



Higher Connectivity in Directed Graphs

Giuseppe F. Italiano  

Luiss University, Rome, Italy

Abstract

The computation of edge-connected components in directed and undirected graphs is a well studied problem that is motivated by several applications (see, e.g., [28]). Let $G = (V, E)$ be a strongly connected directed graph with m edges and n vertices. An edge $e \in E$ is a *strong bridge* if $G \setminus e$ is not strongly connected. More generally, a set of edges $C \subseteq E$ is a *cut* if $G \setminus C$ is not strongly connected. If $|C| = k$ then we refer to C as a k -sized cut of G . Hence, a strong bridge is a 1-sized cut of G . A digraph G is k -edge-connected if it has no $(k - 1)$ -cuts. We say that two vertices v and w are k -edge-connected, and we denote this relation by $v \leftrightarrow_k w$, if there are k edge-disjoint directed paths from v to w and k edge-disjoint directed paths from w to v . (Note that a path from v to w and a path from w to v need not be edge-disjoint). By Menger's theorem [25], $v \leftrightarrow_k w$ if and only if the removal of any set of at most $k - 1$ edges leaves v and w in the same strongly connected component. We define a k -edge-connected component of a digraph $G = (V, E)$ as a maximal subset $U \subseteq V$ such that $u \leftrightarrow_k v$ for all $u, v \in U$. The k -edge-connected components of G form a partition of V , since $v \leftrightarrow_k w$ is an equivalence relation [13].

Connectivity-related problems are known to be much more difficult in directed graphs than in undirected graphs (see, e.g., [9, 19, 22]). Indeed, there is a fundamental difference in the structure of the cuts in the two scenarios. Specifically, it has been established more than 60 years ago [17] that edge cuts in undirected graphs have a nice structure, as defined by the *Gomory-Hu tree* (or *cut tree*), which plays a special role in identifying, for any k , the k -edge-connected components of undirected graphs. Furthermore, many efficient algorithms for computing Gomory-Hu trees are available (see e.g., [1–3, 6, 18, 24]). On the contrary, in directed graphs edge cuts have a more complicated structure, and it was proved by Benczúr [4] that in this case cut trees do not even exist. It is thus not surprising that, while it is known how to compute the k -edge-connected components of undirected graphs in linear time for $k \leq 5$ [8, 10, 11, 20, 23, 26, 27, 31, 33], the situation is more challenging for directed graphs, where linear-time algorithms are only known for $k \leq 2$ [14, 31]. Also, as argued in [15], there is a substantial increase in the inherent difficulty of the problem of computing k -edge-connected components in digraphs for $k = 3$ compared to $k = 2$. Indeed, for $k = 2$ any pair of vertices s, t that are not 2-edge-connected can be separated by only $O(n)$ s - t min-cuts of size 1, for which we can define a total order [21]. For $k = 3$, any pair of vertices s, t that are 2-edge-connected but not 3-edge-connected, can be separated by as many as $O(n^2)$ s - t min-cuts of size 2, which are also not totally ordered. This makes it difficult to explore the effect of removing each such cut of size 2 on the strong connectivity of the graph, similar to what was done for the case of $k = 2$ [14]. Until recently, the best-known bound for computing the k -edge-connected components of a digraph, for constant $k \geq 3$, was $O(mn)$ by Nagamochi and Watanabe [29]. Georgiadis et al. [15] presented a randomized (Monte-Carlo) algorithm that computes the 3-edge-connected components of a digraph with m edges in $\tilde{O}(m^{3/2})$ time. Their algorithm involves a nontrivial extension of the framework of [7, 30] for deciding whether a digraph is $(k + 1)$ -edge-connected. It applies a local search procedure [5, 7] for identifying 2-in or 2-out sets, i.e., vertex sets $S \subseteq V$ such that there are at most 2 edges from $V \setminus S$ to S or from S to $V \setminus S$. After finding such a set S , [15] applies an efficient graph operation for replacing S with a gadget of small size that preserves the pairwise connectivity among the vertices of $V \setminus S$. As in [7, 30], local search is initiated from sampled edges, but the overall scheme is more complicated to guarantee that enough 2-in sets or 2-out sets are identified that separate vertices that are not 3-edge-connected.

