A Bad Example for Jain's Iterative Rounding Theorem for the Cover Small Cuts Problem

Miles Simmons
Department of Combinatorics and Optimization, University of Waterloo, Canada

Ishan Bansal
Amazon, Bellevue, WA, USA

Joe Cheriyan
Department of Combinatorics and Optimization, University of Waterloo, Canada

Abstract

The Cover Small Cuts problem is a key problem in Network Design. In the Cover Small Cuts problem, we are given a capacitated (undirected) graph $G = (V, E, u)$ and a threshold value λ, as well as a set of links L with end-nodes in V and a non-negative cost for each link $\ell \in L$; the goal is to find a minimum-cost set of links such that each non-trivial cut of capacity less than λ is covered by a link.

Jain's iterative rounding theorem is a well-known result in the area of approximation algorithms and, more broadly, in combinatorial optimization. The theorem asserts that LP relaxations of several problems in network design and combinatorial optimization have the following key property: for every basic feasible solution x there exists a variable x_e that has value at least a constant (e.g., $x_e \geq \frac{1}{2}$).

We construct an example showing that this property fails to hold for the standard LP relaxation of the Cover Small Cuts problem. This indicates that the polyhedron of feasible solutions to the LP (for Cover Small Cuts) differs in an essential way from the polyhedrons associated with several problems in combinatorial optimization. Moreover, our example shows that a direct application of Jain's iterative rounding algorithm does not give an $O(1)$ approximation algorithm for Cover Small Cuts.

We mention that Bansal et al. [6] showed that the WGMV primal-dual algorithm, due to Williamson et al. [19], applied to the same standard LP relaxation achieves approximation ratio 16 for the Cover Small Cuts problem. That is, the WGMV primal-dual algorithm, applied to the same LP relaxation, finds an integer solution of cost ≤ 16 times the optimal value of the LP.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases approximation algorithms, capacitated network design, covering small cuts, edge-connectivity of graphs, f-connectivity problem, iterative rounding method, primal-dual method

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.9

Category APPROX

Related Version *Previous Version*: https://doi.org/10.48550/arXiv.2504.13105

Funding *Ishan Bansal*: This work is external and does not relate to the position at Amazon.
Joe Cheriyan: Supported in part by NSERC grant RGPIN-2024-04473.

Acknowledgements We thank the anonymous reviewers and the Approx 2026 PC for their comments. We are grateful to Sharat Ibrahimpur for reading a preliminary version, and for valuable comments.

1 Introduction

The Cover Small Cuts problem has emerged as a key problem in Network Design since the publication of the first major result in ICALP 2023, see [5, 6]. In the Cover Small Cuts problem, we are given a capacitated (undirected) graph $G = (V, E, u)$ and a threshold value λ, as well as a set of links L with end-nodes in V and a non-negative cost c_ℓ for each link $\ell \in L$; the goal is to find a minimum-cost set of links such that each non-trivial cut of

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2026).
Editors: Mohit Singh and Tom Gur; Article No. 9; pp. 9:1–9:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

capacity less than λ is covered by a link. In more detail, a subset $F \subseteq L$ of links is said to *cover* a node-set S if there exists a link $\ell \in F$ with exactly one end-node in S, and the objective is to find a minimum-cost $F \subseteq L$ that covers each nonempty $S \subsetneq V$ such that $u(\delta_E(S)) < \lambda$. Bansal et al. [6] formulated this problem, and they showed that the WGMV primal-dual algorithm, due to Williamson et al. [19], achieves approximation ratio 16 for this problem. Later, Bansal [2, 3] and Nutov [15] proved that the same algorithm achieves approximation ratio 6. Simmons et al. [18, 17] improved the approximation ratio to 5. Recently, Nutov [16] constructed an example of the Cover Small Cuts problem such that the WGMV primal-dual algorithm finds a solution of cost $(5-\epsilon)$OPT, where OPT denotes the cost of an optimal (integral) solution and $\epsilon > 0$ is an arbitrarily small number. This shows that the 5-approximation analysis of [18, 17] is tight We mention that the same problem is called by other names in the recent literature, e.g., AugSmallCuts [6], Small Cut Augmentation [3], Small Cuts Cover [15], etc.

We may use abbreviations for some standard terms, e.g., we may use Cover Small Cuts as an abbreviation for "the Cover Small Cuts problem".

Given a graph $G = (V, E)$ whose edges have both costs and capacities, a fundamental task in network design is to find a spanning subgraph of minimum cost that satisfies some specified edge-connectivity requirements. *Capacitated network design* refers to the more general (and more challenging) setting where edges have non-uniform capacities. This topic has attracted intense research efforts over the last few years; see below for discussion and citations. In the context of algorithmic research and combinatorial optimization, the Cap-k-ECSS problem sits in the core of capacitated network design. In this problem, we are given an (undirected) graph $G = (V, E)$, where each edge has a non-negative cost and a positive integer capacity, and the algorithmic goal is to find a minimum-cost spanning subgraph such that every non-trivial cut has capacity k or more. Let $n = |V|$ and let u_{min} (respectively, u_{max}) denote the minimum (respectively, maximum) capacity of an edge; assume that $u_{max} \le k$. One of the earliest approximation algorithms for Cap-k-ECSS is due to Goemans et al. [10], and it achieves an approximation ratio of $\min\{2k, |E|\}$. Over the last three decades, there have been several advances, and the best approximation ratio known is $\min(O(\log n), O(\log(k/u_{min})), k, 2u_{max})$, due to, respectively, Chakrabarty et al. [8], Bansal et al. [4], and Boyd et al. [7]. Let us mention that there are no known hardness results that rule out the possibility of better asymptotic approximation ratios. Most of the advances of this decade on Cap-k-ECSS are based on $O(1)$ approximation algorithms for Cover Small Cuts.

For the benefit of readers who are not familiar with the papers cited above, let us sketch a simple reduction from Cap-k-ECSS to Cover Small Cuts that incurs a multiplicative factor of $\lceil \frac{k}{u_{min}} \rceil$ in the approximation ratio, see [6, section 5] for further details, and see section 1.3 below for our notation. The algorithm is as follows: Initialize the "solution" edge-set F to be the empty set; during the execution of the algorithm, $G_F = (V, F)$ is our current "solution graph." While the minimum capacity of a cut $\delta_F(S)$, $\emptyset \neq S \subsetneq V$, in (V, F) is less than k, run an α-approximation algorithm for Cover Small Cuts with input $G_F = (V, F)$ and with link-set $L := E - F$, to augment all cuts $\delta_F(S)$, $\emptyset \neq S \subsetneq V$, with $u(\delta_F(S)) < k$ and obtain a valid augmentation $F' \subseteq L$. Update F by adding F', that is, $F := F \cup F'$. On exiting the while loop, output the edge-set F. Bansal et al. [6, Theorem 1.3] show that the cost of F is $\le \alpha \cdot \lceil \frac{k}{u_{min}} \rceil \cdot$ OPT, where OPT denotes the optimal cost for the Cap-k-ECSS instance.

1.1 Jain's iterative rounding method and the f-connectivity problem

Jain's iterative rounding method [12] and its extensions such as iterative relaxation [14] form a key theoretical tool for proving approximation guarantees, as well as characterizing optimal solutions of well-known (poly-time solvable) problems in combinatorial optimization.

Several connectivity augmentation problems can be formulated in a general framework called f-connectivity. In this problem, we are given an undirected graph $G = (V, E)$ on n nodes with nonnegative costs $c \in \mathbb{Q}^E_{\geq 0}$ on the edges and a requirement function $f : 2^V \to \mathbb{Z}_{\geq 0}$ on subsets of nodes. The algorithmic goal is to find an edge-set $J \subseteq E$ with minimum cost $c(J) := \sum_{e \in J} c_e$ such that for all cuts $\delta(S)$, $S \subseteq V$, we have $|\delta(S) \cap J| \geq f(S)$. A function f is called *weakly supermodular* if $f(V) = 0$, and for all $A, B \subseteq V$, either $f(A)+f(B) \leq f(A-B)+f(B-A)$, or $f(A)+f(B) \leq f(A\cap B)+f(A\cup B)$. The following is an LP relaxation for the f-connectivity problem.

$$
\begin{aligned}
\min \quad & \sum_{e\in E} c_e x_e & & \text{(LP: f-connectivity)}\\
\text{subject to:} \quad & \sum_{e\in E\cap\delta(S)} x_e \geq f(S) & \forall\, S \subset V & \\
& 0 \leq x_e \leq 1 & \forall\, e \in E. &
\end{aligned}
$$

Assuming that the function f is weakly supermodular, integral, and has a positive value for some $S \subset V$, Jain [12] presented a 2-approximation algorithm for the f-connectivity problem, based on the following key result.

