

On Algorithms for L -bounded Cut Problem^{*}

Petr Kolman

Department of Applied Mathematics
Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
kolman@kam.mff.cuni.cz

Abstract. Given a graph $G = (V, E)$ with two distinguished vertices $s, t \in V$ and an integer parameter $L > 0$, an L -bounded cut is a subset F of edges (vertices) such that the every path between s and t in $G \setminus F$ has length more than L . The task is to find an L -bounded cut of minimum cardinality.

Though the problem is very simple to state and has been studied since the beginning of the 70's, it is not much understood yet. The problem is known to be \mathcal{NP} -hard to approximate within a small constant factor even for $L \geq 4$ (for $L \geq 5$ for the vertex cuts). On the other hand, the best known approximation algorithm for general graphs has approximation ratio only $\mathcal{O}(n^{2/3})$ in the edge case, and $\mathcal{O}(\sqrt{n})$ in the vertex case, where n denotes the number of vertices.

We show that for planar graphs, it is possible to solve both the edge- and the vertex-version of the problem optimally in time $\mathcal{O}(L^{3L}n)$. That is, the problem is fixed parameter tractable (FPT) with respect to L on planar graphs. Furthermore, we show that the problem remains FPT even for bounded genus graphs, a super class of planar graphs.

Our second contribution deals with approximations of the vertex version of the problem. We describe an algorithm that for a given a graph G , its tree decomposition of treewidth τ and vertices s and t computes a τ -approximation of the minimum L -bounded $s - t$ vertex cut; if the decomposition is not given, then the approximation ratio is $\mathcal{O}(\tau\sqrt{\log \tau})$. For graphs with treewidth bounded by $\mathcal{O}(n^{1/2-\epsilon})$ for any $\epsilon > 0$, but not by a constant, this is the best approximation in terms of n that we are aware of.

1 Introduction

The subject of this paper is a variation of the classical $s-t$ cut problem, namely the *minimum L -bounded edge (vertex) cut problem*: given a graph $G = (V, E)$ with two distinguished vertices $s, t \in V$ and an integer parameter $L > 0$, find a subset F of edges (vertices) of minimum cardinality such that every path between s and t in $G \setminus F$ has length more than L .

^{*} This research was partially supported by project GA15-11559S of GA ČR.

The problem has been studied in various contexts since the beginning of the 70's (e.g., [AK71,LNP78,BEH⁺10]) and occasionally it appears also under the name the *short paths interdiction problem* [KBB⁺08].

Closely related is another \mathcal{NP} -hard problem, namely the *shortest path most vital edges and vertices problem* (e.g. [BGV89,BNKS95,BNN15]): given a graph G , two distinguished vertices s and t and an integer k , the task is to find a subset F of k edges (vertices) whose removal maximizes the increase in the length of the shortest path between s and t . If we introduce an additional parameter – the desired minimum distance of s and t – we obtain a parameterized version of the L -bounded cut problem: given a graph G , two distinguished vertices s and t and integers k and L , does there exist a subset F of at most k edges (vertices) such that every path between s and t in $G \setminus F$ has length more than L ? We also note that \mathcal{NP} -hardness of the shortest path most vital edges (vertices) problem immediately implies \mathcal{NP} -hardness of the L -bounded edge (vertex) cut problem, and vice versa.

In contrast to many other cut problems on graphs (e.g., multiway cut, multicut, sparsest cut, balanced cut, maximum cut, multiroute cut), the known approximations of the minimum L -bounded cut problem are substantially weaker. In this work we focus on algorithms for restricted graph classes, namely planar graphs, bounded genus graphs and graphs with bounded, yet not constant, treewidth, and provide new results for the L -bounded cut problem on them; the results for planar graphs solve one of the open problems suggested by Bazgan et al. [BNN15]. We also remark that the generic approximation scheme of Czumaj et. al [CHLN05] for \mathcal{NP} -hard problems in graphs with superlogarithmic treewidth is not applicable for the L -bounded cut problem.

