Finding Minimum-Cost Paths with Minimum Sharability

S.Q. Zheng
Dept. of Computer Science
University of Texas at Dallas
Richardson, TX 75083, USA
Email: sizheng@utdallas.edu

Bing Yang
Cisco Systems
2200 E. President G. Bush HW
Richardson, TX 75082, USA
Email: binyang@cisco.com

Mei Yang
Dept. of Elec. and Comput. Engg.
University of Nevada
Las Vegas, NV 89154, USA
Email: meiyang@egr.unlv.edu

Jianping Wang
Dept. of Computer Science
City University of Hong Kong
Tat Chee Ave., Kowloon, HK
Email: jianwang@cityu.edu.hk

***Abstract*— In communication networks, multiple communication paths sharing minimum number of links or/and nodes may be desirable for improved performance, resource utilization and reliability. We introduce the notion of link sharability and node sharability, and consider the problems of finding minimum-cost k paths subject to minimum link/node sharability constraints. We identify 65 different link/node sharability constraints, and consider the fundamental problem of finding minimum-cost k paths between a pair of nodes under these constraints. We present a unified polynomial-time algorithm scheme for solving this problem subject to 25 of these different sharability constraints.**

Keywords: Network, graph, routing, network planning, protocol, algorithm, protection, reliability, survivability, disjoint paths, multiple paths, network flow, complexity.

I. Introduction

In a communication network the connection between a source node and a destinations node is a path between them. Currently, most networks employ protocols based on shortest path routing algorithms which determine a single path of minimum cost. Finding multiple paths between a source and a destination has been proposed. Potential benefits of multiple paths include improved reliability (e.g. [7], [12], [15], [17], [18], [19], [20], [21], [22]), load balancing (e.g. [6], [16]), higher network throughput (e.g. [8], [16]), and alleviation of congestion (e.g.[2], [7]).

It is desirable that multiple paths are link or/and node disjoint. Assume that a network is modeled as a weighted graph $G = (V, E)$, where V is the set of nodes, E is the set of links connecting nodes, and each link is associated with a nonnegative cost. In the literature, the complexities of various versions of the problem of finding optimal disjoint paths between two nodes $s, t \in G$ have been investigated. Ford and Fulkerson proposed a polynomial-time algorithm for finding two paths with minimum total cost (named the *Min-Sum 2-Path Problem*) based on minimum-cost network flow model [4]. Suurballe and Tarjan provided a different treatment, and presented algorithms that are more efficient [13], [14]. Li *et al.* proved that the problem of finding two disjoint paths such that the cost of the longer path is minimized (named the *Min-Max 2-Path Problem*) are strongly NP-complete [10]. They also considered a generalized min-sum problem (referred as the *G-Min-Sum k-Path Problem*) assuming that each link is associated with k different lengths. The objective of this problem is to find k disjoint paths such that the total cost of the paths is minimized, where the jth link-cost is associated with the jth path. They showed that the G-Min-Sum k-Path Problem are strongly NP-complete for $k \geq 2$ [11]. In [18]-[21], a set of optimal disjoint 2-path problems with different objective functions, including the *Min-Min 2-path problem*, the *α-MIN-SUM 2-path problem*, and the *MinSum-MinMin 2-path problem*, are considered and proved to be NP-complete.

Given a pair of nodes, finding k, $k > 1$, disjoint paths, though desirable, may not always be possible in practical network applications for at least two reasons. First, if the network is too sparse, such paths may not physically exist. Second, if some links are overly saturated, additional traffic on these links may be prohibited so that two disjoint paths without using these prohibited links do not exist. When k disjoint paths do not exist, alternatively k paths from the source to the destination with minimum shared links/nodes should be found. In the context of network reliability, these paths can provide partial protection[22].

In the literature, limited work on multiple paths with minimum number of shared links/nodes has been reported. In [3], an algorithm based on minimum-cost network flow (MCNF) is given for finding the k-best paths (i.e., k paths with minimum node sharing). However, the algorithm can only be applied to trellis graphs. In [12], an algorithm is provided to transform an arbitrary graph to a trellis graph and then to obtain the k-best paths by the algorithm shown in [3]. As analyzed in [9], this solution is only a heuristic one and the complexity of the transformation is quite high. In [9], an MCNF-based algorithm is proposed for finding k-best paths in arbitrary networks. However, only best link-disjoint paths are considered in [9].

In this paper, we introduce the notion of *link sharability* and *node sharability*, which are a variation of the concept of vulnerability defined in [15]. We use this notion to characterize the degree of link/node sharing among different paths. Larger link/node sharability implies more link/node sharing among a set of paths. A set of paths are link-disjoint if the link sharability of the paths is 0, and they are node-disjoint if their node sharability is 0 (in this case, their link sharability is also 0). We define five basic sharability constraints: min-sum link sharability constraint, min-sum node sharability constraint, min-max link sharability constraint, min-max node sharability constraint, and empty constraint (no restriction on sharabilities). Based on these basic sharability constraints, we identify 65 composite sharability constraints, which are obtained by selections and permutations of basic sharability constraints. We investigate the problem finding minimum-cost paths subject to these sharability constraints by considering the problem of finding minimum-cost k paths from node s to node t in network G. This is a generalization of the classical fundamental problem of finding minimum-cost k paths from node s to node t which has received considerable attention in the context of protecting a network against link/node failure. By presenting a general algorithm scheme, we show that 25 versions of the problem finding minimum-cost k paths from node s to node t in network G subject to link or/and node sharability constraints are solvable in polynomial time.

The rest of the paper is organized as follows. In Section II, we define link sharability and node sharability, introduce 65 different sharability constraints. We show that a subset of 25 sharability constraints are mutually inequivalent. Section III presents a unified algorithm scheme that can be used to generate different algorithms for solving the problem of finding minimum-cost k paths from s to t subject to the 25 different sharability constraints in polynomial time. In Section IV, we generalize our algorithm scheme to solve the problem of finding minimum-cost k paths subject to constraints on allowable individual link and node sharabilities, and the problem of finding minimum-cost one-to-many and many-to-one paths subject to various sharability constraints. Section V concludes the paper.

II. Link and Node Sharability Constraints

We restrict our discussions to directed graphs. All our algorithms and claims are applicable to undirected graphs, since undirected graphs can be converted to directed graphs easily. In the rest of the paper, the terms graph and network are used interchangeably. So are the terms of edge and link.

Let $G = (V, E)$ be a directed graph with non-negative *cost* $l(e)$ or $l(u, v)$ defined for link $e = (u, v)$. Further, we assume that G is simple; i.e. it has no self-loop and parallel links. Given a source s and a destination t in G, let $P = \{P_1, P_2, \cdots, P_k\}$ be a set of k paths (s-t paths) in graph $G = (V, E)$ from s to t.

We define

$$\delta(e, P_i) = \begin{cases} 1, & e \in P_i \\ 0, & e \notin P_i \end{cases}$$

and

$$\delta(e, P) = \sum_{i=1}^{k} \delta(e, P_i).$$

Then $\delta(e, P)$ represents the number of times e appears in P. We define

$$\Delta(P) = \sum_{e \in P} (\delta(e, P) - 1),$$

and

$$\delta(P) = \max_{e \in P} (\delta(e, P) - 1).$$

$\Delta(P)$ is called *total link sharability* of P, and $\delta(P)$ is called the *maximum link sharability* of P. Clearly, $\Delta(P) = 0$ if and only if $\delta(P) = 0$, and either $\Delta(P) = 0$ or $\delta(P) = 0$ indicates that all paths in P are link-disjoint.

Similarly, we define

$$\gamma(v, P_i) = \begin{cases} 1, & v \in P_i \\ 0, & v \notin P_i \end{cases}$$

and

$$\gamma(v, P) = \sum_{i=1}^{k} \gamma(v, P_i).$$

Then $\gamma(v, P)$ represents the number of times node v is shared among paths in P. We define

$$\Gamma(P) = \sum_{v \in P, v \neq s,t} (\gamma(v, P) - 1),$$

and

$$\gamma(P) = \max_{v \in P, v \neq s,t} (\gamma(v, P) - 1).$$

$\Gamma(P)$ is called the *total node sharability* of P and $\gamma(P)$ is called the *maximum node sharability* of P. Clearly, $\Gamma(P) = 0$ if and only if $\gamma(P) = 0$, and either $\Gamma(P) = 0$ or $\gamma(P) = 0$ indicates that all paths in P are node-disjoint.