▶ **Theorem 1** (Jain [12, Theorem 3.1]). *Assume that the function f is weakly supermodular, integral, non-negative and non-zero. Then, in any basic feasible solution x to the LP relaxation, for at least one edge, e, x_e is at least $1/2$.*

The Jain property

For the sake of notational convenience, let us define a particular LP relaxation of a combinatorial problem to have the *Jain property* if there exists a constant c_0 such that every basic feasible solution x of the LP relaxation has an entry x_e such that $x_e \geq 1/c_0$ (i.e., the infinity norm of x is lower-bounded by a constant).

1.2 Our main result: A bad extreme point for the LP relaxation of Cover Small Cuts

We construct an example showing that the Jain property fails to hold for the standard LP relaxation of the Cover Small Cuts problem, and we mention that the WGMV primal-dual algorithm applied to the *same* LP relaxation finds an integer solution of cost $\leq 6 \cdot \mathrm{LP}_{\mathrm{opt}}$. This indicates that the polyhedron of feasible solutions to the LP relaxation (for Cover Small Cuts) differs in an essential way from the polyhedrons associated with several problems in combinatorial optimization. Moreover, our example shows that a direct application of Jain's iterative rounding algorithm does not give an $O(1)$ approximation algorithm for Cover Small Cuts.

Consider an instance $G = (V, E, u),\ \lambda$ of the Cover Small Cuts problem. Let $\mathcal{C} = \{\emptyset \neq S \subsetneq V : u(\delta_E(S)) < \lambda\}$. Then we have the following covering LP relaxation of the problem.

$$\begin{aligned} \mathrm{LP_{opt}} := \min \quad & \sum_{f\in L} c_f x_f & & \text{(LP: Cover Small Cuts)} \\ \text{subject to:} \quad & \sum_{f\in L\cap\delta(S)} x_f \geq 1 & \forall\ S \in \mathcal{C} & \\ & 0 \leq x_f & \forall\ f \in L. & \end{aligned}$$

▶ Remark 2. Let $f_{\mathrm{CSC}} : 2^V \to \{0,1\}$ be the requirement function for Cover Small Cuts; thus, for $S \subseteq V$, $f_{\mathrm{CSC}}(S) = 1$ iff $S \in \mathcal{C}$. Although f_{CSC} is *not* weakly supermodular, f_{CSC} is a pliable function that satisfies property (γ), see [6] for definitions, and the $O(1)$ approximation analysis of [6] relies on this.

▶ **Theorem 3.** *Let $k \geq 4$ be an even integer. There is an instance of the* Cover Small Cuts *problem $G = (V, E), L, u$ such that the LP relaxation* (LP: Cover Small Cuts) *has a basic feasible solution x^* such that every positive variable of x^* has value $1/k$.*

Discussion and contrast with previous results

Jain's iterative rounding theorem [12] (see Theorem 1 above) asserts that for the LP relaxation of f-connectivity, whenever f is weakly supermodular, every basic feasible solution x has at least one variable $x_e \geq \frac{1}{2}$. This Jain property is the lever behind Jain's 2-approximation algorithm and has been applied widely in network design; see [14] for a comprehensive treatment. A natural question is whether the Jain property extends beyond the weakly supermodular setting, in particular, to LP relaxations of problems with *node-connectivity* requirements. The short answer is no, and the previous literature shows that there are (at least) two distinct reasons.

The first, and fundamental, reason comes from the area of hardness of approximation. Kortsarz, Krauthgamer, and Lee [13] proved that the node-connectivity Survivable Network Design Problem (NC-SNDP) admits no polynomial-time $O(1)$-approximation algorithm unless $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{O(\log\log n)})$. Since the natural LP relaxation of NC-SNDP can be solved in polynomial time via a separation oracle based on max-flow computations, this hardness result implies – by a direct contrapositive argument – that the Jain property cannot hold for that LP: if every basic feasible solution x had some variable $x_e \geq 1/c_0$ for a fixed positive constant c_0, then iterating Jain's rounding scheme would yield a polynomial-time $O(1)$-approximation algorithm, contradicting the hardness result of [13].

The second reason is constructive: one can show directly that the standard LP relaxations for certain node-connectivity problems do not have the Jain property; nevertheless, it is possible that the underlying problem may admit a good approximation algorithm via a different LP relaxation or method. Fleischer, Jain, and Williamson [9] studied iterative rounding for minimum-cost node-connectivity problems and showed that it gives a 2-approximation for the *element connectivity* problem – a relaxation of node connectivity in which two "source-sink" paths are permitted to share so-called terminal nodes (but are required to be disjoint on the edges and the Steiner nodes). Their analysis establishes that the Jain property holds for the standard LP relaxation of element connectivity. Crucially, however, their result does not extend to the node-connectivity setting: the requirement function for node connectivity is not weakly supermodular, and the LP does not inherit the Jain property. For the mincost k-node-connected spanning subgraph problem (mincost k-NCSS), which is the core problem of node-connectivity network design, Aazami, Cheriyan, and Laekhanukit [1] gave an explicit family of bad examples for the standard iterative rounding method. Specifically, they constructed instances for which the setpair LP relaxation has a basic feasible solution x satisfying $\max_e x_e \leq O(1)/\sqrt{k}$, thereby showing that the standard iterative rounding method cannot achieve an approximation ratio better than $\Omega(\sqrt{k})$ for mincost k-NCSS.

Our main result, Theorem 3, differs from the previous results in two essential ways.

(a) Edge-connectivity versus node-connectivity. The results of [13, 9, 1] all concern problems with *node-connectivity* requirements: the requirement function counts internally node-disjoint paths (or a relaxation thereof). Our bad example, by contrast, applies to the Cover Small Cuts problem, which has *edge-connectivity* requirements; recall that the goal is to find a minimum-cost set of links such that every cut of edge-capacity less than λ is covered by at least one link. The standard LP relaxation of Cover Small Cuts (see (LP: Cover Small Cuts)) is a covering LP of the same type as the LP for f-connectivity, yet our construction shows that the Jain property does *not* hold for it. To the best of our knowledge, there is no prior bad example for the Jain property for any problem defined by uniform edge-connectivity requirements.

(b) Hardness versus polyhedral structure. A key distinction between our result and the hardness-based argument of [13] is that Cover Small Cuts *does* admit a polynomial-time $O(1)$-approximation algorithm [6, 3, 15], achieved via the primal-dual method of Williamson et al. [19], rather than iterative rounding. Thus the failure of the Jain property for Cover Small Cuts cannot be attributed to hardness of approximation. This separates Cover Small Cuts from NC-SNDP: for NC-SNDP, the Jain property fails (in part) because the problem is hard to approximate; for Cover Small Cuts, the problem allows an $O(1)$ approximation algorithm, but the known iterative rounding methods via the natural covering LP cannot achieve an $O(1)$ approximation ratio. In the context of network design on undirected graphs with uniform edge-connectivity requirements, our construction indicates that the "limits" of the iterative rounding method are "tighter" than what is indicated by the previous literature.

1.3 Basic notation and definitions

For a positive integer k, we use $[k]$ to denote the set $\{1, 2, \ldots, k\}$. A pair of subsets A, B of V (the ground-set) is said to *cross* if each of the four sets $A \cap B, V - (A \cup B), A - B, B - A$ is nonempty.

