Length-Bounded Cuts and Flows*

Georg Baier[1**], Thomas Erlebach[2], Alexander Hall[3], Ekkehard Köhler[1], Heiko Schilling[1], and Martin Skutella[4]

[1] Institute of Mathematics, TU Berlin, Germany
[2] Department of Computer Science, U Leicester, England
[3] Institute of TCS, ETH Zurich, Switzerland
[4] Department of Mathematics, U Dortmund, Germany

Abstract. An L-length-bounded cut in a graph G with source s and sink t is a cut that destroys all s-t-paths of length at most L. An L-length-bounded flow is a flow in which only flow paths of length at most L are used. We show that the minimum length-bounded cut problem in graphs with unit edge lengths is $\mathcal{NP}$-hard to approximate within a factor of 1.1377 for $L \geq 5$ in the case of node-cuts and for $L \geq 4$ in the case of edge-cuts. We also give approximation algorithms of ratio $\min\{L, n/L\}$ in the node case and $\min\{L, n^2/L^2, \sqrt{m}\}$ in the edge case. We discuss the integrality gaps of the LP relaxations of length-bounded flow and cut problems, analyze the structure of optimal solutions, and present further complexity results for special cases.

1 Introduction

In a classical article Menger [1] shows a strong relation between cuts and systems of disjoint paths: let G be a graph and s, t two nodes of G, then the maximum number of edge-/node-disjoint s-t-paths equals the minimum size of an s-t-edge-/node-cut (Menger's Theorem). Originally, Menger showed the result for node-disjoint paths and node-cuts, i. e., the cut is a node set which is removed. The edge-disjoint version has later been formulated by Dantzig and Fulkerson [2] for directed graphs and by Kotzig [3] for undirected graphs. A generalization of the concept of edge-disjoint s-t-paths is an *s-t-flow*, which is a mapping $f : \mathcal{P}_{s,t} \to \mathbb{R}^+$ from the set of all s-t-paths $\mathcal{P}_{s,t}$ in G into the nonnegative real numbers. In 1956 both Ford and Fulkerson [4] and Elias, Feinstein, and Shannon [5] generalized the theorem of Menger to flows and they provided algorithms to find an s-t-flow and an s-t-cut both of the same value. Throughout this paper we will refer to a path-based formulation of flows in contrast to the more commonly used edge-based version of flows. This is due to the fact that we are interested in flows with constraints on the flow carrying paths.

The first research for path related constraints we are aware of was done in 1978 by Lovász, Neumann-Lara, and Plummer [6]. They consider the maximum length-bounded node-disjoint s-t-paths problem. For length-bounds 2, 3, and 4 an analogous relation as Menger's theorem with a new suitable cut definition holds. For length-bounds greater than 4 they give upper and lower bounds for the gap between the maximum number of length-bounded node-disjoint paths and the minimum cardinality of a cut. Furthermore they provide examples showing that some of the bounds are tight. The results were extended independently to edge-disjoint paths by Exoo [7] and Niepel and Safaríková [8]. Length-bounded edge-disjoint s-t-paths can be interpreted as 0, 1-valued length-bounded flows in unit-capacity graphs therefore they are a special sub-problem of integral length-bounded flows.

* This work was partly supported by the Federal Ministry of Education and Research (BMBF grant 03-MOM4B1), by the European Commission - Fet Open project DELIS IST-001907 (SBF grant 03.0378-1), and by the German Research Foundation (DFG grant MO 446/5-2).

** since 12/04 at Corporate Technology, Information & Communication, Siemens AG, Germany

According to Bondy and Murty [9], Lovász conjectured that there is a constant C such that the size of a minimum L-length-bounded s-t-node-cut, i. e., a minimum node-set disjoint to $\{s, t\}$ which hits each L-length-bounded s-t-path, is at most a factor of $C \cdot \sqrt{L}$ larger than the cardinality of a maximum system of node-disjoint s-t-paths of length at most L. Exoo and Boyles [10] disprove this conjecture. They construct for each length-bound $L > 0$ a graph and a node pair s, t, such that the minimum L-length-bounded s-t-node-cut has a size greater than $C \cdot L$ times the maximum number of node-disjoint s-t-paths of length at most L; the constant C is roughly 1/4.

Itai, Perl, and Shiloach [11] give algorithms to find the maximum number of node-/edge-disjoint s-t-paths with at most 2 or 3 edges; the node-disjoint case is also solved for length-bound 4. On the complexity side they show that the node- and edge-disjoint length-bounded s-t-paths problem is $\mathcal{NP}$-complete for length-bounds greater than 4. Instead of fixing the path length, one can fix the number of paths and look for the minimal value bounding all path lengths. Again both the node- and edge-disjoint version is already $\mathcal{NP}$-complete for two paths.

For fractional length-bounded multi-commodity flows in weighted graphs with edge lengths in $\mathbb{R}^+$ a fully polynomial time approximation scheme (FPTAS) has been given by Fleischer and Skutella [12]: for each $\epsilon > 0$ one can find a fractional $(1 + \epsilon)L$-length-bounded multi-commodity flow with a total flow value at least as large as the maximum fractional L-length-bounded multi-commodity flow value. This FPTAS results in a polynomial time algorithm for fractional length-bounded multi-commodity flows and fractional length-bounded edge-(multi-)cuts in unit-length graphs.

Guruswami et al. [13] study the length-bounded edge-disjoint path problem for multiple node pairs s_i, t_i, $1 \leq i \leq k$, i.e. find a largest subset of node pairs that can be simultaneously connected in an edge-disjoint manner. They show that, for any $\epsilon > 0$ the problem is hard to approximate within $n^{\frac{1}{2}-\epsilon}$ (they actually claim $m^{\frac{1}{2}-\epsilon}$, which was corrected later) even in undirected networks, and give an $\mathcal{O}(\sqrt{m})$-approximation algorithm for it, where n denotes the number of nodes and m the number of edges. For directed networks, they show that even the s-t-case (i.e. find the maximum number of length-bounded paths between two nodes s and t) is hard to approximate within $n^{\frac{1}{2}-\epsilon}$, for any $\epsilon > 0$.

Mahjoub and McCormick [14] present a polynomial algorithm for the 3-length-bounded edge-cut in undirected graphs. Furthermore, they show that the fractional versions of the length-bounded flow- and cut problem are polynomial even if L is part of the input and that the integral versions are strongly $\mathcal{NP}$-Hard even if L is fixed.

Length-bounded path problems arise naturally in a variety of real world optimization problems and therefore many heuristics for finding large systems of length-bounded paths have been developed, e.g. see [15,16,17]. For instance, in the field of telecommunication networks one is interested in the so called x-fault-distance, i. e., the maximal distance between two nodes if any set of at most x nodes/edges is removed, see Hsu [18]. An x-(edge)-fault-distance for nodes s and t of value greater than L implies the existence of an s-t-cut with respect to a length-bound L of size at most x.

Our Contribution. In this paper, we present various results concerning the complexity and approximability of length-bounded cut and flow problems. In Section 3, we show that the minimum length-bounded cut problem in graphs with unit edge lengths is $\mathcal{NP}$-hard to approximate within a factor of at least 1.1377 for $L \geq 5$ in the case of node-cuts and for $L \geq 4$ in the case of edge-cuts; see Table 1 for an overview of known and new complexity results. We also

Table 1. Known and new (bold type) complexity results; $\varepsilon \in \mathbb{R}^+$ and $c \in \mathbb{N}$ are constants, ε can be arbitrarily small.

