

Tight Lower Bounds for the Number of Inclusion-Minimal st-Cuts

Alessio Conte, Roberto Grossi, Andrea Marino, Romeo Rizzi, Takeaki Uno,
Luca Versari

► **To cite this version:**

Alessio Conte, Roberto Grossi, Andrea Marino, Romeo Rizzi, Takeaki Uno, et al.. Tight Lower Bounds for the Number of Inclusion-Minimal st-Cuts. WG 2018 - International Workshop on Graph-Theoretic Concepts in Computer Science, Jun 2018, Cottbus, Germany. pp.100-110, 10.1007/978-3-030-00256-5_9 . hal-01964710

HAL Id: hal-01964710

<https://hal.inria.fr/hal-01964710>

Submitted on 23 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Tight Lower Bounds for the Number of Inclusion-Minimal st -Cuts

Alessio Conte¹, Roberto Grossi², Andrea Marino²,
Romeo Rizzi³, Takeaki Uno¹, and Luca Versari²

¹ National Institute of Informatics, Japan, {conte,uno}@nii.ac.jp

² Università di Pisa, {grossi,marino,luca.versari}@di.unipi.it

³ Università di Verona, rizzi@di.univr.it

Abstract. We study the number of inclusion-minimal cuts in an undirected connected graph G , also called st -cuts, for any two distinct nodes s and t : the st -cuts are in one-to-one correspondence with the partitions $S \cup T$ of the nodes of G such that $S \cap T = \emptyset$, $s \in S$, $t \in T$, and the subgraphs induced by S and T are connected. It is easy to find an exponential upper bound to the number of st -cuts (e.g. if G is a clique) and a constant lower bound. We prove that there is a more interesting lower bound on this number, namely, $\Omega(m)$, for undirected m -edge graphs that are biconnected or triconnected (2- or 3-node-connected). The wheel graphs show that this lower bound is the best possible asymptotically.

1 Introduction

Cuts are among the fundamental notions in graphs. A cut in a graph G represents a bipartition S, T of its node set $V(G)$, and the corresponding cutset is the set of edges in $E(G)$ having one endpoint in S and the other in T . Cutsets have a wide range of applications, such as switching functions, sensitivity analysis of optimization problems, vertex packing, and network reliability [22, 23]. Due to the sheer number of cuts, it makes sense to focus on those whose cutsets are minimal under inclusion (i.e. any subset of their edges is not a cutset): these cuts corresponds to those having both induced subgraphs $G[S]$ and $G[T]$ connected.⁴

For any two given distinct nodes s and t in $V(G)$, we consider their st -cuts or, equivalently, the bipartitions S, T for which $s \in S$, $t \in T$ and both $G[S]$ and $G[T]$ are connected (a.k.a. bonds). In the following, we refer to just S as a cut, meaning the bipartition S and $T = V(G) \setminus S$ as it is clear from the context.

In this paper, we investigate the number of st -cuts in an undirected connected graph G with $n = |V(G)|$ nodes and $m = |E(G)|$ edges. This is useful to recursively generate st -cuts, as knowing a lower bound on their number can help to better amortize the cost of recursive calls. For general graphs, we can face two kinds of extreme situations. If G is a clique, any choice of $S \subseteq V(G)$ such that $s \in S$ and $t \notin S$ gives rise to an st -cut; this yields 2^{n-2} st -cuts as

⁴ Since G is connected, also $G[S]$ and $G[T]$ are connected, otherwise we could remove at least one edge from the minimal cutset to reconnect $G[S]$ or $G[T]$.

each such S can be obtained by adding s to any subset of $V(G) \setminus \{s, t\}$. At the other extreme is a single st -cut when G is made up of two cliques connected by an edge $\{s, t\}$, where S is the node set of one of the cliques (see Fig. 1(left)).



Fig. 1. Left: connected graph with one st -cut. Right: 2-edge-connected graph with two st -cuts. The clouds correspond to cliques.