For a graph $G = (V, E)$ and $S \subseteq V$, the cut $\delta_E(S)$ refers to the set of edges that have exactly one end-node in S; $\delta_E(S)$ is called a non-trivial cut if $\emptyset \neq S \subsetneq V$.

For each of the minimization problems (in network design or flexible graph connectivity), we use OPT to denote the optimal value (i.e., the minimum cost of an integer solution), and $\mathrm{LP_{opt}}$ to denote the optimal value of an LP relaxation. The context will resolve potential ambiguities.

2 The Capacitated Graph

Let $k \geq 4$ be an even integer. The capacitated graph $G = (V, E), u$ has $n = 2 + \binom{k}{2}$ nodes, and $m = (n-1) + (k-1) = \binom{k}{2} + k$ edges. The edge capacities are chosen from the set $\{1, 2, 3\}$ and λ (the threshold value for small cuts) is chosen to be 5.

We denote the nodes of G by $v_1, v_2, \ldots, v_n$, as well as by the indices $1, 2, \ldots, n$. For notational convenience, let $s = v_1$ and let $t = v_n$.

The family of small cuts consists of two sub-families. One sub-family corresponds to the node-sets $N_1 = \{v_1\}, N_2 = \{v_1, v_2\}, \ldots, N_{n-1} = \{v_1, v_2, \ldots, v_{n-1}\}$. In other words, for each of the sets $N_i = \{v_1, v_2, \ldots, v_i\}, i = 1, \ldots, n-1$, $\delta_E(N_i)$ is a small cut. We call the sets N_i $(i = 1, \ldots, n-1)$ the *nested sets*. The other sub-family corresponds to the $(k-1)$ node-sets $Q_1 = \{v_2, v_3, \ldots, v_{1+\frac{k}{2}}\}, Q_2 = \{v_{2+\frac{k}{2}}, v_{3+\frac{k}{2}}, \ldots, v_{1+2\cdot\frac{k}{2}}\}, \ldots, Q_{(k-1)} = \{v_{2+(k-2)\cdot\frac{k}{2}}, v_{3+(k-2)\cdot\frac{k}{2}}, \ldots, v_{1+(k-1)\cdot\frac{k}{2}}\}$. Note that these sets form a partition of $V(G)-\{s,t\}$ into $(k-1)$ sets, each of cardinality $k/2$. In other words, for each of the $(k-1)$ "intervals" Q_j $(j = 1, \ldots, k-1)$ formed by picking the node with index $2 + (j-1)\cdot\frac{k}{2}$ and the next $(\frac{k}{2} - 1)$ consecutively indexed nodes, $\delta_E(Q_j)$ is a small cut. We call the sets Q_j $(j = 1, \ldots, k-1)$ the *Q-sets*; moreover, by the *first node* of Q_j we mean the node of lowest index in Q_j (i.e., node $2 + (j-1)\cdot\frac{k}{2}$), and by the *last node* of Q_j we mean the node of highest index in Q_j (i.e., node $1 + (j)\cdot\frac{k}{2}$). For notational convenience, let $Q_0 = \{s\} = \{v_1\}$ and let $Q_k = \{t\} = \{v_n\}$; moreover, let $s = v_1$ be the *last node* of Q_0 and let $t = v_n$ be the *first node* of Q_k. (Consequently, in the proofs below, s may only serve as the last node of a Q-set, and t may only serve as a first node. Thus, if we examine the case that a specified node v is a first node, it is guaranteed that $v \neq s$ (i.e., the Q-set containing v is not Q_0); similarly, if we examine the case that a specified node v is a last node, it is guaranteed that $v \neq t$ (i.e., the Q-set containing v is not Q_k)).

For a nested set N_i $(i = 1, \ldots, n-1)$ and a Q-set Q_j $(j = 1, \ldots, k-1)$, observe that $s = v_1 \in N_i - Q_j$ and $t = v_n \notin N_i \cup Q_j$; moreover, if $N_i \cap Q_j$ is nonempty, then either Q_j crosses N_i (i.e., both $Q_j - N_i$ and $Q_j \cap N_i$ are nonempty) or $Q_j \subsetneq N_i$.

There are two types of edges in G. There are $n-1$ edges that form the s,t-path $v_1, v_2, \ldots, v_n$; we use E_1 to denote the edge set of this path. The first edge and the last edge of this path have capacity 2, thus, $u_{v_1 v_2} = 2 = u_{v_{n-1} v_n}$. Consider any other edge $v_i v_{i+1}$ $(i = 2, \ldots, n-2)$ of this path. If both v_i and v_{i+1} are in the same Q-set then $v_i v_{i+1}$ has capacity 3, otherwise, $v_i v_{i+1}$ has capacity one. Moreover, there are $(k-1)$ other edges (that are not in the s,t-path), and each of these edges has capacity one; we use E_2 to denote this set of $(k-1)$ edges. The nodes incident to the j-th edge of E_2 $(j = 1, \ldots, k-1)$ have indices $1 + (j-1)\cdot\frac{k}{2}$ and $2 + j\cdot\frac{k}{2}$. Thus, E_2 has an edge between $s = v_1$ and the first node of Q_2, an edge between the last node of Q_{k-2} and $t = v_n$, and, for each $j = 1, \ldots, k-3$, E_2 has an edge between the last node of Q_j and the first node of Q_{j+2}.

The following figure illustrates the construction of the capacitated graph for $k = 4$; the graph has $\binom{k}{2} + 2 = 8$ nodes and $\binom{k}{2} + k = 10$ edges.

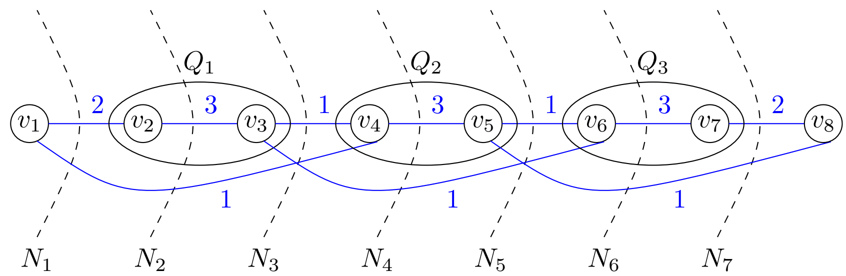

Figure 1 The capacitated graph for $k = 4$. The small cuts are given by the nested sets $N_1, \ldots, N_7$ (indicated by dashed lines) and the sets Q_1, Q_2, Q_3 (indicated by ovals).

We defer the proofs of the next two results to section 5; our proofs are straightforward and use case analysis.

▶ **Proposition 4.** *Each of the nested cuts $\delta(N_i), i = 1, \ldots, n-1$ and each of the Q-cuts $\delta(Q_j), j = 1, \ldots, k-1$ has capacity less than $\lambda = 5$.*

▶ **Proposition 5.** *Every non-trivial cut $\delta(S)$, $\emptyset \subsetneq S \subsetneq V$, of G that is neither a nested cut nor a Q-cut has capacity $\geq \lambda = 5$.*

3 The Links Graph

Let (V, L) denote the graph of the links; thus L is the set of links, and we denote the links by the symbol ℓ, e.g., ℓ_i, ℓ', etc. The number of links is $m = n + k - 2 = \binom{k}{2} + k$. The links are partitioned into k internally node disjoint s,t-paths that we denote by $P_1, \ldots, P_{k-1}, P_k$. The s,t-path P_k has one link $st = v_1 v_n$. Each of the s,t-paths $P_1, \ldots, P_{k-1}$ has $k/2$ internal nodes and has $1 + \frac{k}{2}$ links. Moreover, the sequence of node indices of each of these s,t-paths is an increasing sequence; that is, the node sequence has the form $v_1, v_{i_2}, v_{i_3}, \ldots, v_{i_{(\frac{k}{2}+1)}}, v_n$, where $1 < i_2 < i_3 < \cdots < i_{(\frac{k}{2}+1)} < n$. Each of the nodes $v_2, \ldots, v_{n-1}$ is in one of the s,t-paths $P_1, \ldots, P_{k-1}$; thus, each of these nodes is incident to two links.

