Enhancing ε-Approximation Algorithms With the Optimal Linear Scaling Factor

Gang Cheng, Nirwan Ansari, *Senior Member, IEEE*, and Li Zhu

***Abstract*—Finding a least-cost path subject to a delay constraint in a network is an NP-complete problem and has been extensively studied. Many works reported in the literature tackle this problem by using ε-approximation schemes and scaling techniques, i.e., by mapping link costs into integers or at least discrete numbers, a solution which satisfies the delay constraint and has a cost within a factor of the optimal one, that can be computed with pseudopolynomial computational complexity. In this paper, having observed that the computational complexities of the ε-approximation algorithms using the linear scaling technique are linearly proportional to the linear scaling factor, we investigate the issue of finding the optimal (the smallest) linear scaling factor to reduce the computational complexities, and propose two algorithms, the optimal linear scaling algorithm (OLSA) and the transformed OLSA. We analytically show that the computational complexities of our proposed algorithms are very low, as compared with those of ε-approximation algorithms. Therefore, incorporating the two algorithms can enhance the ε-approximation algorithms by granting them a practically important capability: self-adaptively picking the optimal linear scaling factors in different networks. As such, ε-approximation algorithms become more flexible and efficient.**

***Index Terms*—Delay-constrained least cost (DCLC), NP-complete, ε-approximation, linear scaling factor.**

I. Introduction

THE continuous growth in both commercial and public network traffic with various quality-of-service (QoS) requirements calls for better services than the Internet's best-effort mechanism. One of the challenging issues is to select feasible paths that satisfy the different requirements of various applications. This problem is known as QoS routing. In general, there are two issues related to QoS routing: state distribution, and routing strategy [1]. State distribution addresses the issue of exchanging the state information throughout the network [2]. Routing strategy is used to find a feasible path that meets the QoS requirements. It has been proved in [3] and [4] that the routing problems involving the minimization of two or more additive QoS parameters are NP-complete. In this paper, we focus on a relatively simplified problem: the delay-constrained least-cost (DCLC) path-selection problem [5], defined below.

Definition 1: DCLC path selection [5]: Assume a network is modeled as a directed graph $G(N, E)$, where N is the set of all nodes and E is the set of all links. Each link connected from node u to v, denoted by $e_{u,v} = (u, v) \in E$, is associated with a cost $c(u, v)$ and a delay $d(u, v) \geq 0$. Given a delay constraint $d > 0$, and a pair of nodes s and t, the objective of DCLC is to find the path p that has the least cost among the paths from s to t, subject to $D(p) = \sum_{e_{u,v} \in p} d(u, v) \leq d$.

Many QoS routing algorithms have been proposed in the literature. The limited path heuristic proposed by Yuan [6] maintains a limited number of candidate paths, say x, at each hop. The computational complexity is $O(x^2nm)$ for the extended Bellman–Ford (EBF) algorithm with two constraints, where n and m are the number of links and nodes, respectively. For the particular case of two constraints, Yuan [7] studied two EBF algorithm-based heuristics, the limited granularity and limited path heuristics. Yuan showed that under the condition of uniform mapping, the limited granularity heuristic is optimal in the sense that it provides optimal worst-case guarantee in finding the paths that satisfy the QoS constraints from among all limited granularity heuristic schemes. Guo studied the minimum-cost QoS multicast and unicast problems in [8] and [9]. He presented two efficient algorithms, respectively, to approximate the minimum-cost QoS trees and minimum-cost QoS unicast paths in a communication network. For the purpose of improving the response time and reducing the computation load on the network, precomputation-based methods [10] have been proposed. Korkmaz and Krunz [11] provided a heuristic with computational complexity comparable to that of the Dijkstra algorithm, to find the least-cost path subject to multiple constraints. An algorithm, called A*Prune [12], is capable of locating multiple shortest feasible paths from the maintained heap in which all candidate paths are stored. For the case that only inaccurate link state information is available to nodes, approximate solutions [13] have been proposed for the most-probable bandwidth delay-constrained path (MP-BDCP) selection problem by decomposing it into two subproblems: the most-probable delay-constrained path (MP-DCP) and the most-probable bandwidth-constrained path (MP-BCP). A Lagrange relaxation-based aggregated cost (LARAC) was proposed in [5] for the DCLC path problem. This algorithm is based on a linear cost function $c_\lambda = c + \lambda d$, where c denotes the cost, d the delay, and λ an adjustable parameter. It was shown that the computational complexity of this algorithm is $O(m^2 \log^4 m)$. In [14], a heuristic algorithm was proposed based on a linear cost function for two additive constraints; this is a multiple constrained path (MCP) selection problem with two additive constraints. A binary search

Paper approved by T. T. Lee, the Editor for Switching Systems and Network Performance of the IEEE Communications Society. Manuscript received April 28, 2004; revised February 9, 2006. This work was supported in part by the National Science Foundation under Grant 0435250, and in part by the New Jersey Commission on Science and Technology via NJWINS. This paper was presented in part at the IEEE International Conference on Communications, Seoul, Korea, 2005.

G. Cheng is with VPISystems Inc. Corp., Holmdel, NJ 07733 USA.

N. Ansari is with the Advanced Networking Laboratory, Electrical and Computer Engineering Department, New Jersey Institute of Technology, Newark, NJ 07012 USA (e-mail: ansari@njit.edu).

L. Zhu is with the Advanced Networking Laboratory, Electrical and Computer Engineering Department, New Jersey Institute of Technology, Newark, NJ 07012 USA, and also with Voicenet Inc., Philadelphia, PA 19114 USA.

Digital Object Identifier 10.1109/TCOMM.2006.878832

strategy for finding the appropriate value of β in the linear cost function $w_1(p) + \beta w_2(p)$ or $\beta w_1(p) + w_2(p)$, where $w_i(p)(i = 1, 2)$ are the two respective weights of the path p, was proposed, and a hierarchical Dijkstra algorithm was introduced to find the path. It was shown that the worst-case complexity of the algorithm is $O(\log B(m + n \log n))$, where B is the upper bound of the parameter β. The authors in [15] simplified the multiple constrained QoS routing problem into the shortest-path selection problem, in which the weighted fair queuing (WFQ) service discipline is assumed. Hence, this routing algorithm cannot be applied to networks where other service disciplines are employed. Widyono [16] introduced a constrained Bellman–Ford (CBF) algorithm, which deploys a breadth-first-search to locate paths of monotonically increasing delay, while recording and updating the lowest-cost path to the visited nodes. This approach yields the optimal path, the least-cost path from among all the paths satisfying the delay constraint. However, its worst-case computational complexity is exponentially increasing with the network size. Many researchers have posed the QoS routing problem as the k-shortest-path problem , [12]. The authors in [17] proposed an algorithm, called TAMCRA, for MCP by using a nonlinear cost function and a k-shortest-path algorithm. The computational complexity of TAMCRA is $O(kn \log(kn) + k^3 mM)$, where k is the number of shortest paths and M is the number of constraints. To solve the delay-cost-constrained routing problem, Chen and Nahrstedt [18] proposed an algorithm which maps each constraint from a positive real number to a positive integer. By doing so, the mapping offers a "coarser resolution" of the original problem, and the positive integer is used as an index in the algorithm. The computational complexity is reduced to pseudopolynomial time, and the performance of the algorithm can be improved by adjusting a parameter, but with a larger overhead.

