Proceedings of the 17th IASTED International Conference
Parallel and Distributed Computing and Systems
November 14-16, 2005, Phoenix, AZ, USA

FINDING TWO DISJOINT PATHS IN A NETWORK WITH NORMALIZED α^--MIN-SUM OBJECTIVE FUNCTION

Bing Yang†, S. Q. Zheng† and Enyue Lu‡
† Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083-0688, USA
‡ Mathematics and Computer Science Department, Salisbury University, Salisbury, MD 21801, USA

ABSTRACT
Given a number α with $0 < \alpha < 1$, a graph $G = (V, E)$ and two nodes s and t in G, we consider the problem of finding two disjoint paths P_1 and P_2 from s to t such that $length(P_1) + \alpha \cdot length(P_2)$ with $length(P_1) \geq length(P2)$ is minimized. The paths may be node-disjoint or edge-disjoint, and the network may be directed or undirected. This problem has applications in reliable communication. We show that all four versions of this problem are NP-complete. Then we give an approximation of $\frac{2}{1+\alpha}$, and show that this bound is best possible for directed graphs. For acyclic directed graphs, we also give a pseudo-polynomial-time algorithm for finding optimal solutions.

KEY WORDS
network, graph, reliability, shortest path, disjoint paths, NP-complete, approximation

1 Introduction

A reliable telecommunication network, which is modeled by a graph $G = (V, E)$, is designed in such a way that multiple connections exist between every pair of communicating nodes. Usually, paths are selected according to an objective function. Each edge e in G is assigned a non-negative length $l(e)$, which reflects the resource and/or performance associated with the edge, such as cost, distance, latency, etc. The length $l(P)$ of a path P is defined as the sum of the lengths of its edges. To avoid single point of failure, the paths may be node-disjoint or edge disjoint, and the network may be directed or undirected. Thus, a problem of finding disjoint paths have four versions.

Various problems of finding optimized disjoint paths between two nodes s and t in G have been investigated. Ford and Fulkson gave gave a polynomial-time algorithm for finding two paths with minimum total length (called *MIN-MAX 2-Path Problem*), using the algorithm of finding minimum weighted network flows [7]. Suurballe and Tarjan provided different treatment, and presented algorithms that are more efficient [18] [19]. Li *et al.* proved that all four versions of the problem of finding two disjoint paths such that the length of the longer path is minimized (called the *MIN-MAX 2-Path Problem*) are strongly NP-complete [9]. They also considered a generalized MIN-SUM problem (which we call the *G-MIN-SUM k-Path Problem*) assuming that each edge is associated with k different lengths. The objective of this problem is to find k disjoint paths such that the total length of the paths is minimized, where the jth edge-length is associated with the jth path. They showed that all four versions of the G-MIN-SUM k-path problem are strongly NP-complete even for $k = 2$ [10].

In [12], we considered the problem of finding two disjoint paths such that the length of the shorter path is minimized (named the MIN-MIN 2-path problem). We showed that all four versions of the MIN-MIN 2-path problem are strongly NP-complete[12]. In the same paper, we also showed there does not exist any polynomial-time approximation algorithm with a constant approximation bound for any of these four versions of the MIN-MIN 2-path problem unless $P = NP$. In this paper we consider a generalized weighted 2-path problem. Let P_1 and P_2 be two disjoint paths from s to t in a given graph G, and α a non-negative value. Define

$$L_\alpha(P_1, P_2) = l(P_1) + \alpha \cdot l(P_2)$$

.
Our objective is to find two disjoint paths P_1 and P_2 such that $L_\alpha(P_1, P_2)$ is minimized. Graph G may be directed or undirected, and the paths may be node-disjoint or edge-disjoint. We call this problem the *α-MIN-SUM 2-path problem*. According to the relative values of P_1 and P_2, this problem can be treated as having two versions. One is to minimize

$$L_{\alpha^-}(P_1, P_2) = \max\{l(P_1), l(P_2)\} + \alpha \cdot \min\{l(P_1), l(P_2)\},$$

and the other is to minimize

$$L_{\alpha^+}(P_1, P_2) = \min\{l(P_1), l(P_2)\} + \alpha \cdot \max\{l(P_1), l(P_2)\}.$$

We name the former as the *α^--MIN-SUM 2-path problem*, and the latter as the *α^+-MIN-SUM 2-path problem*. It is clear that if $\alpha = 0$, the α^--MIN-SUM 2-path problem, and the α^+-MIN-SUM 2-path problem degenerates to the MIN-MAX 2-path problem and MIN-MIN 2-path problem, respectively, which are NP-complete for all four versions. But if $\alpha = 1$, both degenerate to the MIN-SUM 2-path problem, which is polynomial-time solvable. Investigation on the α-MIN-SUM 2-path problem is of theoretical interest: what are the computational complexities of the α-MIN-SUM 2-path problem with $0 < \alpha < 1$?

