Lowering Eccentricity of a Tree by Node-Upgrading

Toshihide Ibaraki

Department of Informatics, School of Science and Technology, Kwansei Gakuin University, 2-1 Gakuen, Sanda, Japan 669-1337

Yann Vaxès

Laboratoire d'Informatique Fondamentale de Marseille, Université de la Mèditerranée, Faculté des Sciences de Luminy, F-13288 Marseille Cedex 9, France

Xiao-guang Yang

Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China 100080

The eccentricity lowering problem is to reduce the eccentricity of a network by upgrading some nodes (i.e., shrinking the lengths of the edges incident to such nodes). We consider two types of node-upgrading strategies, i.e., a continuous upgrading strategy and a discrete upgrading strategy, where the improvement under the first strategy is a continuous variable, and the improvement under the second strategy is a fixed amount. These problems are hard even to approximate, for general graphs. Therefore we restrict our attention to graphs with simple structures. Assuming that the graph $G = (V, E)$ **is a tree, we show that the**

[0] The research is supported by the 21 Century COE Program of Japan "Informatics Research Center for Development of Knowledge Society Infrastracture", and National Key Research and Development Program of China (Grant No. 2002CB312004) and NSFC (Grant No. 70425004).

eccentricity lowering problem under the continuous node-upgrading strategy can be reduced to the eccentricity lowering problem under the continuous edge-upgrading strategy, and can be solved by an $O(|V|\log|V|)$ time algorithm. We also show that the problem for a tree is NP-hard under the discrete upgrading strategy, but admits a fully polynomial approximation scheme, if the graph is a line.

Keywords: eccentricity, node-upgrading, edge-upgrading, discrete upgrading strategy, continuous upgrading strategy, tree, line

1 Introduction

In a communication network, the transmission delay over the edges incident with a node can be reduced by node-upgrading (i.e., installing faster communication equipment at the node). This strategy is sometimes applied to improve the communication quality of the network.

Given a network, the eccentricity is defined as the largest distance from one designated node to the other nodes. Defining the transmission delay on an edge as its length, we consider the problem of reducing the eccentricity by node-upgrading at minimum cost. The problem has many potential applications in the real world such as broadcasting, facility-accessing and so forth.

Let a graph $G = (V, E)$ represent a communication network, where the node set V stands for transmission stations, and the edge set E stands for the links between transmission stations. The transmission delay $d(u, v)$ associated with an edge (u, v) from u to v can be decomposed into three parts: $s(u)$ the sending time from node u, $t(u, v)$ the transmission time over link (u, v), and $r(v)$ the receiving time at v; i.e., $d(u, v) = s(u) + t(u, v) + r(v)$. The transmission delay between any two nodes (adjacent or nonadjacent) is given by the length of the shortest path (in terms of edge lengths $d(u, v)$) between the two nodes. For any node v, upgrading it will shorten the sending time and the receiving time at v. We consider two types of upgrading strategies, a continuous upgrading strategy and a discrete upgrading strategy.

Under the continuous upgrading strategy, each node v is given two positive shrinking coefficients $\lambda_s(v) > 0$, $\lambda_r(v) > 0$ and a cost coefficient $c(v) \geq 0$. Also associated with v is a continuous variable $0 \leq x(v) \leq b(v)$ (called the improvement made at v), where $b(v)$ is the upper bound on the improvement. The sending time from node v under improvement $x(v)$ becomes $\max\{s(v) - \lambda_s(v)x(v), 0\}$ and the receiving time at v becomes $\max\{r(v) - \lambda_r(v)x(v), 0\}$. The eccentricity lowering problem under the continuous upgrading strategy is to find a vector $x = (x(v) \mid v \in V)$ that minimizes the total cost $\sum_{v \in V} c(v)x(v)$ under the constraints that the eccentricity from a designated node v_0 after the upgrading is at most U and $x(v) \leq b(v)$ holds for all nodes v. This problem is denoted as CELP (continuous version of the eccentricity lowering problem).

Under the discrete upgrading strategy, each node v is given three fixed values $0 \leq p_s(v) \leq s(v)$, $0 \leq p_r(v) \leq r(v)$ and $c(v) \geq 0$. If a node v is upgraded, the sending time will become $s(v) - p_s(v)$, and the receiving time will become $r(v) - p_r(v)$ at the cost of $c(v)$. The eccentricity lowering problem under the discrete upgrading strategy is to find a subset $S \subseteq V$ such that upgrading all nodes in S reduces the eccentricity of the designated node v_0

to at most U at the minimum total cost $\sum_{v \in S} c(v)$. We denote the problem as DELP (discrete version of the eccentricity lowering problem).

The difference between continuous and discrete upgrading strategies is that the cost and reduction under the continuous upgrading strategy depend linearly on the magnitude of the improvement $x(v)$ made at each node v, while those under the discrete upgrading strategy depend only on whether the nodes are upgraded or not.

It easily follows from the known results (as will be briefly surveyed in the next section) that both DELP and CELP are NP-hard. Furthermore it is not difficult to show that approximating CELP and DELP on a general graph is as hard as approximating the problem SETCOVER (see Appendix). So our attention focuses on graphs with simpler structures. The paper is organized as follows. In the next section we review the literature related to network upgrading problems, and give a brief comparison between our model and existing models. In Section 3 and Section 4, we discuss the two problems CELP and DELP respectively, and present the main results of this paper. In Section 5, we give concluding remarks.