Many ε-approximation algorithms (the solution has a cost within a factor of $(1 + \varepsilon)$ of the optimal one) subject to DCLC have been proposed in the literature. Lorenz *et al.* [20] presented several ε-approximation solutions for both the DCLC and the multicast tree. Hassin [21] presented two ε-approximations algorithms for the restricted shortest path (RSP) problem. Raz and Shavitt [22] proposed an efficient dynamic programming solution for the case in which the QoS parameters are integers, and a sublinear algorithm for the case in which all link costs use the (same) function of their corresponding delays. Different from [20]–[22], in which link costs are scaled into integers, the DSA algorithm [23] scales the link delay into integers, and provides a path having the cost and delay no larger than the cost of the optimal feasible path and $(1 + \varepsilon)d$, respectively. Since the path computed by [23] may have a delay larger than the delay constraint, it cannot guarantee a 100% success ratio.

The basic idea behind DCLC ε-approximation approaches is to use some functions, referred to as scaling functions in this paper, to approximate the link costs with bounded finite ranges, thus reducing the original NP-complete problem to a simpler problem that can be solved in polynomial time. In this paper, we focus our discussion on a particular kind of ε-approximation algorithm, the linear scaling ε-approximation algorithms, defined below.

Definition 2: A linear scaling ε-approximation algorithm refers to an algorithm that can provide an ε-approximation solution by first linearly increasing or decreasing the link costs, and then quantizing them into integers. In particular, given a nondecreasing quantization function $f_\varepsilon(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{Z}^+$, where $\mathbb{R}^+$ is the set of positive real numbers and $\mathbb{Z}^+$ is the set of positive integers, and a network $G(N, E)$ in which each link $e_{u,v}$ is associated with a cost $c(u, v) \in \mathbb{R}^+$, a delay $d(u, v) \in \mathbb{R}^+$, and a delay constraint $d > 0$, an algorithm which yields the optimal solution in $G'(N, E)$ that is an ε-approximation solution in $G(N, E)$, where $G'(N, E)$ is constructed from $G(N, E)$ by mapping the link cost $c(u, v)$ to $c'(u, v) = f_\varepsilon(\lambda c(u, v))$ and λ is the linear scaling factor, is referred to as a linear scaling ε-approximation algorithm.

As reviewed above, it can be observed that except those proposed for the case in which link costs are already integers or discrete, most ε-approximation algorithms, if not all, use the linear scaling functions, e.g., $f(x) = \lceil \lambda x \rceil$. Moreover, we find that their computational complexities are linearly proportional to the linear scaling factor. Note that since the linear scaling factor of all ε-approximation algorithms reported in the literature is linearly proportional to $(1/\varepsilon)$, their computational complexities are linearly proportional to $(1/\varepsilon)$. For instance, many ε-approximation algorithms use Bellman–Ford-like algorithms (e.g., RSP [20] and DAD [23]) to find an optimal solution (the least-cost path satisfying the delay constraint) in the network where link costs are integers. Since the computational complexities of these Bellman–Ford-like algorithms are linearly proportional to the cost of the optimal feasible path (if it exists), which is, in turn, also linearly proportional to the linear scaling factor, the computational complexities of these algorithms are consequently linearly proportional to the linear scaling factor. Hence, our task in this paper is to minimize the linear scaling factor so that the computational complexity of ε-approximation algorithms can be reduced. *It should be noted that although the algorithms presented in this paper are tailored for the DCLC ε-approximation algorithms, they can be readily applied to all other cases where linear ε-approximation techniques are deployed, except those in which $\lfloor \cdot \rfloor$ is used for scaling* (*not* $\lceil \cdot \rceil$). Furthermore, we analytically show that the computational complexities of our proposed algorithms are very low with respect to those of ε-approximation algorithms. Therefore, incorporating the two algorithms into ε-approximation algorithms does not increase their computational complexities, but can, in fact, effectively reduce their computational complexities, because the optimal linear scaling factor can always be computed by our proposed algorithms.

It should be noted that this paper presents, to our best knowledge, the first attempt at minimizing the linear scaling factor of the ε-approximation algorithms. Accordingly, the optimal linear scaling algorithm (OLSA) and the transformed OLSA (T-OLSA) are the first two algorithms proposed specifically to computing the optimal linear scaling factor.

II. A Framework of ε-Approximation Approaches for DCLC

In this section, a framework for optimizing linear scaling ε-approximation algorithms is presented. Observe that in an ε-approximation approach, the nondecreasing quantization

function plays the key role. Hence, *Definition 3* is provided below to formulate the set of functions that can be used to design an ε-approximation algorithm.

Definition 3: Given an instance of DCLC and a nondecreasing function $f_\varepsilon(\cdot), \forall x \in \mathbb{R}^+, f_\varepsilon(x) \in \mathbb{Z}^+$, a delay constraint $d > 0$, and a network $G(N,E)$, in which each link $e_{u,v}$ is associated with a cost $c(u,v) \in \mathbb{R}^+$ and a delay $d(u,v) \in \mathbb{R}^+$, $G'(N,E)$ is constructed from $G(N,E)$ by mapping the cost of link $e_{u,v}$ to $c'(u,v) = f_\varepsilon(c(u,v))$. If the optimal feasible path in $G'(N,E)$ from the source to the destination has a cost no greater than a factor of $1+\varepsilon$ from that of the optimal feasible path between the corresponding pair of nodes in $G(N,E)$, $f_\varepsilon(\cdot)$ is called a feasible ε-approximation function.

Based on *Definition 3*, we present the next proposition, from which feasible ε-approximation functions may be derived. Note that as long as a function satisfies the next proposition, it satisfies *Definition 2* (for any $G(N,E)$), i.e., the functions satisfying the next proposition are universally feasible.

Proposition 1: Given any instance of DCLC, assume there exists a nondecreasing function $f_\varepsilon(\cdot)$ such that $\forall x \in \mathbb{R}^+, f_\varepsilon(x) \in \mathbb{Z}^+$, and for any two sets of positive numbers, $\{\alpha_i > 0, i = 1,2,\ldots,n\}$ and $\{\beta_j > 0, j = 1,2,\ldots,m\}$, if

$$\sum_{i=1}^{n} \alpha_i \le \sum_{j=1}^{m} \beta_j \tag{1}$$

$$\sum_{i=1}^{n} f_\varepsilon(\alpha_i) \ge \sum_{j=1}^{m} f_\varepsilon(\beta_j) \tag{2}$$

imply that

$$(1+\varepsilon)\sum_{i=1}^{n} \alpha_i \ge \sum_{j=1}^{m} \beta_j \tag{3}$$

then the function $f_\varepsilon(\cdot)$ is a feasible ε-approximation function.

Proof: Given an instance of DCLC, assume an optimal feasible path between nodes s and t is path p, and the optimal feasible path is $\hat{p}$ after the costs of links have been scaled to integers via the function $f_\varepsilon(\cdot)$. Therefore

$$\sum_{e_{u,v} \in p} c(u,v) \le \sum_{e_{u,v} \in \hat{p}} c(u,v) \tag{4}$$

$$\sum_{e_{u,v} \in p} f_\varepsilon(c(u,v)) \ge \sum_{e_{u,v} \in \hat{p}} f_\varepsilon(c(u,v)). \tag{5}$$

By the definition of $f_\varepsilon(\cdot)$

$$(1+\varepsilon)\sum_{e_{u,v} \in p} c(u,v) \ge \sum_{e_{u,v} \in \hat{p}} c(u,v) \Leftrightarrow (1+\varepsilon)C(p) \ge C(\hat{p}). \tag{6}$$

Hence, $f_\varepsilon(\cdot)$ is a feasible ε-approximation function. ■

Finding such universally feasible ε-approximation functions for any instance of DCLC solely based on *Proposition 1* is difficult. Instead, we focus on deriving a solution for an easier case: find a feasible linear scaling ε-approximation function for a given instance of DCLC. Since the computational complexities of ε-approximation algorithms subject to DCLC are linearly proportional to the linear scaling factor (λ), the smaller the linear scaling factor, the better. Therefore, for the purpose of reducing the computational complexities of ε-approximation algorithms, our objective is to find the smallest linear scaling factor.