It is natural to investigate how the number of st -cuts changes if we add some more stringent requirement on the connectivity of G . For 2-edge-connectivity, where G remains connected by any single edge removal, the situation does not change significantly. The graph in Fig. 1(right) is 2-edge connected and has just two st -cuts. The graph is formed by a triangle where s and t are its nodes, and the remaining node is part of a clique of $n - 2$ nodes: the only possible choices for S are the singleton s or the clique extended with s . On the other hand, choosing both s and t inside the clique (instead of the triangle) would give rise to an exponential number of st -cuts, as discussed before.

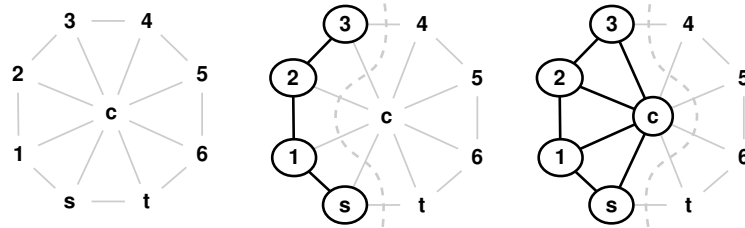


Fig. 2. Left: a wheel graph (triconnected, thus also biconnected). Center and right: st -cuts with respectively $c \notin S$ and $c \in S$.

What if G is biconnected (i.e. 2-node-connected)? For 2-node-connectivity, G remains connected by any single node removal, and the clique is clearly biconnected, thus we still have an exponential number of st -cuts. But this is not the interesting question to pose. What we investigate is the *minimum* number of st -cuts that a biconnected graph can have. For example, the wheel graph in Fig. 2 has $2(n - 2)$ st -cuts: the graph is a cycle of $n - 1$ nodes, all connected to a center node, thus $m = 2(n - 1)$. The center node either belongs to S or not: in either cases, we have $n - 2$ ways of choosing the remaining nodes, as they must bipartite the cycle in two sectors of adjacent nodes. One may wonder if it is possible to find a biconnected graph with a constant number of st -cuts: the answer to this question is negative.

Indeed the contribution of this paper is to give a proof that there are $\Omega(m)$ st -cuts in any biconnected graph G for any choice of distinct node s, t . This provides an interesting *gap*, from $\Omega(1)$ to $\Omega(m)$, when we move from either connectivity or 2-edge-connectivity to 2-node-connectivity. The wheel in Fig. 2 shows that the lower bound is tight. It is an open problem to study higher connectivity or other requirements on G : we observe that our lower bound extends to k -node-connected graphs with $k \geq 2$, and matches for $k = 2, 3$ as the wheel is triconnected, but we do not know if the lower bound is tight for $k > 3$.

Related work. In the literature, the generation of all cuts in undirected graphs has been studied by Abel and Bicker [1], Beltmore and Jensen [3], Tsukiyama et al. [27], Golberg [11]. Others are [15, 21, 28]. Among these approaches, Tsukiyama et al. is the most efficient as it requires $O(m)$ time per cutset. Algorithms for generating minimum cardinality and minimum weight cuts have been proposed by Ball and Provan [2], Gardner [10], and Picard and Queyranne [20]. In [29], all cuts of G are returned by non-decreasing weights ordering. Other variations include the k -best cuts problem which have been considered in [12]. (For the case of directed graphs see [25].) Enumerating the cutsets between all pairs of nodes reduces to the problem of solving a system of linear equations [19]. The notion of cutset has been generalized to cut conjunctions in [17].

From the above works we see that over the years a lot of listing algorithms have been proposed for generating minimal cutsets. On the other hand, studies about bounds on the number of cutsets have been focused on minimum cutsets without fixing s and t , i.e. the minimum number of edges to be removed to disconnect a graph. Deciding the maximum size of such a minimum cutset has been called the maximum connectivity problem, one of the 14 questions of Berge [4]. This question has been answered by Harary [14], giving lower and upper bounds on the size of this cutset for any graph as a function of the number of nodes and edges. Bixby [6] has found the minimum number of edges and nodes in a k -edge-connected graph for a given number b of minimum cardinality cutsets. This problem turned out to have an essentially closed form solution for all k and b , and for many values of k and b it is possible to build a graph achieving this minimum.

