# How to Cut a Ball without Separating: Improved Approximations for Length Bounded Cut

Eden Chlamtáč* Petr Kolman†

**Abstract**

The Minimum Length Bounded Cut problem is a natural variant of Minimum Cut: given a graph, terminal nodes $s, t$ and a parameter $L$, find a minimum cardinality set of nodes (other than $s, t$) whose removal ensures that the distance from $s$ to $t$ is greater than $L$. We focus on the approximability of the problem for bounded values of the parameter $L$.

The problem is solvable in polynomial time for $L \leq 4$ and NP-hard for $L \geq 5$. The best known algorithms have approximation factor $\lceil (L-1)/2 \rceil$. It is NP-hard to approximate the problem within a factor of 1.1715 and Unique Games hard to approximate it within $\Omega(L)$, for any $L \geq 5$. Moreover, for $L = 5$ the problem is $4/3 - \varepsilon$ Unique Games hard for any $\varepsilon > 0$.

Our first result matches the hardness for $L = 5$ with a 4/3-approximation algorithm for this case, improving over the previous 2-approximation. For 6-bounded cuts we give a 7/4-approximation, improving over the previous best 3-approximation. More generally, we achieve approximation ratios that always outperform the previous $\lceil (L-1)/2 \rceil$ guarantee for any (fixed) value of $L$, while for large values of $L$, we achieve a significantly better $((11/25)L + O(1))$-approximation.

All our algorithms apply in the weighted setting, in both directed and undirected graphs, as well as for edge-cuts, which easily reduce to the node-cut variant. Moreover, by rounding the natural linear programming relaxation, our algorithms also bound the corresponding bounded-length flow-cut gaps.

## 1 Introduction

In the Minimum Length Bounded Cut problem, we are given a directed (undirected) graph $G = (V, E)$, two distinguished vertices $s, t \in V$, called the source and the sink, and an integer parameter $L > 0$, and wish to find a minimum cardinality *L-bounded node cut* (edge cut, resp.): a subset $F \subseteq V \setminus \{s, t\}$ of vertices (a subset $F \subseteq E$ of edges, resp.) such that the every path between the source $s$ and the sink $t$ in $G \setminus F$ has length strictly greater than $L$; the path length is the number of edges in it.

Various aspects of the problem have been studied for almost half a century: its relationship to the maximum $L$-bounded flow [1, 2, 3, 17] and to the maximum number of disjoint $L$-bounded $s-t$ paths [1, 7, 16], its complexity [5, 3, 15] and approximability [3, 14], fixed-parameter tractability [6, 9, 10, 11], and polynomial time solvability for various graph classes [4, 9, 18, 14]. In this paper we focus on the approximability of the problem.

There are four basic versions of the $L$-Bounded Cut problem, depending on whether the graph $G$ is directed or undirected, and whether we cut nodes or edges. As observed by Golovach and Thilikos [11], an $\alpha$-approximation algorithm for Directed $L$-Bounded Node Cut yields

*Ben Gurion University, Israel. e-mail: chlamtac@cs.bgu.ac.il

†Charles University, Faculty of Mathematics and Physics, Czech Republic. e-mail: kolman@kam.mff.cuni.cz

an $\alpha$-approximation algorithm for Undirected $L$-Bounded Node Cut, Undirected $(L-1)$-Bounded Edge cut, and Directed $(L-1)$-Bounded Edge Cut. Thus, we describe all our algorithms just for the directed node version, and in the rest of the paper we focus exclusively on this setting.

On the complexity and approximability side, the node (edge, resp.) version of Minimum $L$-Bounded Cut is solvable in polynomial time for $L \leq 4$ [16] (for $L \leq 3$ [17], resp.) and is NP-hard for $L \geq 5$ (for $L \geq 4$, resp.) [3]. With respect to the length parameter $L$, there are several simple $O(L)$-approximation algorithms [3, 17], and the best known algorithm has approximation ratio $\lceil (L-1)/2 \rceil$ [3].

For $L \geq 4$, $L$-Bounded Edge Cut is known to be NP-hard to approximate within a factor of 1.1377 [3]. In fact, we observe that the reduction in that paper also implies the NP-hardness of approximating the problem to within a factor of 1.1715, and Unique Games hardness of approximating it to within $4/3 - \varepsilon$ for any $\varepsilon > 0$ – see Appendix A for details. Note that these results imply the same hardness for $L$-Bounded Node Cut for $L \geq 5$. Recently, Lee [15] showed that for bounded values of $L$ it is Unique Games hard to approximate the undirected edge variant within $\Omega(\sqrt{L})$, and the other three variants within $\Omega(L)$. Thus, assuming the Unique Games Conjecture, the best possible approximation for all but the undirected edge variant is $\Theta(L)$. However, the exact best possible approximation as a function of $L$ is not known.

If the length bound is not a fixed constant but a part of the input, the gap between the known hardness results and approximations is even bigger: there are no stronger hardness results and the best known algorithms [3] have approximation ratio $O(\min\{L, n/L\}) \subseteq O(\sqrt{n})$ in the node case and $O(\min\{L, n^2/L^2, \sqrt{m}\}) \subseteq O(n^{2/3})$ in the edge case where $m$ denotes the number of edges.

## 1.1 Our results

Our first result is an algorithmic upper bound matching the Unique Games hardness of 4-Bounded Edge Cut and 5-Bounded Node Cut; note that these two problems are the *first* hard instances of $L$-Bounded Cut – for smaller length bounds the corresponding problems are in P. For reasons explained earlier we state all our results for the node case only.

**Theorem 1.1.** *There exists a $4/3$-approximation algorithm for* Minimum 5-Bounded Node Cut.

Similarly, for $L = 6$ we also describe a new algorithm with an improved approximation ratio.

**Theorem 1.2.** *There exists a $7/4$-approximation algorithm for* Minimum 6-Bounded Node Cut.

This algorithm is based on the same, yet more involved, techniques as the algorithm for $L = 5$. More generally, we have an approximation algorithm that works for any value of $L$.

**Theorem 1.3.** *For any fixed length bound $L \geq 6$, there exists an $\left(\frac{L-1}{2} - \frac{3}{L-2}\right)$-approximation algorithm for* Minimum $L$-Bounded Node Cut.

This algorithm is based on our algorithm for $L = 6$ and a general observation that an $\alpha$-approximation algorithm for $L$-Bounded Cut with certain properties can be used to design an approximation algorithm for $(L+1)$-Bounded Cut, with a slightly weaker approximation ratio (see Theorem 3.1). This is always better than the previous best known $\lceil (L-1)/2 \rceil$-approximation, but for large values of $L$ it is not significantly better. Though various algorithmic techniques, including the above theorem, all point to $L/2 - o(L)$ being the best possible approximation (cf. [3]), we are able to improve over this bound.

**Theorem 1.4.** *For any fixed length bound $L > 5$, there exists an $((11/25)L + O(1))$-approximation algorithm for* Minimum $L$-Bounded Node Cut.

A succinct summary of the old and the new results on approximability of Minimum $L$-Bounded Cut is provided in Table 1. It is worth mentioning that all our algorithms work also in the more general setting where every node (edge) has a non-negative weight and the objective is to find an $L$-bounded cut of minimum total weight, simply by including these weights in the objective function of the linear program.

