Approximability of 3- and 4-hop bounded disjoint paths problems

Andreas Bley[1⋆] and Jose Neto[2]

[1] TU Berlin, Inst. of Mathematics
Straße des 17. Juni 136, D-10623 Berlin, Germany
[2] Institut TELECOM, TELECOM & Management SudParis,
CNRS UMR 5157, 9 rue Charles Fourier, F-91011 Evry, France

Abstract A path is said to be ℓ-bounded if it contains at most ℓ edges. We consider two types of ℓ-bounded disjoint paths problems. In the maximum edge- or node-disjoint path problems MEDP(ℓ) and MNDP(ℓ), the task is to find the maximum number of edge- or node-disjoint ℓ-bounded (s,t)-paths in a given graph G with source s and sink t, respectively. In the weighted edge- or node-disjoint path problems WEDP(ℓ) and WNDP(ℓ), we are also given an integer $k \in \mathbb{N}$ and non-negative edge weights $c_e \in \mathbb{N}$, $e \in E$, and seek for a minimum weight subgraph of G that contains k edge- or node-disjoint ℓ-bounded (s,t)-paths. Both problems are of great practical relevance in the planning of fault-tolerant communication networks, for example.
Even though length-bounded cut and flow problems have been studied intensively in the last decades, the $\mathcal{NP}$-hardness of some 3- and 4-bounded disjoint paths problems was still open. In this paper, we settle the complexity status of all open cases showing that WNDP(3) can be solved in polynomial time, that MEDP(4) is $\mathcal{APX}$-complete and approximable within a factor of 2, and that WNDP(4) and WEDP(4) are $\mathcal{APX}$-hard and $\mathcal{NPO}$-complete, respectively.
Keywords: Graph algorithms; length-bounded paths; complexity; approximation algorithms

1 Introduction

Two major concerns in the design of modern communication networks are the protection against potential failures and the permanent provision of a guaranteed minimum level of service quality. A wide variety of models expressing such requirements may be found in the literature, e.g. [1,14,15,16]. Coping simultaneously with both requirements naturally leads to length-restricted disjoint paths problems: In order to ensure that a pair of nodes remains connected also after some nodes or edges of the network fail, one typically demands the existence of several node- or edge-disjoint transmission paths between them. Each node on a transmission path, however, may lead to additional packet delay, jitter, and potential transmission errors for the corresponding data stream. To provide a guaranteed level of transmission service quality, these paths thus must not contain more than a certain number of intermediate nodes or, equivalently, of edges.

Mathematically, the task of verifying if a given network satisfies the robustness and quality requirements of a given node pair can be formulated as an edge- or node-disjoint paths problem. Let $G=(V,E)$ be a simple graph with source $s \in V$ and sink $t \in V$ and let $k \in \mathbb{N}$. A path in G is said to be ℓ-bounded for a given number $\ell \in \mathbb{N}$ if it contains at most ℓ edges. In the edge-disjoint paths problem $EDP(\ell)$, the task is to decide if there are k edge-disjoint ℓ-bounded (s,t)-paths in G or not. In the corresponding maximum edge-disjoint paths problem MEDP(ℓ), we wish to find the maximum number of edge-disjoint ℓ-bounded (s,t)-paths. The analogous node-disjoint path problems are denoted as $NDP(\ell)$ and $MNDP(\ell)$. The task of designing a network that satisfies the requirements of a single node pair can be modeled as a weighted edge- or node-disjoint path problems WEDP(ℓ) and WNDP(ℓ). In these problems, we are given the graph G, source s and sink t, the number of paths k, and non-negative edge weights $c_e \in \mathbb{N}$, $e \in E$. The task is to find a minimum cost subset $E' \subseteq E$ such that the subgraph (V,E') contains at least k edge- or node-disjoint ℓ-bounded (s,t)-paths, respectively.

⋆ Supported by the DFG Research Center MATHEON – Mathematics for key technologies.

Table 1. Known and new (bold) complexity results for node- and edge-disjoint ℓ-bounded paths problems.

ℓ	MNDP(ℓ)	WNDP(ℓ)	MEDP(ℓ)	WEDP(ℓ)
≤ 2	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
$= 3$	$\mathcal{P}$	**P**	$\mathcal{P}$	$\mathcal{P}$
$= 4$	$\mathcal{P}$	**APX-hard** (at least)	**APX-complete**	**NPO-complete**
≥ 5	$\mathcal{APX}$-complete	$\mathcal{NPO}$-complete	$\mathcal{APX}$-complete	$\mathcal{NPO}$-complete

Due to their great practical relevance, problems asking for disjoint paths or unsplittable flows between some node pairs have received considerable attention in the literature. Structural results, complexity issues, and approximation algorithms for disjoint paths problems without length restrictions are discussed in [8,9,20], for example.

In a seminal article Menger [24] shows that the maximum number of edge- or node-disjoint (s,t)-paths in a graph is equal to the minimum size of an (s,t)-edge- or (s,t)-node-cut, respectively. Lovász *et al.* [21], Exoo [12], and Niepel *et al.* [25] showed that this strong duality between disjoint paths and (suitably defined) cuts still holds for 4-bounded node-disjoint paths and node-cuts and for 3-bounded edge-disjoint paths and edge-cuts, but that Menger's theorem does not hold for length bounds $\ell \geq 5$. The ratio between the number of paths and the cut size is studied in [5,27]. Generalizations of Menger's theorem and of Ford and Fulkerson's max flow min cut theorem to length-bounded flows are an area of active research [3,23].

Polynomial time algorithms for the minimum ℓ-bounded edge-cut problem with $\ell \leq 3$ have been presented by Mahjoub and McCormick [22]. Baier *et al.* [4] proved that the minimum ℓ-bounded edge-cut problem is $\mathcal{APX}$-hard for $\ell \geq 4$ and that the corresponding node-cut problem is $\mathcal{APX}$-hard for $\ell \geq 5$.

Itai *et al.* [19] and Bley [6] showed that the problems MEDP(ℓ) and MNDP(ℓ) of finding the maximum number of edge- and node-disjoint ℓ-bounded paths are polynomially solvable for $\ell \leq 3$ and $\ell \leq 4$, respectively, and that both problems are $\mathcal{APX}$-complete for $\ell \geq 5$. Heuristics to find large sets of disjoint length bounded paths can be found, e.g., in [19,26,29]. Polyhedral approaches to these problems are investigated in [7,10,18]. The weighted disjoint paths problems WEDP(ℓ) and WNDP(ℓ) are known to be $\mathcal{NPO}$-complete for $\ell \geq 5$ and to be polynomially solvable for $\ell \leq 2$ in the node-disjoint case and for $\ell \leq 3$ in the edge-disjoint case [6]. Further results and a finer analysis of the complexity of disjoint paths problems by means of different parameterizations (namely w.r.t. the number of paths, their length, or the graph treewidth) are presented in [13,17]. The complexity of MEDP(4), WEDP(4), WNDP(3), and WNDP(4), however, has been left open until now.