Over the years, this interest towards bounding cutsets have been mainly motivated by the network reliability problem [7]. One of the fundamental results is due to Kruskal [18] and Katona [16] in terms of F_i , which is the number of sets of i edges which do not contain a cutset. Upper bounds for the number of minimum cutsets in terms of the radius, diameter, minimum degree, maximum degree, chordality, girth and other parameters have been given by Chandran et al. [8] for weighted graphs. Harada et al. [13] have provided lower bounds for the number of cutsets of a given size (not necessarily minimal). We remark that all the above works consider set of edges, rather than partition of nodes, which are eventually minimum but in any case never minimal. Hence, we are not aware of previous work on lower bounds for the number of minimal st -cuts as discussed in this paper.

Preliminaries. All the graphs considered in this paper are undirected, connected, and simple (without multiple edges or self-loops). Hence a sequence of nodes cannot induce more than one path or cycle, thus we may refer to paths and cycles simply as sequences of nodes. Two paths are disjoint if they do not share any node (and consequently any edge), and are *internally* disjoint if they share both the first and last nodes in their sequences, but are otherwise disjoint. Two paths that do not share any edge (but may share nodes) are called *edge* disjoint. We call $\kappa(G)$ the *node connectivity* of G , that is the size of the smallest node cut. If $\kappa(G) = k$ we say that G is k -connected (and thus removing $k - 1$ nodes cannot disconnect the graph). By Menger's theorem, we have that for each pair of nodes $x, y \in V(G)$ there are at least $\kappa(G)$ internally disjoint paths between x and y .

An st -numbering for two adjacent nodes s, t is a numbering of the nodes of G such that each node (except t) is adjacent to a node larger than itself, and each node (except s) to one smaller than itself. When G is biconnected, [9] proves that there is an st -numbering for any pair of adjacent nodes s and t , and that it can be found in linear time. Furthermore, we remark that this ordering can be found even if s and t are not adjacent. Indeed, let G' be the graph obtained by adding the edge $\{s, t\}$ to G . G' is still biconnected, and has an st -numbering. Consider the same numbering for G : s still has a neighbor larger than itself in G (any neighbor, since it has the smallest label), and by the same logic t still has a neighbor smaller than itself. All other nodes have the same neighborhood in G as they had in G' . Thus we can remark the following

Observation 1 *There is an st -numbering on a biconnected graph G for any pair of nodes s and t .*

In the rest of the paper, we assume that the nodes are numbered in st -numbering, and thus $x < y$ for any two nodes x and y means that x appears earlier than y in the st -numbering.

2 Number of st -cuts in a biconnected graph

This section illustrates our main result that, for any undirected biconnected graph G and any two distinct nodes s, t , there are at least $\max(n, m - 2n) = \Omega(m)$ st -cuts. In order to get this lower bound we will attempt at defining, for each edge, a corresponding cut, ensuring that each such cut is valid. However, we also need these cuts to be distinct from each other.

In the following, we will produce two sets of cuts, corresponding to different kind of edges, which may overlap with each other, but each set will contain distinct elements. One of the sets will contain exactly n distinct cuts, and the other at least $m - 2n$: as a result, we obtain that G has at least $\max(n, m - 2n) = \Omega(m)$ st -cuts.

One of the main ingredients of our proof will be defining a *backbone* of a graph, which is based on the st -numbering of G . Its structure immediately leads to a classification of the edges of G that will be crucial to define the st -cuts.

Indeed, it helps us to overcome the fact that it is not possible to identify a distinct st -cut for each edge in a straightforward way, as some edges are not yielding new st -cuts.

For the sake of discussion, we also report some observations on st -numbering that can be partially found in previous work [5, 24, 26].

2.1 Backbone of the graph

Consider a biconnected graph $G = (V(G), E(G))$, and an st -numbering induced by two of its nodes s and t .