Since

$$l(P_1) + \alpha \cdot l(P_2) = \alpha \cdot \left(\frac{1}{\alpha} \cdot l(P_1) + l(P_2) \right)$$

an α-MIN-SUM problem with $\alpha \geq 1$ can be transformed into a problem with $\alpha \leq 1$. Hence, it is sufficient to only consider the problem with $0 < \alpha \leq 1$. We call the α^--MIN-SUM 2-path problem (resp. α^+-MIN-SUM 2-path problem) with $0 \leq \alpha \leq 1$ the *normalized α^--MIN-SUM 2-path problem* (resp. *normalized α^+-MIN-SUM 2-path problem*). The relationship among these 2-path problems is shown in Figure 1.

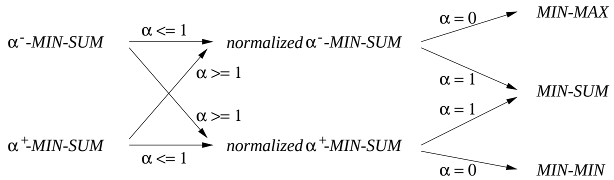

Figure 1. Relations among various 2-path problems.

In this paper, we consider normalized α^--MIN-SUM 2-path problem with $0 < \alpha < 1$. As indicated in Figure 1, the algorithmic issues related to the α^--MIN-SUM 2-path problem with $\alpha > 1$ have to be addressed by considering the normalized α^+-MIN-SUM 2-path problem. Readers should bear in mind that the conclusions on the general α^--MIN-SUM 2-path problem and α^--MIN-SUM 2-path problem can be drawn from the combination of the results on normalized α^--MIN-SUM 2-path problem and normalized α^+-MIN-SUM 2-path problem. We would like to mention that our results on the normalized α^+-MIN-SUM 2-path problem are reported in [13]. It turns out that the properties of these two normalized 2-path problems are quite different.

Apart from its theoretical interest, this α^--MIN-SUM 2-path problem has important applications in survival network design. In many real-world network applications, there requires to provide two disjoint (node-disjoint or edge-disjoint) paths between two nodes to guarantee reliability. One is called *primary route* and the other is called a *secondary route*. In case there is a failure on the primary route, the traffic can be quickly switched to the secondary route. To build a path, the service provider normally charges a fee proportional to the length of the path. The service provider sometimes will give price discount to the shorter path, as α times the normal price. There are similar cases in other areas, like retail business. For example, a shoe store will have "buy one and get 2nd one with 50% off" policy for business promotion. The one with 50% off has to have smaller or equal value as the other one.

In this paper, we first show that all four versions of the normalized α^--MIN-SUM 2-path problem are NP-complete. We show that using MIN-SUM 2-path solutions to approximate normalized α^--MIN-SUM solutions achieves an approximation bound $\frac{2}{1+\alpha}$. Then, we prove that for directed graphs there does not exist any polynomial-time approximation algorithm guaranteeing an approximation bound smaller than $\frac{2}{1+\alpha}$ unless $P = NP$. Finally, give pseudo-polynomial-time algorithms for finding optimal disjoint paths in acyclic directed graphs.

2 NP-Completeness

Two paths are said to be edge-disjoint in $G = (V, E)$ if they have no edge in common, while they are said to be node-disjoint if they have no intermediate node in common. Clearly, node-disjoint paths are also edge-disjoint, but the converse is not true. There are four versions of the normalized α^--MIN-SUM 2-path problem:

- node-disjoint paths in directed graphs (ND-D 2-path problem);
- edge-disjoint paths in directed graphs (ED-D 2-path problem);
- node-disjoint paths in undirected graphs (ND-UD 2-path problem);
- edge-disjoint paths in undirected graphs (ED-UD 2-path problem).

All graphs considered in this paper are simple graphs, i.e. graphs without self-loops and parallel edges between any pair of nodes. We show that all these versions are NP-complete by reducing the partition problem ([1]) to the decision problem corresponding to the normalized α^--MIN-SUM 2-path problem.

2.1 Edge-Disjoint Paths in Directed Graphs

The *Partition Problem* is defined as following ([8]):

INSTANCE: A set of integers: $C = \{c_1, c_2, \cdots, c_n\}$.

QUESTION: Is there a subset $I \subseteq N = \{1, \cdots, n\}$ such that

$$\sum_{i \in I} c_i = \sum_{i \in N-I} c_i \ ?$$

The partition problem is a well-known NP-complete problem[1]. We prove that the ED-D version of the normalized α^--MIN-SUM 2-path problem is NP-complete by reducing the partition problem to it. Let $J = N - I$. We use $(C(I), C(J))$ to denote a partition of C, where $C(I) = \{c_i | i \in I\}$ and $C(J) = \{c_j | j \in J\}$.

For a partion problem instance $\{c_1, c_2, \cdots, c_n\}$, we construct a direct graph $G(C)$ in a way shown in Figure 2.

We call a section corresponding to c_i, shown in Figure 3, a *block* and denote it by B_i.

Lemma 1 *Any two edge-disjoint paths from s to t in $G(C)$ define a unique partition of C.*

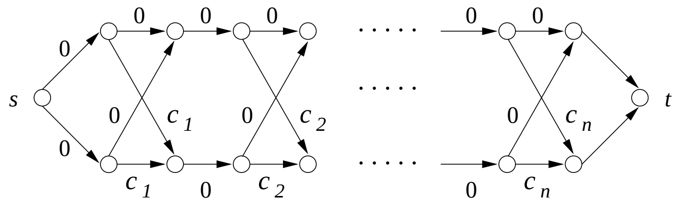

Figure 2. Graph $G(C)$ constructed from C.

