.

We first consider the case in which the average latency of routing all the demands is restricted to be smaller than a value of L. So for the model (P_2), we have an additional constraint

$$\sum_i \sum_e l_e \cdot y_{i,e} \leq L \ . \tag{9}$$

For solving this problem, we can simply introduce a Lagrange multiplier λ and then move the constraint to the objective function as

$$\min \ \sum_e g(x_e) + \lambda(\sum_i \sum_e l_e y_{i,e} - L) \ . \tag{10}$$

Using the Lagrange relaxation, we can solve the constrained minimum cost routing problem we have just talked in previous sections. And by the property of Lagrange relaxation, easily we have that solutions obtained by Lagrange relaxation are always the lower bound of the optimal integer solution. By setting λ to different values, we choose a good solution from the results.

As to better understand the trade-off between energy saving and network latency, we now analyse the bicriteria network routing model. The problem can be described as minimizing both energy consumption and network latency as two objectives. Here we use an aggregated objective function method to deal with it. Recall that in (P_2) we only take energy consumption in consideration, and aim to minimize it. Now we introduce a parameter α to make a convex combination of the two objectives of interest (minimizing both energy consumption and latency). As we have presented, after the interpolation process, the energy consumption cost is given by a convex function. Denote the energy cost as $Cost_e$ and latency cost as $Cost_l$. The total cost is obtained as $Cost = \alpha \cdot Cost_e + (1 - \alpha)Cost_l$.

The convex combination can preserve the convex property of the two individual costs. And thus our routing problem with objective to minimizing the combined cost is still a convex programming. So the methods we proposed in the previous sections are still contributing. At the same time, the introduced parameter α provide flexibility in our model where adjusting α to different values leads to different trade-off effects for the two unrelated metrics. In particular, when α is set to be zero, the total cost only consists of the latency cost. Thus the problem is degraded to be the shortest path routing, which is polynomial time solvable. And $\alpha = 1$ leads to the routing problem with only one objective to minimize the energy cost, which we have just studied before.

We explore now the ratio between the latency of the routes found with our method and the latency of the shortest paths. That is the stretch ratio r_i, that we define for a demand i as the latency of paths we obtain divides by the latency of shortest paths. Then we define the *Stretch* as the maximum stretch ratio among all demands.

$$Stretch = \max_{i \in [1,k]} \{r_i\} . \tag{11}$$

Using this definition, we have

Theorem 4. *There is no bound between the latency of the paths used in the trade-off method and the latency of the shortest paths. In other words, the stretch can not be bounded.*

The proof of this theorem is omitted from this extend abstract. As a result, the only way to obtain good performance for this trade-off is to choose a proper value for α. In practice, we can vary α in $(0, 1)$ to get the relation between the two objectives, which helps to determine the parameter. Usually, satisfying the necessary performance requirements, we aim to maximize the energy savings.

5 Conclusion

In this paper, we investigate the network routing problem with discrete cost functions which aims to route demands under a minimum cost way. The problem comes from the green computing sceneries which are quite important recently. Our contributions are mainly on the following results: for the rate adaptive energy-saving strategy, we give a model expression by an integer program which is believed to be NP-hard; our proposed method for solving this problem consists of two parts. First we provide a particular interpolation method to transform the discrete cost function into a continuous one which makes the complicated integer program solvable. Then a two-step rounding method is developed to give routes to demands and determine link rates by approximately solving the integer program. By using this method, we obtain a constant approximation to the optimal for uniform demands; also we discuss how to extend our original model to a bicriteria network routing which can give trade-off between the two metrics.

References

1. Andrews, M.: Hardness of Buy-at-Bulk Network Design. In: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS), pp 115–124. IEEE Computer Society, Washington, DC (2004).
2. Andrews, M., Antonakopoulos, S., Zhang, L.: Minimum-Cost Network Design with (Dis)economies of Scale. In: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS), pp 585–592. IEEE Computer Society, Washington, DC (2010).

3. Andrews, M., Antonakopoulos, S., Zhang, L.: Energy-aware Scheduling Algorithms for Network Stability. In: Proceedings of 29th Annual IEEE International Conference on Computer Communications (INFOCOM), pp 1359–1367. IEEE Press, Shanghai (2011).
4. Andrews, M., Fernández A.A., Zhang, L., Wenbo Zh.: Routing and Scheduling for Energy and Delay Minimization in the Powerdown Model. In: Proceedings of 29th Annual IEEE International Conference on Computer Communications (INFOCOM). IEEE Press, San Diego, CA (2010).
5. Andrews, M., Fernández A.A., Zhang, L., Wenbo Zh.: Routing for Energy Minimization in the Speed Scaling Model. In: Proceedings of 29th Annual IEEE International Conference on Computer Communications (INFOCOM). IEEE Press, San Diego, CA (2010).
6. Awerbuch, B., Azar, Y.: Buy-at-bulk Network Design. In: Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS), pp 542–550. IEEE Computer Society, Washington, DC (1997).
7. Bansal, N., Chan, H.L., Pruhs, K.: Speed Scaling with an Arbitrary Power Function. In: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp 693–701. Society for Industrial and Applied Mathematics, New York (2009).
8. Bansal, N., Kimbrel, T., Pruhs, K.: Speed Scaling to Manage Energy and Temperature. Journal of the ACM, vol. 54(1), pp 1–39. ACM Press, New York (2007).
9. Brandon H., Srinivasan S., Priya M., Yiannis Y., Puneet S., Sujata B., Nick M.: ElasticTree: Saving Energy in Data Center Networks. In: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, pp 249–264. USENIX Association, Berkeley, CA (2010).
10. Brooks, D.M., Bose, P., Schuster, S.E., Jacobson, H., Kudva, P.N., Buyuktosunoglu, A., Wellman, J., Zyuban, V., Gupta, M., Cook, P.W.: Power-aware Microarchitecture: Design and Modeling Challenges for Next-generation Microprocessors. IEEE Micro, vol. 20(6), pp 26–44. IEEE Press (2000).
11. Chan, H.L. and Chan, W.T., Lam, T.W., Lee, L.K., Mak, K.S., Wong, P.W.H.: Energy Efficient Online Deadline Scheduling. In: Proceedings of the 8th annual ACM-SIAM Symposium on Discrete Algorithms, pp 795–804. Society for Industrial and Applied Mathematics, New Orleans, Louisiana (2007).
12. Gunaratne, C., Christensen, K., Nordman, B., Suen, S.: Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR). IEEE Transaction on Computers, vol 57(4), pp 448–461. IEEE Computer Society, Washington, DC (2008).
13. Gupta, M., Singh, S.: Greening of the Internet. In: Proceedings of the 2003 Conference on Applications, technologies, architectures and protocols for Computer Communications (SIGCOMM), pp. 19–26. ACM Press, Karlsruhe, Germany (2003).
14. Kurp, Patrick: Green Computing. Communication of the ACM. vol. 51(10), pp 11–13. ACM Press, New York (2008).
15. Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., Wetherall, D.: Reducing Network Energy Consumption via Sleeping and Rate-adaptation. In: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI), pp 323–336. USENIX Association, San Francisco, CA (2008).
16. Burt S.R.: Curve fitting to step functions. Agricultural & Applied Economics Association, vol 46(2), pp. 662-672 (1964).
17. Yao, F., Demers, A., Shenker, S.: A Scheduling Model for Reduced CPU Energy. In: 36th Annual Symposium on Foundations of Computer Sciences, pp 374–382 (1995).