The contribution of this paper is to close all these open cases. In Section 2, we prove that the maximum edge-disjoint 4-bounded paths problem MEDP(4) is $\mathcal{APX}$-complete, presenting a 2-approximation algorithm and an approximation preserving reduction from MAX-k-SAT(3) to MEDP(4). This implies that the corresponding weighted edge-disjoint paths problem WEDP(4) is $\mathcal{NPO}$-complete. In Section 3, we then show how to solve the weighted node-disjoint 3-bounded paths problem WNDP(3) via matching techniques in polynomial time and prove that the 4-bounded version of this problem is at least $\mathcal{APX}$-hard. Table 1 summarizes the known and new complexity results regarding these problems. All hardness results and algorithms presented in this paper generalize in a straightforward way to directed graphs and to non-simple graphs containing parallel edges.

2 Edge-Disjoint 4-Bounded Paths

In this section, we study the approximability of the two edge-disjoint 4-bounded problems. First, we consider the problem of maximizing the number of edge-disjoint paths. One easily observes that any inclusion-wise maximal set of edge-disjoint 4-bounded (s,t)-paths, which can be computed in polynomial time by greedily adding disjoint paths to the solution, is a 4-approximate solution for MEDP(4) [6].

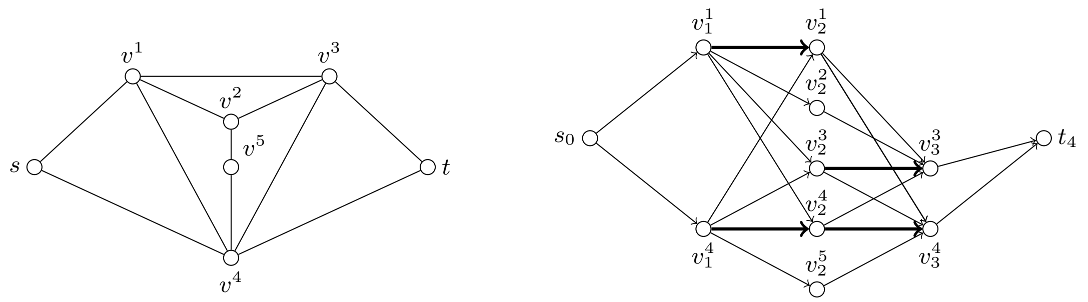

Figure 1. Construction of the hop-extended digraph G' (right) from the given graph G (left) in Step 1 of algorithm ExFlow. Arcs with cost 0 in G' are thick.

A 2-approximation algorithm is obtained as shown in algorithm ExFlow.

ExFlow
1. Compute a minimum cost maximum (s_0, t_4)-flow f in the hop-extended digraph G'. Let $\mathcal{F} := \{P_1, \ldots, P_k\}$ be the corresponding 4-bounded simple paths in G.
2. Create the conflict graph $H := (\mathcal{F}, \{P_iP_j \mid P_i \cap P_j \neq \emptyset\})$.
3. Compute an independent set $\mathcal{S} \subseteq \mathcal{F}$ in H with $\lvert\mathcal{S}\rvert \geq \frac{1}{2}\lvert\mathcal{F}\rvert$.
4. Return $\mathcal{S}$.

In the first step of algorithm ExFlow, we construct the directed graph $G' = (V', E')$ with $V' = \bigcup_{i=0}^{4} V_i$ for $V_0 := \{s_0\}$, $V_4 := \{t_4\}$, and $V_i := \{v_i \mid v \in V \setminus \{s, t\}$ with $dist_G(v, s) \leq i$ and $dist_G(v, t) \leq 4 - i\}$ for all $i \in \{1, 2, 3\}$, and $E' := \bigcup_{i=0}^{4} E_i$ with $E_0 := \{(s_0, t_4)\}$ if $st \in E$, $E_0 := \emptyset$ if $st \notin E$, and $E_i := \{(v_{i-1}, w_i) \in V_{i-1} \times V_i \mid vw \in E$ or $v = w\}$ for $i \in \{1, \ldots, 4\}$, where $dist_G(u, v)$ denotes the distance from node u to node v in G. We assign cost 0 and capacity 1 to all edges $u_iu_{i+1} \in E'$ and capacity 1 and cost 1 to all other edges in E'. Figure 1 illustrates this construction.

In this layered digraph, we compute an (integer) minimum cost maximum (s_0, t_4)-flow and its decomposition into paths $P'_1, \ldots, P'_k$. Each path $P'_i = (s_0, u_1, v_2, w_3, t_4)$ defines a 4-bounded walk (s, u, v, w, t) in G, which can be shortened to a simple 4-bounded path P_i. Let $\mathcal{F} = \{P_1, \ldots, P_k\}$ be the set of these paths. Note that these paths are 4-bounded, but not necessarily edge-disjoint.

In the second step, we create the associated "conflict graph" $H = (\mathcal{F}, \{P_iP_j \mid P_i \cap P_j \neq \emptyset\})$. By Lemma 1, H consists only of disjoint paths and isolated nodes. Choosing all isolated nodes and a maximum independent set in each of these paths, we thus can compute an independent set $\mathcal{S} \subseteq \mathcal{F}$ of size $\lvert\mathcal{S}\rvert \geq \lvert\mathcal{F}\rvert/2$ in H. This is done in the third step of our algorithm.

Finally, we return the path set corresponding to this independent set.

Steps 1, 2, and 4 of this algorithm clearly can be done in polynomial time. The possibility to perform also Step 3 in polynomial time follows from the following lemma.

Lemma 1. *The conflict graph $H = (\mathcal{F}, \{P_iP_j \mid P_i \cap P_j \neq \emptyset\})$ created in Step 3 of algorithm* ExFlow *consists of isolated nodes and disjoint paths only.*

Proof. Let f be the minimum cost maximum (s_0, t_4)-flow in G' computed in Step 1 of ExFlow and let $P'_1, \ldots, P'_k$ be its path decomposition. Note that the paths P'_i are edge-disjoint in G'.