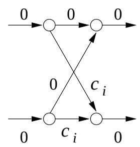

Figure 3. Block B_i

Proof: In $G(C)$, any two edge-disjoint paths going through a block B_i must be in only one of the two cases shown in Figure 4. Thus, one and only one edge with cost c_i is used in the two paths. Count the non-zero edges used in each path, we obtain a partition of $\{c_1, c_2, \cdots, c_n\}$. ■

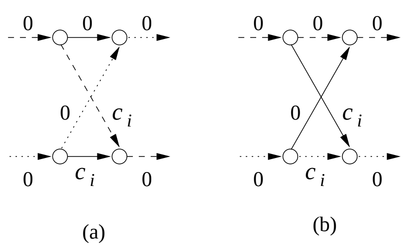

Figure 4. Edge-disjoint paths to go through block B_i.

Lemma 2 *Any partition $(C(I), C(J))$ of $\{c_1, c_2, \cdots, c_n\}$ defines two unique edge-disjoint paths P_1 and P_2 from s to t in $G(C)$ such that $l(P_1) = \sum_{c_i \in C(I)} c_i$ and $l(P_2) = \sum_{c_j \in C(J)} c_j$.*

Proof: Given $(C(I), C(J))$ of C, we construct P_1 and P_2 corresponding to $C(I)$ and $C(J)$, respectively, as follows. In Figures, P_1 is represented by dashed edges, and P_2 by dotted edges.

The edges from s to B_1 are shown in Figure 5, with Figure 5 (a) and (b) corresponding to $c_1 \in C(I)$ and $c_1 \in C(J)$, respectively.

If $c_i \in C(I)$, $i > 1$, we have four cases, as shown in Figure 6 (a), (b), (g) and (h), depending on if c_{i+1} is in $C(I)$ or not. If $c_i \in C(J)$, $1 \geq i$, we have four cases, as shown in Figure 6 (c), (d), (e) and (f), depending on if c_{i+1} is in $C(I)$ or not. Clearly, these include all possible cases.

As an example, Figure 7 shows two edge-disjoint paths constructed from $G(C)$ of $C = \{c_1, c_2, c_3, c_4, c_5\}$, $C(I) = \{c_1, c_3, c_4\}$ and $C(J) = \{c_2, c_5\}$.

This way, we get two unique edge-disjoint paths P_1 and P_2 such that $l(P_1) = \sum_{c_i \in C(I)} c_i$ and $l(P_2) = \sum_{c_j \in C(J)} c_j$. ■

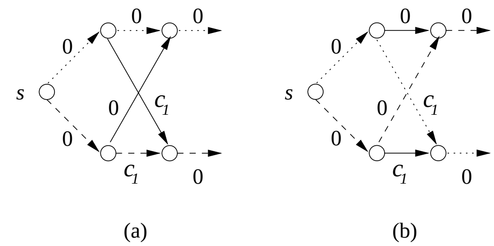

Figure 5. Two paths: (a) $c_1 \in C(I)$, and (b) $c_1 \in C(J)$.

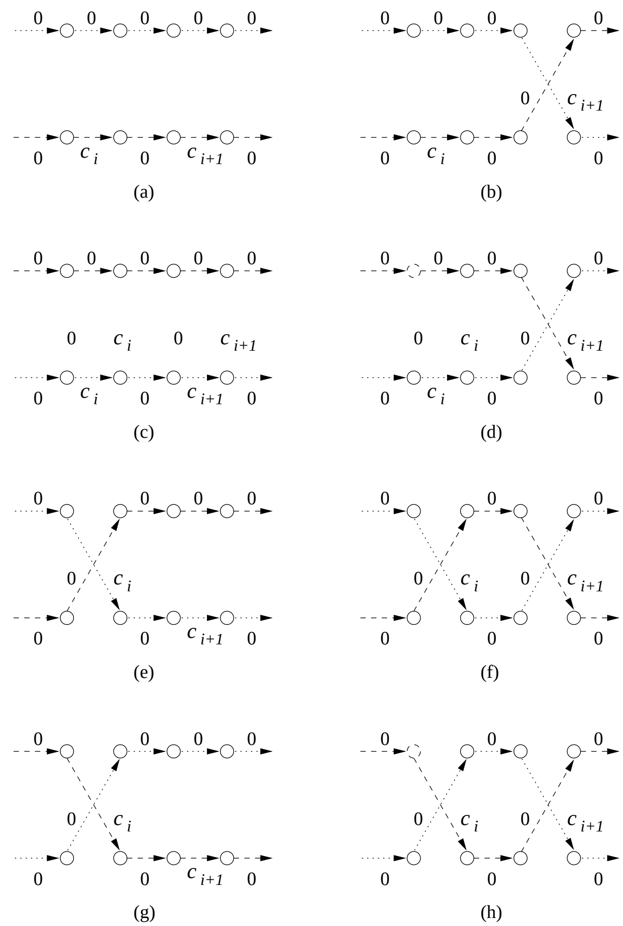

Figure 6. Cases for the proof of Lemma 2.