Definition 1 (backbone). *The backbone of G is the graph $\text{BB} = (V(G), E(\text{BB}))$ where $E(\text{BB}) \subseteq E(G)$ is defined as $\{i, j\} \in E(\text{BB})$ if and only if j is either the largest or smallest neighbor of i in the st -numbering.*

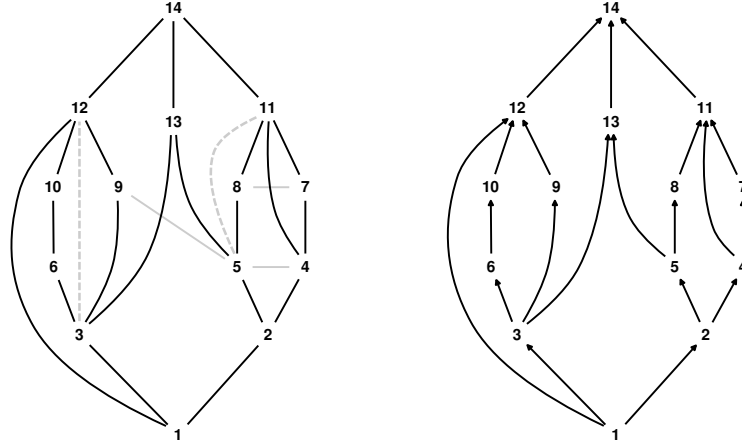


Fig. 3. Left: a biconnected graph G , labeled in st -numbering ($s = 1, t = 14$). Backbone edges in bold, shortcut edges dashed, and cross edges in gray. Right: the corresponding graph $\vec{\text{BB}}$ showing predecessor-successor relationships of G .

In other words, the backbone is obtained by taking for each node just the edge to its largest and its smallest neighbor, as illustrated in Fig. 3(left). We observe that BB fulfills the following properties.

- All edges having s or t as an extreme are in BB , as s and t are respectively the smallest and largest node.
- BB has at most $2n$ edges, since we take at most 2 edges for each node.
- The st -numbering of G is also an st -numbering for BB , since each node has at least one smaller neighbor (except s) and a larger one (except t).

For the mere purpose of definitions, we consider the oriented version of BB, called bipolar orientation $\vec{\text{BB}}$, where an arc (x, y) belongs to $\vec{\text{BB}}$ iff $\{x, y\} \in E(\text{BB})$ and $x < y$, as illustrated in Fig. 3(right). Note that s and t are respectively the only source and target in $\vec{\text{BB}}$. For each node $v \in V(G)$, we say that $x \in V(G)$ is a *predecessor* of v iff there is an oriented path from x to v in $\vec{\text{BB}}$, and $y \in V(G)$ is a *successor* of v iff the oriented path in $\vec{\text{BB}}$ is from v to y . Note that x is a predecessor of v iff there is a monotone increasing path from x to v in BB, and y is a successor of v iff there is a monotone decreasing path from y to v in BB. As it can be seen, there can be pairs of nodes such that they are not one predecessor of the other.

In the following, we drop $\vec{\text{BB}}$ and focus on BB alone, keeping the sets of predecessors and successors of each node v , respectively denoted as $\text{ANC}(v)$ and $\text{DESC}(v)$. Note that $v \notin \text{ANC}(v)$ and $v \notin \text{DESC}(v)$. Clearly $x \in \text{ANC}(v)$ iff $v \in \text{DESC}(x)$, and $\text{ANC}(v) \cap \text{DESC}(v) = \emptyset$. We remark that s and t are respectively a predecessor and a successor of all nodes in $V(G)$ except themselves.

The edges $\{x, y\} \in E(G)$ can be classified in the given BB as three types (see Fig. 3(left)).

- $\{x, y\}$ is a *backbone* (type B) edge iff $\{x, y\} \in E(\text{BB})$.
- $\{x, y\}$ is a *shortcut* (type S) edge iff $\{x, y\} \in E(G) \setminus E(\text{BB})$ and $x \in \text{ANC}(y)$.
- $\{x, y\}$ is a *cross* (type C) edge iff $\{x, y\} \in E(G) \setminus E(\text{BB})$ and $x \notin \text{ANC}(y)$.