By construction of G', each edge $e \in \delta(s) \cup \delta(t)$ corresponds to at most one arc (s_0, v_1), (v_3, t_4), or (s_0, t_4) in G'. Thus, any such edge is contained in at most one path in $\mathcal{F}$. Furthermore, for each edge $e = uv \in E \setminus \delta(s) \setminus \delta(t)$, the paths $P'_1, \ldots, P'_k$ in G' contain at most one of the arcs (u_1, v_2) and (v_1, u_2) and at most one of the arcs (u_2, v_3) and (v_2, u_3). Otherwise, these paths do not correspond to a *minimum cost* maximum flow: If there were two paths $P'_1 = (s_0, u_1, v_2, w_3, t_4)$ and $P'_2 = (s_0, v_1, u_2, q_3, t_4)$, for example, then replacing these paths by the paths $P''_1 = (s_0, u_1, u_2, q_3, t_4)$ and $P''_2 = (s_0, v_1, v_2, w_3, t_4)$

would reduce the cost of the corresponding flow. Consequently, any edge $e \in E \setminus \delta(s) \setminus \delta(t)$ can be contained in at most two of the paths in $\mathcal{F}$ and, further on, a path in $\mathcal{F}$ can intersect with at most two other paths in $\mathcal{F}$. This implies that the conflict graph H constructed in Step 2 of ExFlow consists only of isolated nodes and disjoint paths and cycles.

To see that H cannot contain cycles, let $C = \{P_1, \ldots, P_n\}$ be the shortest cycle in H. Then each edge in $M := \bigcup_{i \in C} P_i \setminus \delta(s) \setminus \delta(t)$ must appear in exactly two paths in C, once as the second and once as the third edge. If there were two paths P_1 and P_2 in C that traverse one of the edges $e = uv \in M$ in opposite directions, then the corresponding paths in G' would be of the form $P_1' = (s_0, u_1, v_2, w_3, t_4)$ and $P_2' = (s_0, q_1, v_2, u_3, t_4)$. In this case, replacing P_1' and P_2' by $P_1'' = (s_0, u_1, u_2, u_3, t_4)$ and $P_2'' = (s_0, q_1, v_2, w_3, t_4)$ would reduce the cost of the corresponding flow in G' (and the size of the remaining cycle in C), which is a contradiction to our assumption that the paths P_i' correspond to a *minimum cost* maximum flow in G'.

So, we may assume that the paths in C traverse each edge $e \in M$ in the same direction. Then, for each $e = uv \in M$, there is exactly one path of the form (s, u, v, w, t) and exactly one path of the form (s, q, u, v, t) in C. In this case, however, we can replace each path $P_i' = (s_0, u_1, v_2, w_3, t_4)$ that corresponds to a path in C by the less costly path $P_i'' = (s_0, u_1, v_2, v_3, t_4)$ without exceeding the edge capacities in G'. This is again a contradiction to our assumption that the paths P_i' define a minimum cost maximal flow in G'. Consequently, there are no cycles in H. □

Theorem 2. ExFlow *is a* 2*-approximation algorithm for MEDP(4).*

Proof. By Lemma 1, all steps of the algorithm can be executed in polynomial time. The paths in $\mathcal{S}$ are derived from the 4-bounded (s_0, t_4)-flow paths in G', so they are clearly 4-bounded. As $\mathcal{S}$ is an independent set in the conflict graph H, the paths in $\mathcal{S}$ are also edge-disjoint.

Furthermore, any set of edge-disjoint 4-bounded (s, t)-paths in G defines a feasible (s_0, t_4)-flow in G'. Hence, $k = |\mathcal{F}|$ is an upper bound on the maximum number k^* of edge-disjoint 4-bounded (s, t)-paths in G, which immediately implies $|\mathcal{S}| \geq \frac{1}{2} k^*$. □

In order to show that MEDP(4) is $\mathcal{APX}$-hard, i.e., that there is some $c > 1$ such that approximating MEDP(4) within a factor less than c is $\mathcal{NP}$-hard, we construct an approximation preserving reduction from the Max-k-Sat(3) problem to MEDP(4). Given a set X of boolean variables and a collection C of disjunctive clauses such that each clause contains at most k literals and each variable occurs at most 3 times as a literal, the Max-k-Sat(3) problem is to find a truth assignment to the variables that maximizes the number of satisfied clauses. Max-k-Sat(3) is known to be $\mathcal{APX}$-complete [2].

Theorem 3. *MEDP(*4*) is* $\mathcal{APX}$*-hard.*

Proof. We construct an approximation preserving reduction from Max-k-Sat(3) to MEDP(4). Let x_i, $i \in I$, be the boolean variables and C_l, $l \in L$ be the clauses of the given Max-k-Sat(3) instance. Without loss of generality we may assume that each variable x_i occurs exactly 3 times as a literal and denote these occurrences by x_i^j, $j \in J := \{1, \ldots, 3\}$.

We construct an undirected graph $G = (V, E)$ that consists of $|I| + |L|$ subgraphs, one for each variable and one for each clause, as follows. For each $i \in I$, we construct a variable graph $G_i = (V_i, E_i)$ as shown in Figure 2. G_i contains the nodes and edges

$$\begin{aligned} V_i :=& \{s,\, t\} \cup \{u_i^j,\, v_i^j,\, w_i^j,\, \bar{w}_i^j,\, a_i^j,\, \bar{a}_i^j \mid j \in J\} \quad \text{and} \\ E_i :=& \{su_i^j,\, u_i^j v_i^j,\, v_i^j w_i^j,\, v_i^j \bar{w}_i^j,\, w_i^j t,\, \bar{w}_i^j t,\, sa_i^j,\, s\bar{a}_i^j,\, \bar{w}_i^j w_i^{j+1} \mid j \in J\} \\ & \cup \{a_i^j w_i^{j+1} \mid j \in J : x_i^j \text{ occurs as unnegated literal } x_i^j\} \\ & \cup \{\bar{a}_i^j \bar{w}_i^j \mid j \in J : x_i^j \text{ occurs as negated literal } \bar{x}_i^j\}\,, \end{aligned}$$

where $w_i^4 = w_i^1$ for notational simplicity. The nodes s and t are contained in all subgraphs and serve as source and destination for all paths. For each $l \in L$, we construct a clause graph $H_l = (W_l, F_l)$ as shown

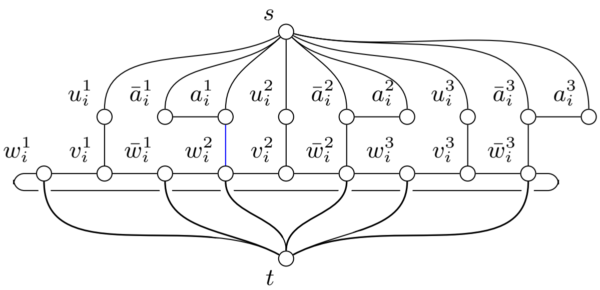

Figure 2. Graph G_i for variable x_i occurring as literals x_i^1, $\bar{x}_i^2$, $\bar{x}_i^3$.