Definition 4: Given a network $G(N,E)$, the feasible linear scaling ε-approximation function $f_\varepsilon(\cdot)$ that has the lowest scaling factor among all feasible functions is called the optimal linear scaling ε-approximation function.

III. Optimal Linear Scaling Feasible ε-Approximation Functions

In this section, we will analytically demonstrate how to find the optimal linear scaling ε-approximation function. We first provide the next proposition to simplify the search for a feasible ε-approximation function.

Proposition 2: Given an instance of DCLC and $f_\varepsilon(x), f_\varepsilon(c(u,v)) \in \mathbb{Z}^+$ is a feasible ε-approximation function if there exists a λ such that $\forall e_{u,v} \in E$

$$(1+\varepsilon)c(u,v)\lambda \ge f_\varepsilon(c(u,v)) \ge c(u,v)\lambda. \tag{7}$$

Proof: Given any two sets of links $E_1 \subset E$ and $E_2 \subset E$, if

$$\sum_{e_{u,v} \in E_1} c(u,v) < \sum_{e_{u,v} \in E_2} c(u,v) \tag{8}$$

$$\sum_{e_{u,v} \in E_1} f_\varepsilon(c(u,v)) \ge \sum_{e_{u,v} \in E_2} f_\varepsilon(c(u,v)) \tag{9}$$

then

$$\begin{aligned} &(1+\varepsilon)\lambda \sum_{e_{u,v} \in E_1} c(u,v) \\ &\ge \sum_{e_{u,v} \in E_1} f_\varepsilon(c(u,v)) \\ &\ge \sum_{e_{u,v} \in E_2} f_\varepsilon(c(u,v)) \ge \lambda \sum_{e_{u,v} \in E_2} c(u,v)) \\ &\Rightarrow (1+\varepsilon) \sum_{e_{u,v} \in E_1} c(u,v) \ge \sum_{e_{u,v} \in E_2} c(u,v)). \end{aligned} \tag{10}$$

By *Proposition 1*, $f_\varepsilon(\cdot)$ is a feasible ε-approximation function. ■

Different from the functions satisfying *Proposition 1*, the ones satisfying *Proposition 2* may be only feasible to a given instance of DCLC in which link costs are already given; i.e., they may not be universally feasible. Since $\forall e_{u,v} \in E, f_\varepsilon(c(u,v)) \in \mathbb{Z}^+$ and $f_\varepsilon(c(u,v)) \in [\lambda c(u,v), (1+\varepsilon)\lambda c(u,v)]$, a straightforward solution is $f_\varepsilon(c(u,v)) = \lceil \lambda c(u,v) \rceil$. It should also be noted that for a given λ, $f_\varepsilon(c(u,v)) = \lceil \lambda c(u,v) \rceil$ may not be feasible, since there may exist $e_{u,v} \in E$ such that

$$(1+\varepsilon)c(u,v)\lambda < f_\varepsilon(c(u,v)). \tag{11}$$

Hence, we will try to compute λ such that $\forall e_{u,v} \in E$, (7) is satisfied. As shown in Fig. 1, given a link $e_{u,v}$, because of the

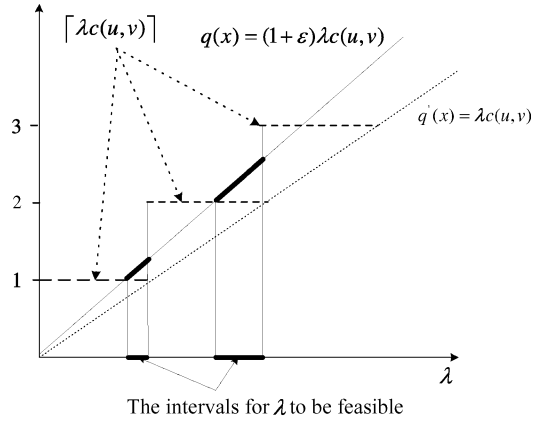

Fig. 1. Illustration of the feasible region of the linear scaling factor for a given link.

discrete nature of $\lceil \lambda c(u,v) \rceil$, there are many intersections between $(1+\varepsilon)\lambda c(u,v)$ and $\lceil \lambda c(u,v) \rceil$, and accordingly, the region for λ satisfying (7) is split into many intervals, which are highlighted with bold lines. We refer to the union of these intervals (of $e_{u,v}$) as the feasible region of $e_{u,v}$, and these intervals as the feasible intervals of $e_{u,v}$. Therefore, our objective is to find a point which exists in all feasible regions of links, i.e., to find λ such that $f_\varepsilon(x) = \lceil \lambda x \rceil$ is a feasible function. In other words, we need to find the λ that is located in the intersection of the feasible regions of all links. From *Definition 4*, the smaller the λ, the better. Hence, the optimal feasible linear scaling factor is the lower bound of the intersection of the feasible regions of all links. We numerically present the feasible intervals of a link by the next theorem.

Theorem 3: Given an instance of DCLC, $f_\varepsilon(x) = \lceil \lambda x \rceil$ is a feasible ε-approximation function if $\forall e_{u,v} \in E, \exists k_{u,v} \in \overline{\mathbb{Z}^-}(\{0\} \cup \mathbb{Z}^+)$ such that

$$k_{u,v} + \max\left\{0, \frac{1-\varepsilon k_{u,v}}{1+\varepsilon}\right\} \le \lambda c(u,v) \le k_{u,v} + 1. \quad (12)$$

Proof: $\forall e_{u,v} \in E, \exists k_{u,v} \in \overline{\mathbb{Z}^-}$ such that

$$\begin{aligned} k_{u,v} + \max\left\{0, \frac{1-\varepsilon k_{u,v}}{1+\varepsilon}\right\} &\le \lambda c(u,v) \\ &\le k_{u,v} + 1 \\ \Rightarrow \lceil \lambda c(u,v) \rceil &\le k_{u,v} + 1. \end{aligned} \quad (13)$$

Consider the case that $k_{u,v} < (1/\varepsilon)$

$$\begin{aligned} \lceil \lambda c(u,v) \rceil &\le k_{u,v} + 1 \\ &= \left[k_{u,v} + \max\left\{0, \frac{1-\varepsilon k_{u,v}}{1+\varepsilon}\right\}\right](1+\varepsilon) \\ &\le (1+\varepsilon)\lambda c(u,v). \end{aligned} \quad (14)$$

Consider the other case that $k_{u,v} \ge (1/\varepsilon)$

$$k_{u,v} + \max\left\{0, \frac{1-\varepsilon k_{u,v}}{1+\varepsilon}\right\} = k_{u,v}. \quad (15)$$

Therefore

$$\begin{aligned} (1+\varepsilon)\lambda c(u,v) &= \lambda c(u,v) + \varepsilon \lambda c(u,v) \\ &\ge k_{u,v} + k_{u,v}\varepsilon \\ &\ge k_{u,v} + 1 \ge \lceil \lambda c(u,v) \rceil. \end{aligned} \quad (16)$$

Therefore, $f_\varepsilon(x) = \lceil \lambda x \rceil$ is a feasible ε-approximation function by *Proposition 2*. ■

Definition 5: λ is feasible if $f_\varepsilon(x) = \lceil \lambda x \rceil$ is a feasible ε-approximation function, and λ is optimal if $f_\varepsilon(x) = \lceil \lambda x \rceil$ is the optimal feasible linear scaling ε-approximation function. λ is said to be feasible to a link $e_{u,v}$ if $\exists k_{u,v} \in \overline{\mathbb{Z}^-}$ such that

$$k_{u,v} + \max\left\{0, \frac{1-\varepsilon k_{u,v}}{1+\varepsilon}\right\} \le \lambda c(u,v) \le k_{u,v} + 1. \quad (17)$$

Theorem 3 defines the constraints for $f_\varepsilon(x) = \lceil \lambda x \rceil$ to be a feasible ε-approximation function. Since our final objective is to minimize the computational complexity of ε-approximation algorithms by finding the least feasible linear scaling factor, it is preferable that the computational complexity introduced by computing the optimal λ is trivial, or negligible, with respect to the overall computational complexity of ε-approximation algorithms.