2 Literature Review

The node-based upgrading problem was first studied by Paik and Sahni [10], in which the authors considered that the delay of an edge is reduced by a factor α if one endpoint of the edge is upgraded, and by α^2 if both endpoints of the edge are upgraded, where $0 \leq \alpha < 1$ is a constant. The objective is to find a minimum number of nodes to be upgraded such that there exists some subgraph with a specific structure to satisfy delay requirements. For example, problem $ShortPath(x, B)$ seeks the smallest number of nodes to be upgraded such that the length of the shortest path between any two nodes is less than or equal to B. This problem is the same as ours except that the upgrading strategy is different. Paik and Sahni showed that several node upgrading problems are NP-hard. Recently Krumke et al. [6, 7, 8, 9] generalized Paik and Sahni's model of [10]. They associated with each edge e three integers $d_0(e) \geq d_1(e) \geq d_2(e)$, where $d_i(e)$ describes the delay of e when exactly i of its endpoints are upgraded. The cost $c(v)$ of upgrading v is also associated with node v. The node upgrading problem considered by them are the bottleneck tree upgrading problem and the minimum length tree upgrading problem, which are to find a minimum cost set of upgraded nodes so that the resulting network has a spanning tree such that the maximum delay and the total delay of the spanning tree do not exceed given bounds. Krumke et al. [7, 8] presented $O(\log |V|)$ approximation algorithms for these problems.

There is a related class of network improvement problems called the edge-upgrading problem, where the lengths of edges are reduced. The definition of one such problem will be given in the next section. This class of problems has been studied in [2, 3, 12, 13]. For a general graph, it is hard to approximate within an $O(\log |V|)$ approximation ratio. If the graph is a tree, however, some polynomial algorithms have been presented [2, 12]. In particular, Zhang et al. [12] contains an $O(|V| \log |V|)$ time algorithm.

In concluding this section, we note two features about our model, which are introduced to describe the real world more precisely. In our node-upgrading model, the transmission delay over an edge is divided into three parts: sending time, receiving time and transmission time. Upgrading a node only shortens the sending and receiving times of this node, and the effects of upgrading a node on the sending time and receiving time can be different. This reflects the situation that installing new communication equipment at a node does not change the quality of the incident links. The discrete upgrading model represents the situation where the amount of improvement is fixed with a fixed cost. On the other hand, the continuous upgrading model represents the situation where the amount of improvement can be chosen but further improvements require larger costs.

3 Problem CELP on a Tree

For a general graph, we can modify the proof in [13] to show that approximating problems DELP and CELP is at least as hard as approximating problem SETCOVER, even on a bipartite graph. We state this result in the appendix to this paper, and restrict our attention to graphs with special structures. In this section we show that when the underlying graph is a tree, CELP under node-upgrading can be reduced to CELP under edge-upgrading (studied in [2, 12]), and hence can be solved in $O(|V| \log |V|)$ time.

Assume that a tree $T = (V(T), E(T))$, a designated node $v_0 \in V(T)$ and a bound U on the eccentricity are given. It is clear that the problem is feasible if and only if the eccentricity of v_0 is at most U when the improvements at all nodes reach their upper bounds. From now on, we assume that the problem is feasible. Also, we regard the given tree T as a directed tree with the orientation from the source node v_0 to all leaves. Let $L(T)$ denote the leaf set of the tree (excluding v_0 if v_0 itself is a leaf node). Clearly the eccentricity is given by the largest distance from v_0 to a leaf node $w \in L(T)$ in this directed tree.

First of all, we note that the upper bound $b(u)$ on the improvement value at node u can be removed by charging the delays remaining in $s(u)$ and/or $r(u)$ after the maximum improvement $x(u) = b(u)$ to the transmission time of the incident edges. Let us consider only $s(u)$ at a node u, since $r(u)$ can be similarly treated. Assume that $s(u) > \lambda_s(u)b(u)$ holds, since otherwise no delay remains in $s(u)$ after the maximum improvement. In this case, replace $s(u)$ by $\lambda_s(u)b(u)$, and $t(u,v)$ by $t(u,v) + [s(u) - \lambda_s(u)b(u)]$ for each edge $(u,v) \in E(T)$ incident from u. It is not difficult to see that the length of any path from v_0 to a leaf node does not change by this modification for any improvement value $0 \leq x(u) \leq b(u)$. Furthermore, as $\lambda_s(u)b(u) - \lambda_s(u)x(u)$ becomes 0 at $x(u) = b(u)$, we do not need the upper bound $x(u) \leq b(u)$ in the formulation.

Moreover, for any intermediate node $u \in V(T) \setminus (L(T) \cup \{v_0\})$, the set of feasible solutions does not change even if we switch the roles of the sending time and the receiving time (by interchanging $s(u)$ and $r(u)$, and interchanging $\lambda_s(u)$ and $\lambda_r(u)$). Therefore, we can assume without loss of generality that $\frac{s(u)}{\lambda_s(u)} \leq \frac{r(u)}{\lambda_r(u)}$ holds for all intermediate nodes u.

Next, we define the corresponding edge-upgrading version of CELP as follows: Given a tree $T' = (V(T'), E(T'))$, a designated node $v'_0 \in V(T')$, non-negative lengths $l'(e')$ and costs $c'(e')$ for all edges $e' \in E(T')$, and a non-negative parameter U', we want to find a vector $y = (y(e') \mid e' \in E(T'))$ such that $0 \leq y(e') \leq l'(e')$ holds for each $e' \in E(T')$ and $\sum_{e' \in E(T')} c'(e')y(e')$ is minimized under the constraint that the eccentricity of v'_0 in T' with edge lengths $l'(e') - y(e')$ is at most U'. As noted above, this problem can be solved in $O(|V'| \log |V'|)$ time.