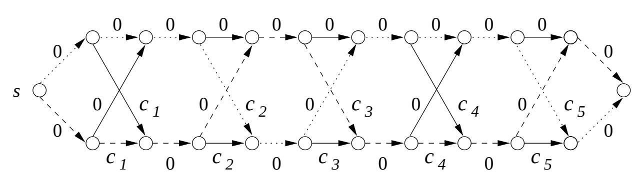

Figure 7. An example.

We define the two paths constructed by the method in the proof of Lemma 2 as *partition defined* 2 *paths*.

Lemma 3 *For the graph $G(C)$ constructed from any partition instance $C = \{c_1, c_2, \cdots, c_n\}$ with $D = \sum_{i \in N} c_i$, the normalized α^--MIN-SUM 2-path problem has the optimal value $\frac{1+\alpha}{2} \cdot D$ if and only if the partition problem has a feasible solution.*

Proof: The proof consists of two parts.

Part I: We show that if the partition problem for C has a feasible solution, then the corresponding graph $G(C)$ has two edge-disjoint paths with normalized α^--MIN-SUM cost $\frac{1+\alpha}{2} \cdot D$. Let the feasible partition be $C(I)$ and $C(J)$. Then $\sum_{c_i \in C(I)} c_i = \sum_{c_j \in C(J)} c_j$. From Lemma 2, we can find 2 edge-disjoint paths P_1 and P_2 corresponds to I and $N - I$ with $l(P_1) = l(P_2) = \frac{D}{2}$. Then $l(P_1) + \alpha \cdot l(P_2) = \frac{1+\alpha}{2} \cdot D$.

We show that $\frac{1+\alpha}{2} \cdot D$ is the normalized α^--MIN-SUM cost. Let P_1' and P_2' be any two edge-disjoint paths in $G(C)$. From the proof in Lemma 1 we have, $l(P_1') + l(P_2') = D$. Assume $l(P_1') \geq l(P_2')$, then $\delta = l(P_1') - l(P_2') \geq 0$. So $P_1' = \frac{D+\delta}{2}$, $P_2' = \frac{D-\delta}{2}$. Hence $l(P_1') + \alpha \cdot l(P_2') = \frac{1+\alpha}{2} \cdot D + \frac{1-\alpha}{2} \cdot \delta$, which achieves its minimum value $\frac{1+\alpha}{2} \cdot D$ at $\delta = 0$.

Part II: We show that if $G(C)$ has two edge-disjoint paths with normalized α^--MIN-SUM cost $\frac{1+\alpha}{2} \cdot D$ then there is a feasible solution for C. Let P_1 and P_2 be two disjoint paths in $G(C)$ such that $l(P_1) \geq l(P_2)$. Then, it requires $l(P_1) = l(P_2) = \frac{D}{2}$ to make $l(P_1) + \alpha \cdot l(P_2)$ to achieve its minimum value $\frac{1+\alpha}{2} \cdot D$. If such two paths exist in $G(C)$, we can use the method of Lemma 1 to map P_1 and P_2 to sets $C(I)$ and $C(J)$ to obtain a a partition with $\sum_{i \in C(I)} c_i = \sum_{j \in C(J)} c_j = \frac{D}{2}$. ■

Theorem 1 *The ED-D version of the normalized α^--MIN-SUM 2-path problem is NP-complete.*

Proof: Obviously, this problem belong to the NP class. Lemmas 1-3 show that the partition problem is polynomial-time reducible to edge-disjoint, directed α^--MIN-SUM 2-path problem.
■

Corollary 1 *The ND-D, ED-UD and ND-UD versions of the normalized α^--MIN-SUM 2-path problem are NP-complete.*

Proof: Since paths used in the proof of previous lemmas are also node-disjoint, the node-disjoint normalized α^--MIN-SUM 2-path problem for directed graphs is also NP-complete. Since the proofs of the previous lemmas also apply undirected graph $G(C)$, edge-disjoint and node-disjoint normalized α^--MIN-SUM 2-path problem for undirected graphs are also NP-complete. ■

How about special graphs? The following result indicates that the normalized α^--MIN-SUM 2-path problem is not easier if graphs are restricted to planar graphs.

Corollary 2 *All four versions of the normalized α^--MIN-SUM 2-path problem on planar graphs are NP-complete.*

Proof: The theorem follows from the fact that graph $G(C)$ is planar. ■

3 Approximation Analysis

We say that an instance of the normalized α^--MIN-SUM 2-path problem is feasible if for which a feasible solution exists. We say that there exists an approximation algorithm with bounded (worst-case) error for the normalized α^--MIN-SUM 2-path problem $\mathbf{\Pi}$ if there exists a polynomial-time algorithm $\mathcal{A}$ such that for the feasible instance space I of $\mathbf{\Pi}$

$$\frac{l(P_1) + \alpha \cdot l(P_2)}{l(P_1^*) + \alpha \cdot l(P_2^*)} \leq E, \tag{1}$$

where (P_1, P_2) is the solution produced by algorithm $\mathcal{A}$, (P_1^*, P_2^*) is an optimal solution, and E is a constant. The next theorem shows that all four versions of the normalized α^--MIN-SUM 2-path problem are approximatable.