Recently, Georgiadis, Italiano and Kosinas [12] improved significantly the bound of [15] by showing how to compute the 3-edge-connected components of a digraph in *linear time* with a *deterministic* algorithm. Their algorithm differs substantially from [15], as it is based on a new characterization of 2-sized cuts in digraphs, which requires new techniques and a suitable combination of the notions of 2-connectivity-light graphs [15] and of maximally edge-disjoint strongly divergent



© Giuseppe F. Italiano;

licensed under Creative Commons License CC-BY 4.0

50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025).

Editors: Paweł Gawrychowski, Filip Mazowiecki, and Michał Skrzypczak; Article No. 2; pp. 2:1–2:4

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

spanning trees [16, 32]. In particular, Georgiadis, Italiano and Kosinas [12] showed how to modify the minset-poset technique of Gabow [9], in order to find the 3-edge-connected components of a digraph with m edges in $O(m)$ time.

In the invited talk, I will survey some of this recent work on higher connectivity on directed graphs.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases Connectivity, Directed graphs, Graph algorithms

Digital Object Identifier 10.4230/LIPIcs.MFCS.2025.2

Category Invited Talk

Funding *Giuseppe F. Italiano*: Partially supported by the Italian Ministry of University and Research under PRIN Project n. 2022TS4Y3N - EXPAND: scalable algorithms for EXploratory Analyses of heterogeneous and dynamic Networked Data.

References

- 1 Amir Abboud, Robert Krauthgamer, and Ohad Trabelsi. Subcubic algorithms for gomory-hu tree in unweighted graphs. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1725–1737. ACM, 2021. doi:10.1145/3406325.3451073.
- 2 Amir Abboud, Robert Krauthgamer, and Ohad Trabelsi. APMF < APSP? Gomory-Hu Tree for Unweighted Graphs in Almost-Quadratic Time. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1135–1146, 2022. doi:10.1109/FOCS52979.2021.00112.
- 3 Amir Abboud, Jason Li, Debmalya Panigrahi, and Thatchaphol Saranurak. All-pairs max-flow is no harder than single-pair max-flow: Gomory–Hu trees in almost-linear time. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2204–2212, 2023. doi:10.1109/FOCS57990.2023.00137.
- 4 András A. Benczúr. Counterexamples for directed and node capacitated cut-trees. *SIAM Journal on Computing*, 24(3):505–510, 1995. doi:10.1137/S0097539792236730.
- 5 Shiri Chechik, Thomas Dueholm Hansen, Giuseppe F. Italiano, Veronika Loitzenbauer, and Nikos Parotsidis. Faster algorithms for computing maximal 2-connected subgraphs in sparse directed graphs. In *Proc. 28th ACM-SIAM Symp. on Discrete Algorithms*, (SODA 2017), pages 1900–1918, 2017. doi:10.1137/1.9781611974782.124.
- 6 Li Chen, Rasmus Kyng, Yang P. Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–623, 2022. doi:10.1109/FOCS54457.2022.00064.
- 7 Sebastian Forster, Danupon Nanongkai, Liu Yang, Thatchaphol Saranurak, and Sorrachai Yingchareonthawornchai. Computing and testing small connectivity in near-linear time and queries via fast local cut algorithms. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2046–2065. SIAM, 2020. doi:10.1137/1.9781611975994.126.
- 8 Harold N. Gabow. Path-based depth-first search for strong and biconnected components. *Information Processing Letters*, 74:107–114, 2000. doi:10.1016/S0020-0190(00)00051-X.
- 9 Harold N. Gabow. The minset-poset approach to representations of graph connectivity. *ACM Transactions on Algorithms*, 12(2):24:1–24:73, February 2016. doi:10.1145/2764909.
- 10 Zvi Galil and Giuseppe F. Italiano. Reducing edge connectivity to vertex connectivity. *SIGACT News*, 22(1):57–61, March 1991. doi:10.1145/122413.122416.