Now we show that CELP under node-upgrading can be reduced to CELP under edge-upgrading. Given a CELP instance $I = (T, v_0, r, s, c, \lambda_r, \lambda_s, t, U)$ under node-upgrading (without upper bounds $b(v)$), we define an instance $I' = (T', v'_0, l', c', U')$ under edge-upgrading as follows.

- For the designated node $v_0 \in V(T)$, create:

  two nodes $\alpha(v_0), \alpha_s(v_0) \in V(T')$,

  edge $(\alpha(v_0), \alpha_s(v_0)) \in E(T')$ of length $l'(\alpha(v_0), \alpha_s(v_0)) = s(v_0)$ and cost $c'(\alpha(v_0), \alpha_s(v_0)) = \frac{c(v_0)}{\lambda_s(v_0)}$.

- For each leaf $w \in L(T)$, create:

  two nodes $\alpha_r(w), \alpha(w) \in V(T')$,

  edge $(\alpha_r(w), \alpha(w)) \in E(T')$ of length $l'(\alpha_r(w), \alpha(w))) = r(w)$ and cost $c'(\alpha_r(w), \alpha(w)) = \frac{c(w)}{\lambda_r(w)}$.

- For each intermediate node $u \in V(T) \setminus (L(T) \cup \{v_0\})$, create:

  three nodes $\alpha_r(u), \alpha(u), \alpha_s(u) \in V(T')$,

  edge $(\alpha_r(u), \alpha(u)) \in E(T')$ of length $l'(\alpha_r(u), \alpha(u)) = s(u) + \lambda_r(u)\frac{s(u)}{\lambda_s(u)}$ and cost $c'(\alpha_r(u), \alpha(u)) = \frac{c(u)}{\lambda_s(u)+\lambda_r(u)}$,

  edge $(\alpha(u), \alpha_s(u)) \in E(T')$ of length $l'(\alpha(u), \alpha_s(u)) = r(u) - \lambda_r(u)\frac{s(u)}{\lambda_s(u)}$ and cost $c'(\alpha(u), \alpha_s(u)) = \frac{c(u)}{\lambda_r(u)}$.

- For each edge $(u, v) \in E(T)$, create:

  edge $(\alpha_s(u), \alpha_r(v)) \in E(T')$ of length $l'(\alpha_s(u), \alpha_r(v)) = t(u, v)$ and cost $c'(\alpha_s(u), \alpha_r(v)) = \infty$.

Finally, let $v_0' = \alpha(v_0)$ and $U' = U$. This transformation is illustrated in Figure 1. One can easily verify that T' is a tree with a set of leaves $\{\alpha(w) | w \in L(T)\}$.

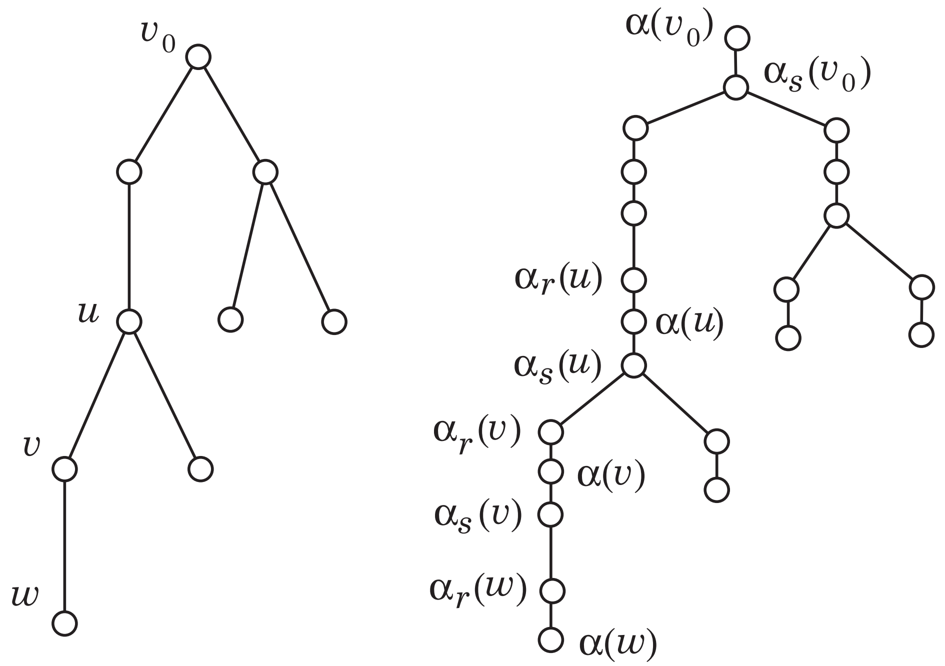

Figure 1: T on the left and T' on the right.

Let us denote the optimum values of instances I and I' by $OPT(I)$ and $OPT(I')$, respectively. The following lemma shows that a feasible solution of instance I corresponds to a feasible solution of instance I' with the same cost, yielding $OPT(I') \leq OPT(I)$.