Related Results. \mathcal{NP} -hardness of the shortest path most vital edges problem (and, thus, as noted above, also of the L -bounded cut problem) was proved by Bar-Noy et al. [BNKS95]. The best known approximation algorithm for the minimum L -bounded cut problem on general graphs has approximation ratio only $\mathcal{O}(\min\{L, n/L\}) \subseteq \mathcal{O}(\sqrt{n})$ for the vertex case and $\mathcal{O}(\min\{L, n^2/L^2, \sqrt{m}\}) \subseteq \mathcal{O}(n^{2/3})$ for the edge case, where m denotes the number of edges and n the number of vertices [BEH⁺10]. On the lower bound side, the edge version of the problem is known to be \mathcal{NP} -hard to approximate within a factor of 1.1377 for $L \geq 4$, and the vertex version for $L \geq 5$ [BEH⁺10]; for smaller values of L the problem is solvable in polynomial time [LNP78,MM10]. Independently, Khachiyan et al. [KBB⁺08] proved that a version of the problem with edge lengths

is \mathcal{NP} -hard to approximate within a factor smaller than 1.36. Recently, assuming the Unique Games Conjecture, Lee proved that the problem is \mathcal{NP} -hard to approximate within *any* constant factor [Lee17].

An instance of the L -bounded cut problem on a graph of treewidth τ can be cast as an instance of constraint satisfaction problem (CSP) with domain of size $L + 1$ and treewidth τ^1 . As CSP instances with treewidth bounded by τ and domain by D can be solved in time $O(D^\tau n)$ [Fre90] (when a tree decomposition of treewidth τ of the constraint graph is given), the problem is fixed parameter tractable with respect to L and τ . Dvořák and Knop [DK15] provide a direct proof of the same result with a slightly worse dependence on L and τ ; they also prove that the problem is $W[1]$ -hard when parameterized by the treewidth only.

From the point of view of parameterized complexity, the problem was also studied by Golovach and Thilikos [GT11], Bazgan et al. [BNN15] and by Fluschnik et al. [FHNN16].

For planar graphs, the problem is known to be \mathcal{NP} -hard [FHNN15,PS16], too, and the problem has no polynomial-size kernel when parameterized by the combination of L and the size of the optimal solution [FHNN16].

For more detailed overview of other related results and applications, we refer to the papers [KBB⁺08,BEH⁺10,MM10].

Our Contribution. We show that on planar graphs, both the edge- and the vertex-version of the problem are solvable in time $\mathcal{O}(L^{3L}n)$. That is, we show that on planar graphs the minimum L -bounded cut problem is fixed parameter tractable (FPT) with respect to L . Furthermore, we show that the problem remains FPT even for bounded genus graphs, a super class of planar graphs. This is in contrast with the situation for general graphs – the problem is NP-hard even for $L = 4$ and $L = 5$, for the edge- and vertex-versions, respectively.

Our second contribution is a τ -approximation algorithm for the vertex version of the problem, if a tree decomposition of width τ is given. If the decomposition is not given, then using the currently best known tree decomposition algorithms, we obtain an $\mathcal{O}(\tau\sqrt{\log \tau})$ -approximation for general graphs with treewidth τ , and an $\mathcal{O}(\tau)$ -approximation for planar graphs, graphs excluding a fixed minor and graphs with treewidth bounded by $\mathcal{O}(\log n)$. For graphs with treewidth bounded by $\tau = \mathcal{O}(n^{1/2-\epsilon})$ for any $\epsilon > 0$, but not by a constant, in terms of n , this is the best approximation we are aware of.

¹ For the sake of completeness, in Appendix A we provide details about this reduction.

Our results are based on a combination of observations about the structure of L -bounded cuts and various known results. The proofs are straightforward but apparently non-obvious, considering the attention given to the problem in recent years.

2 L -Bounded Cut is FPT on Planar Graphs and Bounded Genus Graphs

Throughout the paper, given a graph $G = (V, E)$ and $u, v \in V$, we use $d(u, v)$ to denote the shortest path distance between u and v , that is, the number of edges on the shortest path.