L	node-cut	edge-cut
1	—	poly.
2	poly.	poly.
3	poly.	poly. [14] (undirected)
4	poly. [6] (undirected)	**inapprox. within 1.1377** (directed & undirected)
$5 \ldots \lfloor n^{1-\varepsilon} \rfloor$	**inapprox. within 1.1377** (directed & undirected)	**inapprox. within 1.1377** (directed & undirected)
$n - c$	**poly.** (directed & undirected)	

give approximation algorithms of ratio $\min\{L, n/L\}$ in the node case and $\min\{L, n^2/L^2, \sqrt{m}\}$ in the edge case. Furthermore, we show that the integrality gap of the LP relaxation can be at least $\Omega(\sqrt{n})$. Section 4 discusses the maximum length-bounded flow problem. For series-parallel graphs with unit edge lengths and unit edge capacities, we show a lower bound of $\Omega(\sqrt{n})$ on the integrality gap of the LP formulation. Furthermore, we show that edge- and path-flows are not polynomially equivalent for length-bounded flows. That means, even if the graph is outer-planar, there is no polynomial algorithm to transform an edge-flow which is known to correspond to a length-bounded path-flow into a length-bounded path-flow. We also analyze the structure of optimal solutions and show that there are instances where each maximum flow ships a large percentage of the flow along paths with a very small flow value. The fractionality of these maximum flows can be chosen arbitrarily small.

2 Preliminaries

Our graphs $G = (V, E)$ are *finite*. $V = V(G)$ is the *node* set of G and $E = E(G)$ is the *edge* set of G. The graphs can either be directed or undirected but they are always self-loop free. A graph may contain *multi-edges*, i.e., parallel edges, in which case the graph will be called a *multi-graph*. Sometimes, we call an edge *simple* to distinguish it clearly from multi-edges. The graph G possesses two independent edge-weights, an edge-capacity function $u : E \to \mathbb{Q}_{>0}$ and a (primal) edge-length function $d : E \to \mathbb{Q}_{\geq 0}$. If not stated otherwise we assume unit-length and unit-capacity for our graphs.

2.1 Length-Bounded Cuts

An edge set C_e of a graph $G = (V, E)$ is called an *edge-cut* if $G \setminus C_e = (V, E \setminus C_e)$ has at least one connected component more than G. If two nodes s and t are in the same connected component of G but in different connected components in $G \setminus C_e$ then s and t are called *separated* by C_e. C_e is called a *s-t-edge-cut* and its *value* (or *capacity*) is the number of the edges in C_e (or the total capacity of the edges in C_e, if the edge capacities are not unit). Similarly, a node set C_n of G which separates s and t is defined as an *s-t-node-cut* and its *value* is the number of nodes in C_n.

The *distance* or *length* of an s-t-path is the sum of the lengths of all edges on the path, both in the edge-cut case and in the node-cut case. We call a subset C_e^L of the edge set of a graph G a *length-bounded s-t-edge-cut* with respect to the length-bound L, or *L-length-bounded s-t-edge-cut*, if the nodes s and t have a distance greater than L in $G \setminus C_e^L$. Similarly, a subset C_n^L of the node set of G is called a *L-length-bounded s-t-node-cut* if it destroys all s-t-paths of a length at most L. The *value* (or *capacity*) of a length-bounded cut is defined as in the standard cut case. In the *Minimum Length-Bounded Cut* problem we are looking for a length-bounded cut of minimum value. All of our cuts are s-t-cuts and therefore we will omit the s-t-prefix. If the type of a cut is clear from the context, we will also omit the indices e, n, or L of C.

In the linear programming relaxation of the minimum length-bounded edge-cut problem one has to assign to each edge e a dual length ℓ_e such that the dual length of a shortest s-t-path from $\mathcal{P}_{s,t}(L)$ is at least 1 (the LP relaxation for node-cuts is analogous):

$$\min \sum_{e \in E} u_e \ell_e \qquad \text{s.t.} \quad \sum_{e \in P} \ell_e \geq 1 \quad (P \in \mathcal{P}_{s,t}(L)), \quad \ell_e \geq 0 \quad (e \in E) \tag{1}$$

An integral solution of the linear program in (1) corresponds to a length-bounded s-t-cut, and vice versa. In particular, the minimum length-bounded s-t-cut value and the value of a minimum integral solution are equal. We will refer to feasible solutions of (1) as *fractional cuts* since only a fraction of an edge's capacity may be in the cut.

2.2 Length-Bounded Flows

Length-bounded flows are single- or multi-commodity path-flows in which each path used must obey a length constraint. Let $\mathcal{P}_{s,t}(L)$ denote the set of all s-t-paths with a length at most L. Then, a *L-length-bounded s-t-flow* is defined as a function $f : \mathcal{P}_{s,t}(L) \to \mathbb{R}_{\geq 0}$ assigning a flow value f_P to each s-t-path P in G of length at most $L \geq 0$. The sum $\sum_{P \in \mathcal{P}_{s,t}(L)} f_P$ is called the *s-t-flow value* of f. The flow f is *feasible* if it respects edge capacities, i. e., for each edge $e \in E$ the sum of the flow values of paths containing this edge must be bounded by its capacity u_e. If not stated otherwise our flows are s-t-flows. Therefore we will omit the s-t-prefix.

A natural optimization objective is to find a feasible length-bounded s-t-flow such that the flow value is maximal. We can formulate the problem as a linear program in the following way:

$$\max \sum_{P \in \mathcal{P}_{s,t}(L)} f_P \qquad \text{s.t.} \quad \sum_{P : e \in P} f_P \leq u_e \quad (e \in E), \quad f_P \geq 0 \quad (P \in \mathcal{P}_{s,t}(L)) \tag{2}$$

We will refer to feasible solutions of (2) as *path-flows*. Note that the dual of the linear program in (2) is the linear program in (1) for the minimum length-bounded cut problem. One way to prove the maximum-flow minimum-cut equality for standard flows is to apply duality theory of linear programming. In case of multiple commodities, a source- and sink-node pair (s_i, t_i) and a length-bound $L_i \geq 0$ is given for each commodity $i = 1, \ldots, k$. An *$(L_1, \ldots, L_k)$-length-bounded multi-commodity flow* f is a set of L_i-length-bounded s_i-t_i-flows f_i for $i = 1, \ldots, k$.

3 Length-Bounded Cuts

3.1 Gap: Length-Bounded Disjoint Paths vs. Cut

It follows from linear programming duality that the maximum fractional length-bounded flow value equals the minimum fractional length-bounded cut value. For standard flows this

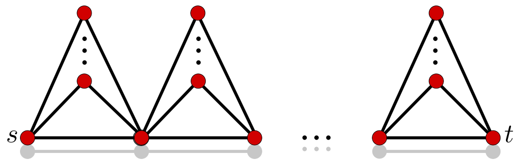

Fig. 1. Example of a large integrality gap for the linear program of the minimum length-bounded cut. The straight s-t-path (in gray) contains $2k+1$ edges. Each of these edges is accompanied by $k+1$ parallel paths of length 2 .

equality holds for (integral) cuts as well. In the presence of a length-bound, the maximum flow value and the minimum cut value may be different. This is an immediate consequence of the integrality gap we state in the following theorem.

Theorem 1. *For (un-)directed series-parallel graphs the ratio of the minimum fractional length-bounded edge-/node-cut value to the minimum integral one can be of order $\Omega(\sqrt{n})$.*

Proof. We give the construction for edge-cuts and note in the end how to adapt it to node-cuts. We construct a family $\{G_k\}_{k\in\mathbb{N}}$ of order $2k^2+5k+3$ graphs with a fractional length-bounded edge-cut value less than 2 and an integral length-bounded cut value $k+1$. The graph G_k is generated from an s-t-path containing $2k+1$ edges; we call them *ground edges*. Parallel to each ground edge we add $k+1$ paths of length 2, see Figure 1 for the undirected case. Note that the proof goes through for directed edges as well (direct edges from left to right).