The total cost of k paths in P is defined as

$$l(P) = \sum_{i=1}^{k} l(P_i),$$

where $l(P_i) = \sum_{e \in P_i} l(e)$.

The problems studied in this paper have a common feature of finding a set of paths $P = \{P_1, P_2, \cdots, P_k\}$ in G with minimum $l(P)$, subject to various combinations of minimum $\Delta(P)$, minimum $\Gamma(P)$, minimum $\delta(P)$ and minimum $\gamma(P)$ as constraints.

Let $\langle C \rangle = \langle C_q, C_{q-1}, \cdots, C_1 \rangle$ be an ordered list of constraints. An optimization problem Π subject to ordered constraint list $\langle C \rangle$ is to find a solution $S(I)$ for an instance I of Π such that

(1) : $S(I)$ satisfies C_1;
(2) : $S(I)$ satisfies C_2 subject to condition (1);
…
(q) : $S(I)$ satisfies C_q subject to condition $(q-1)$; and
($q+1$): $S(I)$ has the optimal solution value among all solutions that satisfy C_q.

We call $\langle C \rangle$ an *ordered composite constraint*, and C_i's the *component constraints* of $\langle C \rangle$. Constraints are defined recursively. An empty list is an ordered composite constraint; it is also called *empty constraint*. A single constraint is an ordered composite constraint. Then, an ordered composite constraint is an ordered pair of two ordered composite constraints. Having defined this recursive structure, we refer to ordered composite constraints simply as constraints.

Define empty constraint, minimum Δ, minimum Γ, minimum δ, and minimum γ (denoted by $\langle\ \rangle$, $\langle \min \Delta \rangle$, $\langle \min \Gamma \rangle$, $\langle \min \delta \rangle$, and $\langle \min \gamma \rangle$, respectively) as *basic sharability constraints*. $\langle \min \Delta \rangle$ and $\langle \min \Gamma \rangle$ are called *min-sum constraints*, and $\langle \min \delta \rangle$ and $\langle \min \gamma \rangle$ are called *min-max constraints*. In general, we can have the following set of constraints: $Z = \{\langle X_i, \cdots, X_1 \rangle | \langle X_1 \rangle, \cdots, \langle X_i \rangle \in \{\langle\ \rangle, \langle \min \delta \rangle, \langle \min \gamma \rangle,$

$\langle \min \Delta \rangle$, $\langle \min \Gamma \rangle \}$, $1 \leq i \leq 4\}$. Clearly, $\langle \langle \ \rangle, \langle \ \rangle \rangle = \langle \ \rangle$, $\langle \langle \ \rangle, \langle X \rangle \rangle = \langle X \rangle$ and $\langle \langle X \rangle, \langle \ \rangle \rangle = \langle X \rangle$. Let $S(n, r)$ denote the number of ordered r-tuples of distinct elements from an n-element set. Then, we have

$$|Z| = \sum_{i=0}^{4} S(4, i) = \sum_{i=0}^{4} \frac{4!}{i!} = 65$$

different constraints. We are able to show that all of these 65 versions of the problem of finding minimum-cost k paths subject to sharability constraints are polynomial-time solvable. In this paper, we focus on 25 constraints listed in Table I, i.e. the composite constraints with min-max component constraints (if any) having priority higher than min-sum component constraints (if any). The version corresponding to constraint $\langle C_0 \rangle$ can be reduced to a problem of finding minimum-cost network flow (MCNF) of flow value k. Thus, such a problem of finding minimum-cost k s-t paths can be solved by an MCNF algorithm, such as the *successive shortest path algorithm* [1], in $O(k \cdot (|E| + |V| \log |V|))$ time. Each constraint $\langle C \rangle$ in the rest of Table I can be partitioned into two component constraints $\langle C' \rangle$ and $\langle C'' \rangle$ such that C' contains min-max constraints and C'' contains min-sum constraints. Then, $\langle C \rangle$ can be considered as a constraint $\langle C'', C' \rangle$ formed by concatenating C'' and C'. For easy reference, we call $\langle C'', C' \rangle$ the *normal form* of C. Based on normal forms, we divide the constraints $\langle C_1 \rangle$ to $\langle C_{24} \rangle$ in Table I into five classes as follows.

Class 1 : C'' is empty. This class contains $\langle C_1 \rangle$ to $\langle C_4 \rangle$.
Class 2 : C'' is $\langle \min \Delta \rangle$. This class contains $\langle C_5 \rangle$ to $\langle C_9 \rangle$.
Class 3 : C'' is $\langle \min \Gamma \rangle$. This class contains $\langle C_{10} \rangle$ to $\langle C_{14} \rangle$.
Class 4 : C'' is $\langle \min \Gamma, \min \Delta \rangle$. This class contains $\langle C_{15} \rangle$ to $\langle C_{20} \rangle$.
Class 5 : C'' is $\langle \min \Delta, \min \Gamma \rangle$. This class contains $\langle C_{20} \rangle$ to $\langle C_{24} \rangle$.

For example, the normal form of constraint $\langle C_{22} \rangle = \langle \min \Delta, \min \Gamma, \min \gamma \rangle$ is $\langle \langle C_{20} \rangle, \langle C_2 \rangle \rangle$, and it is in Class 5. The version corresponding to constraint $\langle C_{22} \rangle$ is to find a set P^* of k paths from s to t such that (1) P^* has min-max node sharability; (2) P^* has min-sum node sharability subject to condition (1); (3) P^* has min-sum link sharability subject to condition (2); and (4) $l(P^*) = \min\{P' | P'$ is a solution that satisfies (3)$\}$. This classification is useful in the analysis of our algorithm scheme.

Remark 1: The problem we are considering is a problem with a prioritized hierarchy of optimization objectives (the total cost of the paths coming the last). The objective of a higher priority narrows feasible solution space of the objectives of lower priorities. According to this feature of hierarchical optimization objective "constraints", we use ordered constraints to refer to ordered optimization objectives only for the sake of easy understanding. Readers should keep in mind that, strictly speaking, this is not a constrained optimization problem in the classical sense.

Consider two versions Π_1 and Π_2 of the minimum-cost k-path problem subject to ordered composite constraint $\langle X \rangle$ and $\langle Y \rangle$, respectively. For the same graph G, their solutions can be different if $\langle X \rangle$ and $\langle Y \rangle$ are different. The following statement is always true: if $\langle Y \rangle$ is a sublist of $\langle X \rangle$, i.e. all constraints in $\langle Y \rangle$ are in $\langle X \rangle$ and they are in the same order as they appear in $\langle X \rangle$, then the cost of the solution subject to $\langle Y \rangle$ is no larger than the cost of the solution subject to $\langle X \rangle$. Clearly, satisfying more sharability constraints tends to reduce link/node sharing with increased cost. Thus, there is a tradeoff between cost and sharability between choosing $\langle X \rangle$ and $\langle Y \rangle$. The 25 different versions of the minimum-cost k-path problem defined by the constraints of Table I provide a wide spectrum of constraint priorities and cost-sharability tradeoffs for the same network.

In the context of finding minimum-cost k s-t paths, we say that two constraints $\langle X \rangle$ and $\langle Y \rangle$ are equivalent if and only if any optimal solution obtained under $\langle X \rangle$ is also an optimal solution obtained under $\langle Y \rangle$ for any network. One question arises: are all the 25 composite constraints given in Table I mutually inequivalent? The following theorem gives a definite answer.

Theorem 1: Constraints of Table I are mutually inequivalent.

Proof: See Appendix. ■

III. An Algorithm Scheme

In this section, we present a unified algorithm scheme for all versions of the problem of finding optimal k s-t paths subject to the constraints defined in Table I. This scheme, which is used to generate slightly different algorithms by reducing the problem of finding a set P^* of minimum-cost k paths in G to finding a minimum-cost flow f^* in G'' using different cost and capacity functions, has three steps.

Step 1 : Compute min-max link sharability k^L or/and min-max node sharability k^N if needed, and construct flow network $G'' = (V'', E'')$ from $G = (V, E)$ according to sharability requirement.
Step 2 : Find a minimum-cost flow f^* of flow value k from s to t in G''.
Step 3 : Construct a set P^* of k s-t paths in G from the flow f^* in G''.

For Step 1, two transformations, *TRANSFORM-1* and *TRANSFORM-2*, are introduced.