Lemma 1 *Let instances I and I' be defined as above. If $(x(u) \mid u \in V(T))$ is a feasible solution of instance I, then the solution $(y(e) \mid e \in E(T'))$ defined by*

$$y(\alpha(v_0), \alpha_s(v_0)) = \lambda_s(v_0)x(v_0),$$

$$y(\alpha_r(w), \alpha(w)) = \lambda_r(w)x(w) \qquad \textit{for all } w \in L(T),$$

$$y(\alpha_r(u), \alpha(u)) = (\lambda_s(u) + \lambda_r(u)) \min\{x(u), \tfrac{s(u)}{\lambda_s(u)}\} \qquad \textit{for all } u \in V(T) \setminus (L(T) \cup \{v_0\}),$$

$$y(\alpha(u), \alpha_s(u)) = \lambda_r(u) \max\{0, x(u) - \tfrac{s(u)}{\lambda_s(u)}\} \qquad \textit{for all } u \in V(T) \setminus (L(T) \cup \{v_0\}),$$

$$y(\alpha_s(u), \alpha(v)) = 0 \qquad \textit{for all } (u, v) \in E(T),$$

is a feasible solution for instance I'. Moreover, the costs of these two solutions are identical.

Proof. Consider a leaf $w \in L(T)$. Let P be the unique path from v_0 to w in T, and P' be the corresponding unique path from $\alpha(v_0)$ to $\alpha(w)$ in T'. The length of P under node-upgrading may be viewed as the sum of the following three contributions:

$$D_1 = \sum_{(u,v)\in E(P)} t(u,v),$$

$$D_2 = s(v_0) - \lambda_s(v_0)x(v_0) + r(w) - \lambda_r(w)x(w),$$

$$D_3 = \sum_{u\in V(P)\setminus\{v_0,w\}} [\max\{0, s(u) - \lambda_s(u)x(u)\} + \max\{0, r(u) - \lambda_r(u)x(u)\}].$$

D_1 consists of all transmission times over the links in P. D_2 represents the sending time from v_0 and the receiving time at w. D_3 includes sending and receiving times at all intermediate nodes in P.

Analogously, the length of P' may be viewed as the sum of the following three contributions:

$$D'_1 = \sum_{(u,v)\in E(P)} [l'(\alpha_s(u), \alpha_r(v)) - y(\alpha_s(u), \alpha_r(u))],$$

$$D'_2 = l'(\alpha(v_0), \alpha_s(v_0)) - y(\alpha(v_0), \alpha_s(v_0)) + l'(\alpha(w), \alpha_r(w)) - y(\alpha(w), \alpha_r(w)),$$

$$D'_3 = \sum_{u\in V(P)\setminus\{v_0,w\}} [l'(\alpha_r(u), \alpha(u)) - y(\alpha_r(u), \alpha(u)) + l'(\alpha(u), \alpha_s(u)) - y(\alpha(u), \alpha_s(u))].$$

Directly from definitions, we get $D'_1 = D_1$ and $D'_2 = D_2$. It remains to prove that $D'_3 = D_3$, by comparing these two sums term by term for the following two cases:

- If $x(u) \le \frac{s(u)}{\lambda_s(u)}$ ($\le \frac{r(u)}{\lambda_r(u)}$ by assumption), then $y(\alpha_r(u), \alpha(u)) = (\lambda_s(u) + \lambda_r(u))x(u)$ and $y(\alpha(u), \alpha_s(u)) = 0$ hold. Therefore, we have

  $$l'(\alpha_r(u), \alpha(u)) - y(\alpha_r(u), \alpha(u)) = s(u) + \lambda_r(u)\tfrac{s(u)}{\lambda_s(u)} - (\lambda_r(u) + \lambda_s(u))x(u),$$

  $$l'(\alpha(u), \alpha_s(u)) - y(\alpha(u), \alpha_s(u)) = r(u) - \lambda_r(u)\tfrac{s(u)}{\lambda_s(u)}.$$

  The sum of these two terms (i.e., those corresponding to node u) is equal to $s(u) +$

$r(u) - (\lambda_s(u) + \lambda_r(u))x(u)$. For the case of $x(u) \le \frac{s(u)}{\lambda_s(u)}$, this is exactly the term in D_3 that corresponds to node u.

- If $\frac{s(u)}{\lambda_s(u)} < x(u) \le \frac{r(u)}{\lambda_r(u)}$, then $y(\alpha_r(u), \alpha(u)) = (\lambda_s(u) + \lambda_r(u))\frac{s(u)}{\lambda_s(u)} = s(u) + \lambda_r(u)\frac{s(u)}{\lambda_s(u)}$ and $y(\alpha(u), \alpha_s(u)) = \lambda_r(u)(x(u) - \frac{s(u)}{\lambda_s(u)})$. Therefore, we have

$$\begin{aligned} l'(\alpha_r(u), \alpha(u)) - y(\alpha_r(u), \alpha(u)) &= 0, \\ l'(\alpha(u), \alpha_s(u)) - y(\alpha(u), \alpha_s(u)) &= r(u) - \lambda_r(u)\tfrac{s(u)}{\lambda_s(u)} - \lambda_r(u)(x(u) - \tfrac{s(u)}{\lambda_s(u)}) \\ &= r(u) - \lambda_r(u)x(u). \end{aligned}$$

  This is exactly the term in D_3 corresponding to node u.

The proof that the costs of instances I and I' are identical is similar. The main point here is to note that for each node $u \in V(P) \setminus (L(T) \cup \{v_0\})$, we have the equality $c(u)x(u) = c'(\alpha_r(u), \alpha(u))y(\alpha_r(u), \alpha(u)) + c'(\alpha(u), \alpha_s(u))y(\alpha(u), \alpha_s(u))$. The proof can proceed again by considering the two cases $x(u) \le \frac{s(u)}{\lambda_s(u)}$ and $\frac{s(u)}{\lambda_s(u)} < x(u) \le \frac{r(u)}{\lambda_r(u)}$ separately. □

We then show that the converse of Lemma 1 is also true.