Theorem 2 *For all four versions of the normalized α^--MIN-SUM 2-path problem, there exists a polynomial-time approximation algorithm with error bound $\frac{2}{1+\alpha}$.*

Proof: Let P_1 and P_2, $l(P_1) \geq l(P_2)$, be the optimal solution of the MIN-SUM 2-path problem, which is polynomial-time solvable [7] [18]). We show that $\frac{l(P_1)+\alpha \cdot l(P_2)}{l(P_1^*)+\alpha \cdot l(P_2^*)} \leq \frac{2}{1+\alpha}$.

As (P_1, P_2) is an optimal solution for MIN-SUM 2-path problem,

$$l(P_1) + l(P_2) \leq l(P_1^*) + l(P_2^*).$$

And as $l(P_1^*) \geq l(P_2^*)$,

$$l(P_1) + l(P_2) \leq l(P_1^*) + l(P_2^*) \leq 2 \cdot l(P_1^*).$$

Then,

$$\begin{array}{ll} & l(P_1^*) + \alpha \cdot l(P_2^*) \\ = & \alpha \cdot (l(P_1^*) + l(P_2^*)) + (1 - \alpha) \cdot l(P_1^*) \\ \geq & \alpha \cdot (l(P_1) + l(P_2)) + (1 - \alpha) \cdot \frac{l(P_1)+l(P_2)}{2} \\ = & \frac{1+\alpha}{2} \cdot (l(P_1) + l(P_2)) \end{array}.$$

Hence,

$$\begin{array}{ll} & \frac{l(P_1)+\alpha \cdot l(P_2)}{l(P_1^*)+\alpha \cdot l(P_2^*)} \\ \leq & \frac{l(P_1)+l(P_2)}{l(P_1^*)+\alpha \cdot l(P_2^*)} \\ \leq & \frac{l(P_1)+l(P_2)}{\frac{1+\alpha}{2} \cdot (l(P_1)+l(P_2))} = \frac{1}{\frac{1+\alpha}{2}} = \frac{2}{1+\alpha} \end{array}$$

■

We now establish the tightness of our approximation bound.

Theorem 3 *With respect to using a MIN-SUM 2-path solution to approximate a normalized α^--MIN-SUM solution, the error bound $\frac{2}{1+\alpha}$ for all four versions of the normalized α^--MIN-SUM 2-path problem is tight.*

Proof: Let P_1, P_2, P_1^* and P_2^* be defined the same way as in the proof of Theorem 2. Consider the graph shown in Figure 8(a). For this graph, the optimal MIN-SUM solution and the optimal α^--MIN-SUM solution are shown in Figure 8(b) and (c), respectively.

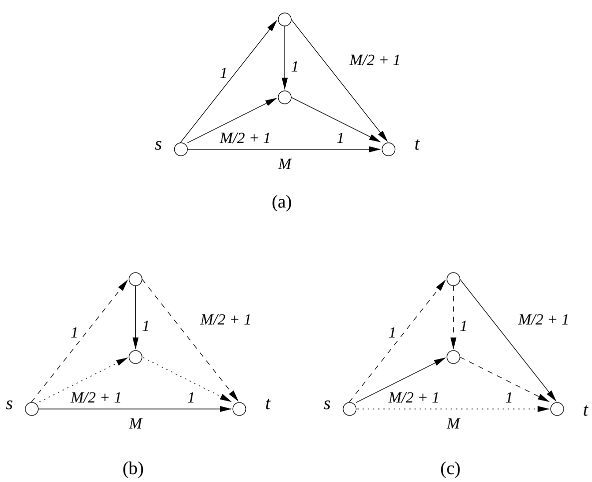

Figure 8. An example.

Clearly,

$$\begin{aligned} & \frac{l(P_1)+\alpha\cdot l(P_2)}{l(P_1^*)+\alpha\cdot l(P_2^*)} \\ = \; & \frac{M+3\cdot\alpha}{(1+\alpha)\cdot(\frac{M}{2}+2)} \\ \to \; & \frac{2}{1+\alpha} \qquad \text{(as } M \text{ approaching } \infty). \end{aligned}$$

Hence, the error bound $\frac{2}{1+\alpha}$ is tight for the two (edge-disjoint and node-disjoint) directed versions. Obviously, this example can be used the two undirected versions. ∎

Theorem 3 states that using optimal MIN-SUM solutions to approximate optimal α^--MIN-SUM solutions, $\frac{2}{1+\alpha}$ is the best possible error bound. A question arises: Is there *any* polynomial-time approximation algorithm for the normalized α^--MIN-SUM with a smaller error bound? The following theorem partially answers this question.