Our main tools are the following two well-known results.

Theorem 1 (Robertson and Seymour [RS84], Bodlaender [Bod88]). *The treewidth of a planar graph with radius d is at most $3d$.*

Theorem 2 (Freuder [Fre90]). *CSP instances with treewidth bounded by τ , domain by D and size by n are solvable in time $O(D^\tau n)$.*

Since the L -bounded cut problem on a graph of treewidth τ can be cast as a CSP instance with treewidth τ , domain of size $L + 1$ and size n , the minimum L -bounded cut problem is solvable in time $O(L^\tau n)$, as already stated in the introduction and explained in Appendix A.

The main result of this section says that the L -bounded cut problem on planar graphs is fixed parameter tractable, with respect to the parameter L .

Theorem 3. *The minimum L -bounded edge (vertex) cut problem on planar graphs is solvable in time $O(L^{3L}n)$.*

Proof. We prove the theorem for the edge version; the proof for the vertex version is analogous.

Given a graph $G = (V, E)$, $s, t \in V$ and an integer L , let $V' = \{v \in V \mid d(s, v) + d(t, v) \leq L\}$. In words, V' is the subset of vertices lying on paths of length at most L between s and t . Without loss of generality we assume that $d(s, t) \leq L$ – otherwise the problem is trivial. Let G' be the subgraph of G induced by V' . Note that the radius of G' is at most L as, by definition, $d(s, v) \leq L$ for every $v \in V'$.

The set V' (and, thus, the subgraph G') can be computed using a linear time algorithm for single-source shortest paths [KRRS94] that we run twice, once for s and once for t . Note that both s and t belong to V' .

Obviously, G' is a planar graph, and by Theorem 1, its treewidth is at most $3L$. We solve the L -bounded problem for G' and s and t by Theorem 2. Let $F \subseteq E(G')$ be the optimal solution. We claim that F is an optimal solution for the original instance of the problem on G as well. To prove *feasibility* of F , assume, for contradiction, that there exists an $s - t$ -path p of length at most L in $(V, E \setminus F)$. As there is no such path in $G' \setminus F$, p has to use at least one vertex v from $V \setminus V'$. However, this yields a contradiction: on one hand, $d(s, v) + d(v, t) \leq L$ as v is on an $s - t$ -path of length at most L , on the other hand, $d(s, v) + d(v, t) > L$ as v is not in V' . Concerning the *optimality* of F , it is sufficient to note that the size of an optimal solution for the subgraph G' is a lower bound on the size of an optimal solution for G . \square

Theorem 1 was generalized by Eppstein [Epp00] to graphs of bounded genus and this result makes it possible to generalize Theorem 3 also to graphs of bounded genus.

Theorem 4 (Eppstein [Epp00]). *There exists a constant \hat{c} such that the treewidth of every graph with radius d and genus g is at most $\hat{c}dg$.*

In the same way as we used Theorem 1 to prove in Theorem 3 fixed parameter tractability of the L -bounded cut for planar graphs, we can use Theorem 4 to prove fixed parameter tractability of the L -bounded cut problem on graphs of bounded genus. Thus, we obtain the following theorem.

Theorem 5. *The minimum L -bounded edge (vertex) cut problem on graphs with genus g is solvable in time $O(L^{\hat{c}g}n)$.*

3 τ -approximation for L -bounded Vertex Cuts

In this section we describe an algorithm for the L -bounded vertex cut problem whose approximation ratio is parametrized by the treewidth of (a given tree decomposition of) the underlying graph. For notions related to the treewidth of a graph and tree decomposition we stick to the standard terminology as given in the book by Kloks [Klo94]. To distinguish vertices of a graph and of its tree decomposition, we call the vertices of the tree decomposition *nodes*.