Recall that we have $(k-1)$ Q-sets, each with $k/2$ nodes, and $(k-1)$ s,t-paths $P_1, \ldots, P_{k-1}$ each with $k/2$ internal nodes. Each of the s,t-paths P_i $(i = 1, \ldots, k-1)$ is constructed such that it is incident to precisely one node of $k/2$ (of the $(k-1)$) Q-sets, and it is disjoint from the other $(k - 1 - \frac{k}{2})$ Q-sets. The s,t-path P_i $(i = 1, \ldots, k-1)$ is incident to the Q-set with index i and the next $(k/2) - 1$ consecutively indexed Q-sets (with "wrap around"); thus, for $i = 1, \ldots, k-1$ and $j = 1, \ldots, k-1$, the s,t-path P_i and the set Q_j have a node in common iff

$$
\begin{aligned}
& j \geq i \quad \textbf{and} \quad j \leq \min(i + \frac{k}{2} - 1,\ k-1), \quad \textbf{or} \\
& j < i \quad \textbf{and} \quad 1 \leq j \leq i - \frac{k}{2}.
\end{aligned}
$$

If an s,t-path P_i and a set Q_j have a node in common, then any one node of Q_j could be assigned to P_i; in other words, our construction does not specify the internal nodes of P_i, it only specifies the Q-sets that intersect P_i.

Let A^{PQ} denote the $(k-1) \times (k-1)$ zero-one matrix that has a row for each s,t-path P_i $(i = 1, \ldots, k-1)$ and a column for each set Q_j $(i = 1, \ldots, k-1)$ such that the entry $A^{PQ}_{i,j}$ is one iff P_i and Q_j have a node in common. Then, by construction, A^{PQ} is a circulant matrix with entries of zero or one such that each row has $k/2$ consecutive ones (with "wrap around") and the remaining $(k - 1 - \frac{k}{2})$ entries of the row are zero. Observe that the number of ones in each row, $k/2$, and the size of the matrix, $k-1$, are relatively prime (i.e., $\gcd(\frac{k}{2}, k-1) = 1$). For an illustration, see the example at the end of section 4.

▶ **Lemma 6.** *The matrix A^{PQ} has rank $(k-1)$. Moreover,* $\det(A^{PQ}) = k/2$*.*

One can prove this lemma directly, by applying row operations (and applying the matrix determinant lemma [1] to a square matrix with diagonal entries 2 and all other entries one). Alternatively, the lemma follows directly from a result of Hariprasad [11, Theorem 2.1].

Figure 2 illustrates the construction of the links graph for $k = 4$. The graph has $\binom{k}{2} + 2 = 8$ nodes and $\binom{k}{2} + k = 10$ links. The links partition into $k = 4$ (internally node disjoint) s,t-paths $P_1, \ldots, P_4$; the links of $P_1, \ldots, P_4$ are indicated by distinct colours.

[1] https://en.wikipedia.org/wiki/Matrix_determinant_lemma

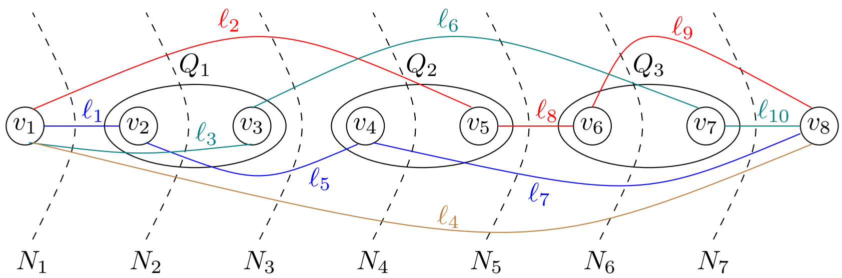

Figure 2 The links graph for $k = 4$. The links partition into s,t-paths $P_1, \ldots, P_4$, and the links of each of these s,t-paths is indicated by a distinct colour. Thus, the links of P_1 are ℓ_1, ℓ_5, ℓ_7, the links of P_2 are ℓ_2, ℓ_8, ℓ_9, the links of P_3 are $\ell_3, \ell_6, \ell_{10}$, and P_4 has the link ℓ_4. The small cuts are given by the nested sets $N_1, \ldots, N_7$ (indicated by dashed lines) and the sets Q_1, Q_2, Q_3 (indicated by ovals).

We construct a basic feasible solution x^* by assigning the value $1/k$ to each link, thus, $x^*_\ell = 1/k, \forall \ell \in L$. Informally speaking, we are assigning a value $1/k$ to each of the s,t-paths $P_1, \ldots, P_k$. It is easy to verify that each of the small cuts, namely, $\delta(N_i)$ $(i = 1, \ldots, n-1)$ and $\delta(Q_j)$ $(j = 1, \ldots, k-1)$, is "tightly covered" by x^*. First, consider a cut $\delta(N_i)$ $(i = 1, \ldots, n-1)$, and observe that each of the s,t-paths $P_1, \ldots, P_k$ has exactly one of its links in $\delta(N_i)$, hence, $x^*(\delta(N_i)) = (k)\frac{1}{k} = 1$. Next, consider a cut $\delta(Q_j)$ $(j = 1, \ldots, k-1)$, and observe that $k/2$ of the s,t-paths $P_1, \ldots, P_k$ have exactly one internal node in Q_j, hence, each of these s,t-paths has exactly two links in $\delta(Q_j)$, therefore, $x^*(\delta(Q_j)) = 2(\frac{k}{2})(\frac{1}{k}) = 1$. Note that the tight constraints at x^* are the covering constraints for the small cuts, and none of the lower bound constraints is tight (since $0 < 1/k < 1$).

The basis matrix A of x^* has one row for each small cut and one column for each link; the entry of A corresponding to a small cut $\delta(S)$ and a link ℓ is one if $\ell \in \delta(S)$, otherwise, the entry is zero. Thus, A is an $m \times m$ matrix with entries of zero or one; recall that $m = n + k - 2 = \binom{k}{2} + k$. In the next section, we show that matrix A has full rank, and that will prove our claim that x^* is a basic feasible solution of the LP relaxation of Cover Small Cuts.

4 The rank of the incidence matrix of small cuts and links

Let A denote the zero-one incidence matrix of the small cuts and the links; A is an $m \times m$ matrix, where $m = n + k - 2 = \binom{k}{2} + k$. The matrix A has rows corresponding to the cuts of the Q-sets followed by rows corresponding to the cuts of the nested sets; the columns of A correspond to the links. In this section, the main goal is to show that $\det(A)$ is non-zero; equivalently, A has full rank. We start by fixing a convenient indexing for the rows and columns of A. The rows $1, \ldots, (k-1)$ of A correspond to the cuts of $Q_1, \ldots, Q_{(k-1)}$, and the rows $k, (k+1), \ldots, m$ of A correspond to the cuts of $N_1, \ldots, N_{(n-1)}$. The links $\ell \in L$ are indexed as follows: The link of the s,t-path P_i $(i = 1, \ldots, k)$ incident to node $s = v_1$ is denoted ℓ_i; thus, each of $\ell_1, \ldots, \ell_k$ is in $\delta(s)$, and there is a bijection between these links and the s,t-path $P_1, \ldots, P_k$. Note that each of the remaining links is incident to one of the nodes $v_2, \ldots, v_{n-1}$, and, for each $i = 2, \ldots, (n-1)$, the unique link that is incident to node v_i and is in the cut $\delta(N_i)$ is assigned the index $(k + i - 1)$. The columns $1, \ldots, m$ of A correspond to the links $\ell_1, \ldots, \ell_m$.

Let us view the matrix A as a 2×2 block matrix $\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ where A_{11} is the $(k-1) \times (k-1)$ sub-matrix of A whose rows correspond to the cuts of $Q_1, \ldots, Q_{(k-1)}$ and whose columns correspond to the links $\ell_1, \ldots, \ell_{(k-1)}$, and A_{22} is the $(m-k+1) \times (m-k+1)$ sub-matrix of A whose rows correspond to the cuts of $N_1, \ldots, N_{(n-1)}$ and whose columns correspond to the links $\ell_k, \ell_{(k+1)}, \ldots, \ell_m$.