Theorem 4 *For the two directed versions of the normalized α^--MIN-SUM 2-path problem, if there exists a polynomial-time approximation algorithm with an error bound smaller than $\frac{2}{1+\alpha}$, then P = NP.*

Proof: We use the same technique Li used in [9] - the polynomial-time reduction from the 2DP problem. The 2DP problem asks if there exists two paths from nodes s_1, s_2 to nodes t_1, t_2 in a given directed graph G. The problem was proven NP-complete by Fortune, Hopcroft and Wyllie in [11]. The polynomial-time reduction is carried out as follows. For $G = (V, E)$ and four distinct nodes s_1, s_2, t_1, t_2, create a new graph $G' = (V', E')$ with

1. set $V' = V \cup \{s, t\}$,
2. set $E' = E \cup \{s \to s_1, s \to s_2, t_1 \to t, t_2 \to t\}$, and
3. length 1 assigned to edges $s \to s_1$ and $t_2 \to t$, and length 0 assigned to $s \to s_2$, $t_1 \to t$ and all edges in E.

Figure 9 shows a general graph G and its corresponding G'.

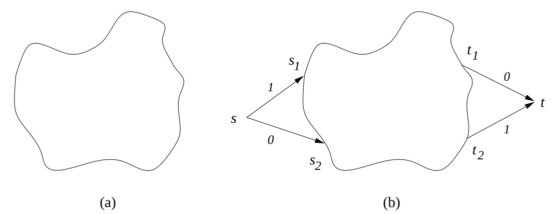

Figure 9. (a) Graph G. (b) Graph G'.

Then the question of "whether or not there exist two disjoint paths from s_1 to t_1 and s_2 to t_2 in G" is equivalent to the question of "whether or not there exist two disjoint paths (P_1, P_2) from s to t in G' with $z = l(P_1)+\alpha\cdot l(P_2) < 1+\alpha$". That is because any two disjoint paths from s to t in G' include edges $s \to s_1, s \to s_2, t_1 \to t, t_2 \to t$. Thus, either $s \to s_1$ and $t_1 \to t$ (resp. $s \to s_2$ and $t_2 \to t$) are in the same path, or $s \to s_1$ and $t_2 \to t$ (resp. $s \to s_2$ and $t_1 \to t$) are in the same path. In the former case, $z = 1+\alpha$ and there exist two disjoint paths from s_1 to t_1 and s_2 to t_2. In the latter case, $z = 2$ and there exist disjoint paths from s_1 to t_2 and s_2 to t_1. Let (P_1^*, P_2^*) be the optimal paths for the normalized α^--MIN-SUM 2-path problem on G'. Then, we have

$$\begin{cases} (l(P_1), l(P_2)) \in \{(1,1),(2,0)\} \\ (l(P_1^*), l(P_2^*)) \in \{(1,1),(2,0)\} \end{cases}$$

$$\Rightarrow \begin{cases} z = l(P_1) + \alpha\cdot l(P_2) \in \{1+\alpha, 2\} \\ z^* = l(P_1^*) + \alpha\cdot l(P_2^*) \in \{1+\alpha, 2\} \end{cases}$$

Suppose that there is a polynomial-time approximation algorithm $\mathcal{A}$ with error bound smaller than $\frac{2}{1+\alpha}$. Then algorithm $\mathcal{A}$ can find (P_1, P_2) in G' such that

$$\frac{l(P_1)+\alpha\cdot l(P_2)}{l(P_1^*)+\alpha\cdot l(P_2^*)} < \frac{2}{1+\alpha},$$

which leads to

$$(z^* = 1+\alpha) \Rightarrow (z < z^* \cdot \frac{2}{1+\alpha} = 2) \Rightarrow (z = 1+\alpha).$$

Then this algorithm can answer the question of "whether or not there exist two disjoint paths (P_1, P_2) from s to t with $z = l(P_1)+\alpha\cdot l(P_2) \le 1+\alpha$". We are led to conclude that algorithm $\mathcal{A}$ can solve 2DP problem in polynomial time. It cannot be true unless $P = NP$. ∎

4 Pseudo-Polynomial-Time Algorithm for Acyclic Directed Graphs

For the special case α^--MIN-SUM 2-path problem on acyclic directed graphs, there exists a pseudo-polynomial-time algorithm to find optimal solutions. The algorithm is obtained by extending the Li's algorithm [9] to find two disjoint paths on acyclic directed graphs for MIN-MAX objectiveh.

4.1 The Node-Disjoint Case

For this case, we adopt a technique used in a pseudo-polynomial-time algorithm of [9]. Given an acyclic directed graph $G = (V, E)$ and source s and destination t, we can relabel nodes with number 1 to $|V|$ to ensure that any edge $u \to v$ in E satisfies $u < v$, $s = 1$ and $t = |V|$ [15] (it is assumed that $s \to t \notin E$; otherwise we can add a node u and replace $s \to t$ by $s \to u$ and $u \to t$). After relabeling, we transform graph G into an acyclic directed graph $\overline{G} = (\overline{V}, \overline{E})$, whose nodes are arranged as a $|V| \cdot |V|$ array, as follows:

$\overline{V} = \{\langle u, v\rangle | u, v \in V$, and $u \neq v$ unless $u = v = s$ or $u = v = t\}$

$\overline{E} = \{\langle u, v\rangle \to \langle u, w\rangle | v \to w \in E$ and $v \leq u\}$
$\cup \{\langle v, u\rangle \to \langle w, u\rangle | v \to w \in E$ and $v \leq u\}$

Then we assign the lengths to edges in $\overline{G}$ as follows:
$l(\langle u, v\rangle \to \langle u, w\rangle) = \alpha \cdot l(v \to w)$, and
$l(\langle v, u\rangle \to \langle w, u\rangle) = l(v \to w)$.