Table 1: Known and **new** (bold type) results for bounded values of $L$. All results hold for both directed and undirected cases, unless stated otherwise. We note that a hardness result for some value of $L$ implies the same hardness result for larger values as well, thus we only state hardness results here for the smallest $L$ for which they hold. NP stands for NP hardness of approximation, UG for Unique Games hardness of approximation.

| $L$ | Node Cuts | | Edge Cuts | |
|---|---|---|---|---|
| | Hardness | Approximations | Hardness | Approximations |
| $\leq 3$ | 1 (in P) | | 1 (in P) [17] | |
| 4 | 1 (in P) [16] | | NP 1.1715 [3, 12]<br>UG 4/3 [3, 13] | 2 [3]<br>**4/3** |
| 5 | NP 1.1715 [3, 12]<br>UG 4/3 [3, 13] | 2 [3]<br>**4/3** | | 3 [3]<br>**7/4** |
| 6 | | 3 [3]<br>**7/4** | | 3 [3]<br>**12/5** |
| any | UG $\Omega(L)$ [15] | $\lceil (L-1)/2 \rceil$ [3]<br>$\boldsymbol{(L-1)/2 - 3/(L-2)}$<br>$\boldsymbol{0.44(L-1) + O(1)}$ | UG $\Omega(L)$ directed,<br>$\Omega(\sqrt{L})$ undir. [15] | $\lceil L/2 \rceil$ [3]<br>$\boldsymbol{L/2 - 3/(L-1)}$<br>$\boldsymbol{0.44L + O(1)}$ |

**Approximate duality of $L$-bounded flows and cuts.** As we have already mentioned earlier, people investigated the relation between the minimum $L$-bounded flow and the maximum $L$-bounded flow (where an $L$-bounded flow is a flow that can be decomposed into flows along paths of length at most $L$) (e.g., [2, 3, 17]); in fact, this was one of the first questions Adámek and Koubek [1] asked back in 1971.

All our algorithms are based on rounding the linear program (1) (given in the next subsection) which is the dual of the exact linear programming formulation of the maximum $L$-bounded flow. Thus, as a corollary of our algorithmic results, we obtain improved approximate duality relations between $L$-bounded cuts and flows that are tighter than those previously known. For the sake of brevity we state the result just for 4-bounded edge cuts and flows.

**Corollary 1.5.** *Given a graph $G = (V, E)$ and nodes $s, t \in V$, let $\mathcal{F}$ denote a maximum 4-bounded flow, and $\mathcal{C}$ a minimum 4-bounded edge cut. Then*

$$|\mathcal{F}| \leq |\mathcal{C}| \leq \frac{4}{3}|\mathcal{F}|,$$

*and these bounds are tight.*

## 1.2 Overview of our approach

All our algorithms are based on a rounding of a natural linear programming relaxation of the problem. This relaxation was studied in earlier works on $L$-Bounded Cut (e.g., [3, 17]) but for the sake of completeness we provide it here again. In the following linear program (solvable via a simple separation oracle), $\mathcal{P}_{s,t}(L)$ denotes the set of all paths between $s$ and $t$ of length at most $L$.

$$\begin{aligned} \min \sum_{v\in V\setminus\{s,t\}} x_v & & & \quad (1) \\ \sum_{v\in p\setminus\{s,t\}} x_v & \geq 1 & \forall p\in\mathcal{P}_{s,t}(L) & \quad (2) \\ x_v & \geq 0 & \forall v\in V & \end{aligned}$$

The previous approximation algorithms were based on cutting only shortest paths, which can be done optimally by taking a minimum cut in the layered graph of shortest paths from $s$ to $t$. By iteratively cutting all shortest $s-t$ paths until the distance between $s$ and $t$ becomes larger then $L$, we get an $L$-approximation.

This approach can also be framed as a rounding of the above linear programming relaxation, via the following classical rounding which cuts all shortest paths while paying at most the LP value:

**Exact-Round**$(G,(x_v)_{v\in V})$

- For every $v\in V\setminus\{s,t\}$, let $y_v := \min\{x(p) \mid p: s\rightsquigarrow v,\ |p|=d(s,v)\}$ and $I_v := [y_v, y_v+x_v]$, where $s\rightsquigarrow v$ stands for an $s-v$ path, $d$ is the hop-distance in $G$ and $x(p)$ is defined to be the total LP value of vertices in $p$ excluding its endpoints.
- Sample $r\in[0,1]$ uniformly at random.
- Return the cut $\{v \mid r\in I(v)\}$.

In words, every vertex $v\in V\setminus\{s,t\}$ is mapped to an interval $I_v$ of length $x_v$ in such a way that for every $s-t$ path $p$ of interest, we have $[0,1]\subseteq\bigcup_{v\in p} I_v$. The algorithm then cuts all vertices whose corresponding intervals lie on the boundary of a ball of uniformly random radius; by cutting these vertices the algorithm separates $s$ from $t$ in the layered subgraph of shortest $s-t$ paths. The property that is important in the analysis, and that will be important later in our new algorithms, is the following:

**Observation 1.6.** *Given a solution $(x_v)_{v\in V}$ of the LP (1), for every $v\in V$,* **Exact-Round** *cuts $v$ with probability at most $x_v$.*

The difficulty with the outlined iterative approach to construct an $L$-bounded cut is that it uses up to $L$ iterations (or up to $\lceil(L-1)/2\rceil$, if we first cut all vertices with $x_v\geq 1/\lceil(L-1)/2\rceil$), where each iteration may yield a cut as large as the LP value. Our idea is to circumvent the need for multiple iterations by mapping every vertex $v$ to *multiple* intervals, each interval representing a possible position of the vertex in an $L$-bounded path. Thus, our algorithms are similar to classical ball cutting algorithms, yet they differ in that they do not (necessarily) separate $s$ from $t$.

However, this is only a framework for our improvements. A naïve random ball growing algorithm would not yield a better approximation ratio when applied to these intervals. To fully take advantage of this framework, we introduce additional ideas such as cutting only a carefully chosen subset of the boundaries of more than one ball, and/or modifying the LP values in order to take better advantage of the structure of our mappings.

## 2 A 4/3-approximation for $L = 5$

Here we prove Theorem 1.1. Specifically, we give a rounding of the natural LP relaxation for 5-Bounded Node Cut which cuts every vertex $v \in V$ with probability at most $(4/3) \cdot x_v$.

Before describing our rounding, we describe here a certain mapping of the vertices $V \setminus \{s,t\}$ to intervals. For a graph $G = (V, E)$ and an LP solution $(x_v)_{v \in V}$, we map every vertex $v$ at distance most 2 from $s$ or to $t$ to two intervals, $I_G^+(v), I_G^-(v)$. For any vertices $u, v$ such that $(s,u),(v,t) \in E$, we define

$$I_G^-(u) = I_G^+(u) = [0, x_u] \qquad I_G^-(v) = I_G^+(v) = [1 - x_v, 1].$$

Next, for any vertex $u$ (for which $(u,t) \notin E$) at distance 2 from $s$, define

$$y_G^+(u) = \min_{u' : (s,u'),(u',u) \in E} x_{u'} \qquad I_G^+(u) = [y_G^+(u), y_G^+(u) + x_u].$$

If $d_G(u,t) > 2$, define $I_G^-(u) = I_G^+(u)$; note that in such a case, within any 5-bounded $s - t$ path containing it, $u$ will be at distance 2 from $s$ and 3 from $t$. Finally, for any vertex $v$ (for which $(s,v) \notin E$) at distance 2 to $t$, define

$$y_G^-(v) = \min_{u : (v,v'),(v',t) \in E} x_{v'} \qquad I_G^-(v) = [1 - y_G^-(v) - x_v, 1 - y_G^-(v)].$$

If $d_G(s,v) > 2$, define $I_G^+(v) = I_G^-(v)$.

We will drop the subscript $G$ from the above definitions when it is clear from the context. Note that $I^+(v), I^-(v)$ have length $x_v$ for every $v$, and $I^-(v) = I^+(v)$ for all vertices $v$ except those which are both at distance 2 from $s$ and at distance 2 to $t$. For $v$ such that $I^-(v) \neq I^+(v)$, there exists a path $\langle s, u', v, v', t \rangle$ such that $y^+(v) = x_{u'}$ and $y^-(v) = x_{v'}$. Thus, the intervals have the same length, and $I^-(v)$ starts to the left of $I^+(v)$ (when plotted on the real number line). Indeed, the left endpoint of $I^+(v)$ will be $x_{u'}$, while the left endpoint of $I^-(v)$ will be $1 - (x_{v'} + x_v)$, which is at most $x_{u'}$ by Constraint (2).