It is important to remark that each edge of $E(G)$ falls under exactly one of the above types.

Observation 2 *Any edge of G is either a backbone edge, a cross edge, or a shortcut edge.*

2.2 Case analysis on the st -cut types

In the following, we will use the classification of the edges to define st -cuts. In particular, for each edge, depending on its type, we define a corresponding cut. For all these edges, the corresponding st -cuts (S, T) we define are always valid, meaning that both S and $T = V \setminus S$ induce connected subgraphs.

Definition 2 (B - backbone cut). *For each node $v \in V(G) \setminus \{t\}$, its type-B cut is $S = \text{ANC}(v) \cup \{v\}$.*

Example (B - backbone cut) On the graph in Fig. 3, $v = 9$ yields the type-B cut $S = \{1, 3, 9\}$ (and $T = V(G) \setminus S$). A visual representation is shown in Fig. 4 (left).

Lemma 3. *Every node except t yields a valid type-B cut, and the type-B cuts are pairwise distinct.*

Proof. Each predecessor of v is on a path from v to s made of predecessors of v , thus $G[S] = G[\text{ANC}(v) \cup \{v\}]$ is connected. $G[T]$ is also connected as any node that is *not* predecessor of v has a path to t made of nodes which are not

predecessors of v . Furthermore, note that all nodes in S (except v itself) are predecessors of v , and v is the only node that satisfies this property, thus v can be uniquely deduced from the set S , meaning that two different nodes may not lead to the same type-B cut. \square

As a consequence of Lemma 3, we get the following.

Observation 3 *Every biconnected graph G has at least n distinct st -cut for any choice of s and t .*

In order to increase the lower bound in Observation 3, in the following we consider edges $\{x, y\}$ which do *not* belong to the backbone of G . Note that there are at least $m - 2n$ such edges since the backbone has at most $2n$ edges.

Suppose $x < y$ without loss of generality in the rest of the section. Note that y cannot be a predecessor of x , but may or may not be a successor.

Definition 4 (C - cross cut). *For each cross edge $\{x, y\}$, its type-C cut is $S = \text{ANC}(x) \cup \text{ANC}(y) \cup \{x, y\}$.*

Example (C - cross cut) On the graph in Fig. 3, $\{x, y\} = \{5, 9\}$ yields the type-C cut $S = \{1, 2, 3, 5, 9\}$ (and $T = V(G) \setminus S$). A visual representation is shown in Fig. 4 (center).

Lemma 5. *Every cross edge yields a valid type-C cut, and the type-C cuts are pairwise distinct.*

Proof. By the proof of Lemma 3, $G[\text{ANC}(x) \cup \{x\}]$ and $G[\text{ANC}(y) \cup \{y\}]$ are connected. As the subgraphs share the node s , their union $G[\text{ANC}(x) \cup \text{ANC}(y) \cup \{x, y\}] = G[S]$ is connected too. $G[T]$ is connected since it is made of nodes which are not predecessors of x nor y , thus have a path to t made of nodes that are not predecessor of x nor y .

Given the set S , the only two nodes who are not successor of any other node in S are x and y , thus the cut is uniquely identified by the cross edge $\{x, y\}$, meaning a different edge may not yield the same cut. \square

Definition 6 (S - shortcut cut). *For each shortcut edge $\{x, y\}$, where $y \in \text{ANC}(x)$, its type-S cut is $S = V(G) \setminus T$, where T is the set of all nodes connected to t in $G[V(G) \setminus (\text{ANC}(y) \cup \{x, y\})]$, including t itself.*

Example (S - shortcut cut) On the graph in Fig. 3, $\{x, y\} = \{3, 12\}$ yields the type-S cut $S = \{1, 3, 6, 10, 12\}$ (and $T = V(G) \setminus S$). A visual representation is shown in Fig. 4 (right).