Figure 10 shows an example of G and its corresponding $\overline{G}$.

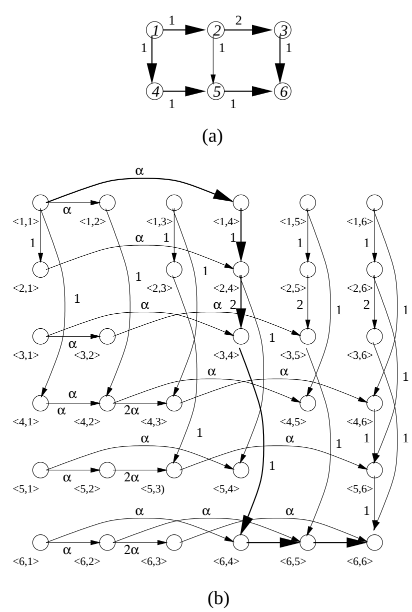

Figure 10. (a) An acyclic graph G. (b) Graph $\overline{G}$.

We call x and y of a node $\langle x, y\rangle$ in $\overline{G}$ the first and second label of the node, respectively. Given a path $\overline{P} = \langle u_1, v_1\rangle \to \langle u_2, v_2\rangle \to \cdots \to \langle u_m, v_m\rangle$ from $\langle 1, 1\rangle$ to $\langle |V|, |V|\rangle$ in $\overline{G}$, let $H(\overline{P})$ and $V(\overline{P})$ be the set of horizontal and vertical edges in $\overline{P}$ respectively. Define $V_1(\overline{P}) = (u_{i,1}, u_{i,2}, \cdots, u_{i,k})$ (resp. $V_2(\overline{P}) = (v_{j,1}, u_{j,2}, \cdots, u_{j,k'})$) as the sequence of distinct first (resp. second) labels of nodes in $V(\overline{P})$ (resp. $H(\overline{P})$). By a straightforward extension of the results of [15], we know that there exist two node-disjoint paths $P_1 = u_{i,1} \to u_{i,2} \to \cdots \to u_{i,k}$ and $P_2 = v_{i,1} \to v_{i,2} \to \cdots \to v_{i,k'}$ from s to t (from 1 to $|V|$ after relabeling) in G if and only if there exists a path $\overline{P}$ from $\langle 1, 1\rangle$ to $\langle |V|, |V|\rangle$ such that $V_1(\overline{P}) = u_{i,1} \to u_{i,2} \to \cdots \to u_{i,k}$ and $V_2(\overline{P}) = v_{i,1} \to v_{i,2} \to \cdots \to v_{i,k'}$. That is, P_1 and P_2 correspond to the vertical edges and horizontal edges in $\overline{P}$, respectively. In the example of Figure 10(a), there are two node-disjoint paths $P_1 = 1 \to 4 \to 5 \to 6$ and $P_2 = 1 \to 2 \to 3 \to 6$ in G. The corresponding path in $\overline{G}$ is $\overline{P} = \langle 1, 1\rangle \to \langle 4, 1\rangle \to \langle 4, 2\rangle \to \langle 4, 3\rangle \to \langle 4, 6\rangle \to \langle 5, 6\rangle \to \langle 6, 6\rangle$, as shown in Figure 10(b).

Then our goal is to find $\overline{P}$ such that

$$\min_{\substack{\overline{P} \in \overline{G} \\ V(\overline{P}) \geq H(\overline{P})}} \{V(\overline{P}) + H(\overline{P})\}$$

We use the similar "scanning and labeling" algorithm[9] to keep multiple labels at each node. The labels for a node are represented by $(x, y, < p, q >)$, where x and y are the total vertical and horizontal distances from $< s, s >$ to the current node along the path through previous node $< p, q >$. Different from [9], we use an array index by x to store these labels, and the size of the array is $L = \sum_{e \in E(G)} l(e)$. The algorithm is as follows:

label $< s, s >$ with $(1, 1, -)$;
for $p = s, \cdots, t$ **do**
  for $q = s, \cdots, t$ **do**
    for each node v adjacent to $< p, q >$ **do**
      let $\delta = l(< p, q > \to v)$;
      if $< p, q > \to v$ is a vertical edge **then**
      update label $(x + \delta, y, < p, q >)$ on v;
      if $< p, q > \to v$ is a horizontal edge **then**
      update label $(x, y + \delta, < p, q >)$ on v;
      (label update procedure: given new label $(x, y, < p, q >)$, if there is no existing label in the x element of the label array, add $(x, y, < p, q >)$ into the array as the x element; if there exists a $(x, y^*, < p^*, q^* >)$ in the x element, replace the element with $(x, y, < p, q >)$ if $y < y^*$).

if $< t, t >$ has no label **then stop** (no path $\overline{P}$ exists in $\overline{G}$);
else select a label $(x, y, < p, q >)$ that $\max(x, y) + \alpha \cdot \min(x, y)$ is minimized; trace the path back to $< s, s >$ to get the optimal path $\overline{P}$.