We are now ready to define our rounding:

**Algorithm 5-Round**

- Let $C_0 = \{v \in V \mid x_v \geq 3/4\}$. Let $G'$ be the remaining graph after deleting all vertices in $C_0$.
- Sample $r \in [0,1]$ uniformly at random.
- Sample $r_1 \in [0, 1/2]$ uniformly at random, and let $r_2 = r_1 + 1/2$.
- Let $C_1$ be the set of all vertices $v$ such that $r \in I_{G'}^-(v) \cup I_{G'}^+(v)$.
- Let $C_2$ be the set of all vertices $v$ for which at least one of the following conditions holds:
  1. $r_1 \in I_{G'}^-(v) \cap I_{G'}^+(v)$ or $r_2 \in I_{G'}^-(v) \cap I_{G'}^+(v)$, or
  2. $r_1, r_2 \in I_{G'}^-(v)$ or $r_1, r_2 \in I_{G'}^+(v)$.
- With probability 2/3, return $C_0 \cup C_1$. Otherwise (with probability 1/3), return $C_0 \cup C_2$.

**Remark 2.1.** *This algorithm can be easily derandomized by trying $O(|V|)$ possible radii $r, r_1$ and choosing the better of the two cuts $C_0 \cup C_1$, $C_0 \cup C_2$.*

First let us see why this is a valid 5-bounded cut.

**Lemma 2.2.** *Algorithm* **5-Round** *cuts all paths of length at most 5 from $s$ to $t$.*

*Proof.* Since vertices in $C_0$ are always cut, we focus on paths not cut by $C_0$. That is, on paths in $G'$.

First, let us see that $C_1$ cuts all 5-bounded $s-t$ paths in $G'$. Let $p = \langle s, u', u, v, v', t\rangle$ be such a path (the proof for paths of length $< 5$ is similar or simpler). Let $u'', v''$ be vertices such that $y^+(u) = x_{u''}$ and $y^-(v) = x_{v''}$. Since $\langle s, u'', u, v, v'', t\rangle$ is also a path in $G'$, it follows from the definition of the intervals and Constraint (2) that $[0,1] \subseteq I^+(u'') \cup I^+(u) \cup I^-(v) \cup I^-(v'')$. Moreover, we clearly have $I^+(u'') \subseteq I^+(u')$ and $I^+(v'') \subseteq I^+(v')$, so $r$ must be contained in one of of the intervals $I^+(u'), I^+(u), I^-(v), I^-(v')$, and so $p$ will be cut by $C_1$.

Now we consider the cut $C_2$. First consider a path $p = \langle s, u', v, v', t\rangle$ of length 4. Note that $I^+(u') = I^-(u')$ and $I^+(v') = I^-(v')$, and by similar reasoning to the above, we have $[0,1] \subseteq I^+(u') \cup (I^+(v) \cap I^-(v)) \cup I^-(v')$, and so at least one of $r_1$ or $r_2$ must be in $I^-(u) \cap I^+(u)$ for some $u \in \{u', v, v'\}$, and this vertex will be cut by $C_2$.

Finally, let $p = \langle s, u', u, v, v', t\rangle$ be a 5-path in $G'$, and let us see that $p$ is cut by $C_2$. Since $I^+(u') = I^-(u')$ and $I^+(v') = I^-(v')$, we may assume that $r_1 > x_{u'}$ and $r_2 < 1 - x_{v'}$ (otherwise, $u'$ or $v'$, respectively, will be cut). Thus, $r_1, r_2 \in I^+(u) \cup I^-(v)$. In particular, $r_1 \in I^+(u)$ or $r_2 \in I^-(v)$; otherwise, we would have $r_2 \in I^+(u)$ and $r_1$ to the left of $I^+(u)$, that is $r_1 < y^+(u) \le x_{u'}$, contradicting our current assumption.

If both radii were in $I^+(u)$ or both in $I^-(v)$, then again the corresponding vertex would be cut by $C_2$, so assume $r_1 \in I^+(u)$ and $r_2 \in I^-(v)$. We claim that in this case we must have $r_1 \in I^-(u) \cap I^+(u)$ or $r_2 \in I^-(v) \cap I^+(v)$, and then the corresponding vertex will be cut in $C_2$. For the sake of contradiction, assume $r_1 \in I^+(u) \setminus I^-(u)$ and $r_2 \in I^-(v) \setminus I^+(v)$. The fact that both of these vertices are mapped to two distinct intervals means that there exist vertices $u'', v''$ in $G'$ such that $\langle s, v'', v\rangle$ and $\langle u, u'', t\rangle$ are paths in $G'$ and $y^-(u) = x_{u''}$ and $y^+(v) = x_{v''}$. As $r_1$ is to the right of $I^-(u)$ and $r_2$ is to the left of $I^+(v)$, we have $r_1 > 1 - x_{u''}$ and $r_1 + 1/2 = r_2 < x_{v''}$. Thus, we get $x_{u''} + x_{v''} > 3/2$, contradicting the fact that the LP value of every vertex in $G'$ is at most $3/4$. □

Next we bound the expected value of the cut.

**Lemma 2.3.** *Every vertex $v \in V$ is cut by Algorithm 5-Round with probability at most $(4/3) \cdot x_v$.*

*Proof.* If $x_v \ge 3/4$, this is trivial. Thus, let us look at vertices in $V \setminus C_0$.

First, consider a vertex $v$ such that $x_v < 1/2$. Note that $v$ is cut by $C_1$ with probability at most $|I^-(v) \cup I^+(v)|$ (it is exactly this probability if both intervals are contained in $[0,1]$, but this is not necessarily the case). Now consider the definition of $C_2$. As $x_v < 1/2$, Condition 2 in the algorithm can never occur. Thus the probability that $v$ is cut by $C_2$ is the probability that it will be cut because of Condition 1, which is exactly $2 \cdot |I^-(v) \cap I^+(v) \cap [0,1/2]| + 2 \cdot |I^-(v) \cap I^+(v) \cap [1/2,1]| = 2 \cdot |I^-(v) \cap I^+(v)|$. Thus the overall probability that $v$ is cut is at most

$$\tfrac{2}{3} \cdot |I^-(v) \cup I^+(v)| + \tfrac{1}{3} \cdot 2 \cdot |I^-(v) \cap I^+(v)| = \tfrac{2}{3} \cdot (|I^-(v)| + |I^+(v)|) = \tfrac{4}{3} \cdot x_v.$$

Now consider a vertex $v \in V$ such that $x_v \ge 1/2$ (but less than $3/4$). If Condition 2 never occurs, or only occurs when Condition 1 also occurs (for example, if $I^-(v) = I^+(v)$), then we are done by the same calculation as above. Otherwise, there exist values of $r_1$ for which Condition 2 occurs but not Condition 1. That is, either

- Case (i): There exists $r_1$ such that $r_1, r_2 \in I^-(v)$ and $r_2$ is to the left of $I^+(v)$, or
- Case (ii): There exists $r_1$ such that $r_1, r_2 \in I^+(v)$ and $r_1$ is to the right of $I^-(v)$.

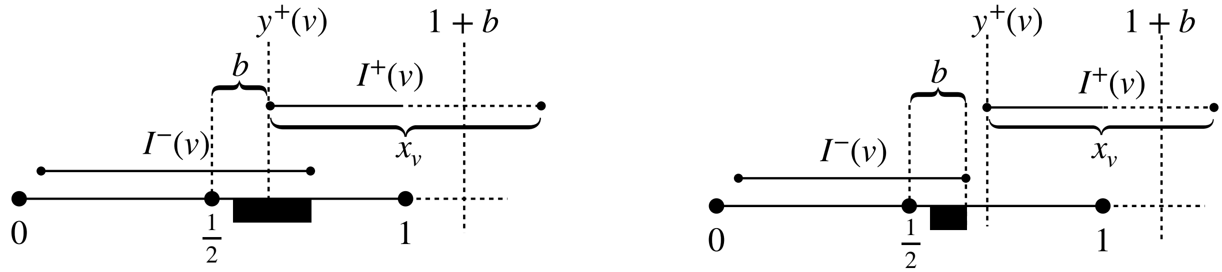


Figure 1: Two examples of the final case in the proof of Lemma 2.3. The black rectangles denote the range of $r_2$ for which $v \in C_2$. This range has length at most $|I^-(v) \cap I^+(v)| + b$, while the range of $r$ for which $v \in C_1$ has length at most $|I^-(v) \cup I^+(v)| - b$.