Given a graph $G = (V, E)$ with two distinguished vertices $s, t \in V$, an integer parameter $L > 0$ and a rooted tree decomposition T of treewidth τ of the graph G , the L -bounded cut is computed using the recursive procedure L -bounded.cut(G, T, s, t, L) described below. Throughout this

section we assume that the vertices s and t are not connected by an edge, as otherwise there is no L -bounded $s - t$ vertex cut in G .

For a node a of the tree decomposition T , we use $B(a) \subseteq V$ to denote the bag of the node a . In the description of the algorithm, $\text{mincut}(G, s, t)$ is a procedure that returns a minimum vertex $s - t$ cut in G ; given in addition a vertex set $C \subset V$, $\text{prune}(G, T, C)$ is a procedure that deletes from G the vertices in C and modifies the decomposition tree T by deleting these vertices from all bags. By $d(G, s, t)$ we denote the distance between s and t in G ; given a connected subtree R of a rooted tree T , a *deepest node* in R is a node in R with no child in R . Given a node b of the rooted tree decomposition T , we denote by T_b the subtree of T consisting of b and of all its descendants, and by G_b the subgraph of G induced by vertices in bags of T_b ; similarly, we denote by \bar{T}_b the subtree of T consisting of all nodes in T except for the descendants of b and by \bar{G}_b the subgraph of G induced by vertices in bags of \bar{T}_b .

Algorithm 1 L -bounded_cut(G, T, s, t, L)

```

1: if  $d(G, s, t) > L$  then return  $\emptyset$ 
2: else if  $\forall a \in V(T) : |B(a) \cap \{s, t\}| \leq 1$  then return  $\text{mincut}(G, s, t)$ 
3: else  $R \leftarrow \{a \in V(T) \mid s, t \in B(a) \text{ and } d(G_a, s, t) \leq L\}$ 
4:    $b \leftarrow$  a deepest node in  $R$ 
5:   if  $B(b) = \{s, t\}$  then
6:      $S_1 \leftarrow L$ -bounded_cut( $G_b, T_b, s, t, L$ )
7:      $S_2 \leftarrow L$ -bounded_cut( $\bar{G}_b, \bar{T}_b, s, t, L$ )
8:     return  $S_1 \cup S_2$ 
9:   else
10:     $(G', T') \leftarrow \text{prune}(G, T, B(b) \setminus \{s, t\})$ 
11:     $S' \leftarrow L$ -bounded_cut( $G', T', s, t, L$ )
12:    return  $S' \cup B(b) \setminus \{s, t\}$ 

```

Theorem 6. *Given a graph G , its rooted tree decomposition T of treewidth τ , vertices s and t and an integer L , the above algorithm finds in polynomial time a τ -approximation of the minimum L -bounded vertex cut.*

Proof. For notational simplicity, we use the term an L -bounded path for a path of length at most L , and we use T^s to denote the subtree of T induced by nodes with s in their bag, that is, induced by the set $\{a \in V(T) \mid s \in B(a)\}$; the subtree T^t is defined analogously.

We start by showing the *correctness* of the algorithm: we claim that the algorithm finds a feasible solution for the L -bounded vertex cut problem. The proof is by induction on the recursion depth. For the final re-

cursive calls, dealt with in steps 1 and 2, the claim is obvious from the description of the algorithm.

Inductive step. Consider a run of the procedure with a graph G and its tree decomposition T , and let R and b be the objects defined by the procedure in steps 3 and 4. We distinguish two cases (step 5): i) the bag $B(b)$ contains the two vertices s and t only, and, ii) the bag $B(b)$ contains in addition to s and t at least one other vertex.

In the first case, realizing that $\{s, t\}$ is a separator in G , we note that any L -bounded $s - t$ path in G either belongs entirely to G_b or belongs entirely to \bar{G}_b . Thus, $S_1 \cup S_2$ is an L -bounded $s - t$ cut in G .

In the second case, let G' be the subgraph defined by the procedure in step 10; recall that $G' = G \setminus (B(b) \setminus \{s, t\})$. As any L -bounded path in G either intersects the set $B(b) \setminus \{s, t\}$ or appears in G' , we conclude that the set $S' \cup B(b) \setminus \{s, t\}$ is an L -bounded $s - t$ cut in G .