Consider a graph G_k for arbitrary k and let $L = 3k+1$. A minimum fractional edge-cut has value less than 2. This can be seen as follows. For a fractional edge-cut we have to assign a dual edge-length to each edge, such that the dual length of each s-t-path with at most $L = 3k+1$ edges is not less than 1. An s-t-path with at most $3k+1$ edges must contain at least $k+1$ ground edges. Thus, assigning all ground edges a dual length of $\frac{1}{k+1}$ and assigning 0 to the remaining edges yields a fractional cut of value $\frac{2k+1}{k+1} < 2$.

Now we give a lower bound of $k+1$ on the size of an (integral) edge-cut. If we take a non-ground edge, we must take at least $k+1$ non-ground edges. Otherwise for any non-ground edge in the cut there would always be another equivalent length 2 path which is not cut and thus the non-ground edges could be removed from the cut without invalidating it. A cut containing only ground edges must have size greater than k, otherwise an s-t-path of length $L = 3k+1$ remains. Since $k+1$ is in $\Theta(\sqrt{n})$ this completes the proof.

For node-cuts one can simply take the line graph (replace each edge by a node, connect two nodes, if the corresponding edges shared a node) of the above construction. This gives the $\Theta(\sqrt{n})$ lower bound on the integrality gap in undirected and directed (direct edges from left to right) graphs. □

Corollary 1. *For (un-)directed series-parallel graphs the ratio of the minimum length-bounded edge-/node-cut size to the maximum number of length-bounded edge-disjoint paths can be of order $\Omega(\sqrt{n})$.*

3.2 Complexity and Polynomially Solvable Cases

Table 1 shows an overview of known and new results concerning the complexity, inapproximability, and polynomially solvable cases of the length-bounded cut problems. Furthermore, we give an $\mathcal{NP}$-hardness proof for the edge version in weighted series-parallel & outer-planar graphs. Note that the polynomial algorithms for L equals $2,3$ and $1,2$ for the node and edge version, respectively, are easy exercises for both directed and undirected graphs (for the case $L = 3$ node-cut or $L = 2$ edge-cut: after directly cutting length 2 or length 1 paths, respectively, Theorem 5 can be applied).

Node-Cuts We first state our result concerning an easy polynomial time algorithm for length-bounded node-cuts with $L = n - c$, where $c \in \mathbb{N}$ is an arbitrary constant and then present the inapproximability result, which is the main result of this section.

Theorem 2. *For $L = n-c$, where $c \in \mathbb{N}$ is an arbitrary constant, a minimum length-bounded node-cut can be computed in polynomial time in (un-)directed graphs.*

Proof. We enumerate all $V' \subseteq V$ with $|V'| \leq c$. We return the smallest V' which is a length-bounded node-cut, if there is any. Otherwise we know that any length-bounded node-cut V' contains $> c$ nodes and therefore less than $n - c$ nodes remain. This gives that the longest remaining path must have a length of less than $n - c$ and therefore V' actually cuts all s-t-paths. In this case returning a standard minimum node-cut suffices. □

Theorem 3. *It is $\mathcal{NP}$-hard to approximate the length-bounded node-cut in an (un-)directed (simple) graph for $L \in \{5, \ldots, \lfloor n^{1-\varepsilon} \rfloor\}$ within a factor of 1.1377, for an arbitrarily small constant $\varepsilon \in \mathbb{R}^+$.*

Proof. We reduce the well known Vertex Cover problem: a vertex cover for an undirected graph $G_{\text{vc}} = (V_{\text{vc}}, E_{\text{vc}})$ is a subset of the nodes $V'_{\text{vc}} \subseteq V_{\text{vc}}$ such that for each edge $\{u, v\} \in E_{\text{vc}}$ at least one of the nodes u, v is in V'_{vc}. The problem to find a minimum vertex cover has been shown $\mathcal{NP}$-hard to approximate within ≈ 1.3606 [19]. We now describe in detail how to reduce a given Vertex Cover instance G_{vc} to a length-bounded node-cut instance $G = (V, E)$ for $L = 5$ and G directed, then we show how to generalize to other values of L, and finally we note why the construction also works for undirected graphs.

Let $n_{\text{vc}} = |V_{\text{vc}}|$ denote the number of nodes in the given instance. Start with $V = \{s, t\}$ and no edges. For each node $v \in V_{\text{vc}}$ we add a *node gadget* to G consisting of seven nodes which are interconnected with s, t and themselves as shown in Figure 2 (left)—the nodes in the bottom half surrounded by a gray box. For each edge $\{u, v\} \in E_{\text{vc}}$ we add an *edge gadget* consisting of four nodes and six edges connecting them to the node gadgets corresponding to u and v as shown in Figure 2 (left). The following lemma will be helpful.

Lemma 1. *From a vertex cover V'_{vc} in G_{vc} of size x one can always construct a node-cut V' in G of size $n_{\text{vc}} + x$ and vice versa, for $x < n_{\text{vc}}$.*

Proof. We start with the easier direction "⇒": Let $V'_{\text{vc}} \subseteq V_{\text{vc}}$ be a vertex cover with $|V'_{\text{vc}}| = x$. For each node $v \in V'_{\text{vc}}$ we add l_v and r_v to our cut $V' \subseteq V$ and for each node $u \in V_{\text{vc}} \setminus V'_{\text{vc}}$ we add m_u to V' (see Figure 2 for an example). Clearly this gives $|V'| = n_{\text{vc}} + x$. To see that no path remains after removing V' from G, first consider the node gadgets for each $v \in V_{\text{vc}}$ individually. In case m_v was added to the cut, no path remains in the gadget. In case l_v and r_v were added, the only remaining path (via m_v) has length 6, which is greater than the length-bound L. Now consider an edge gadget for the edge $\{u, v\} \in E_{\text{vc}}$ and assume that an s-t-path remains. Then either l_v and r_u are not in the cut or l_u and r_v are not in the cut. By construction this means that both u and v were not in V'_{vc}, which is a contradiction to V'_{vc} being a vertex cover. This gives that for none of the edge gadgets paths remain and therefore altogether no path of length at most L remains in G after removing V'.

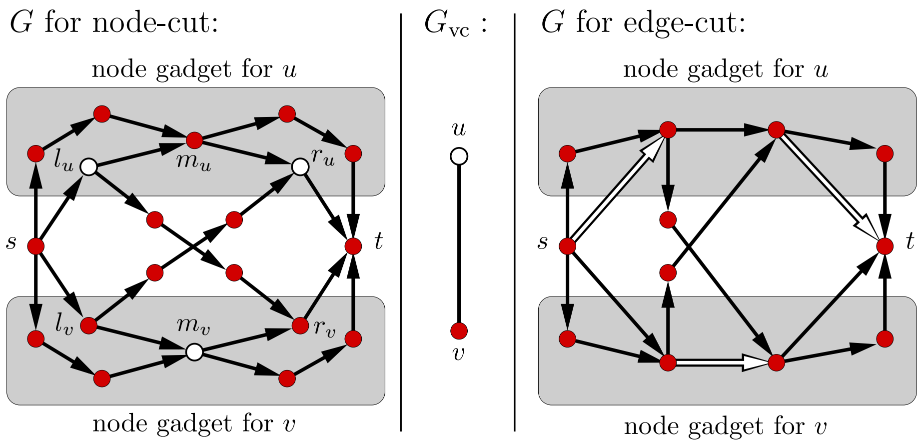

Fig. 2. Gadgets for the reduction of Vertex Cover to length-bounded node-cut (left) and length-bounded edge-cut (right), respectively. Both correspond to two connected nodes u, v of the given Vertex Cover instance, shown in the middle. The highlighted nodes (edges) are in the cut / vertex cover

Now we come to direction "$\Leftarrow$": Let $V' \subseteq V$ be a node-cut with $|V'| = n_{\rm vc} + x$. With two simple transformations we ensure that V' only contains nodes from the node gadgets and for each node gadget either the m-type node is contained or both the l- and the r-type nodes.