Lemma 7. *Every shortcut edge yields a valid type-S cut, and the type-S cuts are pairwise distinct.*

Proof. Let $G' = G[V(G) \setminus (\text{ANC}(x) \cup \{x, y\})]$ (recall that $x < y$). In a shortcut cut, T contains all nodes connected to t in G' , including t itself. Thus $G[T]$ is connected by definition.

Consider now $S = V(G) \setminus T$. In particular, S will contain $\text{ANC}(x) \cup \{x, y\}$, which by definition of ANC is connected and contains s , plus all the nodes that cannot reach t in G' : since these nodes cannot reach t , and G is connected, they must be connected to $\text{ANC}(x) \cup \{x, y\}$ instead, thus $G[S]$ is connected, meaning that S, T is a valid cut. A visual representation of this can be seen in Fig. 4 (right), where $\{x, y\} = \{3, 12\}$, 6 and 10 are the nodes that cannot reach t in $G[V(G) \setminus \text{ANC}(3) \cup \{3, 12\}]$, while 9 can reach t via the edge $\{9, 5\}$.

We now only need to show that any two shortcut edges cannot produce the same cut.

First, all nodes that are *not* predecessors of y have a path to t made of nodes which are not predecessors of y . Since $\text{ANC}(x) \subset \text{ANC}(y)$, these nodes may reach t in G' , and thus are in T . This means that $S \subseteq \text{ANC}(y) \cup \{y\}$. Moreover, by the properties of $\text{ANC}()$, we have that y is the only node that satisfies this property.

We will now prove our claim by contraddiction. Suppose another edge $\{w, z\}$ yields the same S . Then we have that $S \subseteq \text{ANC}(z) \cup \{z\}$, which implies that $y = z$ because of what we said above. Moreover, we must have that $w \in S$ as both extremes of the shortcut edge go in S when defining a type-s cut.

We can now only consider shortcut edges of the form $\{w, y\}$. Without loss of generality, assume $x < w$ (we cannot have $x = w$ since G is not a multigraph). Note that, by definition, $x \in S$. Furthermore, note that by construction of the backbone, the edge between w and its largest neighbor is in the backbone, thus since $\{w, y\}$ is a shortcut edge (not a backbone edge), w must have a neighbor $v > y$.

As $v > y$ and $w > x$, it follows that $v, w \notin \text{ANC}(x) \cup \{x, y\}$. This means that there is a path from w to t that does not use any node from $\text{ANC}(x) \cup \{x, y\}$, thus $w \in T$. As we supposed $w \in S$, this is a contradiction and the thesis follows. \square

Lemma 8. *The sets of type-C cuts and type-S cuts are disjoint.*

Proof. Notice that, as proven in Lemma 7, we have that for each type-s cut there exists a node y that belongs to S and such that $S \subseteq \{y\} \cup \text{ANC}(y)$.

On the other hand, a type-C cut has $S = \{x, y\} \cup \text{ANC}(x) \cup \text{ANC}(y)$, where $\{x, y\}$ is a cross edge. We thus have that $y \notin \text{ANC}(x)$ and $x \notin \text{ANC}(y)$. Moreover, as $x \notin \text{ANC}(z)$ if $z \in \text{ANC}(x)$, any other node of S is not a predecessor of either x or y (or both). This implies that there is no node in S that has the whole S among its predecessors, thus proving that we cannot find the same cut in both the sets of type-s cuts and type-C cuts. \square

We finally give the proof of our main result, which is now an immediate consequence of the properties proved so far.

Theorem 9. *For any biconnected graph G and any two distinct nodes s, t , there are at least $\max(n, m - 2n) = \Omega(m)$ st -cuts.*