For the sake of completeness we also note that every recursive call of the procedure is applied to a smaller graph, implying a finiteness of the algorithm; moreover, as the recursive calls in steps 6 and 7 are applied to almost disjoint subgraphs of G (the two subgraphs intersect in s and t only), the running time of the procedure is polynomial in the size of the input graph.

Now we turn to the *performance* of the algorithm. Let $cost(G)$ be the cost of the solution of our algorithm and let $opt(G)$ be the size of the optimal solution, for the graph G . Similarly as for the proof of correctness, we prove by induction on the recursion depth that $cost(G) \leq \tau \cdot opt(G)$ where τ is the treewidth of T .

We distinguish two cases of the final recursive call: i) for subgraphs with no L -bounded $s - t$ path, dealt with in step 1, the claim is obvious as $cost(G) = 0$ in this case; ii) for subgraphs such that the corresponding tree decomposition does not contain any bag with both s and t , dealt with in step 2, note that for any node c on the unique path connecting the disjoint subtrees T^s and T^t of the decomposition tree T , the set $B(c) \setminus \{s, t\}$ is an $s - t$ cut of size at most τ .

Inductive step. We start with a simple lemma.

Lemma 1. *The L -bounded paths in G_b are internally vertex disjoint with the L -bounded paths in G' .*

Proof. The choice of the node b in step 4 implies that every L -bounded $s - t$ path in G_b uses at least one node from $B(b) \setminus \{s, t\}$. Thus, there is no L -bounded $s - t$ path in $G_b \setminus (B(b) \setminus \{s, t\})$. As the vertex set $B(b)$ is a vertex cut (separator) in G that disconnects the subgraph $G_b \setminus B(b)$ of

G from the rest of the graph, we conclude that every $s - t$ path in G' is internally disjoint with every $s - t$ path in G_b . \square

As in the proof of correctness, we distinguish the two cases $B(a) = \{s, t\}$ and $B(a) \neq \{s, t\}$.

In the first case, exploiting the fact that $B(a) = \{s, t\}$ is a separator in G , we obtain $opt(G) = opt(G_b) + opt(\bar{G}_b)$. Plugging in the inductive assumptions, $cost(G_b) \leq \tau \cdot opt(G_b)$ and $cost(\bar{G}_b) \leq \tau \cdot opt(\bar{G}_b)$ and the earlier observed fact that $cost(G) \leq cost(G_b) + cost(\bar{G}_b)$, we derive that $cost(G) \leq \tau \cdot opt(G)$.

In the second case, from the description of the algorithm and the previous analysis we know that $cost(G) \leq cost(G') + \tau$. Exploiting the fact that by the choice of b there is at least one L -bounded $s - t$ path in G_b , Lemma 1 implies $opt(G) \geq 1 + opt(G')$. Combining these observations with the inductive assumption $cost(G') \leq \tau \cdot opt(G')$, we obtain again $cost(G) \leq \tau \cdot opt(G)$. This concludes the proof of the Theorem. \square

In the case that we are not given a tree decomposition on input, we just start by computing one using one of the best known tree decomposition algorithms. Feige et al. [FHL08] describe a polynomial time algorithm that yields, for a given graph of treewidth τ , a tree decomposition with treewidth $\mathcal{O}(\tau\sqrt{\log \tau})$; for planar graphs and for graphs excluding a fixed minor, the treewidth is $\mathcal{O}(\tau)$ only. Similarly, for graphs with treewidth bounded by $\mathcal{O}(\log n)$, Bodlaender et al. [BDD⁺16] describe how to find in polynomial time a tree decomposition of treewidth $\mathcal{O}(\tau)$. Depending on the input graph, one of these algorithms is used to obtain a desired tree decomposition. Thus, we obtain the following corollary.