No Nodes from Edge Gadgets in Cut. Consider an edge gadget, say for edge $\{u, v\} \in E_{\rm vc}$, for which at least one of its four nodes is in V'. The edge gadget consists of two paths, one from l_v to r_u and one from l_u to r_v. If an inner node of the l_v-r_u-path (l_u-r_v-path) is in V', we replace it by l_v (l_u). This introduces no new paths and does not increase the size of the cut.

In Node Gadget: Either "m" Node or "l" & "r" Nodes. Consider the node gadget for $v \in V_{\rm vc}$. First note that at least one node of the gadget must be in V', otherwise three s-t-paths of length at most L remain (for this gadget). If only one node of the gadget is in V', it must be m_v otherwise at least one path remains. If two or more nodes of the gadget are in V', we replace them by l_v and r_v. Thereby no paths are made possible in the node gadget (only the path of length 6 via m_v remains) and all potential paths via edge gadgets connected to this node gadget are cut.

The two transformations clearly do not increase the size of V'. Let us assume for now that they also do not decrease the cut size, i.e., $|V'| = n_{\rm vc} + x$ still holds.

A vertex cover $V'_{\rm vc}$ can easily be derived from the transformed V': add all nodes $v \in V_{\rm vc}$ to $V'_{\rm vc}$ for which both $l_v, r_v \in V'$. Assume some edge $\{u, v\} \in E_{\rm vc}$ remains uncovered, then both m_u and m_v are in the cut V' (and no other nodes of the two node gadgets). Hence, two s-t-paths via the edge gadget connecting l_v and r_u / l_u and r_v remain, which gives a contradiction to V' being a cut. Since there are always either one or two nodes of each node gadget in the cut, there can be at most $|V'| - n_{\rm vc} = x$ gadgets which contain two cut nodes. This yields $|V'_{\rm vc}| = x$, as desired.

If the size of V' was decreased by the above transformations to, say, $n_{\rm vc} + x'$, we simply add $x - x'$ nodes to the vertex cover $V'_{\rm vc}$ to ensure that its size is exactly x; note $x < n_{\rm vc}$ and $x' \geq 0$. $\square$

The proof of Theorem 1.1 in [19] gives the following gap. There are graphs $G_{\rm vc}$ for which it is $\mathcal{NP}$-hard to distinguish between two cases: the case where a vertex cover of size $n_{\rm vc} \cdot (1 - p + \varepsilon')$ exists and the case where any vertex cover has size at least $n_{\rm vc} \cdot (1 - 4p^3 + 3p^4 - \varepsilon')$, for

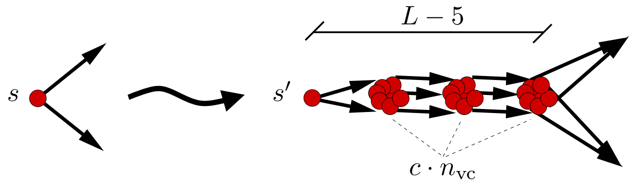

Fig. 3. Replacing s by a path of length $L-5$.

any $\varepsilon' \in \mathbb{R}^+$ and $p = (3-\sqrt{5})/2$. If we plug this into the result of Lemma 1, we have shown that the length-bounded node-cut is hard to approximate within a factor (there is an $\varepsilon' \in \mathbb{R}^+$ for which the inequality holds): $(n_{\mathrm{vc}}+n_{\mathrm{vc}}\cdot(1-4p^3+3p^4-\varepsilon'))/(n_{\mathrm{vc}}+n_{\mathrm{vc}}\cdot(1-p+\varepsilon')) > 1.1377$.

For other values of $L \in \{5,\ldots,\lfloor n^{1-\varepsilon}\rfloor\}$, we modify the construction of G as follows: (1) Add a path of length $L-5$ from a new source node s' to s. Let s' be our new source. (2) Stepwise replace each node on this path after s' and until s (inclusive) by a group of $c \cdot n_{\mathrm{vc}}$ nodes, for some constant c. For each of these groups connect all new nodes with all neighbors of the replaced node (see Figure 3).

(1) ensures the desired path lengths. (2) ensures that none of the new nodes appear in a node-cut computed by a $c/2$-approximation algorithm: a group must be taken completely or not at all; if only a few nodes of the group are in the cut, we can remove these nodes without invalidating the cut. If all nodes of a group are in the cut, this cannot be a $c/2$-approximation. By taking the l- and r-type nodes of all node gadgets but one and taking the m-type node of the last node gadget, we obtain a cut of size $2n_{\mathrm{vc}}-1$. Comparing this with the size of a group yields a factor greater than $c/2$. Thus, we know that any $c/2$-approximation algorithm for length-bound L will still compute a cut in the original construction. The total number of nodes in G depends on L and is $n = \Theta(L \cdot n_{\mathrm{vc}} + n_{\mathrm{vc}} + m_{\mathrm{vc}})$, where $m_{\mathrm{vc}} = |E_{\mathrm{vc}}|$. Therefore, we can create instances for which L is as large as $\lfloor n^{1-\varepsilon}\rfloor$, for arbitrarily small $\varepsilon \in \mathbb{R}^+$.

To see that the reduction also works for undirected graphs, observe that by removing the edge directions in the gadgets, no new undirected paths of length less than L are introduced. □

Edge-Cuts The polynomial time algorithm for node-cuts with length-bound $n-c$ does not carry over for the edge version of the problem, since by removing c edges one cannot guarantee that computing a standard cut suffices. The inapproximability result does carry over, as stated in the following theorem. It can be shown analogously to the proof of Theorem 3 with the difference that the adapted gadgets given in Figure 2 (right) should be used, which already work for length-bound $L=4$.

Theorem 4. *It is $\mathcal{NP}$-hard to approximate the length-bounded edge-cut in an (un-)directed (simple) graph for $L \in \{4,\ldots,\lfloor n^{1-\varepsilon}\rfloor\}$ within a factor of 1.1377, for an arbitrarily small constant $\varepsilon \in \mathbb{R}^+$.*

Lemma 2. *For a series-parallel & outer-planar (un-)directed graph with edge-capacities and lengths it is $\mathcal{NP}$-hard to decide whether there is a length-bounded edge-cut of size less than a given value.*

Proof. We will show a reduction of 2-Partition to the length-bounded cut problem. We are given an arbitrary 2-Partition instance $a_1,\ldots,a_k \in \mathbb{N}$. We have to decide if there exists a partition A_1, A_2 of the ground set $A_1 \cup A_2 = \{a_1,\ldots,a_k\}$ such that $\sum_{i\in A_1} a_i = \sum_{i\in A_2} a_i =: B$ holds.

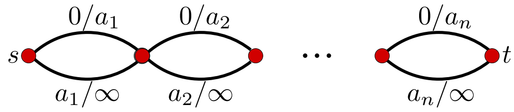

Fig. 4. Reduction of 2-Partition to the length-bounded cut problem. The labels denote length/capacity.

Graph G is a single s-t-path with k multi-edges; each multi-edge consists of two parallel simple edges, see Figure 4. All k upper edges have length 0 and successively $a_1, \ldots, a_k$ as capacity. The lower edges get successively $a_1, \ldots, a_k$ as length and capacity ∞. Note that to obtain a simple graph, we can simply subdivide one of the parallel edges, which still yields a series-parallel & outer-planar graph. For the directed version simply direct the edges from left to right.

Let the length-bound be $L = B - 1$. We will show that there is an edge-cut of size at most B if and only if the instance of 2-Partition is a yes-instance.