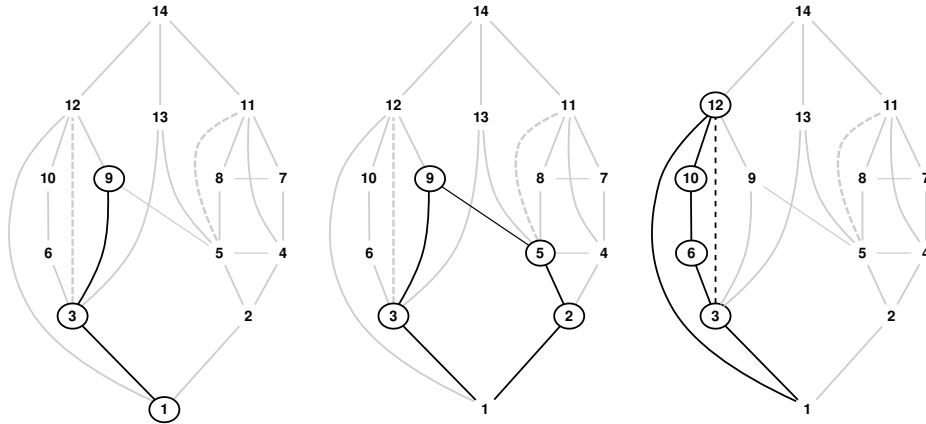


Fig. 4. st -cuts corresponding to respectively a type-B cut with $v = 9$ (left), a type-C cut with $\{x, y\} = \{5, 9\}$ (center), and a type-S cut with $\{x, y\} = \{3, 12\}$ (right)

Proof. Observation 3 proves that we have at least n different cuts. On the other hand, any non-backbone edge gives us either a type-S cut or a type-C cut, and no cut is obtained twice in this way, as proven in Lemma 8. As there are $m - 2n$ non-backbone edges, we have at least $m - 2n$ cuts. \square

2.3 Graphs that allow for an st -numbering

While we considered biconnected graphs, it can be noted that Theorem 9 holds for any graph admitting an st -numbering, as this is sufficient for our proof. This condition is slightly more general than assuming G to be biconnected, and actually corresponds to the biconnected components tree of G being a path, with the components containing s and t in its extremes.

Indeed, as it can be seen in Fig. 5, any node in a biconnected component out of this path (the dashed ones) is separated by both s and t by a single cut node, meaning that there cannot be both a monotone increasing path and a monotone decreasing path from the node to respectively t and s .

On the other hand, it can be easily seen how an st -numbering for a path of biconnected component can be computed by combining a suitable st -numbering of each of the components between its articulation points.

3 Conclusions and further work

In this paper we have proved that there are $\Omega(m)$ st -cuts in any biconnected graph G for any choice of distinct nodes s, t . We have shown that this lower bound is tight for k -node-connected graphs with $k = 2, 3$ as there is a triconnected graph, i.e., the wheel in Fig. 2, matching this lower bound. The natural question which remains open is whether the $\Omega(m)$ bound is tight in k -node-connected graphs for $k > 3$.

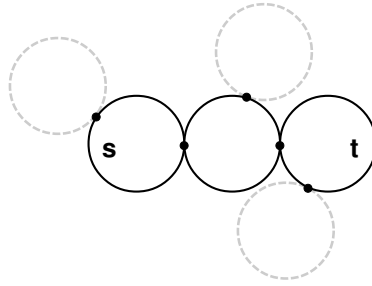


Fig. 5. An example of a graph that does not allow an st -numbering, and of a subgraph (highlighted in black) that does. Circles represent biconnected components.

Acknowledgements. This work was partially supported by JST CREST, grant number JPMJCR1401, Japan, and MIUR, Italy.