Corollary 1. *There exists an $\mathcal{O}(\tau\sqrt{\log \tau})$ -approximation algorithm for the minimum L -bounded vertex cut on graphs with treewidth τ ; for planar graphs, graphs excluding a fixed minor and graphs with treewidth bounded by $\mathcal{O}(\log n)$, there exists an $\mathcal{O}(\tau)$ -approximation algorithm.*

4 Open problems

A natural open problem for planar graphs is whether the shortest path most vital edges (vertices) problem is fixed parameter tractable on them, with respect to the number k of deleted edges (vertices). Despite the close relation of the L -bounded cut problem and the shortest path most vital edges (vertices) problem, fixed parameter tractability of one of them does not seem to easily imply fixed parameter tractability of the other problem.

The τ -approximation for L -bounded vertex cuts is based on the fact that bags in a tree decomposition yield *vertex* cuts of size at most equal the width of the decomposition. Unfortunately, this is not the case for *edge* cuts – one can easily construct bounded treewidth graphs with no small balanced edge cuts. Thus, another open problem is to look for better approximation algorithms for minimum L -bounded *edge* cuts, for graphs with treewidth bounded by τ .

Yet another challenging and more general open problem is to narrow the gap between the upper and lower bounds on the approximation ratio of algorithms for L -bounded cut for general graphs.

Finally, we note that the L -bounded edge cut problem in a graph $G = (V, E)$ is a kind of a *vertex ordering* problem. We are looking for a mapping ℓ from the vertex set V to the set $\{0, 1, \dots, L, L + 1\}$ such that $\ell(s) = 0$, $\ell(t) = L + 1$ and the objective is to minimize the number of edges $\{u, v\} \in E$ for which $|\ell(u) - \ell(v)| > 1$; given a feasible solution $F \subseteq E$, the lengths of the shortest paths from s to all other vertices in $G \setminus F$ yield such a mapping of cost $|F|$. There are plenty of results dealing with *linear* vertex ordering problems where one is looking for a bijective mapping from the vertex set V to the set $\{1, 2, \dots, n\}$ minimizing some objective function. However, the requirement that the mapping be bijective to a set of size n seems crucial in the design and analysis of approximation algorithms for these problems. The question is whether it is possible to obtain good approximations for some nontrivial *non-linear* vertex ordering problems.

Acknowledgments. We thank Martin Koutecký and Hans Raj Tiwary for stimulating discussions and an anonymous referee for suggestions yielding an improvement by a factor $\log n$ in the approximation ratio in Theorem 6.

References

- [AK71] J. Adánek and V. Koubek. Remarks on flows in network with short paths. *Commentationes Mathematicae Universitatis Carolinae*, 12(4):661–667, 1971.
- [BDD⁺16] H. L. Bodlaender, P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshtanov, and M. Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016.
- [BEH⁺10] G. Baier, T. Erlebach, A. Hall, E. Köhler, P. Kolman, O. Pangrác, H. Schilling, and M. Skutella. Length-bounded cuts and flows. *ACM Trans. Algorithms*, 7(1):4:1–4:27, 2010. Preliminary version in Proc. of ICALP 2006.
- [BGV89] M. O. Ball, B. L. Golden, and R. V. Vohra. Finding the most vital arcs in a network. *Operations Research Letters*, 8(2):73 – 76, 1989.