TRANSFORM-1: Obtain $G' = (V', E')$ from $G = (V, E)$ by *node splitting* as follows: replace each node v that is neither s nor t by two nodes v and v' such that all links ending at v in G also end at v in G' and all links originating from v in G originate from v', and then add a link (v, v') (see Figure 1 and Figure 3).

TRANSFORM-2: Obtain graph $G'' = (V'', E'')$ from $G' = (V', E')$ by *link splitting* as follows: replace each link e in G' by two parallel links with the same direction of e. We denote the two links generated from a link (u', v) in G' corresponding to link (u, v) in G by $\underline{(u', v)}$ and $\overline{(u', v)}$, which are called the *primary link-generated* $u - v$ *link* (or simply, *primary* $u - v$ *link*) and the *secondary link-generated* $u - v$ *link* (or simply, *secondary* $u - v$ *link*), respectively (note: u' can be source node s). We denote the two links generated from a link (v, v') in

	constraints		constraints
$\langle C_0\rangle$	$\langle\ \rangle$		
$\langle C_1\rangle$	$\langle\min\delta\rangle = \langle\langle C_0\rangle,\langle C_1\rangle\rangle =$ $\langle\langle\ \rangle,\langle\min\delta\rangle\rangle$	$\langle C_{13}\rangle$	$\langle\min\Gamma,\min\gamma,\min\delta\rangle = \langle\langle C_{10}\rangle,\langle C_3\rangle\rangle =$ $\langle\langle\min\Gamma\rangle,\langle\min\gamma,\min\delta\rangle\rangle$
$\langle C_2\rangle$	$\langle\min\gamma = \langle\langle C_0\rangle,\langle C_2\rangle\rangle =\rangle$ $\langle\langle\ \rangle,\langle\min\gamma\rangle\rangle$	$\langle C_{14}\rangle$	$\langle\min\Gamma,\min\delta,\min\gamma\rangle = \langle\langle C_{10}\rangle,\langle C_4\rangle\rangle =$ $\langle\langle\min\Gamma\rangle,\langle\min\delta,\min\gamma\rangle\rangle$
$\langle C_3\rangle$	$\langle\min\gamma,\min\delta\rangle = \langle\langle C_0\rangle,\langle C_3\rangle\rangle =$ $\langle\langle\ \rangle,\langle\min\gamma,\min\delta\rangle\rangle$	$\langle C_{15}\rangle$	$\langle\min\Gamma,\min\Delta\rangle = \langle\langle C_{15}\rangle,\langle C_0\rangle\rangle =$ $\langle\langle\min\Gamma,\min\Delta\rangle,\langle\ \rangle\rangle$
$\langle C_4\rangle$	$\langle\min\delta,\min\gamma\rangle = \langle\langle C_0\rangle,\langle C_4\rangle\rangle =$ $\langle\langle\ \rangle,\langle\min\delta,\min\gamma\rangle\rangle$	$\langle C_{16}\rangle$	$\langle\min\Gamma,\min\Delta,\min\delta\rangle = \langle\langle C_{15}\rangle,\langle C_1\rangle\rangle =$ $\langle\langle\min\Gamma,\min\Delta\rangle,\langle\min\delta\rangle\rangle$
$\langle C_5\rangle$	$\langle\min\Delta\rangle = \langle\langle C_5\rangle,\langle C_0\rangle\rangle =$ $\langle\langle\min\Delta,\langle\ \rangle\rangle$	$\langle C_{17}\rangle$	$\langle\min\Gamma,\min\Delta,\min\gamma\rangle = \langle\langle C_{15}\rangle,\langle C_2\rangle\rangle =$ $\langle\langle\min\Gamma,\min\Delta\rangle,\langle\min\gamma\rangle\rangle$
$\langle C_6\rangle$	$\langle\min\Delta,\min\delta\rangle = \langle\langle C_5\rangle,\langle C_1\rangle\rangle =$ $\langle\langle\min\Delta\rangle,\langle\min\delta\rangle\rangle$	$\langle C_{18}\rangle$	$\langle\min\Gamma,\min\Delta,\min\gamma,\min\delta\rangle = \langle\langle C_{15}\rangle,\langle C_3\rangle\rangle =$ $\langle\langle\min\Gamma,\min\Delta\rangle,\langle\min\gamma,\min\delta\rangle\rangle$
$\langle C_7\rangle$	$\langle\min\Delta,\min\gamma\rangle = \langle\langle C_5\rangle,\langle C_2\rangle\rangle =$ $\langle\langle\min\Delta\rangle,\langle\min\gamma\rangle\rangle$	$\langle C_{19}\rangle$	$\langle\min\Gamma,\min\Delta,\min\delta,\min\gamma\rangle = \langle\langle C_{15}\rangle,\langle C_4\rangle\rangle =$ $\langle\langle\min\Gamma,\min\Delta\rangle,\langle\min\delta,\min\gamma\rangle\rangle$
$\langle C_8\rangle$	$\langle\min\Delta,\min\gamma,\min\delta\rangle = \langle\langle C_5\rangle,\langle C_3\rangle\rangle =$ $\langle\langle\min\Delta\rangle,\langle\min\gamma,\min\delta\rangle\rangle$	$\langle C_{20}\rangle$	$\langle\min\Delta,\min\Gamma\rangle = \langle\langle C_{20}\rangle,\langle C_0\rangle\rangle =$ $\langle\langle\min\Delta,\min\Gamma\rangle,\langle\ \rangle\rangle$
$\langle C_9\rangle$	$\langle\min\Delta,\min\delta,\min\gamma\rangle = \langle\langle C_5\rangle,\langle C_4\rangle\rangle =$ $\langle\langle\min\Delta\rangle,\langle\min\delta,\min\gamma\rangle\rangle$	$\langle C_{21}\rangle$	$\langle\min\Delta,\min\Gamma,\min\delta\rangle = \langle\langle C_{20}\rangle,\langle C_1\rangle\rangle$ $\langle\langle\min\Delta,\min\Gamma\rangle,\langle\min\delta\rangle\rangle$
$\langle C_{10}\rangle$	$\langle\min\Gamma\rangle = \langle\langle C_{10}\rangle,\langle C_0\rangle\rangle =$ $\langle\langle\min\Gamma\rangle,\langle\ \rangle\rangle$	$\langle C_{22}\rangle$	$\langle\min\Delta,\min\Gamma,\min\gamma\rangle = \langle\langle C_{20},\rangle\langle C_2\rangle\rangle =$ $\langle\langle\min\Delta,\min\Gamma\rangle,\langle\min\gamma\rangle\rangle$
$\langle C_{11}\rangle$	$\langle\min\Gamma,\min\delta\rangle = \langle\langle C_{10}\rangle,\langle C_1\rangle\rangle =$ $\langle\langle\min\Gamma\rangle,\langle\min\delta\rangle\rangle$	$\langle C_{23}\rangle$	$\langle\min\Delta,\min\Gamma,\min\gamma,\min\delta\rangle = \langle\langle C_{20}\rangle,\langle C_3\rangle\rangle$ $\langle\langle\min\Delta,\min\Gamma\rangle,\langle\min\gamma,\min\delta\rangle\rangle$
$\langle C_{12}\rangle$	$\langle\min\Gamma,\min\gamma\rangle = \langle\langle C_{10}\rangle,\langle C_2\rangle\rangle =$ $\langle\langle\min\Gamma\rangle,\langle\min\gamma\rangle\rangle$	$\langle C_{24}\rangle$	$\langle\min\Delta,\min\Gamma,\min\delta,\min\gamma\rangle = \langle\langle C_{20},\rangle,\langle C_4\rangle\rangle =$ $\langle\langle\min\Delta,\min\Gamma\rangle,\langle\min\delta,\min\gamma\rangle\rangle$