Note that at most one of these cases is possible for a given vertex $v$, since Case (i) implies that $I^-(v)$ intersects the interval $[1/2, 1]$, whereas Case (ii) implies that $I^-(v)$ is strictly to the left of 1/2. Without loss of generality, assume (only) Case (i) holds. Let $b = \min\{1/2 - y^-(v), y^+(v) - 1/2\}$, that is, the maximum value such that $1/2 + b \in I^-(v)$ and $1/2 + b \le \min I^+(v)$ (see Figure 1). By our assumption, $b \ge 0$. Thus, $v \in C_2$ *only* if (not iff) $r_2 \in I^-(v) \cap I^+(v)$ or $r_2 \in [1/2, 1/2 + b]$. This happens with probability at most $2|I^-(v) \cap I^+(v)| + 2b$. As we've noted, $v$ is cut by $C_1$ with probability at most $|I^-(v) \cup I^+(v)|$. However a tighter upper bound on this probability is $|(I^-(v) \cup I^+(v)) \cap [0, 1]|$. Since $x_v \ge 1/2$, $b \le y^+(v) - 1/2$, and $I^+(v) = [y^+(v), y^+(v) + x_v]$, we have that $|I^+(v) \setminus [0, 1]| \ge b$, and so the total probability that $v$ will be cut is upper bounded by

$$\begin{aligned}
\frac{2}{3}|(I^-(v) \cup I^+(v)) \cap [0,1]| &+ \frac{1}{3} \cdot \big(2|I^-(v) \cap I^+(v)| + 2b\big) \\
&= \frac{2}{3}\big(|I^-(v) \cup I^+(v)| - |I^+(v) \setminus [0,1]| + |I^-(v) \cap I^+(v)| + b\big) \\
&= \frac{2}{3}\big(|I^-(v)| + |I^+(v)| - |I^+(v) \setminus [0,1]| + b\big) \\
&\le \frac{2}{3}\big(|I^-(v)| + |I^+(v)|\big) = \frac{4}{3} x_v.
\end{aligned}$$

Figure 1 illustrates this final case. □

## 3 Improved approximations for other small values of $L$

In this section, we prove Theorem 1.3. Our approximation is based on an algorithm which takes as a black box a rounding for $(L-1)$-Bounded Cut and converts it into a rounding for $L$-Bounded Cut. In particular, we show the following result:

**Theorem 3.1.** *Given a rounding algorithm for* $(L-1)$-Bounded Cut *which cuts every vertex* $v$ *with probability at most* $\alpha \cdot x_v$*, there is a rounding for* $L$-Bounded Cut *which cuts every vertex* $v$ *with probability at most* $(1 + \alpha \cdot (1 - 1/(L-2))) \cdot x_v$.

Theorem 1.3 now follows easily.

*Proof of Theorem 1.3.* Follows immediately by induction, where the base case is Theorem 1.2 for $L = 6$ (or technically, the more specific Lemma 5.4), and Theorem 3.1 gives the inductive step. □

We now describe our technique for converting a rounding for $(L-1)$-Bounded Cut to a rounding for $L$-Bounded Cut. The following algorithm takes as a black box a rounding algorithm $A_{L-1}$ for $(L-1)$-Bounded Cut:

**$L$-Recurse**$(G, (x_v)_{v\in V}, A_{L-1})$

- Cut every vertex $v \in V$ such that $x_v \geq 1$, and let $V'$ be the remaining vertices.
- Define $(z_v)_{v\in V'}$ as follows:
$$z_v = \left(1 - \tfrac{1}{L-2}\right) \cdot \tfrac{x_v}{1-x_v}.$$
- Run algorithm $A_{L-1}$ on graph $G|_{V'}$ and LP solution $(z_v)_{v\in V'}$. Let $V''$ be the remaining vertices (not cut in this step).
- Run **Exact-Round**$(G|_{V''}, (x_v)_{v\in V''})$.

To understand this algorithm, first let us see the feasibility of the solution $(z_v)_{v\in V'}$ defined in the second step.

**Lemma 3.2.** *If $(x_v)_{v\in V'}$ is a feasible solution for the natural LP relaxation for $(L-1)$-Bounded Cut on a graph with vertex set $V'$, such that $x_v < 1$ for all $v \in V'$, then the solution $(z_v)_{v\in V'}$ defined above is also a feasible LP solution for the same LP relaxation.*

*Proof.* By our assumption that $x_v < 1$ for all $v \in V'$, we have $z_v \geq 0$ for all vertices. Now let $\langle s, v_1, \ldots, v_{L-2}, t\rangle$ be a path of length $L-1$. By the feasibility of $(x_v)_{v\in V'}$ we know that $\sum_{i=1}^{L-2} x_{v_i} \geq 1$. Therefore, by the convexity of the function $f(x) = x/(1-x)$ for all $x < 1$ (and Jensen's inequality), we have

$$\begin{aligned}\sum_{i=1}^{L-2} z_v &= (L-3) \cdot \tfrac{1}{L-2} \textstyle\sum_{i=1}^{L-2} x_{v_i}/(1-x_{v_i}) \\ &\geq (L-3) \cdot \tfrac{1}{L-2} \textstyle\sum_{i=1}^{L-2} x_{v_i} / \left(1 - \tfrac{1}{L-2} \textstyle\sum_{i=1}^{L-2} x_{v_i}\right) \\ &\geq (L-3) \cdot \tfrac{1}{L-2}/(1 - \tfrac{1}{L-2}) = 1.\end{aligned}$$

Thus, Constraint (2) is satisfied for this path. For paths of length $L' < L-1$, the argument follows from the above calculation by taking the LP values along the path and appending additional values $x_{L'-1} = \ldots = x_{L-2} = 0$ which are mapped to $z_{L'-1} = \ldots z_{L-2} = 0$. □

We can now show our main guarantee for this section.

*Proof of Theorem 3.1.* Let us see that Algorithm $L$-Recurse satisfies the requirements. First note that the algorithm returns a feasible $L$-bounded cut. Indeed, by Lemma 3.2, $(z_v)_{v\in V'}$ is a feasible solution for the relaxation for $(L-1)$-Bounded Cut, and so the application of algorithm $A_{L-1}$ will cut all paths of length at most $L-1$ not already cut in the first step. In the remaining graph we have $d(s,t) \geq L$, and so all remaining paths (if any) of length $L$ will be cut in the last step.

As for the approximation guarantee, if $x_v \geq 1$, then the theorem follows trivially. For $v \in V'$, we know that algorithm $A_{L-1}$ will cut $v$ with probability $\alpha'_v \cdot z_v$, for some $\alpha'_v \leq \alpha$. Conditioned on surviving this phase, by Observation 1.6, $v$ will be cut in the last step with probability at most $x_v$. Thus the overall probability that $v$ is cut will be at most

$$\begin{aligned}\alpha'_v \cdot z_v + (1 - \alpha'_v \cdot z_v)x_v &= x_v + (1-x_v)\alpha'_v \cdot z_v \\ &= x_v + \alpha'_v \cdot (1 - 1/(L-2)) \cdot x_v \\ &\leq (1 + \alpha \cdot (1 - 1/(L-2))) \cdot x_v.\end{aligned}$$

□

## 4 An $((11/25) \cdot L + O(1))$-approximation

Here we prove Theorem 1.4, showing that $L(1/2 - o(1))$ is not the best possible approximation for general (bounded) $L$. Similarly to our algorithm for $L = 5$, we will map every vertex to different intervals corresponding to different positions the vertex can have in an $L$-bounded path from $s$ to $t$, and then cut all intervals containing a random radius in $[0, 1]$. Our improvement follows from showing that these intervals will be mapped close together. However, for vertices with small LP value, this will not be sufficient, since the intervals can still be disjoint. To avoid this problem, we will define a new LP solution which will greatly decrease the LP value of vertices which already have a small LP value, giving us an advantage over previous algorithms for these vertices as well.