- 11 Loukas Georgiadis, Giuseppe F. Italiano, and Evangelos Kosinas. Computing the 4-edge-connected components of a graph in linear time. In *Proc. 29th European Symposium on Algorithms*, pages 47:1–47:17, 2021. doi:10.4230/LIPICS.ESA.2021.47.
- 12 Loukas Georgiadis, Giuseppe F. Italiano, and Evangelos Kosinas. Computing the 3-edge-connected components of directed graphs in linear time. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 62–85, 2024. doi:10.1109/FOCS61266.2024.00015.
- 13 Loukas Georgiadis, Giuseppe F. Italiano, Luigi Laura, and Nikos Parotsidis. 2-edge connectivity in directed graphs. *ACM Transactions on Algorithms*, 13(1):9:1–9:24, 2016. doi:10.1145/2968448.
- 14 Loukas Georgiadis, Giuseppe F. Italiano, and Nikos Parotsidis. Strong connectivity in directed graphs under failures, with applications. *SIAM J. Comput.*, 49(5):865–926, 2020. doi:10.1137/19M1258530.
- 15 Loukas Georgiadis, Evangelos Kipouridis, Charis Papadopoulos, and Nikos Parotsidis. Faster computation of 3-edge-connected components in digraphs. In *Proc. 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2489–2531. SIAM, 2023. doi:10.1137/1.9781611977554.CH96.
- 16 Loukas Georgiadis and Robert E. Tarjan. Dominator tree certification and divergent spanning trees. *ACM Transactions on Algorithms*, 12(1):11:1–11:42, November 2015. doi:10.1145/2764913.
- 17 Ralph E Gomory and Tien Chung Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.
- 18 Ramesh Hariharan, Telikepalli Kavitha, and Debmalya Panigrahi. Efficient algorithms for computing all low s-t edge connectivities and related problems. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07*, pages 127–136, USA, 2007. Society for Industrial and Applied Mathematics. URL: <http://dl.acm.org/citation.cfm?id=1283383.1283398>.
- 19 Monika Henzinger, Satish Rao, and Di Wang. Local flow partitioning for faster edge connectivity. *SIAM Journal on Computing*, 49(1):1–36, 2020. doi:10.1137/18M1180335.
- 20 John E. Hopcroft and Robert E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2(3):135–158, 1973. doi:10.1137/0202012.
- 21 Giuseppe F. Italiano, Luigi Laura, and Federico Santaroni. Finding strong bridges and strong articulation points in linear time. *Theoretical Computer Science*, 447:74–84, 2012. doi:10.1016/j.tcs.2011.11.011.
- 22 Ken-Ichi Kawarabayashi and Mikkel Thorup. Deterministic edge connectivity in near-linear time. *Journal of the ACM*, 66(1), December 2018. doi:10.1145/3274663.
- 23 Evangelos Kosinas. Computing the 5-edge-connected components in linear time. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1887–2119, 2024. doi:10.1137/1.9781611977912.76.
- 24 Jason Li, Debmalya Panigrahi, and Thatchaphol Saranurak. A nearly optimal all-pairs min-cuts algorithm in simple graphs. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1124–1134, 2022. doi:10.1109/FOCS52979.2021.00111.
- 25 Karl Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.
- 26 Wojciech Nadara, Mateusz Radecki, Marcin Smulewicz, and Marek Sokołowski. Determining 4-edge-connected components in linear time. In *Proc. 29th European Symposium on Algorithms*, pages 71:1–71:15, 2021. doi:10.4230/LIPICS.ESA.2021.71.
- 27 Hiroshi Nagamochi and Toshihide Ibaraki. A linear time algorithm for computing 3-edge-connected components in a multigraph. *Japan J. Indust. Appl. Math.*, 9(163), 1992. doi:10.1007/BF03167564.
- 28 Hiroshi Nagamochi and Toshihide Ibaraki. *Algorithmic Aspects of Graph Connectivity*. Cambridge University Press, 2008. 1st edition. doi:10.1017/CB09780511721649.

2:4 Higher Connectivity in Directed Graphs

- 29 Hiroshi Nagamochi and Toshimasa Watanabe. Computing k -edge-connected components of a multigraph (special section on discrete mathematics and its applications). *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 76:513–517, 1993.
- 30 Danupon Nanongkai, Thatchaphol Saranurak, and Sorrachai Yingchareonthawornchai. Breaking quadratic time for small vertex connectivity and an approximation scheme. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 241–252, 2019. doi:10.1145/3313276.3316394.
- 31 Robert E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972. doi:10.1137/0201010.
- 32 Robert E. Tarjan. Edge-disjoint spanning trees and depth-first search. *Acta Informatica*, 6(2):171–85, 1976.
- 33 Yung H. Tsin. Yet another optimal algorithm for 3-edge-connectivity. *Journal of Discrete Algorithms*, 7(1):130–146, 2009. Selected papers from the 1st International Workshop on Similarity Search and Applications (SISAP). doi:10.1016/j.jda.2008.04.003.