TABLE I

25 DIFFERENT CONSTRAINTS FOR THE MINIMUM-COST k-PATH PROBLEM.

	cost and capacity		ccost and capacity
$\langle C_0\rangle$	$[(l(e),k),(\times,0)],[(0,k),(\times,0)]$		
$\langle C_1\rangle$	$[(l(e),k^L+1),(\times,0)],[(0,k),(\times,0)]$	$\langle C_{13}\rangle$	$[(l(e),k^L+1),(\times,0)],[(0,1),(M,k^N)]$
$\langle C_2\rangle$	$[(l(e),k),(\times,0)],[(0,k^N+1),(\times,0)]$	$\langle C_{14}\rangle$	$[(l(e),k^L+1),(\times,0)],[(0,1),(M,k^N)]$
$\langle C_3\rangle$	$[(l(e),k^L+1),(\times,0)],[(0,k^N+1),(\times,0)]$	$\langle C_{15}\rangle$	$[(l(e),1),(M'+l(e),k-1)],[(0,1),(M,k-1)]$
$\langle C_4\rangle$	$[(l(e),k^L+1),(\times,0)],(0,k^N+1),(\times,0)]$	$\langle C_{16}\rangle$	$[(l(e),1),(M'+l(e),k^L)],[(0,1),(M,k-1)]$
$\langle C_5\rangle$	$[(l(e),1),(M+l(e),k-1)],[(0,k),(\times,0)]$	$\langle C_{17}\rangle$	$[(l(e),1),(M'+l(e),k-1)],[(0,1),(M,k^N)]$
$\langle C_6\rangle$	$[(l(e),1),(M+l(e),k^L)],[(0,k),(\times,0)]$	$\langle C_{18}\rangle$	$[(l(e),1),(M'+l(e),k^L)],[(0,1),(M,k^N)]$
$\langle C_7\rangle$	$[(l(e),1),(M+l(e),k-1)],[(0,k^N+1),(\times,0)]$	$\langle C_{19}\rangle$	$[(l(e),1),(M'+l(e),k^L)],[(0,1),(M,k^N)]$
$\langle C_8\rangle$	$[(l(e),1),(M+l(e),k^L)],[(0,k^N+1),(\times,0)]$	$\langle C_{20}\rangle$	$[(l(e),1),(M+l(e),k-1)],[(0,1),(M',k-1)]$
$\langle C_9\rangle$	$[(l(e),1),(M+l(e),k^L)],[(0,k^N+1),(\times,0)]$	$\langle C_{21}\rangle$	$[(l(e),1),(M+l(e),k^L)],[(0,1),(M',k-1)]$
$\langle C_{10}\rangle$	$[(l(e),k),(\times,0)],[(0,1),(M,k-1)]$	$\langle C_{22}\rangle$	$[(l(e),1),(M+l(e),k-1)],[(0,1),(M',k^N)]$
$\langle C_{11}\rangle$	$[(l(e),k^L+1),(\times,0)],[(0,1),(M,k-1)]$	$\langle C_{23}\rangle$	$[(l(e),1),(M+l(e),k^L)],[(0,1),(M',k^N)]$
$\langle C_{12}\rangle$	$[(l(e),k),(\times,0)],[(0,1),(M,k^N)]$	$\langle C_{24}\rangle$	$[(l(e),1),(M+l(e),k^L)],[(0,1),(M',k^N)]$

TABLE II

$(cost, capacity)$ ASSIGNMENTS IN G'' FOR THE 25 VERSIONS OF THE k-PATH PROBLEM OF TABLE I.

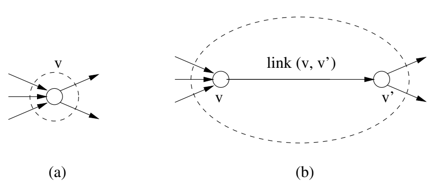

Fig. 1. Node splitting. (a) node v in G. (b) v is replaced by two nodes and an edge (v, v').

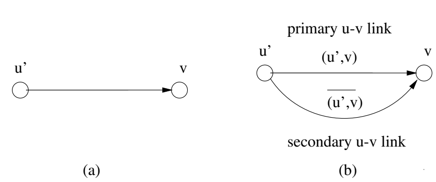

Fig. 2. Link splitting. (a) Original link in G'. (b) Two links obtained for the link of (a).

G' corresponding to node v in G by (v, v') and $\overline{(v, v')}$, which are called the *primary node-generated* $v - v$ *link* (or simply, *primary* v *link*) and the *secondary node-generated* $v - v$ *link* (or simply, *secondary* v *link*), respectively (see Figure 2 and Figure 3).

TRANSFORM-1 is used to construct G' from G and compute min-max link sharability k^L or/and min-max node sharability k^N, which are useful in defining the capacities of links in G''. First, let us consider the versions of the minimum-cost k-path problem corresponding to the constraints in Class 1 (i.e. $\{\langle C_1\rangle, \langle C_2\rangle, \langle C_3\rangle, \langle C_4\rangle\}$). For these versions, we need to

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2007 proceedings.

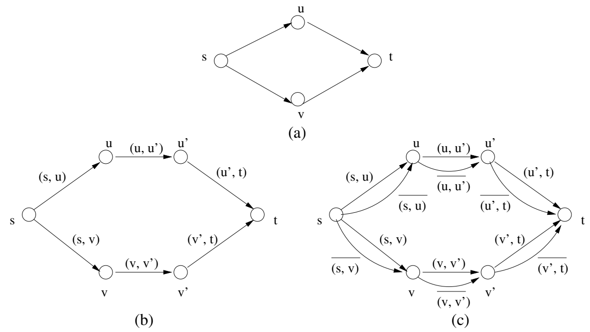

Fig. 3. An example. (a) Original graph. (b) Graph G' constructed from G by *TRANSFORM-1*. (c) Graph G'' constructed from from G' by *TRANSFORM-2*. Capacities for links of (b), and $(cost, capacity)$ pairs of (c) are not shown.

compute min-max link sharability k^L or/and min-max node sharability k^N. The following procedure *MinMax-Sharability* returns k^L and k^N with $flag$ set to 0 and 1, respectively.

procedure *MinMax-Sharability*$(G', flag, k, k')$
begin
$E'_1 :=$ the set of all edges in E' that are resulted from node splitting in the construction of G' by *TRANSFORM-1*;
$E'_2 := E' - E'_1$;
if flag = 0 **then** $E^* := E'_2$ **else** $E^* := E'_1$;
assign capacity k' to all links in $E' - E^*$;
$low := 0$; $high := k$; $lim := \lfloor k/2 \rfloor$;
while $low \neq high$ **do**
for each edge $e \in E^*$ **do** assign e capacity lim **end-for**;
find the maximum flow f in G' from s to t by running a maximum flow algorithm;
if $|f| < k$ **then** $low := lim$ and $lim := low + \lceil (high - low)/2 \rceil$;
else $high := lim$ and $lim := low + \lfloor (high - low)/2 \rfloor$;
end-while
return $lim - 1$;
end

The min-max sharabilities for the problems of Class 1 are computed as follows:

$\langle C_1 \rangle$: $flag := 0$; $k' := k$;
$k^L :=$ *MinMax-Sharability*$(G', flag, k, k')$;
$\langle C_2 \rangle$: $flag := 1$; $k' := k$;
$k^N :=$ *MinMax-Sharability*$(G', flag, k, k')$;
$\langle C_3 \rangle$: $flag := 0$; $k' := k$;
$k^L :=$ *MinMax-Sharability*$(G', flag, k, k')$;
$flag := 1$; $k' := k^L$;
$k^N :=$ *MinMax-Sharability*$(G', flag, k, k')$;
$\langle C_4 \rangle$: $flag := 1$; $k' := k$;
$k^N :=$ *MinMax-Sharability*$(G', flag, k, k')$;
$flag := 0$; $k' := k^N$;
$k^L :=$ *MinMax-Sharability*$(G', flag, k, k')$;

It is important to note that procedure *MinMax-Sharability* is invoked twice for $\langle C_3 \rangle$ and $\langle C_4 \rangle$. For $\langle C_3 \rangle$ (resp. $\langle C_4 \rangle$), k^L (resp. k^N) is computed first, and the value of k^L (resp. k^N) affects the value of k^N (resp. k^L) that is computed by the second call to *MinMax-Sharability*. Thus, the k^L and k^N values computed for $\langle C_3 \rangle$ and $\langle C_4 \rangle$ for the same network G may be different. For all constraints of Table I in the normal form $\langle C'', C' \rangle$ with nonempty min-max sharability component constraint $\langle C' \rangle$, procedure *MinMax-Sharability* is used to compute min-max link sharability k^L or/and min-max node sharability k^N for $\langle C' \rangle$.