Lemma 2 *If $(y(e) \mid e \in E(T'))$ is a feasible solution of instance I' satisfying condition (1), then the following solution $(x(u) \mid u \in V(T))$:*

$$\begin{aligned} x(v_0) &= \tfrac{y(\alpha(v_0),\, \alpha_s(v_0))}{\lambda_s(v_0)}, & \\ x(w) &= \tfrac{y(\alpha_r(w),\, \alpha(w))}{\lambda_r(w)} & \text{for all } w \in L(T), \\ x(u) &= \tfrac{y(\alpha(u),\, \alpha_s(u))}{\lambda_r(u)} + \tfrac{y(\alpha_r(u),\, \alpha(u))}{\lambda_s(u)+\lambda_r(u)} & \text{for all } u \in V(P) \setminus (L(T) \cup \{v_0\}), \end{aligned}$$

is a feasible solution of instance I, and the costs of these two solutions are identical.

Proof. First note that there exists an optimal solution y to instance I' that satisfies

$$y(\alpha(u), \alpha_s(u)) > 0 \Rightarrow y(\alpha_r(u), \alpha(u)) = l'(\alpha_r(u), \alpha(u)), \tag{1}$$

for all $u \in V(P) \setminus (L(T) \cup \{v_0\})$, since $c'(\alpha_r(u), \alpha(u)) \le c'(\alpha(u), \alpha_s(u))$ holds and shrinking either $(\alpha_r(u), \alpha(u))$ or $(\alpha(u), \alpha_s(u))$ has the same effect on the distance in T' between v_0 and any leaf. Based on this, the rest of the proof is similar to that of Lemma 1. □

From Lemma 1 and Lemma 2, we deduce $OPT(I) = OPT(I')$. Moreover, from any optimal solution for instance I' we can derive an optimal solution for instance I. Therefore solving a CELP under node-upgrading can be reduced to solving a CELP under edge-upgrading. Since an $O(|V(T')| \log |V(T')|)$ time algorithm is known for CELP under edge-upgrading [12], and $|V(T')| \le 3|V(T)|$ is obvious, we can conclude the following.

Theorem 1 *When the underlying graph is a tree $T = (V, E)$, CELP under node-upgrading can be solved in $O(|V| \log |V|)$ time.*

4 Problem DELP on a Line

Now let us turn our attention to problem DELP. We show that DELP is NP-hard in the weak sense even if the underlying graph $G = (V, E)$ is a line, but admits a fully polynomial approximation scheme.

Theorem 2 *Problem DELP is NP-hard even on a line.*

Proof. Let $a_1, a_2, \ldots, a_n$ and b be $n+1$ positive integers such that $\sum_{i=1}^{n} a_i = 2b$ holds. Without loss of generality, we assume $b \ge a_i$ for $i = 1, 2, \ldots, n$. Problem PARTITION is to find a subset $I \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in I} a_i = b$ holds, and is known to be NP-complete [4]. We now reduce PARTITION to DELP on a line, which shows that the latter is NP-hard.

From an instance of PARTITION, let us construct an instance of DELP. First we construct the following graph $G = (V, E)$ having $2n + 1$ nodes and $2n$ edges:

$$
\begin{aligned}
V = &\quad \{u_i \mid i = 0, 1, 2, \ldots, n\} \cup \{v_i \mid i = 1, 2, \ldots, n\}, \\
E = &\quad \{(u_{i-1}, v_i), (v_i, u_i) \mid i = 1, 2, \ldots, n\}.
\end{aligned}
$$

This G is a line between u_0 and u_n. For each $e = (u, v) \in E$, we let $s(u) + t(u, v) + r(v) = 2b$. For nodes in G, let $c(u_i) = b + 1$ for $i = 0, 1, 2, \ldots, n$ and $c(v_i) = a_i$ for $i = 1, 2, \ldots, n$. Let $p_s(v_i) = p_r(v_i) = a_i$ for $i = 1, 2, \ldots, n$. Moreover without loss of generality, we assume that $s(u_i), r(u_i), s(v_i), r(v_i)$ and $p_s(u_i), p_r(u_i)$ are properly defined; i.e., $0 \le p_s(u_i) \le s(u_i)$, $0 \le p_r(u_i) \le r(u_i)$, $0 \le p_s(v_i) \le s(v_i)$, and $0 \le p_r(v_i) \le r(v_i)$. Finally, let $v_0 = u_0$ and $U = 4nb - 2b$.

We claim that there exists a subset $I \subseteq \{1, 2, \ldots, n\}$ with $\sum_{i \in I} a_i = b$ if and only if there exists a subset $S \subseteq V$ such that $\sum_{v \in S} c(v) \le b$ holds and the eccentricity of v_0 becomes at most U after upgrading the nodes in S.

If there exists a subset $I \subseteq \{1, 2, \ldots, n\}$ with $\sum_{i \in I} a_i = b$, let $S = \{v_i \mid i \in I\}$. We have

$$\sum_{v \in S} c(v) = \sum_{i \in I} c(v_i) = \sum_{i \in I} a_i = b.$$

By upgrading the nodes in S, for each $v_i \in S$, the delays of the edges incident with v_i are reduced from $2b$ to $2b - a_i$. Denote by $R(v_0)$ the eccentricity of v_0 (i.e., the path length from u_0 to u_n in G) after this upgrading. We have

$$R(v_0) = 2b|E| - 2 \sum_{i \in I} a_i = 4nb - 2b = U.$$

Conversely, suppose that there exists a subset $S \subseteq V$ such that the eccentricity of v_0 becomes at most U after upgrading the nodes in S and its cost satisfies $\sum_{v \in S} c(v) \le b$. First, we note that $\{u_i \mid i = 0, 1, 2, \ldots, n\} \cap S = \emptyset$ holds since upgrading a node in $\{u_i \mid i = 0, 1, 2, \ldots, n\}$ will make the cost more than b. Let $I = \{i \mid v_i \text{ is upgraded}\}$. We have

$$\sum_{v \in S} c(v) = \sum_{i \in I} a_i \leq b,$$

and

$$R(v_0) = 2b|E| - 2\sum_{i \in I} a_i \leq U = 4nb - 2b,$$

i.e., $\sum_{i \in I} a_i \geq b$. Hence we conclude that $\sum_{i \in I} a_i = b$ holds. □