Let

$$\alpha_k^{e(u,v)} = \frac{k}{c(u,v)} + \max\left\{0, \frac{1-k\varepsilon}{(1+\varepsilon)c(u,v)}\right\} \quad (18)$$

$$\beta_k^{e(u,v)} = \frac{k+1}{c(u,v)}. \quad (19)$$

By *Theorem 3*, if $f_\varepsilon(x) = \lceil \lambda x \rceil$ is feasible, $\lambda \in \bigcap_{e_{u,v} \in E} \zeta_{u,v}$, where

$$\zeta_{u,v} = \bigcup_{k=0}^{\infty} \left[\alpha_k^{e(u,v)}, \beta_k^{e(u,v)}\right]. \quad (20)$$

The minimum of $\bigcap_{e_{u,v} \in E} \zeta_{u,v}$ is the optimal λ. Given a link $e(w,x) \in E$

$$\begin{aligned} &\bigcap_{e_{u,v} \in E} \zeta_{u,v} \\ &= \left(\bigcup_{k=0}^{\infty} \left[\alpha_k^{e(w,x)}, \beta_k^{e(w,x)}\right]\right) \bigcap_{e(u,v) \ne e(w,x)} \zeta_{u,v} \\ &= \bigcup_{k=0}^{\infty} \left(\left[\alpha_k^{e(w,x)}, \beta_k^{e(w,x)}\right] \bigcap_{e(u,v) \ne e(w,x)} \zeta_{u,v}\right). \end{aligned} \quad (21)$$

We convert the problem of finding the optimal linear scaling factor into computing the minimum of (21). The remaining problem becomes designing an efficient algorithm for computing the minimum. The most intuitive and straightforward solution is to iteratively search for the feasible λ in $[\alpha_k^{e(w,x)}, \beta_k^{e(w,x)}]$, where k is initialized as zero and increased by one after each iteration. Note that $\lambda \in \mathbb{R}^+$ and there is exactly only one interval $[\alpha_k^{e(u,v)}, \beta_k^{e(u,v)}]$ in $((k-1)/c(u,v), (k/c(u,v))]$ for λ to be feasible to link

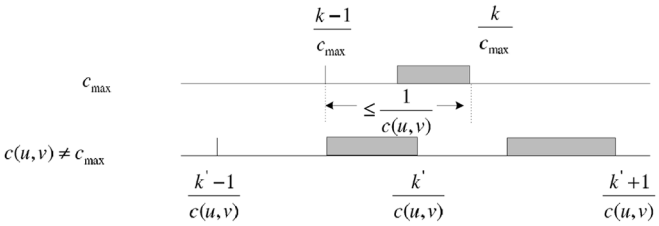

Fig. 2. Shaded regions are the regions for λ to be feasible to the corresponding links.

$e(u,v)$. We divide $\mathbb{R}^+$ into an infinite number of periods, $((k-1)/c(u,v),(k/c(u,v))]$ (each period has a fixed length of $(1/c(u,v))$, and the kth period is from $(k-1)/c(u,v)$ to $(k/c(u,v))$, $k \in \mathbb{Z}^+$ for each link $e(u,v)$. For each particular integer k, we consider the shortest (smallest) period $\varkappa_k$, which is a period of the link with the largest link cost. Intuitively, given any number of consecutive periods of any other link, at most two of them have nonempty intersections with $\varkappa_k$, as each of them has a length larger than that of $\varkappa_k$. Theoretically, as shown in Fig. 2, since $(1/c(u,v)) \geq (1/c_{\max})$, for all $e(u,v) \in E$, at most two consecutive periods of link $e(u,v)$ would have nonempty intersections with one period of the link(s) having costs $c_{\max}$. As such, we can simplify the process of computing the minimum of (21) by the following theorem. Denote $\alpha_k = (k + \max\{0,(1-\varepsilon k)/(1+\varepsilon)\})/c_{\max}, \beta_k = (k+1)/c_{\max}, k^-_{u,v} = \lfloor \alpha_k c(u,v) \rfloor$, and $k^+_{u,v} = \lfloor \beta_k c(u,v) \rfloor$, we can prove the next theorem.

Theorem 4: Given an instance of DCLC and $k \in \overline{\mathbb{Z}^-}$, a feasible ε-approximation function $f_\varepsilon(x) = \lceil \lambda x \rceil, \alpha_k \leq \lambda \leq \beta_k$, exists iff $\bigcap_{e(u,v)\in E} \pi_{u,v} \neq \Phi$, where

$$\pi_{u,v} = \left[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}\right] \cup \left[\alpha_{k^+}^{e(u,v)}, \beta_{k^+}^{e(u,v)}\right] \cup \left[\alpha_k, \frac{k^-_{u,v}}{c(u,v)}\right] \tag{22}$$

and Φ is the empty set.

Proof: Note that given a link $e_{u,v} \in E$, only when $\alpha_k c(u,v) \in \mathbb{Z}^+, [\alpha_k, (k^-_{u,v}/c(u,c))] \neq \Phi$, i.e., $[\alpha_k, (k^-_{u,v}/c(u,c))]$ is a point only when $\alpha_k c(u,v) \in \mathbb{Z}^+$. When $k^-_{u,v} = k^+_{u,v}$

$$\left[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}\right] = \left[\alpha_{k^+}^{e(u,v)}, \beta_{k^+}^{e(u,v)}\right]. \tag{23}$$

Furthermore, $k^+_{u,v} - k^-_{u,v} \in \{0,1\}$ because

$$\begin{aligned} &\beta_k c(u,v) - \alpha_k c(u,v) \\ &\quad = c(u,v)(\beta_k - \alpha_k) \\ &\quad \leq 1 - \max\left\{0, \frac{1-\varepsilon k}{1+\varepsilon}\right\} \leq 1 \\ &\quad \Rightarrow \lfloor \beta_k c(u,v) \rfloor - \lfloor \alpha_k c(u,v) \rfloor \leq 1. \end{aligned} \tag{24}$$

We first prove that if a feasible ε-approximation function $f_\varepsilon(x) = \lceil \lambda x \rceil, \alpha_k \leq \lambda \leq \beta_k$ exists, then $\bigcap_{e_{u,v}\in E} \pi_{u,v} \neq \Phi$. Given a link $e_{u,v}, c(u,v) \neq \max_{e_{w,x}\in E}\{c(w,x)\}$. Consider the case that $k^-_{u,v} = k^+_{u,v}$. By *Theorem 1*, if $f_\varepsilon(x) = \lceil \lambda x \rceil$ is a feasible ε-approximation function, $\exists k_{u,v} \in \overline{\mathbb{Z}^-}$ such that

$$\begin{aligned} &k_{u,v} + \max\left\{0, \frac{1-\varepsilon k_{u,v}}{1+\varepsilon}\right\} \\ &\quad \leq \lambda c(u,v) \\ &\quad \leq k_{u,v} + 1 \\ &\quad \Rightarrow \frac{k_{u,v}}{c(u,v)} + \max\left\{0, \frac{1-\varepsilon k_{u,v}}{(1+\varepsilon)c(u,v)}\right\} \\ &\quad \leq \lambda \leq \frac{k_{u,v}+1}{c(u,v)}. \end{aligned} \tag{25}$$