▷ Claim 7. The sub-matrix A_{22} is lower-triangular and all entries on the diagonal are one; thus, $\det(A_{22}) = 1$.

Proof. The link ℓ_k (with end nodes s and t) is in each of the cuts $\delta(N_i)$, $i = 1, \ldots, (n-1)$. For each of the links $\ell_{(k+i-1)}$ where $i = 2, \ldots, (n-1)$, note that the link is in the cut $\delta(N_i)$ and it is in none of the cuts $\delta(N_1), \ldots, \delta(N_{(i-1)})$. ◁

Our plan is to apply row operations to the first $(k-1)$ rows of A in order to transform the submatrix $\begin{pmatrix} A_{11} & A_{12} \end{pmatrix}$ to a submatrix of the form $\begin{pmatrix} (A^{PQ})^\top & \mathbf{0}_{(k-1)\times(m-k+1)} \end{pmatrix}$, where $(A^{PQ})^\top$ denotes the transpose of the matrix A^{PQ} and $\mathbf{0}_{(k-1)\times(m-k+1)}$ denotes a $(k-1) \times (m-k+1)$ matrix of zeros. Therefore, using row operations, we will rewrite the matrix A as the 2×2 block matrix $\begin{pmatrix} (A^{PQ})^\top & \mathbf{0}_{(k-1)\times(m-k+1)} \\ A_{21} & A_{22} \end{pmatrix}$; then, by applying Lemma 6 as well as a result on the determinant of 2×2 block matrices, we see that the determinant of the re-written matrix A is $\det(A^{PQ}) \det(A_{22}) = \det(A^{PQ}) \neq 0$.

See the example at the end of this section for an illustration.

Let us introduce some notation to describe and analyse our row operations. For any set of links L' let $\chi^{L'}$ denote the zero-one row incidence vector of L'; thus, for any link ℓ, $\chi^{L'}_\ell = 1$ iff $\ell \in L'$. Note that rows $1, \ldots, (k-1)$ of A correspond to $\chi^{\delta(Q_1)}, \ldots, \chi^{\delta(Q_{(k-1)})}$. For any link ℓ, let $\phi(\ell)$ denote the index i of the s,t-path P_i that contains ℓ. For each of the links ℓ_i, $i = 1, \ldots, (k-1)$, note that $\phi(\ell_i) = i$, since we have a bijection between these links and the s,t-paths $P_1, \ldots, P_{(k-1)}$. For any set of links L', let $\Phi(L')$ denote the set of indices of the s,t-paths $P_1, \ldots, P_{(k-1)}$ that contain at least one of the links of L'; thus, $\Phi(L') \subseteq \{1, \ldots, k\}$, and index i is in $\Phi(L')$ iff the s,t-path P_i contains one of the links of L'. For example, if L' is a subset of $\{\ell_1, \ldots, \ell_{(k-1)}\}$, then $\Phi(L') = \{i \;:\; \ell_i \in L'\}$.

Consider any row j, $j = 1, \ldots, (k-1)$ of the matrix A; this row-vector is the same as $\chi^{\delta(Q_j)}$. Let g denote the largest index i such that the nested set N_i is disjoint from Q_j, and let h denote the smallest index i such that the nested set N_i contains Q_j. (By our construction, the $k/2$ nodes of Q_j are in the interval $[(g+1), (g+2), \ldots, h]$.)

▷ Claim 8. $\left(\chi^{\delta(Q_j)} - \chi^{\delta(N_h)} + \chi^{\delta(N_g)}\right) = 2\chi^{L'}$, where L' is the set of $k/2$ links in $\delta(Q_j) \cap \delta(N_g)$.

Proof. By construction, every link has its two end nodes in two different Q-sets. Hence, every link $\ell \in \delta(Q_j)$ that has its lower-index node in Q_j is in $\delta(N_h)$; similarly, every link $\ell \in \delta(Q_j)$ that has its higher-index node in Q_j is in $\delta(N_g)$. Moreover, $\delta(N_g) - \delta(Q_j) = \delta(N_h) - \delta(Q_j)$ is the set of links that have their lower-index node in N_g and their higher-index node in $V - N_h$. The equation in the claim follows from these statements. ◁

In the above claim, note that L' is contained in the cut of a Q-set, hence, the link $\ell_k = st$ cannot be in L'.

Our plan is to (separately) apply the row operations described in the above claim to each of the rows $j = 1, \ldots, (k-1)$ of A. The next lemma explains how to apply row operations to $\chi^{L'}$, where L' is any set of $k/2$ links contained in some nested cut $\delta(N_i)$ and $\ell_k \notin L'$, to transform $\chi^{L'}$ to the vector $\chi^{\widehat{L}}$, where $\widehat{L} = \{\ell_i \;:\; i \in \Phi(L')\}$. Informally speaking, we can apply row operations to "map" any row vector $\chi^{L'}$, where $L' \subset \delta(N_i) - \{\ell_k\}$ and $|L'| = k/2$, to the row incidence vector (padded with zeros on the right) of the $k/2$ s,t-paths among $P_1, \ldots, P_{(k-1)}$ that contain (at least) one of the links of L'. Thus, by applying the above claim and the next lemma, we "map" each row of the submatrix $\begin{pmatrix} A_{11} & A_{12} \end{pmatrix}$ to a distinct row of the submatrix $\begin{pmatrix} (A^{PQ})^\top & \mathbf{0}_{(k-1)\times(m-k+1)} \end{pmatrix}$.

▶ **Lemma 9.** *Let $i = 1, \ldots, n-1$, and let $L' \subset \delta(N_i) - \{\ell_k\}$ be any set of $k/2$ links. By applying row operations, one can transform the vector $\chi^{L'}$ to the vector $\chi^{\widehat{L}}$, where $\widehat{L} = \{\ell_i \,:\, i \in \Phi(L')\}$.*

Proof. By induction on the index of the nested set whose cut contains L'.

Let h be the smallest index such that the nested cut $\delta(N_h)$ contains L'. If $h = 1$, then we have $\widehat{L} = L'$ and the lemma is proved.

Now, suppose $h \geq 2$; thus, L' contains a link ℓ_j with $j > k$. Observe that L' contains the link $\ell_{(h+k-1)}$ which is incident to node v_h and is in the cut $\delta(N_h)$; informally speaking, the presence of $\ell_{(h+k-1)}$ in L' certifies that L' is contained in $\delta(N_h)$ and L' is not a subset of $\delta(N_j)$ for any $j < h$. Let $\overline{L}$ denote the set of links $\delta(N_h) - L'$; $\overline{L}$ is a "complementary set" of $k/2$ links w.r.t. $\delta(N_h)$ and L'. For every link $\ell_j \in \overline{L}$, observe that $j < (h + k - 1)$ (since $\ell_j \in \delta(N_h)$ and $\ell_{(h+k-1)} \notin \overline{L}$).

Let g be the smallest index such that the nested cut $\delta(N_g)$ contains $\overline{L}$. Again, observe that $\overline{L}$ contains the link $\ell_{(g+k-1)}$ which is incident to node v_g and is in the cut $\delta(N_g)$. Clearly, $g < h$ and $\delta(N_g) \neq \delta(N_h)$, since $\ell_{(h+k-1)} \in \delta(N_h)$ and $\ell_{(h+k-1)} \notin \delta(N_g)$.

▷ Claim 10. There exists a set of $k/2$ links of $\delta(N_g) - \{\ell_k\}$, call it L'', such that
(i) $\chi^{L''} = \chi^{L'} - \chi^{\delta(N_h)} + \chi^{\delta(N_g)}$, and
(ii) $\Phi(L'') = \Phi(L')$.