- [BNKS95] A. Bar-Noy, S. Khuller, and B. Schieber. The complexity of finding most vital arcs and nodes. Technical Report CS-TR-3539, Univ. of Maryland, Dept. of Computer Science, November 1995.
- [BNN15] C. Bazgan, A. Nichterlein, and R. Niedermeier. A refined complexity analysis of finding the most vital edges for undirected shortest paths. In *Proc. of Algorithms and Complexity - 9th International Conference (CIAC)*, pages 47–60, 2015.
- [Bod88] H. L. Bodlaender. Planar graphs with bounded treewidth. Technical Report RUU-CS-88-14, Univ. Utrecht, Dept. of Computer Science, 1988.
- [CHLN05] A. Czumaj, M. M. Halldórsson, A. Lingas, and J. Nilsson. Approximation algorithms for optimization problems in graphs with superlogarithmic treewidth. *Information Processing Letters*, 94(2):49–53, 2005.
- [DK15] P. Dvořák and D. Knop. Parameterized complexity of length-bounded cuts and multi-cuts. In *Proc. of 12 Annual Conference on Theory and Applications of Models of Computation (TAMC)*, pages 441–452, 2015.
- [Epp00] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000.
- [FHL08] U. Feige, M. T. Hajiaghayi, and J. R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM J. Comput.*, 38(2):629–657, 2008. Preliminary version in Proc. of STOC 2005.
- [FHNN15] T. Fluschnik, D. Hermelin, A. Nichterlein, and R. Niedermeier. Fractals for kernelization lower bounds, with an application to length-bounded cut problems. *CoRR*, abs/1512.00333, 2015.
- [FHNN16] T. Fluschnik, D. Hermelin, A. Nichterlein, and R. Niedermeier. Fractals for kernelization lower bounds, with an application to length-bounded cut problems. In *Proc. of 43rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 25:1–25:14, 2016.
- [Fre90] E. C. Freuder. Complexity of K -tree structured constraint satisfaction problems. In *Proc. of the 8th National Conference on Artificial Intelligence*, pages 4–9, 1990.
- [GT11] P. A. Golovach and D. M. Thilikos. Paths of bounded length and their cuts: Parameterized complexity and algorithms. *Discrete Optimization*, 8(1):72–86, 2011.
- [KBB⁺08] L. Khachiyan, E. Boros, K. Borys, K. M. Elbassioni, V. Gurvich, G. Rudolf, and J. Zhao. On short paths interdiction problems: Total and node-wise limited interdiction. *Theory Comput. Syst.*, 43(2):204–233, 2008.
- [KK15] P. Kolman and M. Koutecký. Extended formulation for CSP that is compact for instances of bounded treewidth. *Electr. J. Comb.*, 22(4):P4.30, 2015.
- [Klo94] Ton Kloks. *Treewidth: Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994.
- [KRRS94] P. Klein, S. Rao, M. Rauch, and S. Subramanian. Faster shortest-path algorithms for planar graphs. In *Proc. of the 26th ACM Symposium on Theory of Computing (STOC)*, pages 27–37, 1994.
- [Lee17] E. Lee. Improved hardness for cut, interdiction, and firefighter problems. In *Proc. of 44rd International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 92:1–92:14, 2017.
- [LNP78] L. Lovász, V. Neumann-Lara, and M. D. Plummer. Mengerian theorems for paths of bounded length. *Periodica Mathematica Hungarica*, 9:269–276, 1978.

- [MM10] A. R. Mahjoub and S. T. McCormick. Max flow and min cut with bounded-length paths: complexity, algorithms, and approximation. *Math. Program*, 124(1-2):271–284, 2010.
- [PS16] F. Pan and A. Schild. Interdiction problems on planar graphs. *Discrete Applied Mathematics*, 198:215–231, 2016.
- [RS84] N. Robertson and P. D. Seymour. Graph minors. III. planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984.

A Appendix L -bounded Cut as a CSP Instance

An instance $Q = (V, \mathcal{D}, \mathcal{H}, \mathcal{C})$ of CSP [KK15] consists of

- a set of *variables* z_v , one for each $v \in V$; without loss of generality we assume that $V = \{1, \dots, n\}$,
- a set \mathcal{D} of finite *domains* $D_v \subseteq \mathcal{R}$ (also denoted $D(v)$), one for each $v \in V$,
- a set of *hard constraints* $\mathcal{H} \subseteq \{C_U \mid U \subseteq V\}$ where each hard constraint $C_U \in \mathcal{H}$ with $U = \{i_1, i_2, \dots, i_k\}$ and $i_1 < i_2 < \dots < i_k$, is a $|U|$ -ary relation $C_U \subseteq D_{i_1} \times D_{i_2} \times \dots \times D_{i_k}$,
- a set of *soft constraints* $\mathcal{C} \subseteq \{C_U \mid U \subseteq V\}$ where each soft constraint $C_U \in \mathcal{C}$ with $U = \{i_1, i_2, \dots, i_k\}$ and $i_1 < i_2 < \dots < i_k$, is a $|U|$ -ary relation $C_U \subseteq D_{i_1} \times D_{i_2} \times \dots \times D_{i_k}$.