"$\Leftarrow$": Given a solution A_1, A_2 to the 2-Partition instance, we take the upper edges corresponding to set A_1 as our edge-cut. Clearly only s-t-paths of length at least B remain and the cut has size B.

"$\Rightarrow$": We start by showing that any edge-cut must have size at least B. Assume C is a cut of size less than B, then the path which takes the upper edges complementary to C will have length less than B, which gives a contradiction. Thus, a given edge cut of size at most B has size exactly B and yields a two partition in the obvious way. □

A Note on Fractional Length-Bounded Edge-Cuts. We will show in Theorem 8 that it is $\mathcal{NP}$-hard to decide whether a fractional length-bounded flow of given flow value exists even if the graph is outer-planar. Since the primal and dual programs have identical optimal objective function values, the same holds for the fractional length-bounded edge-cut problem.

3.3 Approximation algorithms

If the length-bound L is so large that the system of L-length-bounded s-t-paths contains the set of all s-t-paths ($\mathcal{P}_{s,t}(L) = \mathcal{P}_{s,t}$), then length-bounded cuts and flows reduce to standard cuts and flows. The maximum-flow minimum-cut equality holds and there are many efficient algorithms to compute minimum cuts and maximum flows exactly. Another extreme case is if the length-bound equals the distance between s and t, denoted by $\mathrm{dist}(s,t)$. Lovász, Neumann-Lara, and Plummer [6] show a special version of the following theorem in the context of length-bounded node-disjoint paths.

Theorem 5. *In weighted (un-)directed multi-graphs with edge lengths in $\mathbb{R}^+$, for $L = \mathrm{dist}(s,t)$ the minimum length-bounded edge-/node-cut and the maximum length-bounded flow problem can be solved efficiently. In particular, the max flow value and the min cut value coincide if $L = \mathrm{dist}(s,t)$.*

Proof. We first consider directed graphs. Let G be such a graph with edge-capacities and edge-lengths and let $L = \mathrm{dist}(s,t)$. First we generate the sub-graph $\overline{G}$ induced by all edges which are contained in at least one shortest s-t-path in G. This sub-graph can be found with a slightly modified Dijkstra-labeling algorithm; one has to remember for each node *all* incoming edges generating the smallest label at this node. $\overline{G}$ is acyclic (i.e., a DAG). In particular, each s-t-path in $\overline{G}$ is a shortest s-t-path in G. Therefore, a standard minimum cut and a maximum flow in $\overline{G}$ corresponds to a minimum length-bounded cut and a maximum length-bounded

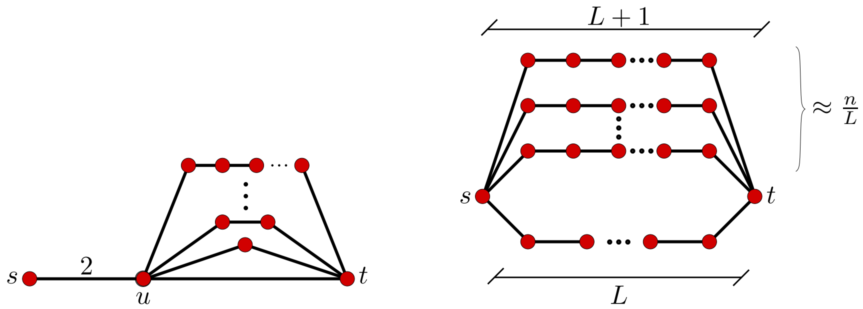

Fig. 5. Left: Graph family showing that the performance ratio is asymptotically tight. Except su all edges have capacity 1. Right: Example of the $\frac{n}{L}$ gap between the standard and the length-bounded cut.

flow in G. The theorem follows from standard flow theory. For undirected graphs we replace each edge by two antiparallel directed edges with the capacity and length of the original edge. The modified Dijkstra algorithm again yields a DAG and any cut or flow in this DAG directly translates into a length-bounded cut or flow in the original graph. $\square$

For suitable length functions, like unit-edge-lengths, we obtain from Theorem 5 an approximation algorithm for the minimum length-bounded cut problem with a performance ratio and complexity depending on the length-bound. For unit-edge-lengths a restricting length-bound is less than the graph's node cardinality and thus polynomially bounded by the input size.

Corollary 2. *In (un-)directed multi-graphs one can find an $(L+1-\mathrm{dist}(s,t))$-approximation to the minimum L-length-bounded cut by at most $L+1-\mathrm{dist}(s,t)$ standard minimum cut calculations.*

Proof. Removing a minimum $\mathrm{dist}(s,t)$-length-bounded cut from the graph increases the distance of s and t by at least 1. Repeating this iteratively increases the s-t-distance to $L+1$ within at most $L+1-\mathrm{dist}(s,t)$ iterations. Since each intermediate cut is minimum an optimal L-length-bounded cut cannot be smaller than any of them. Thus, their union has a size at most $L+1-\mathrm{dist}(s,t)$ times the optimal L-length-bounded cut value. $\square$

The performance ratio bound is tight, as shown by the example in Figure 5 (left). The next theorem establishes bounds on the absolute difference between the sizes of standard minimum cuts and length-bounded minimum cuts.

Theorem 6. *Let $G=(V,E)$ be a (un-)directed multi-graph. A minimum node-cut in G is larger than a minimum length-bounded node-cut by at most $\frac{n}{L}$. If G is a simple graph, a minimum edge-cut is larger than a minimum length-bounded edge-cut by at most $\mathcal{O}(\frac{n^2}{L^2})$.*

Proof. Our arguments apply to directed and undirected graphs in the same way, so we do not distinguish between them.

First, consider the case of node-cuts. Note that the size of a minimum node-cut is equal to the maximum number of node-disjoint s-t-paths by Menger's theorem. Let OPT denote the size of a minimum length-bounded node-cut. Let C^* be a minimum node-cut. We will construct a node-cut C of size at most $\mathrm{OPT}+\frac{n}{L}$. This implies $|C^*| \leq \mathrm{OPT}+\frac{n}{L}$.

Let C_1 be an optimal length-bounded node-cut, $|C_1| = \mathrm{OPT}$. In $G \setminus C_1$, all s-t-paths have length at least $L+1$. The number of node-disjoint s-t-paths in $G \setminus C_1$ is at most

$(n-2)/L \leq n/L$, as each such path contains at least L internal nodes and no two such paths contain the same node. Therefore, a minimum node-cut in $G \setminus C_1$ has cardinality at most n/L. Let C_2 be such a cut. Then $C = C_1 \cup C_2$ is a node-cut in G of the desired cardinality.

Now, consider the case of edge-cuts. Note that the size of a minimum edge-cut is equal to the maximum number of edge-disjoint s-t-paths by the edge version of Menger's theorem. Let OPT denote the size of a minimum length-bounded edge-cut. Let C^* be a minimum edge-cut. We will construct an edge-cut C of size at most OPT $+\mathcal{O}(\frac{n^2}{L^2})$. This implies $|C^*| \leq$ OPT $+\mathcal{O}(\frac{n^2}{L^2})$.

Let C_1 be an optimal length-bounded edge-cut, $|C_1| = \text{OPT}$. In $G \setminus C_1$, all s-t-paths have length at least $L+1$. It follows from [20] that if a (directed or undirected) simple graph contains k edge-disjoint s-t-paths, the shortest of these has length $\mathcal{O}(n/\sqrt{k})$. Therefore, if all s-t-paths have length at least $L+1$, we know that $L+1 = \mathcal{O}(n/\sqrt{k})$ and thus $k = \mathcal{O}(n^2/L^2)$. Consequently, the number of edge-disjoint s-t-paths in $G \setminus C_1$ is at most $\mathcal{O}(n^2/L^2)$. Therefore, a minimum edge-cut in $G \setminus C_1$ has cardinality at most $\mathcal{O}(n^2/L^2)$. Let C_2 be such a cut. Then $C = C_1 \cup C_2$ is an edge-cut in G of the desired cardinality. $\square$

Figure 5 (right) gives an example showing that the bound of $\frac{n}{L}$ on the gap between standard and length-bounded node-cuts given in Theorem 6 is tight. In this example, s and t are connected by one path of length L and by $\frac{n-L-1}{L} \approx \frac{n}{L}$ paths of length $L+1$. A minimum L-bounded node-cut has size one, while the minimum standard node-cut needs to cut all paths and has size approximately $\frac{n}{L}$. As minimum cuts can be computed in polynomial time, Theorem 6 leads to the following corollary.