Proof. Observe that $\overline{L} = \delta(N_h) - L'$, and that $\overline{L}$ is contained in $\delta(N_g)$; also, note that ℓ_k is in $\overline{L}$. Let $L'' = \delta(N_g) - \overline{L}$; observe that L'' is a set of $k/2$ links and $\ell_k \notin L''$. Hence, we have $\chi^{\delta(N_g)} - \chi^{\delta(N_h)} = \chi^{L''} - \chi^{L'}$. This proves part (i) of the claim.

Moreover, note that each of the nested cuts contains exactly one link from each of the s,t-paths $P_1, \ldots, P_k$; that is, for each $i = 1, \ldots, n-1$, $\Phi(\delta(N_i)) = \{1, \ldots, k\}$. Hence, $\Phi(L') = \Phi(\delta(N_h) - \overline{L}) = \{1, \ldots, k\} - \Phi(\overline{L}) = \Phi(\delta(N_g) - \overline{L}) = \Phi(L'')$. This proves part (ii) of the claim. ◁

In the induction step (of the proof of the lemma), we apply the above claim to replace the set L' by the set L''; note that L'' has the properties of L' (listed in the lemma statement) and $\Phi(L'') = \Phi(L')$. Moreover, L'' is contained in the cut of a nested set of smaller index than the index of any nested set whose cut contains L'.

Thus, the lemma follows by induction. ◀

Our main result of this section follows from the above discussion.

▶ **Theorem 3.** *Let $k \geq 4$ be an even integer. There is an instance of the* Cover Small Cuts *problem $G = (V, E), L, u$ such that the LP relaxation* (LP: Cover Small Cuts) *has a basic feasible solution x^* such that every positive variable of x^* has value $1/k$.*

4.1 Example with $k = 4$

In this subsection, we fix $k = 4$, and we illustrate our row operations, in particular, the row operations of Claims 8,10, on the links graph shown in Figure 2. The links partition into s,t-paths $P_1, \ldots, P_4$, and the links of P_1 are ℓ_1, ℓ_5, ℓ_7, the links of P_2 are ℓ_2, ℓ_8, ℓ_9, the links of P_3 are $\ell_3, \ell_6, \ell_{10}$, and P_4 has the link ℓ_4.

The incidence matrix A^{PQ} of the s,t-paths P_1, P_2, P_3 and the Q-sets is given below.

$$
\begin{array}{c}
\begin{array}{ccc} Q_1 & Q_2 & Q_3 \end{array} \\
\left(\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{array}\right)
\end{array}
\begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \end{array}
$$

The incidence matrix A of the small cuts and the links is given below. Note that the submatrix A_{22} (of the nested cuts and the links $\ell_4, \ldots, \ell_{10}$) is lower-triangular with diagonal entries of one.

$$
\begin{array}{c}
\begin{array}{ccc|cccccccc} \ell_1 & \ell_2 & \ell_3 & \ell_4 & \ell_5 & \ell_6 & \ell_7 & \ell_8 & \ell_9 & \ell_{10} \end{array} \\
\left(\begin{array}{ccc||ccccccc}
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
\hline\hline
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1
\end{array}\right)
\end{array}
\begin{array}{c} \\ Q_1 \\ Q_2 \\ Q_3 \\ N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \end{array}
$$

We describe the row operations for $\delta(Q_1), \delta(Q_2), \delta(Q_3)$, i.e., rows 1,2,3 of A; in particular, we describe the row operations of Claims 8,10.

Q_1: $\chi^{\delta(Q_1)} = \chi^{\{\ell_1,\ell_3,\ell_5,\ell_6\}}$. In Claim 8, for $j = 1$, we have $h = 3, g = 1$. $\left(\chi^{\delta(Q_1)} - \chi^{\delta(N_3)} + \chi^{\delta(N_1)}\right) = 2\chi^{\{\ell_1,\ell_3\}}$.
Note that $\Phi(\delta(Q_1)) = \{1, 3\}$.

Q_2: $\chi^{\delta(Q_2)} = \chi^{\{\ell_2,\ell_5,\ell_7,\ell_8\}}$. In Claim 8, for $j = 2$, we have $h = 5, g = 3$. $\left(\chi^{\delta(Q_2)} - \chi^{\delta(N_5)} + \chi^{\delta(N_3)}\right) = 2\chi^{\{\ell_2,\ell_5\}}$.
Next, Claim 10 is applied with $L' = \{\ell_2, \ell_5\}$, and we have $h = 2, g = 1$. $\chi^{\{\ell_2,\ell_5\}} - \chi^{\delta(N_2)} + \chi^{\delta(N_1)} = \chi^{\{\ell_1,\ell_2\}}$.
Note that $\Phi(\delta(Q_2)) = \{1, 2\}$.

Q_3: $\chi^{\delta(Q_3)} = \chi^{\{\ell_6,\ell_8,\ell_9,\ell_{10}\}}$. In Claim 8, for $j = 3$, we have $h = 7, g = 5$. $\left(\chi^{\delta(Q_3)} - \chi^{\delta(N_7)} + \chi^{\delta(N_5)}\right) = 2\chi^{\{\ell_6,\ell_8\}}$.
Next, Claim 10 is applied with $L' = \{\ell_6, \ell_8\}$, and we have $h = 5, g = 4$. $\chi^{\{\ell_6,\ell_8\}} - \chi^{\delta(N_5)} + \chi^{\delta(N_4)} = \chi^{\{\ell_2,\ell_6\}}$.
Next, Claim 10 is applied with $L' = \{\ell_2, \ell_6\}$, and we have $h = 3, g = 2$. $\chi^{\{\ell_2,\ell_6\}} - \chi^{\delta(N_3)} + \chi^{\delta(N_2)} = \chi^{\{\ell_2,\ell_3\}}$.
Note that $\Phi(\delta(Q_3)) = \{2, 3\}$.

5 The Capacitated Graph: Deferred Proofs

In this section, we present the proofs of Propositions 4 and 5, deferred from section 2.

▶ **Proposition 4.** *Each of the nested cuts $\delta(N_i), i = 1, \ldots, n-1$ and each of the Q-cuts $\delta(Q_j), j = 1, \ldots, k-1$ has capacity less than $\lambda = 5$.*

Proof. We examine a few cases.

Nested cut $\delta(N_i)$ such that N_i does not cross any Q-set: Then, either $2 \le i \le n-2$ and there is a largest index $j = 1, \ldots, k-2$ such that Q_j is contained in N_i, or else $i \in \{1,\ (n-1)\}$.

If $i = 1$, then $\delta(N_i)$ has two edges, namely, (i) the E_1-edge v_1v_2 of capacity 2, and (ii) the E_2-edge (of capacity one) between $s = v_1$ and the first node of Q_2; clearly, $\delta(N_1)$ has capacity $3 < \lambda = 5$. Similarly, if $i = (n-1)$, then $\delta(N_i)$ has two edges and has capacity $3 < \lambda = 5$.

Otherwise, $\delta(N_i)$ has three edges, namely, (i) the E_1-edge v_iv_{i+1} of capacity one, (ii) the E_2-edge (of capacity one) between the last node of Q_{j-1} and the first node of Q_{j+1}, and (iii) the E_2-edge (of capacity one) between the last node of Q_j and the first node of Q_{j+2}, Thus, the cut $\delta(N_i)$ has capacity $3 < \lambda = 5$.

Nested cut $\delta(N_i)$ such that N_i crosses a Q-set: Suppose that N_i crosses Q_j, where $j = 1, \ldots, k-1$. Clearly, the first node of Q_j is in N_i and the last node of Q_j is not in N_i. Note that $\delta(N_i)$ has two edges, namely, the E_1-edge v_iv_{i+1} of capacity 3, and the E_2-edge (of capacity one) between the last node of Q_{j-1} and the first node of Q_{j+1}. Thus, the cut $\delta(N_i)$ has capacity $4 < \lambda = 5$.

Q-cut $\delta(Q_j)$: First, suppose that $j = 2, \ldots, (k-2)$. Observe that $\delta(Q_j)$ consists of the following four unit-capacity edges. There are two edges of E_1 in the cut, namely, the edge between the last node of Q_{j-1} and the first node of Q_j, and the edge between the last node of Q_j and the first node of Q_{j+1}. Moreover, there are two edges of E_2 in the cut; namely, the edge between the last node of Q_{j-2} and the first node of Q_j, and the edge between the last node of Q_j and the first node of Q_{j+2}.