The above theorem, however, says that DELP on a line is only NP-hard in the weak sense, since so is PARTITION [4]. We now show that there is a pseudo-polynomial time algorithm for solving DELP on a line.

Consider an instance of DELP on a line $G = (V, E)$ with $V = \{u_i \mid i = 0, 1, 2, \ldots, n\}$ and $E = \{(u_{i-1}, u_i) \mid i = 1, 2, \ldots, n\}$. We treat the following two cases separately.

The first case is that the source node v_0 is located at one end of the line, say $v_0 = u_0$. In this case, let $c_i = c(u_i)$ for $i = 0, 1, 2, \ldots, n$, and let $d_0 = p_s(u_0)$, $d_n = p_r(u_n)$, $d_i = p_s(u_i) + p_r(u_i)$ for $i = 1, 2, \ldots, n-1$, and $d(u_{i-1}, u_i) = s(u_{i-1}) + t(u_{i-1}, u_i) + r(u_i)$ for $i = 1, 2, \ldots, n$. Then DELP is formulated as the following knapsack problem:

$$\begin{array}{ll} \min & \sum_{i=0}^{n} c_i x_i \\ \text{s.t.} & \sum_{i=0}^{n} d_i x_i \geq \sum_{i=1}^{n} d(u_{i-1}, u_i) - U \\ & x_i \in \{0, 1\}, \quad i = 0, 1, \ldots, n \end{array} \tag{2}$$

The second case is that the source node v_0 is located at one of the intermediate nodes on the line, say $v_0 = u_k$ for some $0 < k < n$. Then we divide the line into two segments, $L_1 = \{(u_0, u_1), (u_1, u_2), \ldots, (u_{k-1}, u_k)\}$ and $L_2 = \{(u_k, u_{k+1}), (u_{k+1}, u_{k+2}), \ldots, (u_{n-1}, u_n)\}$, and define the following four knapsack problems:

$$KP_1: \quad \min \sum_{i=0}^{k-1} c_i x_i$$
$$\text{s.t.} \quad \sum_{i=0}^{k-1} d_i x_i \geq \sum_{i=1}^{k} d(u_{i-1}, u_i) - U \tag{3}$$
$$x_i \in \{0,1\}, \quad i = 0, 1, \ldots, k-1,$$

$$KP_2: \quad \min \sum_{i=k+1}^{n} c_i x_i$$
$$\text{s.t.} \quad \sum_{i=k+1}^{n} d_i x_i \geq \sum_{i=k+1}^{n} d(u_{i-1}, u_i) - U \tag{4}$$
$$x_i \in \{0,1\}, \quad i = k+1, k+2, \ldots, n,$$

$$KP_3: \quad \min \sum_{i=0}^{k-1} c_i x_i$$
$$\text{s.t.} \quad \sum_{i=0}^{k-1} d_i x_i \geq \sum_{i=1}^{k} d(u_{i-1}, u_i) - U - p_r(u_k) \tag{5}$$
$$x_i \in \{0,1\}, \quad i = 0, 1, \ldots, k-1,$$

$$KP_4: \quad \min \sum_{i=k+1}^{n} c_i x_i$$
$$\text{s.t.} \quad \sum_{i=k+1}^{n} d_i x_i \geq \sum_{i=k+1}^{n} d(u_{i-1}, u_i) - U - p_s(u_k) \tag{6}$$
$$x_i \in \{0,1\}, \quad i = k+1, k+2, \ldots, n.$$

Denote the optimal values of these four knapsack problems by C_1, C_2, C_3 and C_4 respectively. In the optimal solution of DELP, either the source node u_k is upgraded or not. It is straightforward to see that C_1 and C_2 correspond to the case without upgrading u_k, while C_3 and C_4 correspond to the case when u_k is upgraded. The optimal cost C of DELP is then given by

$$C = \min\{C_1 + C_2, c(u_k) + C_3 + C_4\}.$$

As the knapsack problem has a pseudo-polynomial time algorithm [4], DELP on a line can also be solved in pseudo-polynomial time.

Moreover, as the knapsack problem admits a fully polynomial approximation scheme, we can easily show that DELP on a line can also admits a fully polynomial approximation scheme. This is clearly true for the first case (i.e., source v_0 is one of u_0 and u_n), and we need only consider the second case.

Let $\{A_\rho \mid \rho > 0\}$ be a set of algorithms for the knapsack problem such that each A_ρ is a $(1+\rho)$-approximation algorithm and the running time is polynomially bounded in the input length and $1/\rho$. For any instance KP of the knapsack problem, write by $A_\rho(KP)$ the objective value obtained by A_ρ applied to KP. We construct an algorithm B_ρ for DELP as follows:

$$B_\rho(DELP) = \min\{A_\rho(KP_1) + A_\rho(KP_2),\ c(u_k) + A_\rho(KP_3) + A_\rho(KP_4)\},$$

where we slightly abuse the notation since $DELP$ stands for an instance of DELP, and KP_1, KP_2, KP_3 and KP_4 stand for instances of the associated knapsack problems. Clearly, the running time of B_ρ is bounded by a polynomial in the input length and $1/\rho$.