References

1. U Abel and R Bicker. Determination of all minimal cut-sets between a vertex pair in an undirected graph. *IEEE Transactions on Reliability*, 31(2):167–171, 1982.
2. Michael O Ball and J Scott Provan. Calculating bounds on reachability and connectedness in stochastic networks. *Networks*, 13(2):253–278, 1983.
3. M Bellmore and PA Jensen. An implicit enumeration scheme for proper cut generation. *Technometrics*, 12(4):775–788, 1970.
4. Claude Berge. La theorie des graphes. *Paris, France*, 1958.
5. Therese C. Biedl and Goos Kant. A better heuristic for orthogonal graph drawings. *Comput. Geom.*, 9(3):159–180, 1998.
6. RE Bixby. The minimum number of edges and vertices in a graph with edge connectivity n and m n -bonds. *Networks*, 5(3):253–298, 1975.
7. Timothy B Brecht and Charles J Colbourn. Lower bounds on two-terminal network reliability. *Discrete Applied Mathematics*, 21(3):185–198, 1988.
8. L Sunil Chandran and L Shankar Ram. On the number of minimum cuts in a graph. *SIAM Journal on Discrete Mathematics*, 18(1):177–194, 2004.
9. Shimon Even and Robert Endre Tarjan. Computing an st -numbering. *Theoretical Computer Science*, 2(3):339 – 344, 1976.
10. ML Gardner. Algorithm to aid in the design of large scale networks. *LARGE SCALE SYST.*, 8(2):147–156, 1985.
11. Leslie Ann Goldberg. *Efficient algorithms for listing combinatorial structures*, volume 5. Cambridge University Press, 2009.
12. Horst W Hamacher, Jean-Claude Picard, and Maurice Queyranne. On finding the k best cuts in a network. *Operations Research Letters*, 2(6):303–305, 1984.
13. Hideaki Harada, Zheng Sun, and Hiroshi Nagamochi. An exact lower bound on the number of cut-sets in multigraphs. *Networks*, 24(8):429–443, 1994.
14. Frank Harary. The maximum connectivity of a graph. *Proceedings of the National Academy of Sciences*, 48(7):1142–1146, 1962.
15. GB Jasmon and KW Foong. A method for evaluating all the minimal cuts of a graph. *IEEE transactions on reliability*, 36(5):539–545, 1987.

16. G Katona. A theorem for finite sets. *Theory of Graphs*, pages 187–207, 1968.
17. Leonid Khachiyan, Endre Boros, Konrad Borys, Khaled Elbassioni, Vladimir Gurvich, and Kazuhisa Makino. Generating cut conjunctions in graphs and related problems. *Algorithmica*, 51(3):239–263, 2008.
18. Joseph B Kruskal. The number of simplices in a complex. *Mathematical optimization techniques*, 10:251–278, 1963.
19. Alberto Martelli. A gaussian elimination algorithm for the enumeration of cut sets in a graph. *J. ACM*, 23(1):58–73, January 1976.
20. Jean-Claude Picard and Maurice Queyranne. On the structure of all minimum cuts in a network and applications. *Mathematical Programming*, 22(1):121–121, 1982.
21. VC Prasad, V Sankar, and KS Prakasa Rao. Generation of vertex and edge cutsets. *Microelectronics Reliability*, 32(9):1291–1310, 1992.
22. J Scott Provan and Michael O Ball. Computing network reliability in time polynomial in the number of cuts. *Operations Research*, 32(3):516–526, 1984.
23. J. Scott Provan and Douglas R Shier. A paradigm for listing (s, t)-cuts in graphs. *Algorithmica*, 15(4):351–372, 1996.
24. Pierre Rosenstiehl and Robert Endre Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discrete & Computational Geometry*, 1:343–353, 1986.
25. Douglas R Shier and David E Whited. Iterative algorithms for generating minimal cutsets in directed graphs. *Networks*, 16(2):133–147, 1986.
26. Roberto Tamassia and Ioannis G. Tollis. A unified approach a visibility representation of planar graphs. *Discrete & Computational Geometry*, 1:321–341, 1986.
27. Shuji Tsukiyama, Isao Shirakawa, Hiroshi Ozaki, and Hiromu Ariyoshi. An algorithm to enumerate all cutsets of a graph in linear time per cutset. *Journal of the ACM (JACM)*, 27(4):619–632, 1980.
28. Li Yan, Hamdy A Taha, and Thomas L Landers. A recursive approach for enumerating minimal cutsets in a network. *IEEE transactions on reliability*, 43(3):383–388, 1994.
29. Li-Pu Yeh, Biing-Feng Wang, and Hsin-Hao Su. Efficient algorithms for the problems of enumerating cuts by non-decreasing weights. *Algorithmica*, 56(3):297–312, 2010.