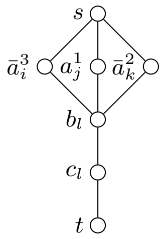

Figure 3. Graph H_l for clause $C_l = (\bar{x}_i^3 \vee x_j^1 \vee \bar{x}_k^2)$.

in Figure 3. In addition to the nodes and edges it shares with the variable graphs, H_l contains 2 nodes and $k' + 2$ edges, where k' is the number of literals in clause C_l. Formally, W_l and F_l are defined as

$$\begin{aligned} W_l :=& \{s,\, t,\, b_l,\, c_l\} \cup \{\bar{a}_i^j \mid i \in I,\, j \in J : \text{negated literal } \bar{x}_i^j \text{ occurs in } C_l\} \\ & \cup \{a_i^j \mid i \in I,\, j \in J : \text{unnegated literal } x_i^j \text{ occurs in } C_l\} \quad \text{and} \\ F_l :=& \{b_l c_l, c_l t\} \cup \{s\bar{a}_i^j,\, \bar{a}_i^j b_l \mid i \in I,\, j \in J : \text{negated literal } \bar{x}_i^j \text{ occurs in } C_l\} \\ & \cup \{s a_i^j,\, a_i^j b_l \mid i \in I,\, j \in J : \text{unnegated literal } x_i^j \text{ occurs in } C_l\}\ . \end{aligned}$$

The goal in the constructed MEDP(4) instance is to find the maximum number of edge-disjoint 4-bounded (s,t)-paths in the simple undirected graph G obtained as the union of all variable and clause (sub)-graphs. It is clear that the constructions can be performed in polynomial time.

For notational convenience, we denote for each $i \in I$ and $j \in J$ the paths

$$P_{ij} = (s, u_i^j, v_i^j, \bar{w}_i^j, t)\ , \qquad P'_{ij} = \begin{cases} (s, \bar{a}_i^j, \bar{w}_i^j, w_i^{j+1}, t) & \text{if } \bar{x}_i^j \text{ occurs} \\ (s, \bar{a}_i^j, a_i^j, w_i^{j+1}, t) & \text{if } x_i^j \text{ occurs} \end{cases}\ ,$$

$$\bar{P}_{ij} = (s, u_i^j, v_i^j, w_i^j, t)\ , \qquad \bar{P}'_{ij} = \begin{cases} (s, a_i^j, \bar{a}_i^j, \bar{w}_i^j, t) & \text{if } \bar{x}_i^j \text{ occurs} \\ (s, a_i^j, w_i^{j+1}, \bar{w}_i^j, t) & \text{if } x_i^j \text{ occurs} \end{cases}\ .$$

For each $i \in I$ and $l \in L$ such that variable x_i occurs in clause C_l, we denote

$$Q_{li} = \begin{cases} (s, a_i^j, b_l, c_l, t) & \text{if literal } x_i^j \text{ occurs in } C_l \\ (s, \bar{a}_i^j, b_l, c_l, t) & \text{if literal } \bar{x}_i^j \text{ occurs in } C_l. \end{cases}$$

Furthermore, we define $\mathcal{P}_i := \{P_{ij},\, P'_{ij} \mid j \in J\}$ and $\bar{\mathcal{P}}_i := \{\bar{P}_{ij},\, \bar{P}'_{ij} \mid j \in J\}$ for all $i \in I$, and $\mathcal{Q}_l := \{Q_{li} \mid i \in I : x_i \text{ occurs in } C_l\}$ for all $l \in L$. Figure 4 illustrates the paths in $\bar{\mathcal{P}}_i$ and path Q_{li}.

In the first part of the proof we show that any truth assignment $\hat{x}$ that satisfies r clauses of the given MAX-k-SAT(3) instance can be transformed into a set $\mathcal{S}(\hat{x})$ of $6|I| + r$ edge-disjoint 4-bounded (s,t)-paths in G. Let $\hat{x}$ be a truth assignment. For each clause C_l that is satisfied by this truth assignment, let $i_l(\hat{x})$ be one of the variables whose literal evaluates to $true$ in C_l. We define

$$\mathcal{S} = \mathcal{S}(\hat{x}) := \bigcup_{i \in I : \hat{x}_i = true} \mathcal{P}_i \cup \bigcup_{i \in I : \hat{x}_i = false} \bar{\mathcal{P}}_i \cup \{Q_{l i_l(\hat{x})} \mid l \in L : C_l(\hat{x}) = true\}\ .$$

Clearly, all paths in $\mathcal{S}$ contain at most 4 edges, $|\mathcal{S}| = 6|I| + r$, and all paths in $\mathcal{S} \cap \bigcup_i (\mathcal{P}_i \cup \bar{\mathcal{P}}_i)$ are edge-disjoint. Note that if some path Q_{li} is contained in $\mathcal{S}$, then either the negated literal $\bar{x}_i^j$ occurring in clause C_l evaluates to $true$, which implies that $x_i = false$ and $P'_{ij} \notin \mathcal{S}$, or the unnegated literal x_i^j occurring in C_l evaluates to $true$ and, hence, $\bar{P}'_{ij} \notin \mathcal{S}$. Furthermore, observe that these paths P'_{ij} and $\bar{P}'_{ij}$ are the only paths that may be contained in $\mathcal{S}$ and share an edge with Q_{li_l}. Consequently, each path $Q_{li} \in \mathcal{S}$ is edge-disjoint to any other path in $\mathcal{S}$ and, thus, all paths in $\mathcal{S}$ are edge-disjoint.

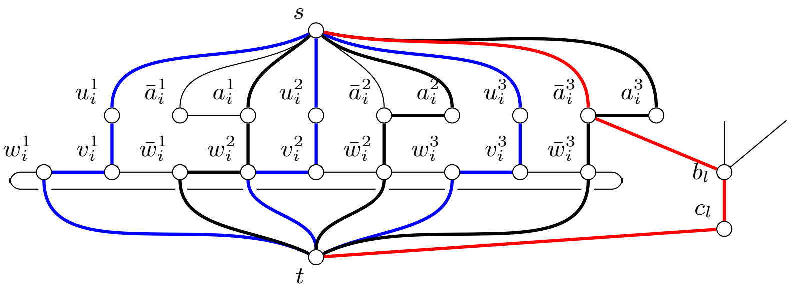

Figure 4. Union of G_i and H_l for variable x_i and clause $C_l = (\bar{x}_i^3 \vee \dots)$. Thick lines are paths in $\bar{\mathcal{P}}_i$ and path Q_{li}.