If $k_{u,v} < \lfloor \alpha_k c(u,v) \rfloor$

$$\frac{k_{u,v}+1}{c(u,v)} \leq \frac{\lfloor \alpha_k c(u,v) \rfloor}{c(u,v)} \leq \frac{\alpha_k c(u,v)}{c(u,v)} = \alpha_k. \tag{26}$$

The equation is held only when $k_{u,v} = \lfloor \alpha_k c(u,v) \rfloor - 1$ and $\alpha_k c(u,v) \in \mathbb{Z}^+$ (it is impossible that $\alpha_k c(u,v) = 0$ because $\alpha_k > 0$ and $c(u,v) > 0$). If $k_{u,v} > \lfloor \beta_k c(u,v) \rfloor$

$$\begin{aligned} \frac{k_{u,v}}{c(u,v)} + \max\left\{0, \frac{1-\varepsilon k_{u,v}}{(1+\varepsilon)c(u,v)}\right\} &\geq \frac{\lfloor \beta_k c(u,v) \rfloor + 1}{c(u,v)} \\ &> \frac{\beta_k c(u,v)}{c(u,v)} = \beta_k. \end{aligned} \tag{27}$$

Since $\alpha_k \leq \lambda \leq \beta_k$, if $\alpha_k c(u,v) \notin \mathbb{Z}^+$

$$\begin{aligned} k^-_{u,v} &= \lfloor \alpha_k c(u,v) \rfloor \\ &\leq k_{u,v} \leq \lfloor \beta_k c(u,v) \rfloor \\ &= k^+_{u,v}. \end{aligned} \tag{28}$$

Therefore

$$\begin{aligned} &k^+_{u,v} - k^-_{u,v} \in \{0,1\} \\ &\quad \Rightarrow \lambda \in [\alpha_k, \beta_k] \cap \left(\left[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}\right] \right. \\ &\qquad \left. \cup \left[\alpha_{k^+}^{e(u,v)}, \beta_{k^+}^{e(u,v)}\right] \cup \left[\alpha_k, \frac{k^-_{u,v}}{c(u,v)}\right] \right). \end{aligned} \tag{29}$$

So, if a feasible ε-approximation function $f_\varepsilon(x) = \lceil \lambda x \rceil, \alpha_k \leq \lambda \leq \beta_k$ exists, $\bigcap_{e_{u,v}\in E} \pi_{u,v} \neq \Phi$, where

$$\begin{aligned} \pi_{u,v} = [\alpha_k, \beta_k] \cap &\left(\left[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}\right] \right. \\ &\left. \cup \left[\alpha_{k^+}^{e(u,v)}, \beta_{k^+}^{e(u,v)}\right] \cup \left[\alpha, \frac{k^-_{u,v}}{c(u,v)}\right] \right). \end{aligned} \tag{30}$$

Note that $\pi_{u,v} = [\alpha_k, \beta_k]$ if $c(u,v) = \max_{e_{w,x}\in E}\{c(w,x)\}$. We can simplify $\pi_{u,v}$ as

$$\pi_{u,v} = \left[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}\right] \cup \left[\alpha_{k^+}^{e(u,v)}, \beta_{k^+}^{e(u,v)}\right] \cup \left[\alpha_k, \frac{k^-_{u,v}}{c(u,v)}\right]. \tag{31}$$

If $\bigcap_{e_{u,v}\in E} \pi_{u,v} \neq \Phi$, by *Theorem 3*, a feasible ε-approximation function $f_\varepsilon(x) = \lceil \lambda x \rceil, \alpha_k \leq \lambda \leq \beta_k$, exists. ■

Next, we will try to find the smallest feasible λ based on *Theorem 4*. Since $[\alpha_k, (k_{u,v}^-/c(u,v))]$ is a point only when $\alpha_k c(u,v) \in \mathbb{Z}^+$, the point can be eliminated by checking if α_k is the solution (optimal λ). If α_k is feasible, the smallest feasible λ in $[\alpha_k, \beta_k]$ must be α_k. Otherwise, we set

$$\pi_{u,v} = \left[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}\right] \cup \left[\alpha_{k^+}^{e(u,v)}, \beta_{k^+}^{e(u,v)}\right]. \tag{32}$$

Let

$$\begin{aligned}\pi &= \bigcap_{k_{u,v}^- = k_{u,v}^+} \pi_{u,v} \\ &= \bigcap_{k_{u,v}^- = k_{u,v}^+} \left[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}\right] \\ &= \left[\max_{k_{u,v}^- = k_{u,v}^+} \left\{\alpha_{k^-}^{e(u,v)}\right\}, \min_{k_{u,v}^- = k_{u,v}^+} \left\{\beta_{k^-}^{e(u,v)}\right\}\right]. \end{aligned} \tag{33}$$

The remaining problem is to compute

$$\pi \cap \left(\bigcap_{k_{u,v}^- \neq k_{u,v}^+} \pi_{u,v}\right). \tag{34}$$

It should be noted that if $\pi_{u,v}(e_{u,v} \in E)$ are independent with each other, the computational complexity of computing $\pi \cap (\bigcap_{k_{u,v}^- \neq k_{u,v}^+} \pi_{u,v})$ could be rather high. Hence, the next theorem is introduced to further reduce the complexity of computing $\pi \cap (\bigcap_{k_{u,v}^- \neq k_{u,v}^+} \pi_{u,v})$. The basic idea behind the next theorem is to find the condition such that only one of $[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}]$ and $[\alpha_{k^+}^{e(u,v)}, \beta_{k^+}^{e(u,v)}]$ has a nonempty intersection with $[\alpha_k, \beta_k]$. As such, we can reduce the worst-case computational complexity of computing (34) to $O(m)$.

Theorem 5: Given a link $e_{u,v} \in E$, if $k_{u,v}^+ - k_{u,v}^- = 1$, and $c(u,v) < c_{\max}/(1+\varepsilon)$, either

$$\left[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}\right] \cap [\alpha_k, \beta_k] = \Phi \tag{35}$$

$$\text{or} \quad \left[\alpha_{k^+}^{e(u,v)}, \beta_{k^+}^{e(u,v)}\right] \cap [\alpha_k, \beta_k] = \Phi. \tag{36}$$

Proof: When $k_{u,v}^+ - k_{u,v}^- = 1$

$$\begin{aligned}&\frac{k_{u,v}^+}{c(u,v)} + \max\left\{0, \frac{1 - k_{u,v}^+ \varepsilon}{(1+\varepsilon)c(u,v)}\right\} \\ &\quad - \frac{k_{u,v}^-}{c(u,v)} - \max\left\{0, \frac{1 - k_{u,v}^- \varepsilon}{(1+\varepsilon)c(u,v)}\right\} \\ &= \frac{1}{c(u,v)} + \max\left\{0, \frac{1 - k_{u,v}^+ \varepsilon}{(1+\varepsilon)c(u,v)}\right\} \\ &\quad - \max\left\{0, \frac{1 - k_{u,v}^- \varepsilon}{(1+\varepsilon)c(u,v)}\right\} \\ &\geq \frac{1}{c(u,v)} - \frac{\varepsilon}{(1+\varepsilon)c(u,v)} \\ &= \frac{1}{(1+\varepsilon)c(u,v)} > \frac{1}{c_{\max}} \end{aligned} \tag{37}$$

while

$$\beta_k - \alpha_k = \frac{1 - \max\left\{0, \frac{1-\varepsilon k}{1+\varepsilon}\right\}}{c_{\max}} \leq \frac{1}{c_{\max}}. \tag{38}$$