**Distort-Round**

- Cut all vertices $v \in V$ with $x_v \geq 25/(11(L-1))$. Let $G'$ be the graph on remaining vertices.
- For every remaining vertex $v$, let $i_{\min}(v) = d(s, v)$ and $i_{\max}(v) = L - d(v, t)$ (the first and last possible positions of $v$ in an $L$-bounded path), where $d$ is the hop-distance in $G'$. Let $V' = \{v \in G \mid i_{\min}(v) \leq i_{\max}(v)\}$, and on these vertices define $(x'_v)_{v \in V'}$ as follows:

$$x'_v = \max\{0, (23/20) \cdot x_v - 3/(20(L-1))\}.$$

- For all $v \in V'$ and $i \in \{i_{\min}(v), \ldots, i_{\max}(v)\}$ define the following intervals, where paths are in $G'$ and as before $x'(p)$ is the total $x'$ value of all vertices in $p$ excluding the endpoints:

$$y_i(v) := \min_{\substack{p: s \rightsquigarrow v \\ |p| \leq i}} x'(p) \qquad \text{and} \qquad I_i(v) = [y_i(v), y_i(v) + x'_v].$$

- Sample $r \in [0, 1]$ uniformly at random, and cut all vertices $v \in V'$ such that $r \in I_i(v)$ for some $i$.

Let us first see the correctness of the algorithm:

**Lemma 4.1.** *Algorithm Distort-Round returns a valid $L$-bounded vertex cut.*

*Proof.* Since the first step cuts all paths not entirely in $G'$, let us focus on paths in $G'$. It is straightforward to see that in this graph, the interior vertices of every $L$-bounded $s - t$ path are all in $V'$, since if $v$ is $i$th vertex in such a path, then $i_{\min}(v) \leq i \leq i_{\max}(v)$. For any such path we have $\sum_{v \in p \setminus \{s,t\}} x'_v \geq \sum_{v \in p \setminus \{s,t\}} (23/20) \cdot x_v - ((|p| - 1)/(L-1))\, 3/20 \geq 1$.

Let us see that such a path will necessarily be cut. Denote $p = \langle s, v_1, \ldots, v_{L'}, t \rangle$ for some $L' \leq L - 1$. By the definition of our intervals, it can be shown by induction that for every $i \in [L']$ we have $[0, y_i(v_i) + x'_{v_i}] \subseteq \bigcup_{j=1}^{i} I_j(v_j)$. Thus, to see that the path is cut in the last step, it suffices to show that $y_{L'}(v_{L'}) + x'_{v_{L'}} \geq 1$. But since $(v_{L'}, t) \in E$, by the definition of $y$, this is the total $x'$ value of some $s - t$ path of length at most $L' + 1$, and so it must be at least 1. $\square$

To bound the approximation guarantee, we need to bound the probability that any vertex $v \in V'$ is cut. Let $D(v) := i_{\max}(v) + 1 - i_{\min}(v)$ denote the number of intervals defined for $v$. Then one trivial bound is the following:

**Observation 4.2.** *Every vertex $v \in V'$ is cut with probability at most $D(v) \cdot x'_v$.*

However, this bound may be too conservative. In fact, if $v$ participates in a large number of intervals, we can show that these intervals cannot be too spread out.

**Lemma 4.3.** *For any given vertex $v \in V'$, all the intervals $I_i(v)$ are contained in a single interval of length at most $161/110 - D(v) \cdot 271/(110(L-1)) + O(x_v)$.*

*Proof.* Let $p$ be a shortest path from $s$ to $v$ in $G'$. By the definition of $G'$, all vertices $u \in p$ have $x_u < 25/(11(L-1))$, and so $x'_u < 271/(110(L-1))$. By definition of $y_i(v)$, this means that for all $i \geq i_{\min}(v)$ we have

$$y_i(v) \leq x'(p) \leq (d(s,v)-1)271/(110(L-1)) = (i_{\min}(v)-1)271/(110(L-1)),$$

and in particular,

$$y_i(v) + x'_v \leq (i_{\min}(v)-1)271/(110(L-1)) + x'_v \tag{3}$$

Similarly, for any $i \in \{i_{\min}(v), \ldots, i_{\max}(v)\}$, let $p_i$ be an $s-v$ path of length $\leq i$ such that $y_i(v) = x'(p_i)$, and let $p'$ be a shortest path from $v$ to $t$. Then $p_i \circ p'$ is an $L$-bounded $s-t$ path with $x'$ value

$$\begin{aligned} x'(p_i) + x'_v + x'(p') &\leq y_i(v) + x'_v + (|p'|-1)271/(110(L-1)) \\ &= y_i(v) + x'_v + (L - i_{\max}(v) - 1)271/(110(L-1)) \\ &= y_i(v) + x'_v - i_{\max}(v)271/(110(L-1)) + 271/110. \end{aligned}$$

However, its $x'$ value must also be at least 1, and so we get

$$y_i(v) \geq i_{\max}(v)271/(110(L-1)) - 161/110 - x'_v. \tag{4}$$

Since $I_i(v) = [y_i(v), y_i(v) + x'_v]$, equations (3) and (4) imply that all these intervals are contained in a single interval of length at most

$$\begin{aligned} &(i_{\min}(v)-1)271/(110(L-1)) + x'_v - (i_{\max}(v)271/(110(L-1)) - 161/110 - x'_v) \\ &\qquad = 161/110 - D(v) \cdot 271/(110(L-1)) + O(x_v). \end{aligned}$$

□

We can now prove our final guarantee for general (bounded) $L$:

**Lemma 4.4.** *Every vertex $v \in V$ is cut by Algorithm Distort-Round with probability at most $((11/25)(L-1) + O(1))x_v$.*

*Proof.* For vertices $v \notin V'$, this is trivial, so assume $v \in V'$. Also, let us assume $x'_v > 0$, since otherwise $v$ would not be cut. First, from Observation 4.2 and Lemma 4.3, we have that the probability that $v$ is cut is at most

$$\min\{D(v) \cdot x'_v,\ 161/110 - D(v) \cdot 271/(110(L-1))\} + O(x_v) \leq \frac{161x'_v}{110x'_v + 271/(L-1)} + O(x_v),$$

where the inequality is obtained by maximizing over all possible values of $D(v)$. Let us scale up $x_v$ and denote $c_v = (L-1)x_v$, which (by our assumption that $x'_v \neq 0$) implies that $x'_v = ((23 - 3/c_v)/20)x_v$. Thus, we can rewrite the above upper bound on the probability as

$$\left(\frac{23 - 3/c_v}{20} \cdot \frac{161(L-1)}{11(23c_v - 3)/2 + 271} + O(1)\right) x_v$$

Ignoring the $O(1)$ term, the expression in parentheses is maximized for $c_v = (3 + \sqrt{1626/11})/23$, and so is at most $0.43954(L-1) < (11/25) \cdot (L-1)$. □

## 5 A 7/4-approximation for $L = 6$

We now prove Theorem 1.2. We note first that the techniques we have seen already give an improvement over the previous 3-approximation for Minimum 6-Bounded Node Cut. Indeed, Lemma 2.3 and Theorem 3.1 together give a 2-approximation.

As a warm-up to our final algorithm, let us first see an alternative 2-approximation which does not use Theorem 3.1 or our algorithm for $L = 5$. We introduce some notation which we will use both for our warm-up algorithm for $L = 6$ as well as our main algorithm.