In the second part of the proof we show that any set $\mathcal{S}$ of $6|I|+r$ edge-disjoint 4-bounded (s,t)-paths in G can be transformed into a truth assignment $\hat{x}(\mathcal{S})$ that satisfies a least r clauses of the given MAX-k-SAT(3) instance. We may ignore path sets with $|\mathcal{S}| < 6|I|$, as the path set $\bigcup_{i\in I}\mathcal{P}_i$ is a feasible solution for the constructed MEDP(4) instance with $6|I|$ paths. Furthermore, we may restrict our attention to path sets $\mathcal{S}$ that satisfy the property that, for each $i \in I$, either $\mathcal{P}_i \subseteq \mathcal{S}$ or $\bar{\mathcal{P}}_i \subseteq \mathcal{S}$. Any path set $\mathcal{S}$ that does not satisfy this property can be turned into a path set $\mathcal{S}'$ with $|\mathcal{S}'| \geq \mathcal{S}$ that does as follows:

Suppose that, for some i, neither $\mathcal{P}_i \subseteq \mathcal{S}$ nor $\bar{\mathcal{P}}_i \subseteq \mathcal{S}$. Let $\mathcal{S}_i \subseteq \mathcal{S}$ be the set of paths in $\mathcal{S}$ that are fully contained in the variable subgraph G_i. As there are only 6 edges adjacent to t in G_i, we have $|\mathcal{S}_i| \leq 6$. Observe that each 4-bounded (s,t)-path in G is either of the form Q_{li} or it is fully contained in one of the variable subgraphs G_i. Furthermore, all (s,t)-paths of length exactly 4 in G_i are contained in $\mathcal{P}_i \cup \bar{\mathcal{P}}_i$. The only other 4-bounded paths in G_i are the three paths of length 3, which we denote $\bar{P}''_{ij} = (s, \bar{a}_i^j, \bar{w}_i^j, t)$ for the negated literals $\bar{x}_i^j$ and $P''_{ij} = (s, a_i^j, w_i^{j+1}, t)$ for the unnegated literals $\bar{x}_i^j$. In terms of edge-disjointness, however, the paths P''_{ij} and $\bar{P}''_{ij}$ conflict with the same 4-bounded (s,t)-paths as the paths P'_{ij} or $\bar{P}'_{ij}$, respectively. Replacing all paths P''_{ij} and $\bar{P}''_{ij}$ in $\mathcal{S}$ by the paths P'_{ij} and $\bar{P}'_{ij}$, respectively, thus yields a set of edge-disjoint 4-bounded path of the same size as $\mathcal{S}$. Hence, we can assume that $\mathcal{S}_i \subseteq \mathcal{P}_i \cup \bar{\mathcal{P}}_i$.

Now consider the paths Q_{il} corresponding to the clauses C_l in which variable x_i occurs. Recall that variable x_i occurs exactly 3 times in the clauses, so there are at most 3 paths Q_{il} in $\mathcal{S}$ that may share an edge with the paths in $\mathcal{P}_i \cup \bar{\mathcal{P}}_i$. If variable x_i occurs uniformly in all 3 clauses negated or unnegated, then these three paths Q_{il} are edge-disjoint from either all 6 paths in $\mathcal{P}_i$ or from all 6 paths in $\bar{\mathcal{P}}_i$. Replacing the paths in $\mathcal{S}_i$ by $\mathcal{P}_i$ or $\bar{\mathcal{P}}_i$, respectively, yields an edge-disjoint path set $\mathcal{S}'$ with $|\mathcal{S}'| \geq |\mathcal{S}|$. If variable x_i occurs non-uniformly, then either the paths in $\mathcal{P}_i$ or the paths in $\bar{\mathcal{P}}_i$ conflict with at most one of the three Q_{il} paths. In this case we have $\mathcal{S}_i \leq 5$, as the only edge-disjoint path sets of size 6 in $\mathcal{P}_i \cup \bar{\mathcal{P}}_i$ are $\mathcal{P}_i$ and $\bar{\mathcal{P}}_i$ themselves. Replacing the at most 5 paths in $\mathcal{S}_i$ and the 1 potentially conflicting path Q_{il} (if it is contained in $\mathcal{S}$ at all) by either $\mathcal{P}_i$ or $\bar{\mathcal{P}}_i$ thus yields a path set $\mathcal{S}'$ with $|\mathcal{S}'| \geq |\mathcal{S}|$ and either $\mathcal{P}_i \subseteq \mathcal{S}'$ or $\bar{\mathcal{P}}_i \subseteq \mathcal{S}'$. Repeating this procedure for all $i \in I$, we obtain a path set with the desired property.

So, suppose we are given a set $\mathcal{S}$ of 4-bounded edge-disjoint (s,t)-paths in G with $|\mathcal{S}| = 6|I| + r$ and $\mathcal{P}_i \subseteq \mathcal{S}$ or $\bar{\mathcal{P}}_i \subseteq \mathcal{S}$ for each $i \in I$. Then we define the truth assignment $\hat{x}(\mathcal{S})$ as

$$\hat{x}_i(\mathcal{S}) := \begin{cases} true & \text{if } \mathcal{P}_i \subset \mathcal{S}, \\ false & \text{otherwise} \end{cases} \quad \text{for all } i \in I.$$

To see that $\hat{x}(\mathcal{S})$ satisfies at least r clauses, consider the (s,t)-cut in G formed by the edges adjacent to node t. As $\mathcal{S}$ contains either $\mathcal{P}_i$ or $\bar{\mathcal{P}}_i$ for each $i \in I$, which amounts to a total of $6|I|$ paths, each of the remaining r paths in $\mathcal{S}$ must be of the form Q_{il} for some $i \in I$ and $l \in L$. Path Q_{il}, however, can be contained in $\mathcal{S}$ only if clause C_l evaluates to $true$. Otherwise it would intersect with the path P'_{ij} or $\bar{P}'_{ij}$ in $\mathcal{S}$ that corresponds to literal x_i^j occurring in clause C_l. Hence, at least r clauses of the given MAX-k-SAT(3) instance are satisfied by the truth assignment $\hat{x}(\mathcal{S})$.