By the definition, we have

$$A_\rho(KP_i) \leq (1+\rho)C_i \quad \text{for } i = 1, 2, 3, 4.$$

Hence

$$B_\rho(DELP) \leq \min\{(1+\rho)C_1 + (1+\rho)C_2,\ c(u_k) + (1+\rho)C_3 + (1+\rho)C_4\} \leq (1+\rho)C,$$

since $\rho > 0$. Therefore $\{B_\rho \mid \rho > 0\}$ is a fully polynomial approximation scheme for DELP in the second case.

Theorem 3 *DELP on a line can be solved by a pseudo-polynomial time algorithm and admits a fully polynomial approximation scheme.*

Remark 1: We call a graph $G = (V, E)$ a star if G is a tree and there is at most one node $u \in V$ whose degree is more than one. Using similar arguments as in this section, it is not difficult to deduce that DELP on a star can also be solved in pseudo-polynomial time and admits a fully polynomial approximation scheme.

Remark 2: If $c_i = 1$ holds for all i, the knapsack problems (2) to (6) are trivially solved. In fact, we select the items in non-increasing order of d_i until the requirement is satisfied. As $\{d_i\}$ can be sorted in $O(|V| \log |V|)$ time, DELP on a line with unit costs can be solved in $O(|V| \log |V|)$ time. This result can also be extended to DELP on a star.

5 Concluding Remarks

In this paper, we considered a new type of network upgrading model. The model divides the communication delay over an edge into three parts: sending time, receiving time and transmission time. Upgrading a node shortens the sending and receiving times of this node, with possibly different effects on them. We considered two upgrading strategies; one assumes that the improvement at a node is variable, while the other assumes that the improvement at a node is fixed. These two problems are denoted by CELP and DELP respectively. We showed that CELP can be solved in polynomial time if the underlying graph is a tree. However DELP is NP-hard in the weak sense even if the underlying graph is a line, but admits a fully polynomial approximation scheme. Note that the eccentricity lowering problems considered in this paper are to minimize the upgrading cost in order to reduce the eccentricity by a given amount. There is a related version of the eccentricity lowering problem, i.e., to minimize the eccentricity within a given budget. It is straightforward to see that the results in this paper can be extended to such a version.

References

[1] S. Arora and C. Lund, "Hardness of approximations", Approximation algorithms for NP-hard problems, D. S. Hochbaum (Editor), Boston, PWS Publishing, 1997.

[2] V. Chepoi, H. Noltemeier, and Y. Vaxès, Upgrading trees under diameter and budget constraints, Networks 41 (2003), 24-35.

[3] K.U. Drangmeister, S.O. Krumke, M.V. Marathe, N. Noltemeier, and S.S. Ravi, Modifying edges of a network to obtain short subgraphs, Theoretical Computer Science 203 (1999), 91-121.

[4] M.R. Garey and D.S. Johnson, Computers and intractability: a guide to the theory of NP-completeness, W.H. Freeman and Company, New York, 1979.

[5] T. Ibaraki and N. Katoh, Resource allocation problems: algorithmic approaches, The MIT Press, Cambridge, Massachusetts, 1988.

[6] S.O. Krumke et al., Approximation algorithms for certain network improvement problems, Journal of Combinatorial Optimization 2 (1998), 257-288.

[7] S.O. Krumke et al., Improving spanning trees by upgrading nodes, Theoretical Computer Science 221 (1999), 139-155.

[8] S.O. Krumke et al., Improving minimum cost spanning trees by upgrading nodes, Journal of Algorithms 33 (1999), 92-111.

[9] S.O. Krumke et al., Upgrading bottleneck constrained forests, Discrete Applied Mathematics 108 (2001), 129-142.

[10] D. Paik and S. Sahni, Network upgrading problems, Networks 26 (1995), 45-58.

[11] R. Raz and S. Safra, A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP, Proc. 29th Ann. ACM Symp. on Theory of Comp., ACM, 1997, pp.475-484.

[12] J.Z. Zhang, X.G. Yang, and M.C. Cai, Inapproximability and a polynomially solvable special case of a network improvement problem, European Journal of Operational Research 155 (2004), 251-257.

[13] J.Z. Zhang, X.G. Yang, and M.C. Cai, A network improvement problem under different norms, Computational Optimization and Applications 27 (2004), 305-319.

Appendix: An outline of the proof for inapproximability of DELP and CELP

First consider problem DELP. We show that problem SETCOVER is L-reducible [1] to DELP.

Problem SETCOVER can be stated as follows: Given a finite ground set $X = \{x_1, x_2, \ldots, x_n\}$ and a collection of its subsets $S_1, S_2, \ldots, S_m$ satisfying $\cup_{i=1}^m S_i = X$, find the size of the minimum subcollection that covers X, i.e., a subset $I \subseteq \{1, 2, \ldots, m\}$ with the minimum size such that $\cup_{i \in I} S_i = X$. It is well known that approximating SETCOVER within a ratio of $c \ln(n)$ for some constant $c > 0$ is NP-hard [11].