Corollary 3. *For (un-)directed multi-graphs $G = (V, E)$ there exists an $\mathcal{O}(\frac{n}{L})$-approximation algorithm for the min. length-bounded node-cut problem. For (un-)directed simple graphs $G = (V, E)$ there exists an $\mathcal{O}(\frac{n^2}{L^2})$-approximation algorithm for the min. length-bounded edge-cut problem.*

Now we show that there are approximation algorithms with ratio $\mathcal{O}(\sqrt{n})$ for length-bounded node-cuts and with ratio $\mathcal{O}(\sqrt{m})$ for length-bounded edge-cuts. This then gives the following theorem.

Theorem 7. *For (un-)directed graphs $G = (V, E)$ there exists an $\mathcal{O}(\min\{L, n/L, \sqrt{n}\})$-approximation algorithm for the minimum length-bounded node-cut problem and an $\mathcal{O}(\min\{L, n^2/L^2, \sqrt{m}\})$-approximation algorithm for the minimum length-bounded edge-cut problem.*

Proof. The upper bounds of $\min\{L, n/L\}$ in the node case and $\min\{L, n^2/L^2\}$ in the edge case follow from Corollaries 2 and Corollary 3. Furthermore, we have $\min\{L, n/L\} \leq \sqrt{n}$, so the claimed ratio for length-bounded node cuts follows directly.

It remains to show that ratio $\mathcal{O}(\sqrt{m})$ can be achieved for length-bounded edge-cuts. The algorithms for directed and undirected graphs are essentially the same, so we do not need to distinguish these cases.

We use the following approach. Let OPT denote the size of a smallest length-bounded edge-cut. If $L \leq \sqrt{m}$, we simply apply the algorithm from Corollary 2. If $L > \sqrt{m}$, we repeatedly find an s-t-path of length at most $\lceil\sqrt{m}\rceil$, add all its edges to the cut, and remove these edges from the graph. This process ends when no such path exists in the remaining graph. Let C_1 denote the set of edges added to the cut in this process. If $G \setminus C_1$ does not contain an s-t-path of length at most L, we output C_1. Otherwise, we compute a minimum edge-cut C in G and output that as the solution.

We analyze the algorithm as follows. Let OPT denote the size of an optimal length-bounded edge-cut. Let k be the number of iterations the algorithm has executed in the computation of C_1. The number of edges added to the cut C_1 by the algorithm is at most $k\lceil\sqrt{m}\rceil$. Furthermore, the optimal length-bounded edge-cut must contain at least k edges, since there are k edge-disjoint paths of length at most $\lceil\sqrt{m}\rceil$ in G. Therefore, $|C_1| \leq (\sqrt{m}+1)\cdot\text{OPT}$. If $G\setminus C_1$ does not contain an s-t-path of length at most L, the algorithm outputs C_1 and produces a $(\sqrt{m}+1)$-approximation. Otherwise, the algorithm computes a minimum edge-cut C in G and output that as the solution. We claim that C is an $\mathcal{O}(\sqrt{m})$-approximation in this case. We show this by proving the existence of an edge-cut of cardinality at most $|C_1| + \sqrt{m}$. Let V_i denote the set of nodes at distance i from s in $G \setminus C_1$. Note that the distance from s to t in $G \setminus C_1$ is at least $\lceil\sqrt{m}\rceil + 1$. Let E_i be the set of edges in $G \setminus C_1$ with tail in V_i and head in V_{i+1}. Let j be such that E_j has minimum cardinality among the sets E_i for $0 \leq i \leq \lceil\sqrt{m}\rceil - 1$. Observe that $|E_j| \leq m/\lceil\sqrt{m}\rceil \leq \sqrt{m}$. Hence, $C_1 \cup E_j$ is an edge-cut of cardinality at most $|C_1| + \sqrt{m} \leq 2\sqrt{m}\cdot\text{OPT}$. □

4 Length-Bounded Flows

4.1 Edge-Based vs. Path-Based Flows: Complexity

Choosing infinity as a length-bound reduces the length-bounded flow problem to the corresponding standard flow problem. In most cases one does not use the linear program in (2) for standard flows since the number of paths and thus the number of variables may be exponential in the input size. It is more common to use an edge-based formulation, since this always uses a polynomially bounded number of variables. However, for length-bounded flows we do not know an edge-flow formulation.

When looking at a given length-bounded flow, we can infer from linear programming theory the existence of a corresponding path-decomposition of small size, where all paths fulfil the length-bound.

Proposition 1. *Given a length-bounded (multi-commodity) path-flow in a graph with m edges. There exists a length-bounded (multi-commodity) path-flow with the same length bound and the same flow value per edge and commodity, that uses at most m paths for each commodity.*

The proof of Proposition 1 follows from the fact, that the linear program in (2) has only m linear constraints. Therefore, the rank of the linear program for a single commodity is at most m. Consequently, there has to be a solution using no more than m paths. We can modify the edge-capacities appropriately and apply this argument to each commodity one after another. The same argument can simultaneously be applied to all commodities of a length-bounded flow. Hence, provided a suitable linear program, a length-bounded flow always possesses an optimal path-flow solution using no more than $|E| + k$ paths in total. However, this transformation in general changes the edge-flow values of the individual commodities with respect to the given flow. Consider the graph in Figure 6.

As shown above, the theory of linear programming can be used to show that there is always a path-flow of maximum flow value which has a small size. Nevertheless, linear programming cannot be used to find maximum fractional length-bounded flows efficiently, unless $\mathcal{P} = \mathcal{NP}$.

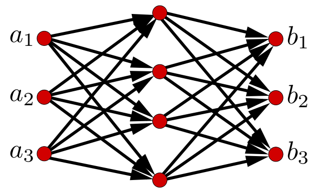

Fig. 6. The graph shown possesses no small multi-commodity path-flow corresponding to a prescribed edge-flow. As set of terminals we choose all 9 pairs (a_i, b_j), $i, j = 1, 2, 3$. The capacity of all 24 edges is set to 1. For each commodity (a_i, b_j), the prescribed edge-flow for this commodity assigns to all edges with a_i as tail node or b_j as head node a flow value of 1/3 and to the remaining edges the value 0 is assigned. This is a feasible multi-commodity edge-flow sending $\frac{4}{3}$ flow units per commodity. However, there is only one path-flow which corresponds to this edge-flow. This path-flow has to use for each terminal pair (a_i, b_j) all four a_i-b_j-paths. Hence, the path-flow uses all 36 paths which is greater than $|E| + k = 33$.

Theorem 8. *For a single-commodity length-bounded flow problem in an undirected outer-planar graph it is $\mathcal{NP}$-complete to decide whether there is a fractional length-bounded flow of given flow value.*

Proof (Sketch.). To prove the theorem one can show that 2-Partition can be reduced to the integral length-bounded flow problem for a flow of value 2. In a second step one shows that a fractional flow of value 2 in this special graph induces an integral flow of value 2. □

Finding a maximum length-bounded flow is computationally more difficult than finding a standard maximum flow. As we have mentioned already, standard flows are usually modeled as edge-flows. Each flow in a path formulation can be transformed into an edge-flow. For standard flows the reverse transformation is also possible and only a polynomial number of paths are needed, see [21]. If length-bounds are present, one may try to use an edge-flow formulation, too. But, as the following theorem shows, edge- and path-flows are not polynomially equivalent for length-bounded flows. More precisely, even if we are given an edge-based flow and we know that there is a length-bounded path-decomposition of it, it is hard to find an path-decomposition. This can be shown by a reduction of 2-Partition and is stated in the following Theorem.