Each link in G'' is assigned a cost-capacity pair. More specifically, for each link $e = (u, v)$ in G, its corresponding primary $u - v$ link (u', v) and secondary $u - v$ link $\overline{(u', v)}$ in G'' are assigned pair $(cost(u', v), capacity(u', v))$ and $(cost\overline{(u', v)}, capacity\overline{(u', v)})$, respectively. Similarly, for each node v in G, its corresponding primary v link (v, v') and secondary v link $\overline{(v, v')}$ are assigned pair $(cost(v, v'), capacity(v, v'))$ and $(cost\overline{(v, v')}, capacity\overline{(v, v')})$, respectively. We use an ordered pair of cost-capacity pairs

$$[(cost(u', v), capacity(u', v)), (cost\overline{(u', v)}, capacity\overline{(u', v)})],$$

$$[(cost(v, v'), capacity(v, v')), (cost\overline{(v, v')}, capacity\overline{(v, v')})]$$

to represent the cost and capacity assignment for the links of G''. Let $l_{max} = \max_{e \in E} l(e)$. Without loss of generality, assume that $l_{max} > 1$. If this is not true, we simply choose a constant c to scale up the cost of each link e from $l(e)$ to $c \cdot l(e)$ so that $l_{max} > 1$. Define $M = k \cdot |V| \cdot l_{max}$ and $M' = M^2 = k^2 \cdot |V|^2 \cdot l_{max}^2$. Then, all versions of the minimum-cost k-path problem considered are reduced to finding minimum-cost flows, and we present algorithms by only presenting the costs and capacities assigned to the links in G''. We list $(cost, capacity)$ assignments for all the 25 constraints of Table I in Table II. We use $\times$ to represent a "don't care" value.

In Step 2, an MCNF algorithm is applied to the flow network G'' constructed in Step 1 to find minimum-cost s-t flow f^* of flow value k. Intuitively, for a constraint normal form $\langle C'', C' \rangle$, the values of k^L or/and k^N found by *MinMax-Sharability* and used in capacity functions guarantee that $\langle C' \rangle$ is satisfied, the values M and M' used in cost functions are used to satisfy $\langle C'' \rangle$ subject to $\langle C' \rangle$.

For Step 3, we introduce the following procedure *PATH-RECOVER*.

procedure *PATH-RECOVER*$(G'' = (V'', E''), s, t, f^*, k)$
begin
$P^* := \emptyset$;
for each $e \in E''$ **do**
if $f^*(e) \leq 0$ **then** remove e from G'';
end-for
for $i = 1$ to k **do**
find a shortest path P''_i from s to t in G'';
obtain an s-t path P_i in G from P''_i by replacing (v, v') and $\overline{(v, v')}$ by node v and replacing (u', v) and $\overline{(u', v)}$ by link (u, v);
$P^* := P^* \cup \{P_i\}$
for each e in G'' such that $e \in P''_i$ **do**
$f^*(e) := f^*(e) - 1$;
if $f^*(e) = 0$ **then** remove e from G'';
end-for
end-for
return P^*;
end

Given a flow f^* in G'' such that $|f^*| = k$ and $f^*(e)$ is an integer flow value for edge e in G'', procedure *PATH-RECOVER* constructs a unique set of k s-t paths corresponding to f^*. Based on flow conservation property, these k paths exist and can be enumerated iteratively by *PATH-RECOVER*.

Theorem 2: For any graph $G = (V, E)$ with non-negative edge cost, source s and destination t in V, if there exists an s-t path in G, then for each of the sharability constraints $\langle C_0 \rangle$ to $\langle C_{24} \rangle$ our algorithm scheme computes a set of k s-t paths $P^* = \{P_1^*, P_2^*, \cdots, P_k^*\}$ such that $l(P^*)$ is minimum subject to the sharability constraint.

Proof: See Appendix. ■

Both G' and G'' have $O(|V|)$ nodes and $O(|E|)$ links, where V and E are the set of nodes and links of G, respectively. Constructing G' and G'' takes $O(|V| + |E|)$ time. Computing maximum flow on G' can be done in $O(k \cdot |E|)$ time by applying Breadth First Search on residual graphs. Thus, procedure *MinMax-Sharability* for computing k^L or/and k^N has time complexity $O(k \cdot \log k \cdot |E|)$. Finding a minimum-cost flow on G'' takes $O(k \cdot (|E| + |V| \log |V|))$ time. Thus, the total time for computing minimum-cost k s-t paths under any of the 25 constraints listed in Table I is $O(k \cdot (|E| \log k + |V| \log |V|))$. Summerizing above discussions, we have the following result.

Theorem 3: Our algorithm scheme takes $O(k \cdot (|E| \log k + |V| \log |V|))$. time for computing minimum-cost k s-t paths in G subject to any sharability constraint of Table I.

In network applications, $k < |V|$, and the complexity of our algorithm scheme is actually $O(k|E| \log |V|)$.

IV. Generalizations

A. Nonuniform Maximum Allowable Sharability

In the problems we considered so far, we assumed that all links and nodes (except s and t) have the same maximum allowable sharability $k - 1$. For many applications, we may want to assign different maximum allowable sharabilities to individual links or/and nodes. For example, if we know that a link is highly reliable (resp. unreliable), we may want to assign maximum allowable sharability $k - 1$ (resp. 0) to the link. For optical networks, the numbers of available wavelengths on links may be different, and consequently we may assume that the maximum allowable sharability of a link to be the number of its available wavelengths less 1.

We generalize the minimum-cost k s-t path problem with sharability constraints by adding two more constraints: each link e (resp. node $v \neq s, t$), is allowed to be shared by at most $s(e) + 1$ (resp. $s(v) + 1$) paths, where $s(e)$ (resp. $s(v)$) is the maximum allowable link (resp. node) sharability of e (resp. v). Clearly, the problems (with uniform maximum allowable sharability constraints) considered in the previous sections are special cases of the problem (with nonuniform maximum allowable sharability constraints) we are discussing. The algorithm scheme for the constraints listed in Table I in Section III can be easily modified to solve the problem with extra nonuniform maximum allowable sharability constraints. For finding the minimum δ (resp. γ), binary search of procedure *MinMax-Sharability* can be slightly modified to find minimum k^L (or k^N) such that a flow f of value k exists without violating the capacity $s(e)+1$ (resp. $s(v)+1$) of each link (resp. node). Then, with respect to Table II, the $capacity$ of a $(cost, capacity)$ pair for a link in G'' for finding optimal

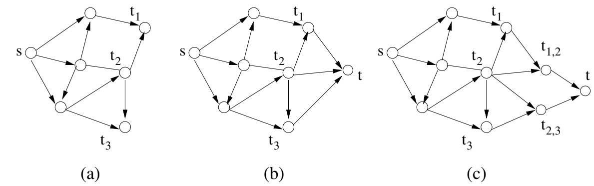

Fig. 4. Transformation used to solve related problems. (a) A given graph G. (b) Graph G^* for finding optimal paths from s to destinations in $T = \{t_1, t_2, t_3\}$. (b) Graph G^* for finding optimal paths from s to destination-pairs (t_1, t_2) and (t_2, t_3).

k paths under a specific composite constraint is modified as follows: value k is replaced by $s(e) + 1$ (resp. $s(v) + 1$), and value k^L (resp. k^N) is replaced by $\min\{s(e), k^L\}$ (resp. $\min\{s(v), k^N\}$). It is easy to see that minimum-cost k s-t paths that satisfy the given (composite) constraint of Table I and nonuniform maximum allowable individual link/node sharabilities can be computed by finding a minimum-cost flow f^* of flow value k from s to t in G''. Hence, all 25 versions of the problem of finding minimum-cost k s-t paths with nonuniform maximum allowable sharabilities can be solved in the same amount of time as their counterparts with uniform maximum allowable sharabilities. It is possible that a feasible solution does not exist. But such a situation can be detected easily.

B. Minimum Cost One-to-Many Paths with Minimum Sharability

Our algorithm scheme can be easily generalized to solve other problems. We mention two problems.

The first is to find minimum-cost paths $P = \{P_1, P_2, \cdots, P_k\}$ from s to a set of destinations $T = \{t_1, \cdots, t_k\}$ in a graph $G = (V, E)$ subject to minimum sharability constraint $\langle C \rangle$. Clearly, the paths in P are useful for multicast communications. Assuming that all nodes of T are reachable from s, we can reduce this problem to finding k s-t paths as follows: We construct a graph $G^* = (V^*, E^*)$ from G by introducing a new node t, and introducing a link from each node t_i to the new node t with cost 0 and maximum allowable link sharability 0 (see Figure 4(b) for an example). Then, we apply our algorithm scheme to find minimum-cost k s-t paths from s to t in G^* satisfying $\langle C \rangle$. By reversing the directions of links in G, this method can also be used to compute minimum-cost k paths with various sharability constraints for many-to-one communications.