Similarly to our algorithm for $L = 5$, we will map every vertex to a small number of possible intervals, corresponding to its possible positions in a 6-bounded path, relative to $s$ and $t$. For all $i \in [3]$, and all vertices $u, v \in V \setminus \{s, t\}$ such that $d_G(s, u) \leq i$ and $d_G(v, t) \leq i$, define

$$y_i^G(u) := \min_{\substack{p: s \rightsquigarrow u \\ |p| \leq i}} x(p) \qquad \text{and} \qquad y_{-i}^G(v) := 1 - \min_{\substack{p: v \rightsquigarrow t \\ |p| \leq i}} x(p).$$

As before, if $(s, u), (v, t) \in E$, for such neighbors of $s, t$ we define

$$I_1^G(u) = [0, x_u] \qquad I_{-1}^G(v) = [1 - x_v, 1].$$

For other vertices $v$, wherever the relevant $y_i^G(v)$ values are defined, we also define intervals

$$I_2^G(v) = \left[y_2^G(v), y_2^G(v) + x_v\right] \qquad I_3^G(v) = \left[y_3^G(v), y_{-3}^G(v)\right] \qquad I_{-2}(v) = \left[y_{-2}^G(v) - x_v, y_{-2}^G(v)\right].$$

We will drop the superscript $G$ from the above definitions when it is clear from the context. If $I_1(v)$ (resp. $I_{-1}(v)$) is defined, we do not define any other interval for $v$, as such an interval would be contained in $I_1(v)$ (resp. $I_{-1}(v)$). Note that every interval associated with a vertex $v$ has length at most $x_v$ (in fact, exactly $x_v$ except for $I_3(v)$). As before, it is not hard to see that the left (resp. right) endpoints of the intervals $I_2(v), I_3(v), I_{-2}(v)$ (or whichever subsequence of intervals is defined for this vertex) form a monotone non-increasing sequence.

Consider the following rounding algorithm.

**Algorithm Simple-6-Round**

- Cut all vertices $v$ with $x_v \geq 1/2$, and let $G^{1/2}$ be the remaining graph after deleting these vertices.
- Sample $r \in [0, 1]$ uniformly at random.
- Cut all vertices $v$ such that $r \in I_i^{G^{1/2}}(v)$ for some $i \in \{1, 2, 3, -2, -1\}$.

One can check easily that this algorithm will cut all 6-bounded $s - t$ paths (the proof is nearly identical to the first part of the proof of Lemma 2.2). To see that it gives a 2-approximation, first note that trivially every vertex $v$ with $x_v \geq 1/2$ or for which at most two of the intervals $I_2^{G^{1/2}}(v), I_3^{G^{1/2}}(v), I_{-2}^{G^{1/2}}(v)$ are defined, will be cut with probability at most $2x_v$. Thus we only need to concern ourselves with vertices for which all three intervals are defined. Since we removed vertices with LP value at least 1/2, this means that in the remaining graph we have $y_2^{G^{1/2}}(v) < 1/2$ and $y_{-2}^{G^{1/2}}(v) > 1/2$, and so by monotonicity, $I_2^{G^{1/2}}(v)$ and $I_{-2}^{G^{1/2}}(v)$ must intersect, and $I_3^{G^{1/2}}(v)$ must be contained in their union. Thus, the union of the three intervals has length at most $2x_v$, which bounds the probability of cutting $v$.

To improve over this algorithm, we must make a number of changes. First of all, we cannot cut all vertices with $x_v \geq 1/2$. We must allow some (at least slightly) costlier vertices to remain. This means that we will not have the nice overlap property described above. We can overcome this by avoiding a small subinterval in the middle of $[0,1]$ in our radius sampling. However, this does not resolve the issue that some vertices will still be mapped to two possibly disjoint intervals, and in fact will exacerbate the problem by increasing the probability of hitting any given interval (since we are restricting our radius to a smaller sample space). We are able to overcome all these pitfalls by choosing at random either a single radius or a two-radius cut as we did in Algorithm 5-Round, and defining our two-radius cut carefully.

**Algorithm 6-Round**

- Let $C_0 = \{v \in V \mid x_v \geq 4/7\}$. Let $G'$ be the remaining graph after deleting all vertices in $C_0$.
- Sample $r \in [0, 3/7] \cup [4/7, 1]$ uniformly at random.
- Sample $r_1 \in [0, 3/7]$ uniformly at random, and let $r_2 = r_1 + 4/7$.
- Let $C_1$ be the set of all vertices $v$ such that $r \in I_i^{G'}(v)$ for some $i \in \{1, 2, 3, -2, -1\}$.
- Let $C_2$ be the set of all vertices $v$ satisfying at least one of the following conditions:
  1. $r_1 \in I_1^{G'}(v)$ or $r_2 \in I_{-1}^{G'}(v)$.
  2. $r_1 \in I_2^{G'}(v)$.
  3. $r_1 \in I_3^{G'}(v)$ and $\frac{3}{7} \in I_3^{G'}(v)$.
  4. $r_2 \in I_{-2}^{G'}(v)$ and $\frac{3}{7} \in I_{-2}^{G'}(v)$.
- Let $C_3$ be the set of all vertices $v$ satisfying at least one of the following conditions:
  1. $r_1 \in I_1^{G'}(v)$ or $r_2 \in I_{-1}^{G'}(v)$.
  2. $r_2 \in I_{-2}^{G'}(v)$.
  3. $r_2 \in I_3^{G'}(v)$ and $\frac{4}{7} \in I_3^{G'}(v)$.
  4. $r_1 \in I_2^{G'}(v)$ and $\frac{4}{7} \in I_2^{G'}(v)$.
- Return a random cut according to the following distribution: $C_0 \cup C_1$ with probability $1/2$, $C_0 \cup C_2$ with probability $1/4$, and $C_0 \cup C_3$ with probability $1/4$.

First let us see why this is a valid 5-bounded cut.

**Lemma 5.1.** *Algorithm 6-Round cuts all paths of length at most* $6$ *from* $s$ *to* $t$.

*Proof.* As we did for $L = 5$, we focus on paths in $G'$. Also, the proof of the correctness of $C_0 \cup C_1$ is essentially the same as the proof of correctness of the corresponding cut in Lemma 2.2.

Since $C_2$ and $C_3$ are symmetrically defined, let us focus on $C_2$. Let $\langle s, u, u', w, v', v, t \rangle$ be a 5-path from $s$ to $t$ in $G'$. The proof for paths of length $< 5$ is similar or simpler. As before, it can easily be seen that $[0,1] \subseteq I_1(u) \cup I_2(u') \cup I_3(w) \cup I_{-2}(v') \cup I_{-1}(v)$. If the first condition in the definition of $C_2$ does not hold w.r.t. $I_1(u)$ or $I_{-1}(v)$, then $r_1$ and $r_2$ must both intersect the intervals $I_2(u') \cup I_3(w) \cup I_{-2}(v')$. If in addition, the second condition does not hold w.r.t. $u$, then both these radii (in fact all points in $[r_1, r_2]$) intersect the intervals $I_3(w) \cup I_{-2}(v')$. Since $r_2 - r_1 = 4/7$ and all intervals corresponding to vertices in $G'$ have length less than $4/7$, this necessarily means that $r_1 \in I_3(w)$ and $r_2 \in I_{-2}(v')$. Since $\frac{3}{7} \in [r_1, r_2]$, either $\frac{3}{7} \in I_3(w)$ or $\frac{3}{7} \in I_{-2}(v')$. In the first case, $w \in C_2$ by Condition 3, and in the second, $\frac{3}{7} \in I_{-2}(v')$ and so $v' \in C_2$ by Condition 4. $\square$

Before bounding the expected value of the cut, we note that, as before (in Algorithm **Simple-6-Round**), Algorithm **6-Round** also has the property that for every $v \notin C_0$, at most two intervals are responsible for $v$ being in the single radius cut.