Hence, either

$$\left[\alpha_{k^-}^{e(u,v)}, \beta_{k^-}^{e(u,v)}\right] \cap [\alpha_k, \beta_k] = \Phi \tag{39}$$

$$\text{or} \quad \left[\alpha_{k^+}^{e(u,v)}, \beta_{k^+}^{e(u,v)}\right] \cap [\alpha_k, \beta_k] = \Phi. \tag{40}$$

By *Theorem 5*, given a network in which $\forall e(u,v)$, either $c(u,v) = c_{\max}$ or $c(u,v) < c_{\max}/(1+\varepsilon)$, we have

$$\left[\alpha_{n(u,v)}^{e(u,v)}, \beta_{n(u,v)}^{e(u,v)}\right] \cap [\alpha_k, \beta_k] \neq \Phi \tag{41}$$

only when $n(u,v) \in \{k_{u,v}^+, k_{u,v}^-\}$, where $n(u,v)$ is the integer such that (41) is satisfied. Hence

$$\begin{aligned}\bigcap_{e_{u,v} \in E} \pi_{u,v} &= \bigcap_{e_{u,v} \in E} \left[\alpha_{n(u,v)}^{e(u,v)}, \beta_{n(u,v)}^{e(u,v)}\right] \\ &= \left[\max_{e_{u,v} \in E} \left\{\alpha_{n(u,v)}^{e(u,v)}\right\}, \min_{e_{u,v} \in E} \left\{\beta_{n(u,v)}^{e(u,v)}\right\}\right]. \end{aligned} \tag{42}$$

In other words

$$\begin{aligned}1 + \varepsilon &< \frac{c_{\max}}{\max_{c(u,v) \neq c_{\max}}\{c(u,v)\}} \\ &\Longrightarrow \bigcap_{e_{u,v} \in E} \pi_{u,v} = \bigcap_{e_{u,v} \in E} \left[\alpha_{n(u,v)}^{e(u,v)}, \beta_{n(u,v)}^{e(u,v)}\right] \\ &= \left[\max_{e_{u,v} \in E} \left\{\alpha_{n(u,v)}^{e(u,v)}\right\}, \min_{e_{u,v} \in E} \left\{\beta_{n(u,v)}^{e(u,v)}\right\}\right]. \end{aligned} \tag{43}$$

Accordingly, the computational complexity of computing a feasible linear scaling factor in $[\alpha_k, \beta_k]$ becomes $O(m)$, which results from computing $\max_{e_{u,v} \in E} \left\{\alpha_{n(u,v)}^{e(u,v)}\right\}$ and $\min_{e_{u,v} \in E} \left\{\beta_{n(u,v)}^{e(u,v)}\right\}$.

IV. Proposed Algorithms for Searching the Optimal Linear Scaling Factor

In this section, we will propose two efficient algorithms for searching the optimal linear scaling factor, which can be incorporated into ε-approximation algorithms to reduce their computational complexities. We first introduce the next theorem, based on which, an upper bound on the optimal linear scaling factor can be derived.

Theorem 6: Given an instance of DCLC, the linear scaling ε-approximation solution computed with a linear scaling factor λ has a cost less than

$$c^* + \frac{h}{\lambda} \tag{44}$$

where c^* and h are the cost and the number of hops of the optimal feasible path of DCLC, respectively.

Proof: Assume the optimal path of DCLC is p, and the path computed by using a linear scaling factor λ is $\hat{p}$. Therefore

$$\begin{aligned}\sum_{e(u,v)\in\hat{p}} c(u,v)\lambda &\le \sum_{e(u,v)\in\hat{p}} \lceil c(u,v)\lambda \rceil \\ &\le \sum_{e(u,v)\in p} \lceil c(u,v)\lambda \rceil \\ &< \sum_{e(u,v)\in p} [c(u,v)\lambda + 1] \\ &= \sum_{e(u,v)\in p} c(u,v)\lambda + h. \end{aligned} \tag{45}$$

Hence, the feasible path computed with linear scaling factor λ has a cost less than $c^* + (h/\lambda)$. ■

Based on *Theorem 6*, the next lemma provides an upper bound on the optimal feasible linear scaling factor of a given instance of DCLC.

Lemma 7: Given an instance of DCLC, the linear scaling ε-approximation solution computed by using a linear scaling factor $\lambda = (n-1)/L\varepsilon$ is an ε-approximation solution, where L is a lower bound on c^*.

Proof: By *Theorem 6*, the path computed with the linear scaling factor $\lambda = (n - 1/L\varepsilon)$ has a cost less than

$$c^* + \frac{h}{\lambda} = c^* + \frac{h}{n-1} L\varepsilon \le c^* + L\varepsilon \le c^*(1+\varepsilon). \tag{46}$$

Hence, the path computed with a linear scaling factor $\lambda = (n - 1)/L\varepsilon$ is a ε-approximation solution. ■

How to compute L is beyond the scope of this paper. Readers can refer to [20] and [24] for related information. We adopt $\mu = (n-1)/L\varepsilon$ as an upper bound on the optimal linear scaling factor by *Lemma 7*.

The first algorithm, OLSA, is recommended for cases where $\forall e(u,v) \in E$, either $c(u,v) = c_{\max}$ or $c(u,v) < c_{\max}/(1+\varepsilon)$, so that *Theorem 5* can be applied to reduce the computational complexity for searching the optimal linear scaling factor. As mentioned before, we use μ as a loose upper bound for the optimal linear scaling factor. The basic idea behind OLSA is that the optimal linear scaling factor λ can be found by *Theorems 4* and *5* by gradually increasing the integer k until λ reaches μ, implying that k is limited by

$$k_{\max} = \lfloor \mu c_{\max} \rfloor \tag{47}$$

in OLSA. Without loss of generality, we assume $c_{\max} = 1$ for the rest of the paper, which can be achieved simply by dividing all the link costs by $c_{\max}$.

Thus, OLSA consists of the following steps.

Step 1) Initialize $k = 0$.

Step 2) Apply *Theorems 4* and *5* to find a feasible linear scaling factor in $[k + \max\{0, (1-\varepsilon k)/(1+\varepsilon)\}, k+1]$. If $\bigcap \pi_{u,v} \neq \Phi$ at the kth iteration, the optimal feasible λ must be the lower bound of $\bigcap \pi_{u,v}$. Hence, set λ to the lower bound of $\bigcap \pi_{u,v}$, and the optimal feasible linear scaling factor is found. End.

Step 3) $k = k + 1$. If $k = k_{\max}$, go to step 4. Otherwise, go to step 2.

Step 4) By *Lemma 7*, let $\lambda = \mu$.

Since the computational complexity of computing a feasible linear scaling factor in $[k + \max\{0, (1-\varepsilon k)/(1+\varepsilon)\}, k+1]$ is, by *Theorems 4* and *5*, $O(m)$, and there are totally $\lceil \lambda c_{\max} \rceil$ iterations in OLSA, the computational complexity of OLSA is

$$O(\lceil \lambda c_{\max} \rceil m) = O(\lceil \lambda \rceil m). \tag{48}$$

It can be observed that the worst-case computational complexity of OLSA is very low. On the other hand, since OLSA always locates the optimal λ, the computational complexities of ε-approximation algorithms deploying OLSA can be reduced, especially in some special cases. For example, for a given network $G(N,E)$, and $\forall e(u,v) \in E, c(u,v) \in \{k/3, k = 1,2,\ldots\}, \lambda = 3$ is a feasible linear scaling factor, which is independent of ε. As a result, the computational complexities of ε-approximation algorithms are greatly reduced when ε is small. Moreover, in this case, the computational complexities become polynomial, no longer pseudopolynomial.