The second related problem is defined as follows: given a graph $G = (V, E)$ with $|V| = n$, $|E| = m$, a source node s, a set T of k pairs (t_i, t_j) with $t_1, t_2 \in V - \{s\}$, and sharability constraint $\langle C \rangle$, find two paths from s to every pair (t_i, t_j) of nodes in T such that $\langle C \rangle$ is satisfied and the total cost of the paths is minimum. This is the problem of finding minimum-cost protection of dual homing architecture considered in [15], [17], [22]. This problem can be reduced to finding k s-t paths as follows: We construct a graph $G^* = (V^*, E^*)$ from G by introducing a new node $t_{i,j}$ for each pair (t_i, t_j), two links from nodes t_i and t_j to the new node $t_{i,j}$ with cost 0 and maximum allowable link sharability 0, a new node t, and a link from each $t_{i,j}$ to t with cost 0 and maximum allowable link sharability 2 (see Figure 4(c) for an example). Then, all we need to do is to find minimum-cost $2k$ paths from s to t satisfying $\langle C \rangle$ in G^*. Clearly, 25 versions of each of these two problems can be solved in polynomial time.

V. Concluding Remarks

In this paper, we characterized the degree of link sharing and node sharing by the notion of link sharability and node sharability. We identified 65 different sharability constraints for the problem of finding minimum-cost k s-t paths in a directed or undirected graph G, and showed that 25 of them are not mutually equivalent. We showed that these 25 versions of the problem finding minimum-cost k paths are polynomial-time solvable by reducing it to the minimum-cost network flow problem. We also showed that finding minimum-cost one-to-many paths subeject to various sharability constraints are polynomial-time solvable. Our algorithms can be used to find link-disjoint and node-disjoint paths if they exist by checking the min-max link sharability and min-max node sharability in the resulting solution. The algorithms presented in this paper are very useful for many network applications.

For all 65 versions of the problem of finding minimum-cost k paths with minimum sharability, we have shown that they are pair-wise inequivalent and developed a new algorithm scheme with slightly higher time complexity. We will report these new results in a subsequent paper.

Acknowledgement

This work was partially supported by NSF grants CCF-0514092 and CNS-0652287.

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows*, Prentice-Hall, 1993.

[2] B. Bank and M.E. Zarki, Dynamic multi-path routing and how it compares with other dynamic routing algorithms for high speed wide area networks, *Proc. of SIGCOMM*, 1992.

[3] D. Castanon, Efficient algorithms for finding the K best paths through a trellis, ÎEEE Trans. AES, vol.26, pp. 405-410, 1990.

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

[5] S. Fortune, J. Hopcroft, and J. Wyllie, The directed subgraph homeomorphism problem, *Theoret. Comput. Sci.*, vol 10, pp. 111-121, 1980.

[6] P. Georgatsos and D. Griffin, A management system for load balancing through adaptive routing in multiservice ATM networks, *Proc. of SIGCOMM*, 1996.

[7] K. Ishida, Y. Kakuda, and T. Kikuno, A routing protocol for finding two node-disjoint paths in computer networks, *Proc. of International Conference on Network Protocols*, pp. 340-347, 1992.

[8] R. Krishnan and J. Silvester, Choice of allocation granularity in multi-path source routing schemes, *Proc. of IEEE INFOCOM*, 1993.

[9] S.-W. Lee and C.-S. Wu, A k-best paths algorithm for highly reliable communication networks, *IEICE Trans. Commun.* E82-B.4, pp. 586-590, 1999.

[10] 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.* vol. 26, no. 1, pp. 105-115, 1990.

[11] C.L. Li, S.T. McCormick, and D. Simchi-Levi, Finding disjoint paths with different path-costs: complexity and algorithms, *Networks*, vol. 22, pp. 653-667, 1 22(1992), pp. 653-667.

[12] S.D. Nikolopoulos, A. Pitsillides, and D. Tipper, Addressing network survivability issues by finding the K-best paths through a trellis graph, *Proc. IEEE INFOCOM*, pp. 370-377,1997.

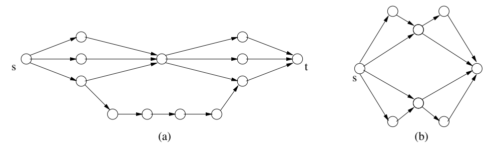

Fig. 5. (a) A graph for which pairs $\{C_i, C_j\}$ of different1 constraints in $\{\langle C_0\rangle, \langle C_1\rangle, \langle C_2\rangle, \langle C_3\rangle, \langle C_4\rangle\}$ such that $\{C_i, C_j\} \neq \{C_2, C_4\}$ are mutually inequivalent. (b) A graph for which constraints $\langle C_2\rangle$ and $\langle C_4\rangle$ are mutually inequivalent.

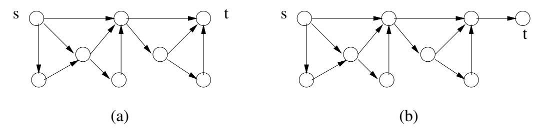

Fig. 6. (a) A graph for which constraints $\langle C_0\rangle, \langle C_5\rangle, \langle C_{10}\rangle, \langle C_{15}\rangle$ and $\langle C_{20}\rangle$ are mutually inequivalent. (b) The graph used for Case (3) in the proof of Theorem 1.

[13] J. W. Suurballe, Disjoint paths in a network, *Networks*, vol. 4, pp. 125-145, 1974.

[14] J. W. Suurballe and R. E. Tarjan, A quick method for finding shortest pairs of disjoint paths, *Networks*, vol. 14, pp. 325-336, 1984.

[15] J. Wang, M. Yang, X. Qi, and R. Cook, Dual-homing multicast protection, *Proc. IEEE Globecom*, pp. 1123-1127, 2004.

[16] H. Suzuki and F.A. Tobagi, Fast bandwidth reservation scheme with multi-link and mult-path routing in ATM networks, *Proc. IEEE INFOCOM*, 1992.

[17] J. Wang, M. Yang, B. Yang, and S.Q. Zheng, Dual-homing based scalable partial multicast protection, *IEEE Trans. on Computers*, vol. 55, no. 9, pp. 1130-1141, 2006.

[18] B. Yang, S.Q. Zheng, and S. Katukam, Finding two disjoint paths in a network with Min-Min objective function, *Proc. of the 15th IASTED Int'l Conf. on Parallel and Distr. Computing Systems*, pp. 75-80, 2003.

[19] B. Yang, S.Q. Zheng, and E. Lu, Finding two disjoint paths in a network with normalized α^+-Min-Sum objective function, *Proceedings of the 16th International Symposium on Algorithms and Computation* (*Lecture Notes in Computer Science, 3827*, Springer), pp. 954-063, 2005. .

[20] B. Yang, S.Q. Zheng, and E. Lu, Finding two disjoint paths in a network with normalized α^--Min-Sum objective function, *Proceedings of the 17th IASTED Int'l Conf. on Parallel and Distr. Computing Systems*, pp. 342-348, 2005.

[21] B. Yang, S.Q. Zheng, and E. Lu, Finding two disjoint paths in a network with MinSum-MinMin objective Function, Technical Report, UTDCS-16-05, Dept. of Computer Science, Univ. of Texas at Dallas, Apr. 2005.

[22] M. Yang, J. Wang, X. Qi, and Y. Jiang, On finding the best partial multicast protection tree under dual-homing architecture, *Proc. IEEE HPSR*, 2005.