Theorem 9. *Unless $\mathcal{P} = \mathcal{NP}$, there is no polynomial algorithm to transform an edge-flow which is known to correspond to a length-bounded path-flow into a length-bounded path-flow, even if the graph is outer-planar.*

4.2 Structure of Optimal Solutions and Integrality Gap

So far, we mostly considered the difference between standard and length-bounded flows with respect to complexity issues. Now we turn our attention to the structure of optimal solutions. For standard single-commodity flows with integral capacities there is always an integral maximum flow. Again, the situation is completely different in the presence of length constraints. We will not only show that there need not exist an integral maximum flow, but also that there are instances where each maximum flow ships a large percentage of the flow along paths with very small flow values. The fractionality of these maximum flows can be chosen arbitrarily small.

Theorem 10. *There are unit-capacity outer-planar graphs of order n such that every maximum length-bounded flow ships more than one half of the total flow along paths with flow values $\mathcal{O}(1/n)$.*

Proof. We construct a family $\{G_k\}_{k\in\mathbb{N}}$ of unit-capacity and unit-length outer-planar graphs such that G_k has order $3k+4$ and a maximum fractional L_k-length-bounded s-t-flow of value less than 2, for a certain length-bound $L_k \in \Theta(k)$. The unique maximum L_k-length-bounded flow in G_k contains $k+1$ paths each with flow value $\frac{1}{k+1}$.

The graph G_k consists of a sequence of $k+1$ triangles preceded by a path of length $k+1$ and a single edge that is parallel to the path; see Figure 7.

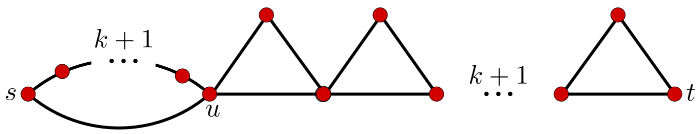

Fig. 7. Graph G_k in which each maximum length-bounded flow has to send more than one half of the flow along paths with small flow values.

In G_k we consider a maximum fractional $(2k+2)$-length-bounded s-t-flow, i. e., $L_k = 2k+2$. There is only one s-t-path $\widetilde{P}$ of length at most $2k+2$ that contains the s-u-path of length $k+1$. Indeed, this path has length exactly $2k+2$ and contains the unique shortest u-t-path. To obtain a total flow value larger than 1, path $\widetilde{P}$ has to be used. For simplicity we call the edges in the shortest u-t-path *ground edges.*

All s-t-paths of length at most $2k+2$ except $\widetilde{P}$ contain the edge su and at least one of the ground edges. Consider the s-t-paths of length exactly $2k+2$ that contain edge su. There are $k+1$ of those paths, one corresponding to each ground edge. Routing a fraction of $\frac{1}{k+1}$ units along each of them yields a feasible flow of value 1. Each ground edge is contained in exactly one of these paths and has therefore a residual capacity of $1-\frac{1}{k+1}$. Thus, along path $\widetilde{P}$ we can route further $1-\frac{1}{k+1}$ units of flow and obtain a feasible $2k+2$-length-bounded s-t-flow of value $2-\frac{1}{k+1}$. We claim that this flow is maximum and unique.

Sending 1 unit of flow along path $\widetilde{P}$ blocks each other path containing a ground edge, i. e., each further feasible s-t-path. Assume, $1-\delta$ units of flow are sent along path $\widetilde{P}$, for an arbitrary $0 < \delta < 1$. Then all remaining paths have a flow value not greater than δ each and thus altogether at most $\min\{1,(k+1)\delta\}$. Therefore, the maximum flow value dependent on δ is $h(\delta) := 1-\delta+\min\{1,(k+1)\delta\}$. This function $h(\delta)$ reaches its unique maximum for $0 < \delta < 1$ at $\delta = \frac{1}{k+1}$. Hence, $2-\frac{1}{k+1}$ is the maximum fractional s-t-flow value for the given length-bound and the above constructed flow is unique. □

In Section 4.1 we showed (in contrast to standard single-commodity flows) it is $\mathcal{NP}$-hard to find a maximum length-bounded path-flow even if an edge-flow corresponding to a maximum length-bounded path-flow is given. For integral length-bounded flows there is a structural difference between path- and edge-flows which is stated in Theorem 11.

Theorem 11. *An integral (maximum) edge-flow corresponding to a (fractional) length-bounded flow in an undirected graph with unit-edge-lengths does not need to have an integral path decomposition.*

Proof. Consider the graph in Figure 8. All edges except vt have length and capacity 1, edge vt has capacity 3 and length 1. The length-bound is 6. Assume there is an integral 6-length-bounded s-t-flow of value 4. Then all edges except vt must have flow value 1 and vt has flow value 3. Since s and v have distance 3, each s-t-path not using edge vt must contain one of the two shortest s-v-paths. An integral 6-length-bounded s-t-flow of value 4 must send one unit of flow along this path. Assume this path uses the upper half of the graph. Then each additional path in the upper half of the graph has length 7 and is therefore infeasible. Thus, no integral 6-length-bounded s-t-flow has value 4.

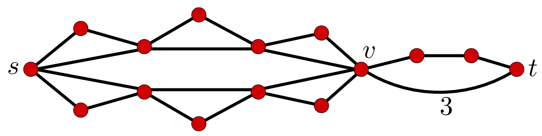

Fig. 8. The graph shown possesses an integral edge-flow that corresponds to a maximum fractional 6-length-bounded path-flow but which has no integral 6-length-bounded path-flow.

It remains to show, that there is a fractional 6-length-bounded s-t-flow of value 4. Along each of the two feasible paths not using edge vt send half a unit of flow. All remaining paths contain edge vt and may use up to two detours from the shortest s-v-paths. Consider only paths using exactly two detours. There are three of them using the upper left side part and three using the lower left side part of the graph. In each triple two of them share a detour and no edge from the shortest s-v-paths. It is feasible to send along each path half a unit of flow. Thus there is a half-integral 6-length-bounded s-t-flow of value $\frac{8}{2} = 4$. $\square$

Hence Theorem 11 shows that, even if there is a length-bounded decomposition of a maximum (non-bounded) flow, an integral length-bounded decomposition for this flow does not have to exist.

In [12] it was shown that the length-bounded flow problem can be approximated within arbitrarily precision. Having this in mind, it becomes interesting how far the value of such a fractional solution is away from the corresponding integral solution. Theorem 12 gives a lower bound on this value.

Theorem 12. *For unit-capacity graphs with n nodes, the integrality gap of the integer program in (2) can be of order $\Omega(\sqrt{n})$ even for unit-edge-lengths and planar graphs. The length-bound used is of order $\Theta(\sqrt{n})$.*

Proof. We construct a graph with n nodes, integral length-bounded flow value 1, and half-integral flow value of order $\Omega(\sqrt{n})$. The construction is inspired by Guruswami et al. [13].

The basic structure is half of a k by k grid, see Figure 9. There are k nodes $s_1, \ldots, s_k$ vertically, all connected to the source s and k nodes $t_1, \ldots, t_k$ horizontally, all connected to the sink t. Each grid node (dashed ellipse) is split into two nodes connected by a single edge. Furthermore, there are diagonal edges connecting the "rightmost grid nodes" of each pair of successive rows as shown in the figure. The new adjacency for the split nodes is also illustrated in the figure. All edges have capacity 1. We assign to all diagonal and splitting edges (dashed) length 0 and 1 to the remaining edges not adjacent to s or t. The edges ss_i and tt_{k-i} get length i for $i = 1, \ldots, k$. As length-bound we choose $L := 2k - 1$.