It now follows in a straightforward way that MEDP(4) is $\mathcal{APX}$-hard. Suppose there is an algorithm ALG to approximate MEDP(4) within a factor of $\alpha > 1$ and denote by $\mathcal{S}$ the solution computed by this algorithm. Let $r(\mathcal{S})$ be the number of clauses satisfied by the truth assignment $\hat{x}(\mathcal{S})$ and let $|\mathcal{S}^*|$ and r^* be optimal solution values of MEDP(4) and MAX-k-SAT(3), respectively. The fact that at least half of the clauses in any MAX-k-SAT(3) instance can be satisfied implies $r^* \geq \frac{1}{2}|L|$ and, further on, $r^* \geq \frac{3}{2k}|I|$. Applying the problem transformation and algorithm ALG to a given MAX-k-SAT(3) instance, we get

$$r(\mathcal{S}) \geq |\mathcal{S}| - 6|I| \geq \frac{1}{\alpha}|\mathcal{S}^*| - 6|I| \geq \frac{1}{\alpha}(r^* + 6|I|) - 6|I| \geq \frac{1 + 4k - 4k\alpha}{\alpha} r^*$$

As there is a threshold $c > 1$ such that approximating MAX-k-SAT(3) within a factor smaller than c is $\mathcal{NP}$-hard, it is also $\mathcal{NP}$-hard to approximate MEDP(4) within a factor less than $c' = \frac{4kc+c}{4kc+1} > 1$. □

Theorem 3 immediately implies the following corollary.

Corollary 4. *Given a graph $G = (V, E)$, $s, t \in V$, and $k \in \mathbb{Z}_+$, it is $\mathcal{NP}$-hard to decide if there are k edge-disjoint 4-bounded (s, t)-paths in G.*

Now consider the weighted problem WEDP(4). By Corollary 4, it is already $\mathcal{NP}$-hard to decide whether a given subgraph of the given graph contains k edge-disjoint (s, t)-path and, thus, comprises a feasible solution or not. Consequently, finding a minimum cost such subgraph is $\mathcal{NPO}$-complete.

Theorem 5. *WEDP(4) is $\mathcal{NPO}$-complete.*

As a consequence of Theorem 5, it is $\mathcal{NP}$-hard to approximate WEDP(4) within a factor $2^{f(n)}$ for any polynomial function f in the input size n of the problem.

3 Node-Disjoint 3- and 4-Bounded Paths

In this section we study the complexity of the node-disjoint paths problems. The maximum disjoint paths problem MNDP(ℓ) is known to be polynomially solvable for $\ell \leq 4$ and to be $\mathcal{APX}$-hard for $\ell \geq 5$ [6,19]. The weighted problem WNDP(ℓ) is solvable in polynomial time for $\ell \leq 2$, and $\mathcal{NPO}$-complete for $\ell \geq 5$. In the special case where $c_e \leq \sum_{f \in C - e} c_f$ holds for every cycle C in G and every edge $e \in C$, the weighted problem can be solved polynomially also for $\ell = 3$ and $\ell = 4$ [6]. For $\ell = 3$, the problem can still be solved efficiently if this condition is not satisfied.

Theorem 6. *WNDP(3) can be solved in polynomial time.*

Proof. Let S and T denote the set of neighbors of node s and t in the given graph G, respectively. We may assume w.l.o.g. that each node in G is contained in $\{s, t\} \cup S \cup T$, for otherwise it may not appear in any 3-bounded (s, t)-path.

We reduce WNDP(3) to the problem of finding a minimum weight matching with cardinality k in an auxiliary graph $G' = (V', E')$, which is constructed as follows: For each node $v \in S$ (resp. $w \in T$), there is an associated node $u_v \in V'$, (resp. $u'_w \in V'$). For each node $v \in S \cap T$, there is an associated edge $e_v = (u_v, u'_v) \in E'$ with weight $c_{sv} + c_{vt}$. Choosing this edge in the matching corresponds to choosing the path (s, v, t) in G. For each edge $(v, w) \in (S \times T) \setminus (S \cap T)^2$, there is an associated edge $(\overline{u}_v, \overline{u}_w) \in E'$, with $\overline{u}_z = u'_z$ if $z \in T$ and $\overline{u}_z = u_z$ otherwise for any $z \in V$. The weight of edge $(\overline{u}_v, \overline{u}_w)$ is set to $c_{sv} + c_{vw} + c_{wt}$. Choosing $(\overline{u}_v, \overline{u}_w)$ in the matching in G' corresponds to choosing path (s, v, w, t) in G. For each edge $(v, w) \in (S \cap T)^2$, there is an associated edge $(\overline{u}_v, \overline{u}_w) \in E'$, with weight $\min\{c_{sv} + c_{vw} + c_{wt},\ c_{sw} + c_{wv} + c_{vt}\}$, which represents the paths (s, v, w, t) and (s, w, v, t) in G. For each edge $(s, t) \in E$, there is an associated edge $(u_s, u_t) \in E'$ with weight c_{uv}.

Clearly, this construction can be performed in polynomial time. One easily verifies that any set of k vertex-disjoint 3-bounded (s, t)-paths in G corresponds to a matching of size k and the same cost in G', and vice versa. Since the cardinality constrained minimum weight matching problem can be solved in polynomial time [11,28], the claim follows. □

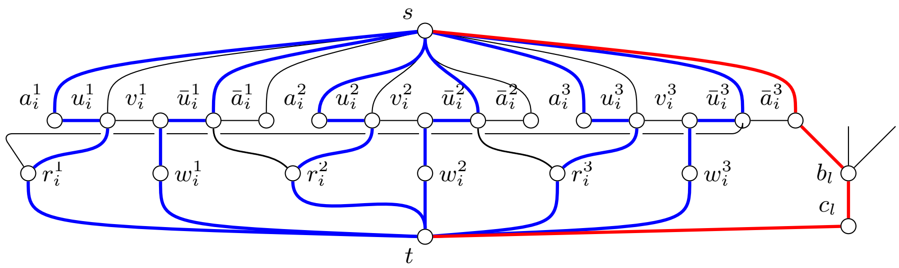

Figure 5. Union of G_i and H_l for variable x_i and clause $C_k = (\bar{x}_i^3 \vee \dots)$. Thick lines are paths in $\bar{\mathcal{P}}_i$ and path Q_{li}.

For $\ell = 4$, the problem becomes at least $\mathcal{APX}$-hard in the general case.

Theorem 7. *WNDP(4) is (at least) $\mathcal{APX}$-hard.*

Proof. We use a construction similar to the one presented in the previous section to reduce MAX-k-SAT(3) to WEDP(4). Again, we let x_i, $i \in I$, be the boolean variables and C_l, $l \in L$ be the clauses of the given MAX-k-SAT(3) instance and we denote the three occurrences of variable x_i by x_i^j, $j \in J := \{1, \dots, 3\}$.