For a vector $z = (z_1, z_2, \dots, z_n)$ and $U = \{i_1, i_2, \dots, i_k\} \subseteq V$ with $i_1 < i_2 < \dots < i_k$, we define the *projection of z on U* as $z|_U = (z_{i_1}, z_{i_2}, \dots, z_{i_k})$. A vector $z \in \mathcal{R}^n$ *satisfies the constraint $C_U \in \mathcal{C} \cup \mathcal{H}$* if and only if $z|_U \in C_U$. We say that a vector $z^* = (z_1^*, \dots, z_n^*)$ is a *feasible solution* for Q if $z^* \in D_1 \times D_2 \times \dots \times D_n$ and z^* satisfies every hard constraint $C \in \mathcal{H}$. In the *maximization* (*minimization*, resp.) version of CSP, the task is to find a feasible solution that maximizes (minimizes, resp.) the number of *satisfied* (unsatisfied, resp.) soft constraints; the *cost* of a feasible solution is the number of satisfied (unsatisfied, resp.) soft constraints.

The *constraint graph* of Q is defined as $H = (V, E)$ where $E = \{\{u, v\} \mid \exists C_U \in \mathcal{C} \cup \mathcal{H} \text{ s.t. } \{u, v\} \subseteq U\}$. We say that a CSP instance Q has *bounded treewidth* if the constraint graph of Q has bounded treewidth.

Given an L -bounded edge cut instance $G = (V, E)$ with $s, t \in V$ and an integer L , we construct the corresponding minimization CSP instance $Q = (V, \mathcal{D}, \mathcal{H}, \mathcal{C})$ as follows. The set of variables of Q coincides with the set V of vertices of G and for each $v \in V$, the corresponding domain D_v is $\{0, 1, \dots, L\}$. The set of hard constraints \mathcal{H} consists of two constraints, $C_{\{s\}} = \{0\}$ and $C_{\{t\}} = \{L\}$. The set of soft constraints \mathcal{C} contains a constraint

$$C_{(i,j)} = \{(0, 0), (0, 1), (L, L-1), (L, L)\} \cup \bigcup_{i=1}^{L-1} \{(i, i-1), (i, i), (i, i+1)\}$$

for each edge $\{i, j\} \in E$, $i < j$, of the graph G .

To see that a feasible solution for the constructed instance Q of CSP corresponds to a feasible solution of the L -bounded cut problem of the same cost, and vice versa, we observe the following.

Given an optimal solution $F \subset E$ of the L -bounded cut problem with s and t in the same component, the vector of shortest path distances from s to all other vertices in $(V, E \setminus F)$ yields a feasible solution for the CSP instance Q (to be more precise, if some of the distances are larger than L , we replace in the vector every such value by L); given an optimal solution $F \subset E$ of the L -bounded cut problem with s and t in the different components, we obtain a feasible solution for Q by assigning the value 0 to every vertex in the s -component and the value 2 to every vertex in the t -component. Note that in both cases the cost of the L -bounded cut and the cost of the CSP instance Q are the same. On the other hand, for every feasible solution (z_1, \dots, z_n) of the instance Q , the set $F = \{\{u, v\} \mid |z_u - z_v| > 1\}$ is an L -bounded cut of the same cost. Finally, we note that the constraint graph of Q coincides with the original graph G .

For the vertex version of the L -bounded cut problem, we extend the domain of every vertex v by an extra element -1 representing the fact that v belongs to the L -bounded cut, and we adjust the hard and soft constraints appropriately.