Remark 1: As mentioned earlier, most ε-approximation approaches adopt Bellman–Ford-like algorithms, e.g., RSP [20] and DAD [23]. Therefore, their worst-case computational complexities are linearly proportional to the upper bound of the path costs (the computational complexity of DAD is linearly proportional to the delay bound). For example, given a delay bound d and an upper bound U on the cost of the optimal feasible path, the computational complexity of RSP [20] is $O(mU)$ (it should be noted that link costs must be integers to deploy RSP). Given a linear scaling factor λ and an upper bound U on path costs, in order to achieve a 100% success ratio in finding the optimal feasible path with a cost less than U, the upper bound is adjusted to $O(\lambda U)$ after all link costs are linearly scaled by λ. Accordingly, the computational complexities of ε-approximation algorithms are proportional to $O(m\lceil \lambda U \rceil)$. Note that U must be larger than $c_{\max}$. Otherwise, we can prune the links whose costs are larger than U. Therefore

$$O(m\lceil \lambda U \rceil) > O(\lceil \lambda c_{\max} \rceil m) = O(\lceil \lambda \rceil m) \tag{49}$$

which implies that adopting OLSA will not increase the computational complexities of ε-approximation algorithms. Note that the upper bound of λ adopted in this paper is $\mu = (n-1)/L\varepsilon$, and in [24], efficient methods have been introduced to estimate U and L such that $(U/L) \le \alpha$, where α is a constant. For instance, the authors provided an efficient algorithm in ε-OPQR (optimal QoS partition and routing) which gurantees $(U/L) \le 8$. Therefore, the computational complexity of OLSA is bounded by

$$O(m\lceil \lambda U \rceil) < O\left(m \left\lceil U \frac{n-1}{L\varepsilon} \right\rceil\right) = O\left(\frac{mn}{\varepsilon}\right) \tag{50}$$

which is, to our best knowledge, no larger than the computational complexity of any DCLC ε-approximation solution,

implying that OLSA can be adopted by ε-approximation algorithms to minimize their linear scaling factors without increasing their computational complexities. On the other hand, since the computational complexities of ε-approximation algorithms are linearly proportional to the linear scaling factor, their computational complexities can be reduced by adopting OLSA. Finally, we need to point out that since OLSA is proposed under the assumption that $\lceil\cdot\rceil$ is used for scaling, it may not be directly applicable to algorithms, e.g., DSA [23] and S-OPQR [20], in which $\lfloor\cdot\rfloor$ is deployed.

Note that OLSA is not applicable to cases where $\exists e(u,v) \in E$ such that $c(u,v) \neq c_{\max}$ and $c(u,v) \geq c_{\max}/(1+\varepsilon)$. Thus, we propose our second algorithm, T-OLSA, by first setting the costs of the links with costs no less than $c_{\max}/(1+\varepsilon)$ to $c_{\max}$, and then applying OLSA.

Theorem 8: Given an instance of DCLC, construct $G'(V,E)$ by setting the costs of the links in $G(V,E)$ whose costs are not less than $c_{\max}/(1+\varepsilon)$ to $c_{\max}$, and then construct $\tilde{G}(V,E)$ from $G'(V,E)$ by linearly scaling the link costs by λ, where λ is the optimal linear scaling factor computed by OLSA in $G'(V,E)$. Assume the least-cost feasible path in $\tilde{G}(V,E)$ is $\tilde{p}$; $\tilde{p}$ has a cost in $G(V,E)$ no larger than $(1+\varepsilon)^2 c^*$, where c^* is the cost of the optimal feasible path in $G(V,E)$.

Proof: Assume p is the optimal feasible path of $G(V,E)$, p' is the optimal feasible path in $G'(V,E)$, and $c'(p')$ is the cost of p' in $G'(V,E)$. Since $\tilde{G}(V,E)$ is constructed from $G'(V,E)$ by linearly scaling the link costs by λ, and $\tilde{p}$ is the optimal feasible path of $\tilde{G}(V,E)$ (or $\tilde{p}$ is the solution of an ε-approximation algorithm computed in $\tilde{G}(V,E)$)

$$c'(\tilde{p}) \leq (1+\varepsilon)c'(p'). \tag{51}$$

Define $c'(u,v)$ as the cost of link $e_{u,v}$ in $G'(V,E)$. Since p' is the optimal feasible path in $G'(V,E)$, and $G'(V,E)$ is constructed by setting the costs of the links in $G(V,E)$ whose costs are no less than $(c_{\max}/1+\varepsilon)$ to $c_{\max}$, implying that for any link $e(u,v)$, $c'(u,v) \leq (1+\varepsilon)c(u,v)$, and thus

$$\begin{aligned} c'(p') \leq c'(p) &= \sum_{e_{u,v}\in p} c'(u,v) \\ &\leq \sum_{e_{u,v}\in p} c(u,v)(1+\varepsilon) \\ &= (1+\varepsilon)c^* \end{aligned}$$

where $c'(p)$ is the cost of p in $G'(V,E)$. Thus

$$c'(\tilde{p}) \leq (1+\varepsilon)c'(p') \leq (1+\varepsilon)^2 c^*. \tag{52}$$

Therefore

$$c(\tilde{p}) \leq c'(\tilde{p}) \leq (1+\varepsilon)^2 c^* \tag{53}$$

where $c(\tilde{p})$ is the cost of $\tilde{p}$ in $G(V,E)$, i.e., $\tilde{p}$ has a cost in $G(V,E)$ no larger than $(1+\varepsilon)^2 c^*$. ■

Thus, T-OLSA consists of the following steps.

Step 1) Set the costs of the links whose cost are no less than $c_{\max}/(1+\varepsilon')$ as $c_{\max}$, where $\varepsilon' = \sqrt{1+\varepsilon} - 1$.

Step 2) Initialize $k = 0$.

Step 3) Apply *Theorems 4* and *5* to find a feasible linear scaling factor in $[k+\max\{0,(1-\varepsilon' k)/(1+\varepsilon')\}, k+1]$. If $\bigcap \pi_{u,v} \neq \Phi$ at the kth iteration, the optimal feasible λ must be the lower bound of $\bigcap \pi_{u,v}$. Hence, set λ to the lower bound of $\bigcap \pi_{u,v}$, and the smallest feasible linear scaling factor is found. End.

Step 4) $k = k+1$. If $k = k_{\max}$, go to step 5. Otherwise, go to step 2.

Step 5) By *Lemma 7*, let $\lambda = \mu$ and set $\varepsilon' = \varepsilon$.

Lemma 9: Given an instance of DCLC and ε, by letting $\varepsilon' = \sqrt{1+\varepsilon} - 1$, a path with cost no larger than $(1+\varepsilon)c^*$ can be found by an ε-approximation algorithm (with parameter ε') if T-OLSA is adopted for finding the linear scaling factor.

Proof: Assume the path computed by the ε-approximation algorithm with parameter $\varepsilon' = \sqrt{1+\varepsilon} - 1$ is p. Therefore, by *Theorem 8*

$$c(p) \leq (1+\varepsilon')^2 c^* = (1+\varepsilon)c^* \tag{54}$$

where c^* is the cost of the optimal feasible path. ■

Observe that T-OLSA does not totally rely on OLSA. Since $\varepsilon' < \varepsilon$, by *Lemma 1*, the computational complexity may unnecessarily increase if λ is larger than μ. In this case, we can directly solve the original problem (link costs are not modified) by setting

$$\lambda = \mu. \tag{55}$$

Therefore, the worst-case computational complexity of T-OLSA is the same as that of OLSA.

V. Conclusion

In this paper, having observed that the computational complexity of many ε-approximation algorithms is linearly proportional to the linear scaling factor, we have investigated the issue of finding the optimal linear scaling factor in order to reduce their computational complexities. Two algorithms, OLSA and T-OLSA, have been proposed. We have analytically shown that incorporating the two algorithms into DCLC ε-approximation solutions not only does not increase, but, in fact, reduces their computational complexities because the optimal linear scaling factor can always be found. Besides the DCLC ε-approximation solutions, our proposed algorithms can be applied to all linear ε-approximation solutions in which $\lceil\cdot\rceil$ is used for scaling. Similarly, algorithms can also be developed for cases in which $\lfloor\cdot\rfloor$ is deployed for scaling.