**Lemma 5.2.** *For $v \notin C_0$, If $I^{G'}_{-2}(v)$ and $I^{G'}_{2}(v)$ are both defined, then the random radius $r$ is in $C_1$ iff it is in $I^{G'}_{-2}(v) \cup I^{G'}_{2}(v)$.*

*Proof.* Note that in this case, $I^{G'}_{3}(v)$ is also defined. Since all three intervals are defined for $v$ (and $I^{G'}_{1}(v), I^{G'}_{-1}(v)$ are not defined – so $d(s,v), d(v,t) > 1$), there must exist paths $\langle s,u,v\rangle$ and $\langle v,v',t\rangle$ in $G'$, and by the bound on LP values of vertices in $G'$, we have $y_2(v) \leq x_u < 4/7$ and $y_{-2}(v) \geq 1 - x_{v'} > 3/7$. Thus, by the monotonicity of the interval sequence, $I_3(v) \setminus (3/7, 4/7)$ is contained in $(I_{-2}(v) \cup I_2(v)) \setminus (3/7, 4/7)$, and the lemma follows. $\square$

Since this is the only case in which $v$ will be mapped to more than two intervals, this immediately bounds the probability that such a vertex participates in $C_1$:

**Corollary 5.3.** *For every $v \notin C_0$, the probability that $v \in C_1$ is at most $\frac{7}{6} \cdot 2x_v$.*

Now let us bound the expected value of the cut in our final algorithm.

**Lemma 5.4.** *Every vertex $v \in V$ is cut by Algorithm* **6-Round** *with probability at most $(7/4) \cdot x_v$.*

*Proof.* If $x_v \geq 4/7$, this is trivial. Thus, let us look at vertices cut by $C_1$, $C_2$, or $C_3$.

For convenience, define a random set

$$C' = \begin{cases} C_2, & \text{with probability } 1/2 \\ C_3, & \text{otherwise.} \end{cases}$$

Thus the final step in the algorithm can be equivalently stated as returning $C_0 \cup C_1$ w.p. 1/2 and $C_0 \cup C'$ w.p. 1/2.

To analyze the probability that a vertex $v$ in $G'$ will be cut overall, we will consider a number of different of cases for this vertex.

**Case 1: $v$ is mapped to only one interval.** Suppose $I_1(v)$ or $I_{-1}(v)$ are defined. Then $v \in C'$ with probability at most $\frac{7}{3} \cdot x_v$ (by Condition 1 in both $C_2$ and $C_3$). In total, it will be cut with probability at most $\left(\frac{1}{2} \cdot \frac{7}{6} + \frac{1}{2} \cdot \frac{7}{3}\right) x_v = \frac{7}{4} \cdot x_v$. This holds similarly for any vertex $v$ which is not necessarily a neighbor of $s$ or $t$ but which is mapped to only one interval.

**Case 2: $v$ is mapped to at least two intervals, and both $I_{-2}(v)$ and $I_2(v)$ are defined for $v$.** Again, note that this implies that $I_3(v)$ is also defined and $I_1(v), I_{-1}(v)$ are not defined. Recall that $I_3(v) = [y_3(v), y_{-3}(v)]$ and as noted above, in this case we must have $y_{-3}(v) \geq y_{-2}(v) \geq \frac{3}{7}$ and $y_3(v) \leq y_2(v) \leq \frac{4}{7}$. Thus, $I_3(v)$ cannot be entirely to the left of 3/7 or entirely to the right of 4/7. This together with monotonicity of intervals gives us the following implications: Conditions 2 and 3 of $C_2$ each imply $r_1 \in I_3(v) \cap \left[0, \frac{3}{7}\right)$ and Conditions 2 and 3 of $C_3$ each imply $r_2 \in I_3(v) \cap \left[\frac{4}{7}, 1\right)$, giving two possible ways $v$ may be in $C'$:

- $r_1 \in I_3(v) \cap \left[0, \frac{3}{7}\right)$ and $C' = C_2$, or
- $r_2 \in I_3(v) \cap \left[\frac{4}{7}, 1\right)$ and $C' = C_3$.

Note that the probability that at least one of these events occurs is at most $\frac{7}{6} \cdot |I_3(v)|$.

The final reason we may have $v \in C'$ is due to Condition 4 in either $C_2$ or $C_3$. That is, if one of the following occurs:

(a) $r_1 \in I_2(v) \cap \left[0, \frac{3}{7}\right)$ and $C' = C_3$, or

(b) $r_2 \in I_{-2}(v) \cap \left[\frac{4}{7}, 1\right)$ and $C' = C_2$.

Note that each of these two events implies that the given radius ($r_1$ in (a) and $r_2$ in (b)) is in fact in $I_2(v) \cap I_{-2}(v)$. Indeed, take event (b) for example. As we've noted, $y_2(v)$ (the left endpoint of $I_2(v)$) is at most $\frac{4}{7}$, so $r_2$ cannot be to the left of $I_2(v)$. On the other hand, by monotonicity of intervals, since $r \in I_{-2}(v)$, it also cannot be to the right of $I_2(v)$. Thus, $r_2 \in I_2(v) \cap I_{-2}(v)$. Thus, the probability that (a) or (b) occurs is at most $\frac{7}{6} \cdot |I_{-2}(v) \cap I_2(v)|$.

Combining these bounds with Lemma 5.2 for $C_1$, we can bound the total probability that $v$ will be cut by

$$\frac{1}{2} \cdot \frac{7}{6} \cdot |I_2(v) \cup I_{-2}(v)| + \frac{1}{2} \cdot \left(\frac{7}{6} \cdot |I_3(v)| + \frac{7}{6} \cdot |I_2(v) \cap I_{-2}(v)|\right) = \frac{1}{2} \cdot \frac{7}{6} \cdot (|I_3(v)| + |I_2(v)| + |I_{-2}(v)|) \leq \frac{7}{4} \cdot x_v.$$

**Case 3: $v$ is mapped to at two distinct intervals, $I_3(v)$ is defined, but only one of $I_{-2}(v)$, $I_2(v)$ is defined.** As before, because $v$ is mapped to two distinct intervals, Condition 1 for both $C_2$ and $C_3$ is irrelevant. Without loss of generality, assume only $I_2(v)$ and $I_3(v)$ are defined for $v$, and note that Condition 4 for $C_2$ and Condition 2 for $C_3$ are now also irrelevant. Thus, $v$ can be in $C'$ only if Conditions 2 or 3 for $C_2$ or Conditions 3 or 4 for $C_3$ occur. We further divide this case (under this assumption) into several subcases.

**Subcase 3a: $\frac{4}{7} \notin I_2(v)$.** As we've noted, $I_2(v)$ cannot lie entirely to the right of $\frac{4}{7}$, so in this subcase it must lie entirely to the left of $\frac{4}{7}$, and by monotonicity, so must $I_3(v)$. Thus, in fact $v$ can only be in $C'$ when $C' = C_2$. For $C_2$ consider two possibilities. If $\frac{3}{7} \notin I_3(v)$, then only Condition 2 can apply. Otherwise, we have $\frac{3}{7} \in I_3(v)$ and so by monotonicity Condition 2 would imply Condition 3, and so we need only consider Condition 3. Either way, only one interval is responsible for $v$ being in $C'$, and the probability that this occurs is at most $\frac{7}{6} \cdot x_v$. This together with Corollary 5.3 gives the desired bound on the probability that $v$ is cut.