The cuts $\delta(Q_1)$ and $\delta(Q_{k-1})$ have capacity four since each of these cuts has only one edge of E_2 (since no E_2-edge is incident to the first node of Q_1 or to the last node of Q_{k-1}), and has two edges of E_1, one edge of capacity 2 (i.e., $sv_2 \in \delta(Q_1)$ and $v_{n-1}t \in \delta(Q_{k-1})$) and the other edge of capacity one. ◀

▶ **Proposition 5.** *Every non-trivial cut $\delta(S)$, $\emptyset \subsetneq S \subsetneq V$, of G that is neither a nested cut nor a Q-cut has capacity $\ge \lambda = 5$.*

Proof. We denote the nodes $v_1, \ldots, v_n$ by their indices $1, \ldots, n$ for this proof. (Notational remark: In this proof, the symbols ℓ_1, ℓ_2, etc., refer to the indices $1, \ldots, n$ rather than to links in L; in fact, links are not relevant in this section.) W.l.o.g., assume that S is a set of nodes such that $1 \in S$ (thus, S contains v_1), $S \neq N_i, i = 1, \ldots, n-1$, and $S \neq (V - Q_j), j = 1, \ldots, k-1$. Let $\ell_1 = 1$ and let r_1 denote the smallest index such that $r_1 \in S$ and $(r_1 + 1) \notin S$; note that r_1 is well defined, since $S \neq V$; possibly, $r_1 = \ell_1$. Thus, $[\ell_1, \ldots, r_1]$ is a maximal interval contained in S. Clearly, $S \neq [\ell_1, \ldots, r_1]$, otherwise, we would have $S = N_{r_1}$ (contradicting our assumption that S is not a nested set). Thus, $r_1 \le (n-2)$. Let ℓ_2 be the smallest index $> r_1$ that is in S; thus, $\ell_2 = \min_i\{i \in S, i \in [(r_1+1), \ldots, n]\}$. Moreover, let r_2 be the smallest index $\ge \ell_2$ such that either $r_2 = n$ or else $(r_2 + 1) \notin S$; thus, $r_2 = n$ if $[\ell_2, \ldots, n] \subset S$, and $r_2 = \min\{i \in S, i+1 \notin S, i \in [\ell_2, \ldots, n]\}$ if $[\ell_2, \ldots, n] \not\subset S$. Then, $[\ell_2, \ldots, r_2]$ is another maximal interval contained in S. Clearly, both the E_1-edges $v_{r_1}v_{(r_1+1)}$ and $v_{(\ell_2-1)}v_{\ell_2}$ are in $\delta(S)$.

For any node i $(i = 1, \ldots, n)$, we use $q(i)$ to denote the index of the (unique) Q-set that contains i; thus, $q(i) = \lceil (i-1)/\frac{k}{2} \rceil$.

We examine a few cases and show that $\delta(S)$ has capacity $\ge \lambda = 5$ in each case.

- Node r_1 is not the last node of its Q-set, $Q_{q(r_1)}$, and node ℓ_2 is not the first node of its Q-set, $Q_{q(\ell_2)}$:
  Note that $r_1 \neq s = 1$ and $\ell_2 \neq t = n$.

Then the E_1-edge $v_{r_1}v_{(r_1+1)}$ has capacity 3, and the E_1-edge $v_{(\ell_2-1)}v_{\ell_2}$ has capacity 3, hence, $\delta(S)$ has capacity $\geq 6 > \lambda = 5$.

- Node r_1 is not the last node of its Q-set, $Q_{q(r_1)}$, and node ℓ_2 is the first node of its Q-set, $Q_{q(\ell_2)}$:
  Note that $r_1 \neq s = 1$, and, possibly, $\ell_2 = t = n$.
  Thus, $r_1 \geq 2$. Then the E_1-edge $v_{r_1}v_{(r_1+1)}$ has capacity 3.
  Suppose $\ell_2 = t = n$. Then the E_1-edge $v_{(\ell_2-1)}v_{\ell_2}$ has capacity two. Thus, $\delta(S)$ has capacity $\geq 5 = \lambda = 5$.
  Now, suppose $\ell_2 \neq t = n$. Then the E_1-edge $v_{(\ell_2-1)}v_{\ell_2}$ has capacity one. Note that $q(r_1) \geq 1$ and $2 \leq q(\ell_2) \leq (k-1)$. We have two subcases, depending on $q(\ell_2) - q(r_1)$.
  $q(\ell_2) = q(r_1) + 1$: Then we have two further subcases.
  (i) First, suppose that $r_2 = n$; thus, $[\ell_2, \ldots, n]$ is a maximal interval contained in S. Then the last node of $Q_{q(r_1)}$ is not in S and the first node of $Q_{(q(r_1)+2)}$ is in S. Then the E_2-edge between the last node of $Q_{q(r_1)}$ and the first node $Q_{(q(r_1)+2)}$ is in $\delta(S)$. Thus, the cut $\delta(S)$ has capacity $\geq 5 = \lambda = 5$.
  (ii) Now, suppose that $r_2 < n$. Then the E_1-edge $v_{r_2}v_{(r_2+1)}$ is in $\delta(S)$, and it has capacity one or more. Thus, the cut $\delta(S)$ has capacity $\geq 5 = \lambda = 5$.
  $q(\ell_2) \geq q(r_1) + 2$: Then the last node of $Q_{(q(r_1)-1)}$ is in S and the first node of $Q_{(q(r_1)+1)}$ is not in S. Then the E_2-edge between the last node of $Q_{(q(r_1)-1)}$ and the first node $Q_{(q(r_1)+1)}$ is in $\delta(S)$. Thus, the cut $\delta(S)$ has capacity $\geq 5 = \lambda = 5$.
- Node r_1 is the last node of its Q-set, $Q_{q(r_1)}$, and node ℓ_2 is not the first node of its Q-set, $Q_{q(\ell_2)}$:
  Possibly, $r_1 = s = 1$, and recall that $r_1 \leq (n-2)$. Note that $\ell_2 \neq t = n$.
  First, suppose $r_1 = 1$. Then the E_1-edge $v_{r_1}v_{(r_1+1)}$ has capacity two, and the E_1-edge $v_{(\ell_2-1)}v_{\ell_2}$ has capacity 3. Hence, $\delta(S)$ has capacity $\geq 5 = \lambda = 5$.
  Now, suppose $r_1 \geq 2$. Then, $1 \leq q(r_1) \leq (k-2)$. Then the E_1-edge $v_{r_1}v_{(r_1+1)}$ has capacity one, and the E_1-edge $v_{(\ell_2-1)}v_{\ell_2}$ has capacity 3. Clearly, $\delta(S)$ has both these edges, as well as the E_2-edge between the last node of $Q_{q(r_1)-1}$ and the node (r_1+1) (first node of $Q_{q(r_1)+1}$). Hence, $\delta(S)$ has capacity $\geq 5 = \lambda = 5$.
- Node r_1 is the last node of its Q-set, $Q_{q(r_1)}$, and node ℓ_2 is the first node of its Q-set, $Q_{q(\ell_2)}$: Possibly, $r_1 = s = 1$, and recall that $r_1 \leq (n-2)$. Possibly, $\ell_2 = t = n$.
  Note that $q(r_1) \geq 0$ and $2 \leq q(\ell_2) \leq (k)$.
  The E_1-edge $v_{(\ell_2-1)}v_{\ell_2}$ has capacity one or two.
  If $r_1 = s$, then the E_1-edge $v_{r_1}v_{(r_1+1)}$ has capacity two, otherwise, this edge has capacity one.
  If $r_1 \neq s$, then $q(r_1) \geq 1$; moreover, $(r_1+1) \notin S$ and (r_1+1) is the first node of its Q-set, $Q_{q(r_1+1)}$; hence, the E_2-edge between the last node of $Q_{(q(r_1)-1)}$ and (r_1+1) is in $\delta(S)$.
  At this point in the analysis, it is clear that $\delta(S)$ has capacity at least 3.
  We have two subcases, depending on $q(\ell_2) - q(r_1)$.
  $q(\ell_2) \geq q(r_1) + 3$: Then the interval $[(r_1+1), \ldots, (\ell_2-1)]$ (which is a subset of $V - S$) contains two or more Q-sets. We will show that the second and the last-but-one of the Q-sets each contribute an E_2-edge to $\delta(S)$.
  Since r_1 is the last node of its Q-set, $r_1 \in S$, and the first node of $Q_{(q(r_1)+2)}$ is not in S, the E_2-edge between these two nodes is in $\delta(S)$.
  Similarly, since ℓ_2 is the first node of its Q-set, $\ell_2 \in S$, and the last node of $Q_{(q(\ell_2)-2)}$ is not in S, the E_2-edge between these two nodes is in $\delta(S)$.
  Thus, $\delta(S)$ has capacity $\geq 5 = \lambda = 5$.

