Annual IEEE Symposium on Foundations of Computer Science 2024

Matej Belšak [2,*], Nik Čadež [2], Klemen Kavčič [2], Tomaž Leonardis [1], Bor Pangeršič [1], Marko Rozman [2], Pia Sotlar [2], Vanja Stojanović [1]

[1] Faculty of Mathematics and Physics, University of Ljubljana
[2] Faculty of Computer and Information Science, University of Ljubljana

Abstract

We will write this at the end

Introduction

The FOCS conference or IEEE Annual Symposium on Foundations of Computer Science is an academic conference that coveres a broad range of theoretical computer science. It is sponsered by the IEEE Computer Science Technical Committe on the Mathematical Foundations of Computing (TCMF) [1].

The FOCS 2024 took place in Chicago - Voco Chicago Downtown, from October 27-30, 2024 [2]. It covered a variety of topics, for submissions the following were mentioned [2]:

- Communication complexity
- Circuit complexity
- Average-case algorithms and complexity
- High-dimensional algorithms
- Online algorithms
- Parametrized algorithms
- Spectral methods
- Streaming algorithms
- Randomized algorithms
- Cryptography
- Computational complexity
- Algorithms and data structures
- Quantum computing
- Foundations of machine learning
- Algorithmic coding theory
- Sublinear algorithms
- Algorithmic graph theory
- Continuous optimization
- Foundations of fairness, privacy and databases
- Pseudorandomness and derandomization
- Markov chains
- Analysis of Boolean functions
- Economics and computation
- Combinatorial optimization
- Algebraic computation
- Approximation algorithms
- Parallel and distributed algorithms
- Computational learning theory
- Computational geometry
- Algorithmic game theory
- Combinatorics

The official welcome message of FOCS 2024 states that nearly 500 papers were submitted, but does not specify the exact number, out of which 133 were accepted and 131 were presented as talks during the event.

In the following three sections we present three curated papers from the conference... (dopisemo na koncu ko vemo katere)

Paper 1

Hi, I'm paper 1!

Computing the 3-Edge-Connected Components of Directed Graphs in Linear Time

Abstract

The paper describes a significant improvement of a *randomized* (Monte-Carlo) algorithm for computing the 3-edge-connected components of a digraph with m edges in polylogarithmic time $\widetilde{O}(m^{3/2})$. The algorithm described bests the previous one by being deterministic and computable in linear time.

Preliminaries and primary problem

This algorithm solves the problem of finding **3-edge-connected components in directed graphs**, for preliminaries; Let $G = (V, E)$ be a strongly connected directed graph with $|V(G)| = n$ and $|E(G)| = m$.

*Corresponding author: mb4390@student.uni-lj.si
Ljubljana, March 2025

Generally a set of edges $C \subseteq E$ is a **cut** if $G \setminus C$ is not strongly connected i.e. there does not exist a directed path between every pair of vertices, if $|C| = k$ we refer to C as a k**-sized cut** of G. Hence 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. We define a k**-edge-connected component** of a digraph G as a maximal subset $U \subseteq V(G)$ such that $u \leftrightarrow_k v, \forall u, v \in U$.

How they achieved this improvement

The authors derive and prove that instead of a randomized polylogarithmic algorithm, a deterministic linear one exists. Their method relies on a substructure of digraphs, known as 2**-connectivity-light graph** (denoted 2CLG). This is because the decomposition of digraphs into a collection of 2CLGs exists in linear time, and it maintains the 3-edge-connected components of the original graph. Besides 2CLGs they rely on the definition of minimal 2-in and -out sets, which contain vertices with out- or in-degree of 2. Formally, we define both here.

Definition 1. *A* 2***-connectivity-light*** *graph G is a strongly connected digraph that contains two types of vertices;* ***ordinary*** *and* ***auxiliary****, that satisfy the following conditions:*

1. *Any two ordinary vertices are* 2*-edge-connected,*
2. *each auxiliary vertex has an in- or out-degree of 1,*
3. *for every vertex u with out-degree > 1 and every vertex v with in-degree > 1, there are 2 edge-disjoint paths from u to v,*
4. *for each auxiliary vertex v with out-degree (resp. in-degree) of one, all paths from v to any vertex in G (resp. from any vertex in G to v), we have exactly one common edge, the unique out-edge (resp. in-edge).*

Definition 2. *For a strongly connected digraph G we arbitrarily choose a start vertex s. For any vertex $v \neq s$ we define $M(v)$ as a* ***minimal*** 2***-in set*** *that contains v, i.e. a minimal set of vertices which contains v, does not contains s, and has two incoming edges from $V(G) \setminus M(v)$, we denote by $M_R(v)$ the analogous sets in G^R, which is the reverse graph of G, i.e., the graph obtained from G after reversing the orientation of its edges.*

The technique is then based on the following proposition and theorem.

Proposition I.3. *Let G be a 2GLC with a fixed ordinary start vertex s. Then for any two ordinary vertices u and v, we have $v \leftrightarrow_k u$ if and only if $M(u) = M(v)$ and $M_R(u) = M_R(v)$.*

Theorem II.5. *Let G be a strongly connected digraph. In linear time, we can construct a collection $H_1, ..., H_t$ of 2CLG graphs, such that:*

- *For every vertex of G there is exactly one graph among $H_1, ..., H_t$, that contains it as an ordinary vertex.*
- *Every two vertices u and v of G are* 3*-edge-connected if and only if there is an $i \in \{1, ..., t\}$ such that u and v are 3-edge-connected.*

Paper 3

Hi, I'm paper 3!

References

[1] IEEE Symposium on Foundations of Computer Science, *Ieee focs conference*, Accessed: 2025-02-27, 2024. [Online]. Available: `https://ieee-focs.org`.

[2] IEEE Symposium on Foundations of Computer Science, *Focs 2024 - 65th annual ieee symposium on foundations of computer science*, Accessed: 2025-02-27, 2024. [Online]. Available: `https://focs.computer.org/2024/`.