For each $l \in L$, we construct a clause graph $H_l = (W_l, F_l)$ exactly as in the proof of Theorem 3 and shown in Figure 3. For each $i \in I$, we construct a variable graph $G_i = (V_i, E_i)$ as

$$\begin{aligned} V_i :=& \{s,\, t\} \cup \{a_i^j,\, \bar{a}_i^j,\, u_i^j,\, \bar{u}_i^j,\, v_i^j,\, w_i^j,\, r_i^j \mid j \in J\} \quad \text{and} \\ E_i :=& \{s a_i^j,\, s\bar{a}_i^j,\, s u_i^j,\, s\bar{u}_i^j,\, a_i^j u_i^j,\, \bar{a}_i^j \bar{u}_i^j,\, u_i^j v_i^j,\, \bar{u}_i^j v_i^j, \\ & v_i^j w_i^j,\, u_i^j r_i^j,\, \bar{u}_i^j r_i^{j+1},\, r_i^j t,\, w_i^j t \mid j \in J\}\,, \end{aligned}$$

where $r_i^4 = r_i^1$. Figure 5 illustrates these graphs. The graph G is obtained as the union of all G_i and H_l (sub-)graphs. Finally, we assign weight 1 to all edges su_i^j and $s\bar{u}_i^j$ and weight 0 to all other edges in G. The goal in the constructed WNDP(4) instance is to find a minimum cost subgraph of G that contains (at least) $6|I| + |L|$ node-disjoint 4-bounded (s,t)-paths.

For each $i \in I$ and $j \in J$, we denote the paths

$$\begin{aligned} P_{ij} &= (s, u_i^j, v_i^j, w_i^j, t)\,, & P'_{ij} &= (s, \bar{a}_i^j, \bar{u}_i^j, r_i^{j+1}, t)\,, & P''_{ij} &= (s, \bar{u}_i^j, r_i^{j+1}, t)\,, \\ \bar{P}_{ij} &= (s, \bar{u}_i^j, v_i^j, w_i^j, t)\,, & \bar{P}'_{ij} &= (s, a_i^j, u_i^j, r_i^j, t)\,, & \bar{P}''_{ij} &= (s, u_i^j, r_i^j, t)\,. \end{aligned}$$

For each variable x_i that occurs in clause C_l, we denote

$$Q_{li} = \begin{cases} (s, a_i^j, b_l, c_l, t) & \text{if literal } x_i^j \text{ occurs in } C_l, \\ (s, \bar{a}_i^j, b_l, c_l, t) & \text{if literal } \bar{x}_i^j \text{ occurs in } C_l. \end{cases}$$

Note that these are the only 4-bounded (s,t)-paths in G. Furthermore, we let $\mathcal{P}_i := \{P_{ij},\, P'_{ij} \mid j \in J\}$, $\bar{\mathcal{P}}_i := \{\bar{P}_{ij},\, \bar{P}'_{ij} \mid j \in J\}$, and $\mathcal{Q}_l := \{Q_{li} \mid i \in I : x_i \text{ occurs in } C_l\}$. Figure 5 illustrates the paths in $\bar{\mathcal{P}}_i$ and path Q_{li}.

As in the proof of Theorem 3, one finds that a truth assignment $\hat{x}$ that satisfies r clauses of the given MAX-k-SAT(3) instance corresponds to a path set

$$\mathcal{S} = \mathcal{S}(\hat{x}) := \bigcup_{i \in I : \hat{x}_i = true} \mathcal{P}_i \cup \bigcup_{i \in I : \hat{x}_i = false} \bar{\mathcal{P}}_i \cup \{Q_{l i_l(\hat{x})} \mid l \in L : C_l(\hat{x}) = true\}$$

with $|\mathcal{S}| = 6|I| + r$ and cost $c(\mathcal{S}) = 3|I|$. In order to obtain a set of $6|I| + |L|$ paths, we modify $\mathcal{S}$ as follows: For each $l \in L$ with $C_l(\hat{x}) = false$, we arbitrarily chose one i such that x_i^j or $\bar{x}_i^j$ occurs in C_l, add the path Q_{li} to $\mathcal{S}$, and replace the path P'_{ij} or $\bar{P}'_{ij}$ in $\mathcal{S}$ with P''_{ij} or $\bar{P}''_{ij}$, respectively. This modification maintains the node-disjointness of the paths in $\mathcal{S}$ and increases both the size and the cost of $\mathcal{S}$ by $|L|-r$. The cost of the resulting path set $\mathcal{S}$ thus is

$$c(\mathcal{S}(\hat{x})) = 3|I| + |L| - r\ . \tag{1}$$

Conversely, one finds that any set $\mathcal{S}$ of $6|I| + |L|$ node-disjoint 4-bounded (s,t)-paths must contain one path from each set $\mathcal{Q}_l$ and 6 paths within each variable subgraph G_i. The only way to have 6 node-disjoint 4-bounded path within G_i, however, is to have either all 3 paths P_{ij} or all 3 paths $\bar{P}_{ij}$, complemented with 3 paths of the type P'_{ij} and P''_{ij} or with 3 paths of the type $\bar{P}'_{ij}$ and $\bar{P}''_{ij}$, respectively. The cost of such a path set is equal to the number of P_{ij} and $\bar{P}_{ij}$ paths it contains, which amounts to a total of $3|I|$, plus the number of P''_{ij} and $\bar{P}''_{ij}$ paths. Note that the paths P''_{ij} and $\bar{P}''_{ij}$ contain only a subset of the nodes in P'_{ij} and $\bar{P}'_{ij}$, respectively, and that the cost induced by P''_{ij} and $\bar{P}''_{ij}$ is 1, while the cost induced by P'_{ij} and $\bar{P}'_{ij}$ is 0. Thus, we may assume that $\mathcal{S}$ contains path P''_{ij} or $\bar{P}''_{ij}$ only if it contains path Q_{li} for the clause l in which literal x_i^j occurs. Let $\hat{x}(\mathcal{S})$ be the truth assignment defined as

$$\hat{x}_i(\mathcal{S}) := \begin{cases} true & \text{if } P_{i1} \in \mathcal{S}, \\ false & \text{otherwise,} \end{cases} \quad \text{for all } i \in I.$$

Consider a path $Q_{li} \in \mathcal{S}$ and suppose $C_l(\hat{x}(\mathcal{S})) = false$. Then also the literal x_i^j or $\bar{x}_i^j$ occurring in C_l evaluates to $false$. Since $\mathcal{S}$ contains either P_{ij} or $\bar{P}_{ij}$, it also must contain P''_{ij} or $\bar{P}''_{ij}$, respectively. As these paths induce a cost of one, the number of clauses satisfied by $\hat{x}(S)$ is

$$r(\hat{x}(\mathcal{S})) \geq |L| + 3|I| - c(\mathcal{S})\ . \tag{2}$$

As in the proof of Theorem 3, it follows straightforward from (1) and (2) that approximation ratios are transformed linearly by the presented reduction and, hence, WNDP(4) is $\mathcal{APX}$-hard. □

Unfortunately, it remains open if WNDP(4) is approximable within a constant factor or not. The best known approximation ratio for WNDP(4) is $\mathcal{O}(k)$, which is achieved by a simple greedy algorithm.