$q(\ell_2) \leq q(r_1) + 2$: In this case, $q(\ell_2) = q(r_1) + 2$, since the first node of $Q_{(q(r_1)+1)}$, which is $(r_1 + 1)$, is not in S, and ℓ_2 (which is the first node of its Q-set) is in S.
Observe that $r_2 \neq n$; that is, $r_2 \leq (n - 1)$ and the node $(r_2 + 1)$ is not in S. Otherwise, we would have $S = V - Q_{(q(r_1)+1)}$, contradicting our assumption that S is not the complement of a Q-set.
First, suppose $r_2 = (n - 1)$. Then $\delta(S)$ has the edge $v_{(n-1)}v_n$ of capacity two. Hence, overall, $\delta(S)$ has capacity $\geq 5 = \lambda = 5$.
Now, suppose $r_2 \leq (n - 2)$. If the E_1-edge $v_{r_2}v_{(r_2+1)}$ has capacity 3, then, overall, $\delta(S)$ has capacity $\geq 6 = \lambda = 5$. Otherwise, the E_1-edge $v_{r_2}v_{(r_2+1)}$ has capacity one. Observe that the node r_2 is the last node of its Q-set; moreover, the node (r_2+1) is the first node of its Q-set and $q(r_2 + 1) \leq (k - 1)$. If the interval $[(r_2 + 1), (r_2 + 2), \ldots, n]$ has one or more nodes of S, then $\delta(S)$ has one more E_1-edge (besides $v_{r_2}v_{(r_2+1)}$) of capacity one or more, hence, overall, $\delta(S)$ has capacity $\geq 5 = \lambda = 5$. Finally, suppose that the interval $[(r_2 + 1), (r_2 + 2), \ldots, n]$ has no nodes of S. Then, the node r_2 is in S and the first node of $Q_{(q(r_2)+2)}$ is not in S, hence, the E_2 edge between these two nodes is in $\delta(S)$. Thus, overall, $\delta(S)$ has capacity $\geq 5 = \lambda = 5$. ◀

References

1 Ashkan Aazami, Joseph Cheriyan, and Bundit Laekhanukit. A bad example for the iterative rounding method for mincost k-connected spanning subgraphs. *Discret. Optim.*, 10(1):25–41, 2013. doi:10.1016/j.disopt.2012.10.002.

2 Ishan Bansal. A Global Analysis of the Primal-Dual Method for Pliable Families. *CoRR*, abs/2308.15714, 2024. doi:10.48550/arXiv.2308.15714.

3 Ishan Bansal. A global analysis of the primal-dual method for edge augmentation problems. In Nicole Megow and Amitabh Basu, editors, *Integer Programming and Combinatorial Optimization - 26th International Conference, IPCO 2025, Baltimore, MD, USA, June 11-13, 2025, Proceedings*, Lecture Notes in Computer Science, pages 58–71. Springer, 2025. doi:10.1007/978-3-031-93112-3_5.

4 Ishan Bansal, Joe Cheriyan, Sanjeev Khanna, and Miles Simmons. Improved approximation algorithms for capacitated network design and flexible graph connectivity. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming, ICALP 2025, Aarhus, Denmark, July 8-11, 2025*, LIPIcs, pages 20:1–20:20, n.p., 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2025.20.

5 Ishan Bansal, Joseph Cheriyan, Logan Grout, and Sharat Ibrahimpur. Improved Approximation Algorithms by Generalizing the Primal-Dual Method Beyond Uncrossable Functions. In *Proceedings of the 50th International Colloquium on Automata, Languages, and Programming*, volume 261, article 15, pages 1–19, 2023. doi:10.4230/LIPIcs.ICALP.2023.15.

6 Ishan Bansal, Joseph Cheriyan, Logan Grout, and Sharat Ibrahimpur. Improved approximation algorithms by generalizing the primal-dual method beyond uncrossable functions. *Algorithmica*, 86(8):2575–2604, 2024. doi:10.1007/s00453-024-01235-2.

7 Sylvia C. Boyd, Joseph Cheriyan, Arash Haddadan, and Sharat Ibrahimpur. Approximation algorithms for flexible graph connectivity. *Math. Program.*, 204(1):493–516, 2024. doi:10.1007/s10107-023-01961-5.

8 Deeparnab Chakrabarty, Chandra Chekuri, Sanjeev Khanna, and Nitish Korula. Approximability of Capacitated Network Design. *Algorithmica*, 72(2):493–514, 2015. doi:10.1007/s00453-013-9862-4.

9 Lisa Fleischer, Kamal Jain, and David P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *J. Comput. Syst. Sci.*, 72(5):838–867, 2006. doi:10.1016/j.jcss.2005.05.006.

10 Michel X. Goemans, Andrew V. Goldberg, Serge A. Plotkin, David B. Shmoys, Éva Tardos, and David P. Williamson. Improved Approximation Algorithms for Network Design Problems. In *Proceedings of the 5th Symposium on Discrete Algorithms*, pages 223–232, 1994. URL: `http://dl.acm.org/citation.cfm?id=314464.314497`.

11 M. Hariprasad. Determinant of binary circulant matrices. *Special Matrices*, 7(1):92–94, 2019. `doi:10.1515/spma-2019-0008`.

12 Kamal Jain. A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. *Combinatorica*, 21(1):39–60, 2001. `doi:10.1007/s004930170004`.

13 Guy Kortsarz, Robert Krauthgamer, and James R. Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM J. Comput.*, 33(3):704–720, 2004. `doi:10.1137/S0097539702416736`.

14 Lap Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, UK, 2011. `doi:10.1017/CBO9780511977152`.

15 Zeev Nutov. Tight Analysis of the Primal-Dual Method for Edge-Covering Pliable Set Families. In Pawel Gawrychowski, Filip Mazowiecki, and Michal Skrzypczak, editors, *50th International Symposium on Mathematical Foundations of Computer Science, MFCS 2025, Warsaw, Poland, August 25-29, 2025*, LIPIcs, pages 82:1–82:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. `doi:10.4230/LIPIcs.MFCS.2025.82`.

16 Zeev Nutov. A tight example for approximation ratio 5 for covering small cuts by the primal-dual method. *CoRR*, abs/2512.08350, 2025. `doi:10.48550/arXiv.2512.08350`.

17 Miles Simmons. Cover Small Cuts and Flexible Graph Connectivity Problems. Master's thesis, University of Waterloo, Waterloo, ON, Canada, September 2025. URL: `https://hdl.handle.net/10012/22426`.

18 Miles Simmons, Ishan Bansal, and Joe Cheriyan. A 5-Approximation Analysis for the Cover Small Cuts Problem. *CoRR*, abs/2602.01462, 2026. `doi:10.48550/arXiv.2602.01462`.

19 David P. Williamson, Michel X. Goemans, Milena Mihail, and Vijay V. Vazirani. A Primal-Dual Approximation Algorithm for Generalized Steiner Network Problems. *Combinatorica*, 15(3):435–454, 1995. `doi:10.1007/BF01299747`.