**Subcase 3b: $\frac{4}{7} \in I_2(v)$ and $\frac{3}{7} \in I_3(v)$.** In this subcase, we need to be slightly more precise for $C_1$, and note that the probability that $v \in C_1$ is at most $\frac{7}{6}|I_2(v) \cup I_3(v)|$. As above, because $\frac{3}{7} \in I_3(v)$, Condition 2 for $C_2$ implies Condition 3 for $C_2$, so the relevant conditions are Condition 3 in both $C_2$ and $C_3$, and Condition 4 in $C_3$. The probability that $v \in C'$ because of Condition 3 in either set is clearly at most $\frac{7}{6} \cdot x_v$. Similarly, Condition 4 for $C_3$ also implies that $r_1 \in I_3(v)$, so the probability that $C' = C_3$ and Condition 4 occurs is at most $\frac{7}{6} \cdot |I_2(v) \cap I_3(v)|$. Putting these three bounds together, the probability that $v$ is cut is at most

$$\frac{1}{2} \cdot \frac{7}{6} \cdot |I_2(v) \cup I_3(v)| + \frac{1}{2} \cdot \left(\frac{7}{6} \cdot x_v + \frac{7}{6} \cdot |I_2(v) \cap I_3(v)|\right) = \frac{1}{2} \cdot \frac{7}{6} \cdot (x_v + |I_2(v)| + |I_3(v)|) \leq \frac{7}{4} \cdot x_v.$$

**Subcase 3c: $\frac{4}{7} \in I_2(v)$ and $I_3(v)$ lies to the right of $\frac{3}{7}$.** In this subcase, only Condition 3 for $C_3$ is relevant, and so $v \in C'$ with probability at most $\frac{7}{6} \cdot x_v$, which as we've noted (see Case 3a) is enough.

**Subcase 3d: $\frac{4}{7} \in I_2(v)$ and $I_3(v)$ lies to the left of $\frac{3}{7}$.** Note that in this case we will have $v \in C'$ precisely when $r_1 \in I_2(v) \cap [0, \frac{3}{7}]$ (regardless of whether $C' = C_2$ or $C' = C_3$), which occurs with probability $\frac{7}{3} \cdot |I_2(v) \cap [0, \frac{3}{7}]|$. Thus, we may also assume $\frac{3}{7} \in I_2(v)$, since otherwise we will never have $v \in C'$, and the probability that $v$ is cut will be much better than we require. However, this means that $[\frac{3}{7}, \frac{4}{7}] \subseteq I_2(v)$, so we can improve our probability bounds by not charging for this omitted interval. That is, the probability that $v \in C_1$ is at most $\frac{7}{6} \cdot \left(|(I_2(v) \cup I_3(v)) \cap [0,1]| - \frac{1}{7}\right)$, and the probability that $v \in C'$ is $\frac{7}{3} \cdot (|I_2(v) \cap [0, \frac{4}{7}]| - \frac{1}{7})$.

Now, suppose first that $x_v \leq \frac{3}{7}$. Then by the above bounds, the probability that $v$ is cut is at most

$$\begin{aligned}
\frac{1}{2} \cdot \frac{7}{6} \cdot \left(|I_2(v)| + |I_3(v)| - \frac{1}{7}\right) + \frac{1}{2} \cdot \frac{7}{3} \cdot \left(|I_2(v)| - \frac{1}{7}\right) &\leq \frac{7}{3} \cdot x_v - \frac{1}{4} \\
&\leq \frac{7}{4} \cdot x_v. \qquad \text{since } x_v \leq \frac{3}{7}
\end{aligned}$$

Otherwise, we have $x_v \geq \frac{3}{7}$ (and, since $v \notin C_0$, $x_v < \frac{4}{7}$), and so the proof will follow if we can show that $v$ is cut with probability at most $\frac{3}{4}$. To see that this holds, recall that $I_2(v) = [y_2(v), y_2(v) + x_v]$. There are three ways that $v$ might be cut:

- In $C_1$:
  - If $r \in [0, \frac{3}{7}]$ (where possibly $r \in I_2(v) \cup I_3(v)$) – with probability $\frac{1}{2}$.
  - If $r \in [\frac{4}{7}, y_2(v) + x_v]$ (where $r \in I_2(v)$) – with probability $\frac{7}{6} \cdot \left(y_2(v) + x_v - \frac{4}{7}\right) < \frac{7}{6} \cdot y_2(v)$.
- In $C'$:
  - If $r_1 \in [y_2(v), \frac{3}{7}]$ (where $r_1 \in I_2(v)$) – with probability $\frac{7}{3} \cdot (\frac{3}{7} - y_2(v)) = 1 - \frac{7}{3} \cdot y_2(v)$.

And indeed, as required, the probability that at least one of these events happens is at most

$$\frac{1}{2} \cdot \left(\frac{1}{2} + \frac{7}{6} \cdot y_2(v)\right) + \frac{1}{2} \cdot \left(1 - \frac{7}{3} \cdot y_2(v)\right) = \frac{3}{4} - \frac{7}{6} \cdot y_2(v) \leq \frac{3}{4}.$$

□

# References


[1] J. Adámek and V. Koubek. Remarks on flows in network with short paths. *Commentationes Mathematicae Universitatis Carolinae*, 12(4):661–667, 1971.

[2] K. Altmanová, P. Kolman, and J. Voborník. On polynomial-time combinatorial algorithms for maximum l-bounded flow. In *Proc. of Algorithms and Data Structures Symposium WADS*, pages 14–27, 2019.

[3] 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.

[4] 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.

[5] 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.

[6] 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*, volume 9079 of *LNCS*, pages 47–60, 2015.

[7] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications.* North Holland, New York, 1976.

[8] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439 – 485, 2005.

[9] P. Dvořák and D. Knop. Parametrized 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.

[10] T. Fluschnik, D. Hermelin, A. Nichterlein, and R. Niedermeier. Fractals for kernelization lower bounds. *SIAM J. Discrete Math*, 32(1):656–681, 2018.

[11] 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.

[12] S. Khot, D. Minzer, and M. Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. In *59th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 592–601, 2018.

[13] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2. *Journal of Computer and System Sciences*, 74(3):335 – 349, 2008. Computational Complexity 2003.

[14] P. Kolman. On algorithms employing treewidth for $l$-bounded cut problems. *Journal of Graph Algorithms and Applications*, 22(2):177–191, 2018.

[15] E. Lee. Improved hardness for cut, interdiction, and firefighter problems. In *Proc. of 44rd International Colloquium on Automata, Languages, and Programming (ICALP)*, 2017.

[16] L. Lovász, V. Neumann-Lara, and M. D. Plummer. Mengerian theorems for paths of bounded length. *Periodica Mathematica Hungarica*, 9:269–276, 1978.

[17] 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.

[18] F. Pan and A. Schild. Interdiction problems on planar graphs. *Discrete Applied Mathematics*, 198:215–231, 2016.


# A Hardness of $L$-Bounded Cut for $L \geq 5$

The 1.1377-hardness of approximation result of Baier et al. [3] is based on a reduction from Vertex Cover. Given an instance $G = (V, E)$ of Vertex Cover, they construct an instance $(G', s, t)$ of $L$-Bounded Cut such that, given a vertex cover of size $x$ in $G$, one can efficiently construct an $L$-bounded cut of size $|V| + x$ in $G'$, and vice versa. This yields the following general hardness:

**Theorem A.1** (Implicit in [3])**.** *For any $0 \leq c \leq s \leq 1$, approximating $L$-Bounded Edge Cut for $L \geq 4$ or $L$-Bounded Node Cut for $L \geq 5$ to within $(1+s)/(1+c)$ is at least as hard as Gap Vertex Cover$(c, s)$.*

The 1.1377-hardness of approximation of $L$-Bounded Cut is in fact the hardness one gets from the above theorem by plugging in the NP-hardness of Gap Vertex Cover for $c = (\sqrt{5}-1)/2+\varepsilon$ and $s = (71-31\sqrt{5})/2-\varepsilon$, implied by the work of Dinur and Safra [8].

However, we can also plug in newer or different hardness results for Vertex Cover in the above theorem. For instance, the NP-hardness of Gap Vertex Cover has since been improved to include the case of $c = 1/\sqrt{2}+\varepsilon$, $s = 1-\varepsilon$ [12], which improves the hardness of approximation of $L$-Bounded Cut to $2/(1+1/\sqrt{2}) - \varepsilon < 1.1715$. Also, plugging in the Unique Games hardness of Gap Vertex Cover$(1/2+\varepsilon, 1-\varepsilon)$ [13] gives a $(4/3-\varepsilon)$ Unique Games hardness of approximation for $L$-Bounded Cut, which is now matched by our algorithm for 4-Bounded Edge Cut and 5-Bounded Node Cut.