Theorem 8. *WNDP(4) can be approximated within a factor of* $4k$.

Proof. Consider the algorithm, which adds the edges in order of non-decreasing cost until the constructed subgraph contains k node-disjoint 4-bounded (s,t)-paths and then returns the subgraph defined by these paths. As, in each iteration, we can check in polynomial time whether such paths exist or not [19], this algorithms runs in polynomial time. Furthermore, the optimal solution must contain at least one edge whose cost is at least as big as the cost of the last edge added by the greedy algorithm. Therefore, the total cost of the greedy solution is at most $4k$ times the optimal solution's cost. □

4 Conclusion

In this paper we show that the maximum edge-disjoint 4-bounded paths problem MEDP(4) is $\mathcal{APX}$-complete and that the corresponding weighted edge-disjoint paths problem WEDP(4) is $\mathcal{NPO}$-complete. The weighted node-disjoint ℓ-bounded paths problem was proven to be polynomially solvable for $\ell = 3$ and to be at least $\mathcal{APX}$-hard for $\ell = 4$. This closes all basic complexity issues that were left open in [19,6]. In addition, we presented a 2-approximation algorithm for WEDP(4) and a $4k$-approximation algorithm WNDP(4). It remains open whether WNDP(4) is approximable within a factor better than $\mathcal{O}(k)$ or if there is a stronger, non-constant approximation threshold.

The hardness results and algorithms presented in this paper also hold for directed graphs and for graphs containing parallel edges.

References

1. D. Alevras, M. Grötschel, and R. Wessäly. Capacity and survivability models for telecommunication networks. ZIB Technical Report SC-97-24, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1997.
2. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*. Springer-Verlag, 1999.
3. G. Baier. *Flows with path restrictions*. PhD thesis, Technische Universität Berlin, 2003.
4. G. Baier, T. Erlebach, A. Hall, E. Köhler, P. Kolman, O. Panagrác, H. Schilling, and M. Skutella. Length-bounded cuts and flows. *ACM Transactions on Algorithms*, To appear.
5. W. Ben-Ameur. Constrained length connectivity and survivable networks. *Networks*, 36:17–33, 2000.
6. A. Bley. On the complexity of vertex-disjoint length-restricted path problems. *Computational Complexity*, 12:131–149, 2003.
7. Q. Botton, B. Fortz, and L. Gouveia. The k-edge 3-hop constrained network design polyhedron. In *Proceedings of the 9th INFORMS Telecommunications Conference*, 2008. Available as preprint at Université Catholique de Louvain: Le polyedre du probleme de conception de reseaux robustes K-arête connexe avec 3 sauts.
8. T. Chakraborty, J. Chuzhoy, and S. Khanna. Network design for vertex connectivity. In *Proceedings of the 40th Annual ACM Symposium on the Theory of Computing (STOC'08)*, pages 167–176, 2008.
9. C. Chekuri, S. Khanna, and F.B. Shepherd. An $\mathcal{O}(\sqrt{n})$ approximation and integrality gap for disjoint paths and unsplittable flow. *Theory of Computing*, 2:137–146, 2006.
10. G. Dahl, D. Huygens, A.R. Mahjoub, and P. Pesneau. On the k edge-disjoint 2-hop-constrained paths polytope. *Operations Research Letters*, 34:577–582, 2006.
11. J. Edmonds. Maximum matching and a polyhedron with 0-1 vertices. *Journal of Research of the National Bureau of Standards*, 69B:125–130, 1965.
12. G. Exoo. On line disjoint paths of bounded length. *Discrete Mathematics*, 44:317–318, 1983.
13. P.A. Golovach and D.M. Thilikos. Paths of bounded length and their cuts: Parameterized complexity and algorithms. In *Proceedings of the 4th International Workshop on Parametrized and Exact Computation (IWPEC 2009)*, LNCS. Springer, To appear.
14. L. Gouveia, P. Patricio, and A. De Sousa. Compact models for hop-constrained node survivable network design: An application to MPLS. In G. Anandaligam and S. Raghavan, editors, *Telecommunications Planning: Innovations in Pricing, Network Design and Management*. Springer, 2005.
15. L. Gouveia, P. Patricio, and A. De Sousa. Hop-constrained node survivable network design: An application to MPLS over WDM. *Networks and Spatial Economics*, 8(1), 2008.
16. M. Grötschel, C. Monma, and M. Stoer. *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, chapter 10, pages 617–672. Elsevier, North-Holland, Amsterdam, 1995. M. Ball, T. Magnanti, C. Monma, G. Nemhauser (Eds.).
17. V. Guruswami, S. Khanna, B. Shepherd, R. Rajaraman, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. In *Proceedings of the Thirty-first Annual ACM Symposium on the Theory of Computing, Atlanta, GA*, pages 19–28, 1999.
18. D. Huygens and A.R. Mahjoub. Integer programming formulations for the two 4-hop constrained paths problem. *Networks*, 49(2):135–144, 2007.
19. A. Itai, Y. Perl, and Y. Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12:277–286, 1982.
20. Y. Lando and Z. Nutov. Inapproximability of survivable networks. *Theoretical Computer Science*, 410:2122–2125, 2009.
21. L. Lovász, V. Neumann-Lara, and M. Plummer. Mengerian theorems for paths of bounded length. *Periodica Mathematica Hungarica*, 9(4):269–276, 1978.
22. A.R. Mahjoub and T. McCormick. Max flow and min cut with bounded-length paths: Complexity, algorithms and approximation. *Mathematical Programming*, To appear.
23. M. Martens and M. Skutella. Length-bounded and dynamic k-splittable flows. In *Operations Research Proceedings 2005*, pages 297–302, 2006.
24. K. Menger. Zur allgemeinen Kurventheorie. *Fund. Mathematicae*, 10:96–115, 1927.
25. L. Niepel and D. Safarikova. On a generalization of Menger's theorem. *Acta Mathematica Universitatis Comenianae*, 42:275–284, 1983.
26. Y. Perl and D. Ronen. Heuristics for finding a maximum number of disjoint bounded paths. *Networks*, 14, 1984.
27. L. Pyber and Z. Tuza. Menger-type theorems with restrictions on path lengths. *Discrete Mathematics*, 120:161–174, 1993.
28. A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.
29. D. Wagner and K. Weihe. A linear-time algorithm for edge-disjoint paths in planar graphs. *Combinatorica*, 15(1):135–150, 1995.