Appendix

Proof of Theorem 1:

Given graph G, nodes s and t, constant k, and constraints $\langle X\rangle$ and $\langle Y\rangle$, let $S_{\langle X\rangle}$ and $S_{\langle Y\rangle}$ denote the feasible solution space, the set of sets of k s-t paths, for $\langle X\rangle$ and $\langle Y\rangle$, respectively. To show that $\langle X\rangle$ and $\langle Y\rangle$ are not equivalent, i.e. $\langle X\rangle \not\equiv \langle Y\rangle$, we only need to find a graph G and a k value for which $S_{\langle X\rangle} \neq S_{\langle Y\rangle}$. For a given graph G and a given constraint $\langle X\rangle$, we use $\delta|\langle X\rangle$ to denote the δ value of feasible s-t k-path solutions under $\langle X\rangle$. Since there can be multiple possible δ values for different feasible solutions under $\langle X\rangle$, we use $range(\delta|\langle X\rangle) = [\delta_1, \delta_2]$ to denote the closed integer interval of these δ values, where δ_1 and δ_2 are minimum and maximum possible value with respect to $\langle X\rangle$, respectively. A closed integer interval of a single value δ is denoted by

constraint $\langle C\rangle$	$range(\Gamma\|\langle C\rangle)$	$range(\Delta\|\langle C\rangle)$
$\langle\ \rangle$	$[k-1, 6k-6]$	$[2k-5, 7k-7]$
$\langle \min\Delta\rangle$	$[k, k+1]$	$[2k-5]$
$\langle \min\Gamma\rangle$	$[k-1]$	$[2k-4, 2k-2]$
$\langle \min\Gamma, \min\Delta\rangle$	$[k]$	$[2k-5]$
$\langle \min\Delta, \min\Gamma\rangle$	$[k-1]$	$[2k-4]$

TABLE IV

POSSIBLE Γ AND Δ VALUES FOR FEASIBLE k-PATH SOLUTIONS UNDER DIFFERENT MIN-SUM CONSTRAINTS FOR NETWORK OF FIGURE 6.

$[\delta]$. With respect to a given constraint $\langle X\rangle$, $range(\Delta|\langle X\rangle)$, $range(\gamma|\langle X\rangle)$, and $range(\Gamma|\langle X\rangle)$ are similarly defined for a graph G. Then, finding G and k for which $S_{\langle X\rangle} \neq S_{\langle Y\rangle}$ is equivalent to finding G and k for which

$$range(\lambda|\langle X\rangle) \neq range(\lambda|\langle Y\rangle)$$

or

$$range(\Lambda|\langle X\rangle) \neq range(\Lambda|\langle Y\rangle),$$

where $\lambda \in \{\delta, \gamma\}$ and $\Lambda \in \{\Delta, \Gamma\}$. We assume that all rational numbers $\frac{x}{y}$ used are integers; i.e. x is a multiple of y for $\frac{x}{y}$. We compare two constraints $\langle X\rangle$ and $\langle Y\rangle$ in Table I by comparing their normal froms $\langle X\rangle = \langle\langle X''\rangle, \langle X'\rangle\rangle$ and $\langle Y\rangle = \langle\langle Y''\rangle, \langle Y'\rangle\rangle$. For any two distinct constraints $\langle X\rangle$ and $\langle Y\rangle$, we have three cases.

Case (1): $\langle X''\rangle = \langle Y''\rangle = \langle\ \rangle$ and $\langle X'\rangle \neq \langle Y'\rangle$. Consider the graphs shown in Figure 5(a) and Figure 5(b). We list $\delta|\langle C\rangle$ and $\gamma|\langle C\rangle$ values of Figure 5(a) in the top part of Table III. Clearly, for some k either $range(\delta|\langle X'\rangle) \neq range(\delta|\langle Y'\rangle)$ or $range(\gamma|\langle X'\rangle) \neq range(\gamma|\langle Y'\rangle)$ if $\{\langle X'\rangle, \langle Y'\rangle\} \neq \{C_2, C_4\} = \{\langle\min\gamma\rangle\}, \langle\delta, \min\gamma\rangle\}$. For Figure 5(b), $range(\langle\min\gamma\rangle) \neq range(\langle\min\delta, \min\gamma\rangle)$, as indicated in the bottom part of Table III.

Case (2): $\langle X'\rangle = \langle Y'\rangle = \langle\ \rangle$ and $\langle X''\rangle \neq \langle Y''\rangle$. Consider the graph shown in Figure 6(a). We list $\Delta|\langle C\rangle$ and $\Gamma|\langle C\rangle$ values for this graph in Table IV. For any two distinct $\langle X\rangle = \langle X''\rangle$ and $\langle Y\rangle = \langle Y''\rangle$ of the five constraints, either $range(\Delta|\langle X''\rangle) \neq range(\Delta|\langle Y''\rangle)$ or $range(\Gamma|\langle X''\rangle) \neq range(\Gamma|\langle Y''\rangle)$ for some k .

Case (3): $\{\langle X'\rangle, \langle Y'\rangle\} \neq \{\langle\ \rangle\}$ and $\{\langle X''\rangle, \langle Y''\rangle\} \neq \{\langle\ \rangle\}$. If $\langle X''\rangle = \langle Y''\rangle$, then by Table III $range(\Delta|\ \langle X\rangle) \neq range(\Delta|\ \langle Y\rangle)$ or $range(\Gamma|\langle X\rangle) \neq range(\Gamma|\ \langle Y\rangle)$. If $\langle X''\rangle \neq \langle Y''\rangle$, then consider the graph shown in Figure 6(b). For this graph, $range(\delta|\ \langle X'\rangle) = range(\gamma|\ \langle X'\rangle) = range(\delta|\ \langle Y''\rangle) = range(\gamma|\ \langle Y''\rangle) = k-1$. By Case (2), we know that either $range(\Delta|\ \langle X''\rangle) \neq range(\Delta|\langle Y''\rangle)$ or $range(\Gamma|\ \langle X''\rangle) \neq range(\Gamma|\langle Y''\rangle)$ for some k. Proof is completed.

Proof of Theorem 2:

For constraint C_0, the theorem obviously holds. We prove the theorem by considering four remaining classes. For constraints in Class 1, our algorithm first finds min-max link sharability k^L or/and min-max node sharability k^N, which are used to restrict the feasible solution space. The MCNF algorithm finds an optimal solution within this restricted space.

For Class 2, the solutions have to satisfy min-max constraints, which are enforced by k^L or/and k^N. Suppose for the sake of contradiction the claim is not true, then there exists a different set of k paths from s to t, $P' = \{P'_1, P'_2, \cdots, P'_k\}$ in G, such that one of the following conditions holds:

(1) $\Delta(P') < \Delta(P^*)$;
(2) $\Delta(P') = \Delta(P^*)$, and $l(P') < l(P^*)$.

We create the unique network flow f' in G'' according to P' as follows:

```
for every e ∈ E' do f'(e) := 0 end-for
for i = 1 to k do
    for each link (u, v) in P'_i do
        case
            :f'(u', v) = 0: f'(overline(u', v)) := 1;
            :f'(u', v) = 1: f'(overline(u', v)) := f'(overline(u', v)) + 1;
        end-case
    end-for
end-for
```

Let f^* denote the flow corresponding to P^* in G''. Note that $\delta(e, P^*) - 1$ is exactly the value of f^* on $\overline{e} = \overline{(u, v)}$. Let $c(f^*)$ denote the cost of flow f^* in G''. We have

$$\begin{aligned} c(f^*) &= \sum_{e\in P^*} l(e) + \sum_{e\in P^*} (\delta(e, P^*) - 1)\cdot(M + l(e)) \\ &= \sum_{e\in P^*} \delta(e, P^*) l(e) + M \cdot \sum_{e\in P^*} (\delta(e, P^*) - 1) \\ &= \sum_{e\in P^*} \delta(e, P^*) l(e) + M \cdot \Delta(P^*) \end{aligned}$$

Similarly, we have $\Delta(P') = \sum_{e\in P'}(\delta(e, P')-1)$, and $c(f') = \sum_{e\in P'} \delta(e, P') l(e) + M \cdot \Delta(P')$. Then,

$$\begin{aligned} & c(f^*) - c(f') \\ = & (\sum_{e\in P^*} \delta(e, P^*) l(e) - \sum_{e\in P'} \delta(e, P') l(e)) \\ & + M \cdot (\Delta(P^*) - \Delta(P')). \end{aligned}$$

Since $0 \leq \sum_{e\in P^*} \delta(e, P^*) l(e) < M$ and $0 \leq \sum_{e\in P'} \delta(e, P') l(e) < M$, $\sum_{e\in P^*} \delta(e, P^*) l(e) - \sum_{e\in P'} \delta(e, P') l(e) > -M$. Now we consider the two possibilities.

Case (1): $\Delta(P') < \Delta(P^*)$. Then $\Delta(P^*) - \Delta(P') \geq 1$. Thus, $c(f^*) - c(f') = (\sum_{e\in P^*} \delta(e, P^*) l(e) - \sum_{e\in P'} \delta(e, P') l(e)) + M \cdot (\Delta(P^*) - \Delta(P')) > 0$, which contradicts the assumption that f^* is a minimum-cost flow.