Given an instance of SETCOVER, we construct the following graph $G = (V, E)$ (shown in Figure 2).

$$V = \{u_0, u_1, u_2, \ldots, u_m\} \cup \{v_0, v_1, v_2, \ldots, v_n\},$$

$$E = \{(v_0, u_0)\} \cup \{(u_0, u_i) \mid i = 1, 2, \ldots, m\} \cup \{(u_i, v_j) \mid x_j \in S_i\}.$$

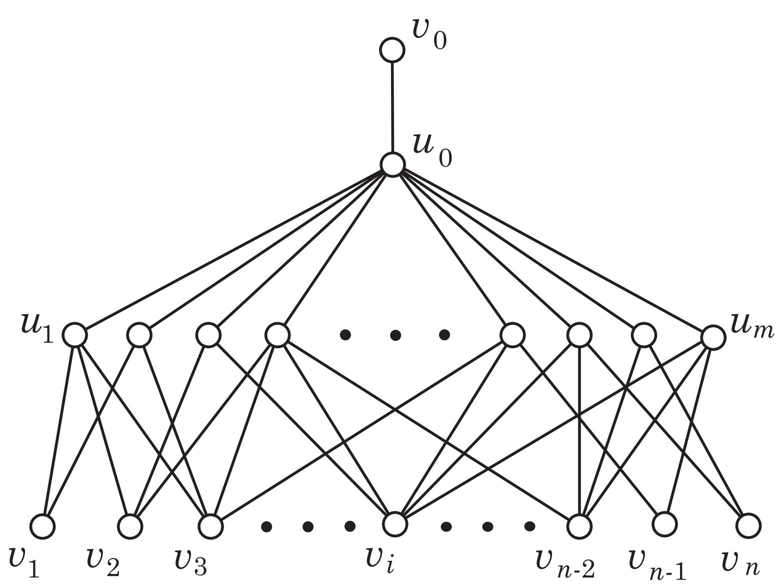

Figure 2: Graph G.

Let $U = 1$, $c(v) = 1$ for all $v \in V$, $t(e) = 0$ for all $e \in E$, $s(v_0) = r(v_0) = p_s(v_0) = p_r(v_0) = 0$ and $s(v) = r(v) = p_s(v) = p_r(v) = 1$ for $v \in V \setminus \{v_0\}$. This completes the construction of the instance of DELP. In this case, we have $d(v_0, u_0) = 1$, and $d(e) = 2$ for all $e \in E \setminus \{(v_0, u_0)\}$, so the eccentricity from v_0 is 5. Moreover, if all nodes in a subset $V' \subseteq V$ are upgraded, the cost incurred is $|V'|$.

If $\mathcal{C}^* = \{S_{i_1}, S_{i_2}, \ldots, S_{i_k}\}$ is a minimum set cover of X, we upgrade u_0 and nodes u_{i_h} for $S_{i_h} \in \mathcal{C}^*$, where we denote the set of upgraded nodes by V'. It is not difficult to see that the upgraded distance d' satisfies $d'(v_0, u_0) = 0$, $d'(u_0, u_{i_h}) = 0$ for $S_{i_h} \in \mathcal{C}^*$, $d'(u_0, u_i) = 1$ for

$S_i \notin \mathcal{C}^*$, $d'(u_{i_h}, v_j) = 1$ for $S_{i_h} \in \mathcal{C}^*$, and $d'(u_i, v_j) = 2$ for $S_i \notin \mathcal{C}^*$. As $\mathcal{C}^*$ is a set cover, we have $d'(v_0, v_j) = 1$ for all j, $d'(v_0, u_i) = 1$ for $S_i \notin \mathcal{C}^*$, and $d'(v_0, v) = 0$ for all other nodes v. Hence the eccentricity from v_0 is 1, implying that V' is feasible. The cost of this solution is $|V'| = 1 + k = 1 + |\mathcal{C}^*|$. Therefore, any optimal solution V^* of DELP satisfies

$$|V^*| \leq |V'| = 1 + |\mathcal{C}^*|. \tag{7}$$

Conversely, assume that upgrading $V' \subseteq V$ is feasible. If $u_0 \notin V'$ holds, then the length from v_0 to any node u_i or v_j is not less than 2. Thus u_0 must be upgraded.

Next if, for a v_j, $u_i \notin V'$ holds for all i with $(u_i, v_j) \in E$, then

$$d'(v_0, v_j) = \min_{(u_i,v_j)\in E} [d'(u_0, u_i) + d'(u_i, v_j)] \geq 1 + 1 = 2,$$

showing that this is not feasible. Therefore for each v_j, one of the u_i adjacent to v_j must be upgraded to make $d'(v_0, v_j) \leq 1$. In this case, as u_0 and one of the u_i adjacent to v_j are upgraded, we have $d'(v_0, v_j) = 1$, and upgrading v_j is unnecessary. Also the set of upgraded u_i gives a set cover $\mathcal{C}$ as each v_j must be incident to one of the upgraded u_i. Hence we obtain a set cover $\mathcal{C}$ from a feasible solution of DELP, such that

$$1 + |\mathcal{C}| \leq |V'|. \tag{8}$$

Combining (7) and (8), we obtain $|\mathcal{C}| - |\mathcal{C}^*| \leq |V'| - |V^*|$. Moreover, since $|\mathcal{C}^*| \geq 1$ holds, we have $|V^*| \leq 2|\mathcal{C}^*|$ by (7). Hence we obtain

$$\frac{|\mathcal{C}| - |\mathcal{C}^*|}{|\mathcal{C}^*|} \leq 2\frac{|V'| - |V^*|}{|V^*|}. \tag{9}$$

This shows that the reduction from an instance of problem SETCOVER to the instance of DELP is an L-reduction. It is not difficult to see that (9) implies that approximating DELP is as hard as approximating problem SETCOVER.

For problem CELP, we can also use a similar argument to show its inapproximability.