From the proof in [9], the above algorithm covers all valid non-redundant paths from $< s, s >$ to $< t, t >$. Thus, the $\overline{P}$ is the optimal path.

Complexity Analysis:

Let L be sum of all edge lengths in G. The scanning stage takes $O(|V|^3)$ time, as there are $|V|^2$ nodes and each node has at most $2|V|$ neighbors. The update procedure takes constant time. The final optimization stage takes $O(L)$ time. So the total time complexity is $O(|V|^3 + L)$. And the algorithm takes $O(|V|^2 L)$ space.

4.2 The Edge-Disjoint Case

For this case, we directly apply a technique of [9] that transforms the acyclic edge-disjoint case to the acyclic node-disjoint case, and then apply the algorithm presented in the previous section. First, we transform the given graph $G = (V, E)$ to a corresponding directed line-graph $\overline{G}$ [2], which contains $O(|E|)$ nodes and $O(|V|)$ edges. This can be done in $O(|E|)$ time. Then, finding two node-disjoint paths in $\overline{G}$ can be done in $O(|E|^3 + L)$ time. For more details, refer to [9].

5 Concluding Remarks

We have shown that for all four versions of the normalized α^--MIN-SUM 2-path problem one cannot obtain polynomial-time algorithms for finding optimal solutions, unless P = NP. We showed that MIN-SUM solutions can be used as good approximations of optimal normalized α^--MIN-SUM solutions. We also showed that for acyclic directed case, pseudo-polynomial-time algorithms exist for finding optimal normalized α^--MIN-SUM solutions. We showed that $\frac{2}{1+\alpha}$ is the best possible approximation bound for normalized α^--MIN-SUM solutions in directed graphs. A challenging question is if $\frac{2}{1+\alpha}$ is also the optimal approximation bound for undirected graphs. Polynomial-time algorithms may exist for finding optimal normalized α^--MIN-SUM 2-path solutions or achieving smaller approximation bound for graphs of some special properties.

References

[1] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman (1979).

[2] F. Harary, *Graph Theory*, Addison-Wesley, 1972.

[3] S. Even, *Graph Algorithms*, Computer Science (1979).

[4] T.H. Cormen, C.E. Leiserson and R.L. Rivest, it Introduction to Algorithms, The MIT Press (1990).

[5] R. Bhandari, *Survivable Networks - Algorithms for Diverse Routing*, Kluwer (1999).

[6] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization - Algorithms and Complexity*, Dover (1998).

[7] L.R. Ford and D.R. Fulkerson, *Flows in Networks*, Princeton, 1962.

[8] A. Itai, Y. Perl and Y. Shiloach, "The Complexity of finding Maximum Disjoint Paths with Length Constraints", *Networks* 12 (1982) 277-286.

[9] C. L. Li, S. T. McCormick, and D. Simchi-Levi, "The Complexity of Finding Two Disjoint Paths with Min-Max Objective Function", *Discrete Appl. Math.* 26(1) (1990), 105-115.

[10] C. L. Li, S. T. McCormick, and D. Simchi-Levi, "Finding Disjoint Paths with Different Path-Costs: Complexity and Algorithms", *Networks*, 22 (1992), 653-667.

[11] S. Fortune, J. Hopcroft, and J. Wyllie, "The directed subgraph homeomorphism problem", *Theoret. Comput. Sci.*, 10 (1980), 111-121.

[12] B. Yang, S.Q. Zheng and S. Katukam, "Finding Two Disjoint Paths in a Network with Min-Min Objective Function", *Proceedings of the 15th IASTED International Conference on Parallel and Distributed Computing and Systems*, (2003), 75-80.

[13] B. Yang, S.Q. Zheng and E. Lu, "Finding Two Disjoint Paths in a Network with Normalized α^+-Min-Sum Objective Function", to appear in *Proceedings of the 16th International Symposium on Algorithms and Computation*, (ISAAC 2005).

[14] Y. Liu, D. Tipper, "Successive Survivable Routing for Node Failures", *IEEE Globecom* (2001) 2093-2097.

[15] Y. Perl and Y. Shiloach, "Finding Two Disjoint Paths between Two Pairs of Vertices in a Graph", *J. ACM*, 25(1) (1978), 1-9.

[16] Y. Shiloach, A polynomial solution to the undirected two paths problem. J. ACM 27(3) (1980) 445-456.

[17] M.R. Garey, D.S. Johnson and L. Stockmeyer, "Some Simplified NP-Complete Graph Problems", *Theoret. Comput. Sci.* 1 (1976) 237-267.

[18] J. W. Suurballe, "Disjoint Paths in a Network", *Networks* 4 (1974), 125-145.

[19] J. W. Suurballe and R. E. Tarjan, "A Quick Method for Finding Shortest Pairs of Disjoint Paths", *Networks* 14 (1984), 325-336.