Case (2): $\Delta(P') = \Delta(P^*)$ and $l(P') < l(P^*)$. Then, we have $c(f^*) - c(f') = l(P^*) - l(P') > 0$, which contradicts the assumption that f^* is a minimum-cost flow. This completes the proof for Class 2.

The proof for Class 3 is about the same as that for Class 2, except that $c(f^*) = \sum_{e\in P^*} l(e) + \sum_{v\in P^*, v\neq s,t} (\gamma(e, P^*) - 1) = \sum_{e\in P^*} l(e) + \Gamma(P^*)$ and $c(f') = \sum_{e\in P^*} l(e) + \sum_{v\in P', v\neq s,t} (\gamma(v, P') - 1) = \sum_{e\in P'} l(e) + \Gamma(P')$.

For Class 4, the solutions have to satisfy min-max constraints, which are enforced by k^L or/and k^N. Suppose for the sake of contradiction the claim is not true, then there exists a different set of k paths from s to t, $P = \{P'_1, P'_2, \cdots, P'_k\}$ such that one of the following conditions must hold:

constraint $\langle C\rangle$	$range(\gamma\|\langle C\rangle)$	$range(\delta\|\langle C\rangle)$	$range(\Gamma\|\langle C\rangle)$	$range(\Delta\|\langle C\rangle)$	graph
$\langle\ \rangle$	$[\frac{k}{2}-1, k-1]$	$[\frac{k}{3}-1, k-1]$	$[3k-7, 6k-6]$	$[4k-12, 7k-7]$	Fig. 5(a)
$\langle\min\delta\rangle$	$[\frac{2k}{3}-1, k-1]$	$[\frac{k}{3}-1]$	$[3k-7, 4k-11]$	$[4k-12, \frac{19k}{3}-19]$	Fig. 5(a)
$\langle\min\gamma\rangle$	$[\frac{k}{2}-1]$	$[\frac{k}{2}-1]$	$[\frac{9k}{2}-11]$	$[\frac{11k}{2}-15]$	Fig. 5(a)
$\langle\min\gamma,\min\delta\rangle$	$[\frac{2k}{3}-1]$	$[\frac{k}{3}-1]$	$[4k-11]$	$[5k-15]$	Fig. 5(a)
$\langle\min\delta,\min\gamma$	$[\frac{k}{2}-1]$	$[\frac{k}{2}-1]$	$[\frac{9k}{2}-11]$	$[\frac{11k}{2}-15]$	Fig. 5(a)
$\langle\min\gamma\rangle$	$[\frac{k}{2}-1]$	$[\frac{k}{4}-1, \frac{k}{2}-1]$	$[k-2, 3k-10]$	$[2k-8, 4k-16]$	Fig. 5(b)
$\langle\min\delta,\min\gamma\rangle$	$[\frac{k}{2}-1]$	$[\frac{k}{4}-1]$	$[2k-6]$	$[3k-12]$	Fig. 5(b)

TABLE III

POSSIBLE γ AND δ VALUES FOR FEASIBLE k-PATH SOLUTIONS UNDER DIFFERENT MIN-MAX CONSTRAINTS FOR NETWORK OF FIGURE 5.

(1) $\Delta(P') < \Delta(P^*)$;
(2) $\Delta(P') = \Delta(P^*)$, and $\Gamma(P') < \Gamma(P^*)$;
(3) $\Delta(P') = \Delta(P^*), \Gamma(P') = \Gamma(P^*)$, and $l(P') < l(P^*)$.

We create a network flow f' in G'' according to P' as follows:

```
for every e ∈ E'' do f'(e) := 0 end-for
for i = 1 to k do
    for each node v that is either s nor t in P'_i do
      case
        :f'(v, v') = 0: f'(v, v') := 1;
        :f'(v, v') = 1: f'(overline(v, v')) := f'(overline(v, v')) + 1;
      end-case
end-for
    for each link (u, v) in P'_i do
      case
        :f'(u', v) = 0: f'(u', v) := 1;
        :f'(u', v) = 1: f'(overline(u', v)) := f'(overline(u', v)) + 1;
      end-case
    end-for
end-for
```

We have

$$
\begin{aligned}
c(f^*) &= \sum_{v\in P^*, v\neq s,t} (\gamma(v,P^*)-1)\cdot M + \sum_{e\in P^*} l(e) \\
&\quad + \sum_{e\in P^*} (\delta(e,P^*)-1)\cdot(M'+l(e)) \\
&= \Gamma(P^*)\cdot M + \sum_{e\in P^*} \delta(e,P^*)l(e) + \\
&\quad \sum_{e\in P^*} (\delta(e,P^*)-1)\cdot M' \\
&= M\cdot\Gamma(P^*) + M'\cdot\Delta(P^*) + \sum_{e\in P^*} \delta(e,P^*)l(e).
\end{aligned}
$$

Similarly, we have

$$c(f') = M\cdot\Gamma(P') + M'\cdot\Delta(P') + \sum_{e\in P'} \delta(e,P')l(e).$$

Then,

$$
\begin{aligned}
&c(f^*) - c(f') \\
= \ & M\cdot(\Gamma(P^*)-\Gamma(P')) + M'\cdot(\Delta(P^*)-\Delta(P')) + \\
&(\sum_{e\in P^*} \delta(e,P^*)l(e) - \sum_{e\in P'} \delta(e,P')l(e)).
\end{aligned}
$$

For any set of k s-t paths with minimum total cost, there is no loop of positive cost in any path. Then, $0 \le \Gamma(P^*) \le k\cdot(|V|-2)$ and $0 \le \Gamma(P') \le k\cdot(|V|-2)$, and $\Gamma(P^*) - \Gamma(P') > -k\cdot(|V|-2)$ and $M\cdot(\Gamma(P^*)-\Gamma(P')) > -M' + 2kM$. Since $\sum_{e\in P^*} \delta(e,P^*)l(e) \le k\cdot(|V|-1)l_{max} < M$ and $\sum_{e\in P'} \delta(e,P')l(e) \le k\cdot(|V|-1)l_{max} < M$, we have $\sum_{e\in P^*} \delta(e,P^*)l(e) - \sum_{e\in P'} \delta(e,P')l(e) > -M$. Now we consider the three possibilities.

Case (1): $\Delta(P') < \Delta(P^*)$. Then $\Delta(P^*) - \Delta(P') \ge 1$, and

$$
\begin{aligned}
&c(f^*) - c(f') \\
= \ & M\cdot(\Gamma(P^*)-\Gamma(P')) + M'\cdot(\Delta(P^*)-\Delta(P')) + \\
&(\sum_{e\in P^*} \delta(e,P^*)l(e) - \sum_{e\in P'} \delta(e,P')l(e)) \\
\ge \ & M\cdot(\Gamma(P^*)-\Gamma(P')) + M' + (\sum_{e\in P^*} \delta(e,P^*)l(e) - \\
&\sum_{e\in P'} \delta(e,P')l(e)) \\
> \ & (-M' + 2kM) + M' - M \\
= \ & (2k-1)M > 0,
\end{aligned}
$$

which contradicts the assumption that f^* is a minimum-cost flow.

Case (2): $\Delta(P') = \Delta(P^*)$, and $\Gamma(P') < \Gamma(P^*)$. Then

$$
\begin{aligned}
&c(f^*) - c(f') \\
= \ & M\cdot(\Gamma(P^*)-\Gamma(P')) + (\sum_{e\in P^*} \delta(e,P^*)l(e) - \\
&\sum_{e\in P'} \delta(e,P')l(e)) > 0,
\end{aligned}
$$

which contradicts the assumption that f^* is a minimum-cost flow.

Case (3): $\Delta(P') = \Delta(P^*), \Gamma(P') = \Gamma(P^*)$, and $l(P') < l(P^*)$. Then,

$$c(f^*) - c(f') = \sum_{e\in P^*} \delta(e,P^*)l(e) - \sum_{e\in P'} \delta(e,P')l(e) > 0,$$

which contradicts the assumption that f^* is a minimum-cost flow. This completes the proof for Class 4.

The proof for constraints in Class 5 is about the same as that for Class 4, except that $c(f^*) = M'\cdot\Gamma(P^*) + M\cdot\Delta(P^*) + \sum_{e\in P^*} \delta(e,P^*)l(e)$ and $c(f') = M'\cdot\Gamma(P') + M\cdot\Delta(P') + \sum_{e\in P'} \delta(e,P')l(e)$ are used. Proof is completed.