References

[1] S. Chen and K. Nahsted, "An overview of quality-of-service routing for next-generation high-speed network: Problems and solutions," *IEEE Netw.*, vol. 12, no. 6, pp. 64–79, Jun. 1998.

[2] A. Shaikh, J. Rexford, and K. G. Shin, "Evaluating the impact of stale link state on quality-of-service routing," *IEEE/ACM Trans. Netw.*, vol. 9, no. 2, pp. 162–176, Apr. 2001.

[3] M. S. Garey and D. S. Johnson, *Computers and Intractability: Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.
[4] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 7, pp. 1228–1234, Jul. 1996.
[5] A. Juttner, B. Szyiatovszki, I. Mecs, and Rajko, "Lagrange relaxation based method for the QOS routing problem," in *Proc. IEEE INFOCOM*, 2001, vol. 2, pp. 859–868.
[6] X. Yuan, "Heuristic algorithm for multiconstrained quality-of-service routing," *IEEE/ACM Trans. Netw.*, vol. 10, no. 2, pp. 244–256, Apr. 2002.
[7] ——, "On the extended Bellman–Ford algorithm to solve two constrained quality of service routing problems," in *Proc. IEEE ICCCN*, 1999, pp. 304–310.
[8] G. Xue, "Minimum cost QoS multicast and unicast routing in communication networks," *IEEE Trans. Commun.*, vol. 51, no. 5, pp. 817–824, May 2003.
[9] ——, "Primal-dual algorithms for computing weight-constrained shortest paths and weight-constrained minimum spanning trees performance," in *Proc. IEEE IPCCC*, 2000, pp. 271–277.
[10] A. Orda and A. Sprintson, "Precomputation schemes for QoS routing," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 578–591, Aug. 2003.
[11] T. Korkmaz and M. Krunz, "Routing multimedia traffic with QoS guarantees," *IEEE Trans. Multimedia*, vol. 5, no. 3, pp. 429–443, Jun. 2003.
[12] G. Liu and K. G. Ramakrishnan, "A$*$Prune: An algorithm for finding K shortest paths subject to multiple constraints," in *Proc. IEEE INFOCOM*, 2001, vol. 2, pp. 743–749.
[13] T. Korkmaz and M. Krunz, "Bandwidth-delay constrained path selection under inaccurate state information," *IEEE/ACM Trans. Netw.*, vol. 11, no. 3, pp. 384–398, Jun. 2003.
[14] T. Korkmaz, M. Krunz, and S. Tragoudas, "An efficient algorithm for finding a path subject to two additive constraints," in *Proc. ACM SIGMETRICS*, 2000, pp. 318–327.
[15] C. Pomavalzi, G. Chakraborty, and N. Shiratori, "QoS based routing algorithm in integrated services packet networks," in *Proc. IEEE Conf. Netw. Protocols*, 1997, pp. 167–174.
[16] R. Widyono, "The design and evaluation of routing algorithms for real-time channels" Univ. California, Berkeley, Tech. Rep. TR-94-024, 1994.
[17] D. Eppstein, "Finding the k shortest path," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 154–165.
[18] H. De Neve and P. Van Mieghem, "A multiple quality of service routing algorithm for PNNI," in *Proc. IEEE ATM Workshop*, 1998, pp. 324–328.
[19] S. Chen and K. Nahrsted, "On finding multi-constrained path," in *Proc. IEEE ICC*, 1998, vol. 2, pp. 874–899.
[20] D. H. Lorenz, A. Orda, D. Raz, and Y. Shavitt, "Efficient QoS partition and routing of unicast and multicast," in *Proc. 8th Int. Workshop Quality of Service*, 2000, pp. 75–83.
[21] R. Hassin, "Approximation schemes for the restricted shortest path problem," *Math. Oper. Res.*, vol. 2, no. 2, pp. 36–42, 1992.
[22] D. Raz and Y. Shavitt, "Optimal partition of QoS requirements with discrete cost functions," *IEEE J. Sel. Areas Commun.*, vol. 12, no. 12, pp. 2593–2602, Dec. 2000.
[23] A. Goel, K. G. Ramakrishnan, D. Kataria, and D. Logothetis, "Efficient computation of delay-sensitive routes from one source to all destinations," in *Proc. IEEE INFOCOMM*, 2001, pp. 854–858.
[24] D. H. Lorenz and D. Raz, "A simple efficient approximation scheme for the restricted shortest path problem," *Oper. Res. Lett.*, vol. 28, no. 5, pp. 213–219, 2001.

Gang Cheng received the B.S. degree in 1997 in information engineering, and the M.S. degree in information and signal processing in 2000, both from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, and the Ph.D. degree in 2005 from the New Jersey Institute of Technology (NJIT), Newark.

He joined Lucent Technologies in 2000. Currently, he is with VPIsystems Corporation, Holmdel, NJ, focusing on network planning optimization algorithm design and development. His research interests include Internet routing protocols and service architectures, information-theory-based network optimization and protocol design, and modeling and performance evaluation of computer and communication systems.

Dr. Cheng was the recipient of the Hashimoto Prize from NJIT, which is awarded annually to the best doctoral graduate in electrical and computer engineering.

Nirwan Ansari (S'78–M83–SM'94) received the B.S.E.E. degree (*summa cum laude*) from the New Jersey Institute of Technology (NJIT), Newark, in 1982, the M.S.E.E. degree from the University of Michigan, Ann Arbor, in 1983, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1988.

He joined the Department of Electrical and Computer Engineering, NJIT, as an Assistant Professor in 1988, and has been a Full Professor since 1997. He authored *Computational Intelligence for Optimization* (Norwell, MA: Kluwer, 1997) with E.S.H. Hou and translated into Chinese in 2000, and coedited *Neural Networks in Telecommunications* (Norwell, MA: Kluwer, 1994) with B. Yuhas. He is a Senior Technical Editor of the *IEEE Communications Magazine*, and also serves on the editorial board of *Computer Communications*, the *ETRI Journal*, and the *Journal of Computing and Information Technology*. His current research focuses on various aspects of broadband networks and multimedia communications. He has also contributed approximately 100 refereed journal articles, plus numerous conference papers and book chapters.

Dr. Ansari initiated (as the General Chair) the First IEEE International Conference on Information Technology: Research and Education (ITRE'03), was instrumental, while serving as its Chapter Chair, in rejuvenating the North Jersey Chapter of the IEEE Communications Society, which received the 1996 Chapter of the Year Award and a 2003 Chapter Achievement Award, served as Chair of the IEEE North Jersey Section and in the IEEE Region 1 Board of Governors during 2001–2002, and has been serving in various IEEE committees, such as TPC Chair/Vice Chair of several conferences. He was the 1998 recipient of the NJIT Excellence Teaching Award in Graduate Instruction, and a 1999 IEEE Region 1 Award. He is frequently invited to deliver keynote addresses, tutorials, and talks. He has been selected as an IEEE Communications Society Distinguished Lecturer (2006–2007).

Li Zhu received the B.S. and M.S. degrees in physics from Nanjing University, Nanjing, China, in 1994 and 1997, respectively, the M.S. degree in electrical engineer from Georgia Institute of Technology, Atlanta, in 2001, and the Ph.D. degree in electrical engineering from the New Jersey Institute of Technology, Newark, in 2006.

He is currently with Voicenet Inc., Philadelphia, PA, as a Senior Network Engineer. His current research interests include QoS in the Internet, overlay networks, and ad hoc networks.