Consider an s-t-path P of length-bounded by L. Assume P does not contain a diagonal edge. Let ss_i and t_jt be edges of P. Thus, P has to skip $k+1-i$ rows and j columns from s_i to reach t_j. If P does not use a diagonal edge of length 0 it has to pay for each skip at least 1

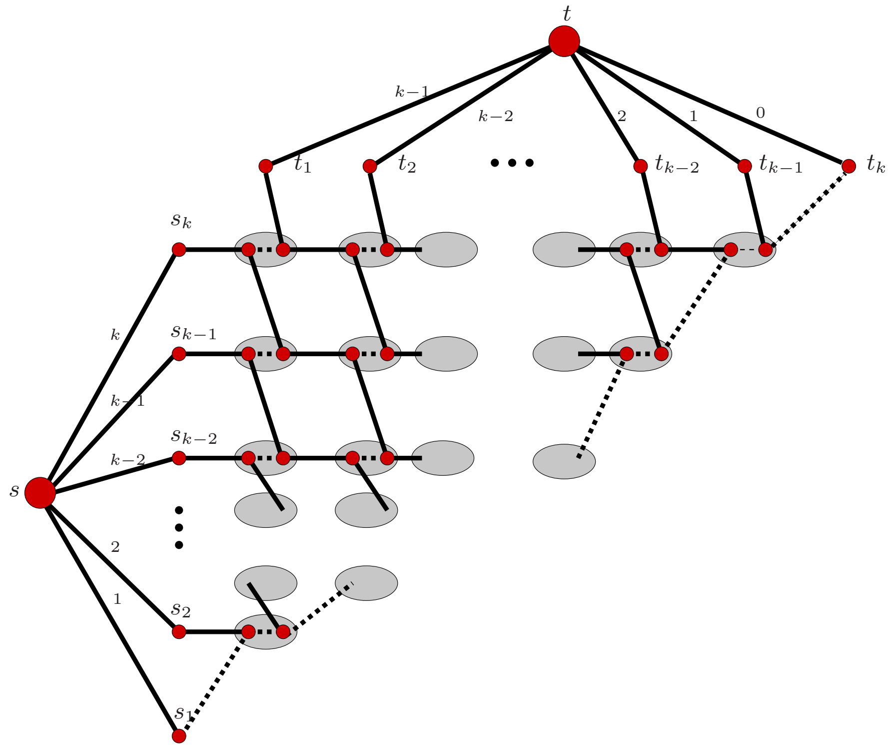

Fig. 9. Example-graph for a large integrality gap for the maximum integral length-bounded s-t-flow.

length unit. Together with the lengths of the edges ss_i and t_jt path P must have length at least $i + (k + 1 - i) + j + (k - j) = 2k + 1 > L$, a contradiction.

Each L-length-bounded s-t-path must contain a diagonal edge. Therefore, there are no two edge disjoint s-t-paths having a length bounded by L.

The i-th canonical path via ss_i horizontally to the row end, one diagonal edge, and vertically up to t via t_it has length exactly L. Since each pair of such canonical paths shares a different single edge we can feasibly sent along each of them half a unit. That is, there is a fractional L-length-bounded s-t-flow of value $k/2$.

Since there is no pair of edge-disjoint L-length-bounded s-t-paths the gap between a maximum integral and a maximum half integral flow is at least $k/2$. Since k is of size $\Theta(\sqrt{n})$, this shows the lemma for integral edge-lengths. If we subdivide each edge of length ℓ into a path of length $\ell + 1$, we increase the number of nodes by a constant factor only and obtain the same result for unit-edge-lengths. $\square$

The big integrality gap in Theorem 12 is tied to the unit-capacities of the graph used in the proof. In fact, rising the edge capacities in this graph up to 2 brings the integrality gap down to 2. Indeed, the integrality gap is constant for high capacity graphs. This can be shown by randomized rounding, a constructive technique introduced by Raghavan and Thompson [22].

Theorem 13. *Consider a graph with minimal edge-capacity of at least* $c \log |E|$*, for a suitable constant* c*. Using randomized rounding one can convert a (maximum) fractional solution into an integral solution, which is feasible and has a value that is at most a constant factor smaller with high probability. In particular, the integrality gap is constant for high capacity graphs.*

References

1. Menger, K.: Zur allgemeinen Kurventheorie. Fundamenta Mathematicae (1927) 96–115
2. Dantzig, G.B., Fulkerson, D.R.: On the max flow min cut theorem of networks. In Kuhn, H.W., Tucker, A.W., eds.: Linear Inequalities and Related Systems. Volume 38 of Annals of Math. Studies. Princeton University Press, Princeton, New Jersey (1956) 215–221
3. Kotzig, A.: Connectivity and Regular Connectivity of Finite Graphs. PhD thesis, Vysoká Škola Ekonomická, Bratislava (1956)
4. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canadian Journal of Mathematics **9** (1956) 399–404
5. Elias, P., Feinstein, A., Shannon, C.E.: A note on the maximum flow through a network. IRS Trans. Inf. Th. IT **2** (1956)
6. Lovász, L., Neumann Lara, V., Plummer, M.D.: Mengerian theorems for paths of bounded length. Periodica Mathematica Hungarica **9** (1978) 269–276
7. Exoo, G.: On line disjoint paths of bounded length. Discrete Mathematics **44** (1983) 317–318
8. Niepel, L., Safaríková, D.: On a generalization of Menger's Theorem. Acta Mathematica Universitatis Comenianae **42** (1983) 275–284
9. Bondy, J.A., Murty, U.: Graph Theory with Applications. North Holland (1976)
10. Boyles, S.M., Exoo, G.: A counterexample to a conjecture on paths of bounded length. J. of Graph Theory **6** (1982) 205–209
11. Itai, A., Perl, Y., Shiloach, Y.: The complexity of finding maximum disjoint paths with length constraints. Networks **12** (1982) 277–286
12. Fleischer, L., Skutella, M.: The quickest multicommodity flow problem. In: IPCO. LNCS 2337, Springer (2002) 36–53
13. Guruswami, V., Khanna, S., Rajaraman, R., Shepherd, B., Yannakakis, M.: Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. In: Proceedings of the 31st Annual ACM Symposium on Theory of Computing. (1999) 19–28
14. Mahjoub, A.R., McCormick, S.T.: The complexity of max flow and min cut with bounded-length paths. Manu. (2003)
15. Perl, Y., Ronen, D.: Heuristics for finding a maximum number of disjoint bounded paths. Networks **14** (1984) 531–544
16. Brandes, U., Neyer, G., Wagner, D.: Edge-disjoint paths in planar graphs with short total length. Technical Report 19, Universität Konstanz (1996)
17. Wagner, D., Weihe, K.: A linear-time algorithm for edge-disjoint paths in planar graphs. Combinatorica **15**(1) (1995) 135–150
18. Hsu, D.: On container width and length in graphs, groups, and networks. IEICE Trans. Fundamental **E77-A**(4) (1994)
19. Dinur, I., Safra, S.: On the hardness of approximating minimum vertex cover. Annals of Math. **162**(1) (2005) 439–486
20. Galil, Z., Yu, X.: Short length versions of Menger's Theorem (extended abstract). In: Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, ACM (1995) 499–508
21. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows : Theory, Algorithms, and Applications. Prentice Hall, New Jersey, NJ, USA (1993)
22. Raghavan, P., Thompson, C.: Randomized rounding: A technique for provably good algorithms and algorithmic proofs. Combinatorica **7** (1987) 365–374