Faster Algorithms for Maximal 2-Connected Subgraphs in Directed Graphs

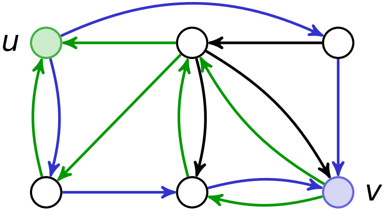
Veronika Loitzenbauer

Bar-Ilan University
veronika@datalab.cs.biu.ac.il

joint work with
Monika Henzinger and Sebastian Krinninger (ICALP'15) and
Shiri Chechik, Thomas D. Hansen, Giuseppe F. Italiano, and
Nikos Parotsidis (SODA'17)

2-Edge Connected

Can u and v still reach each other when an arbitrary edge is deleted?



Yes: u and v are 2-edge (strongly) connected

Graph is 2-edge-connected if all pairs of vertices are

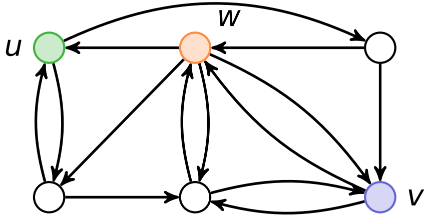
2-Vertex Connected

u and *v* are 2-vertex (strongly) connected

$\Leftrightarrow$

u and *v* still strongly connected

after any vertex other than *u* or *v* removed

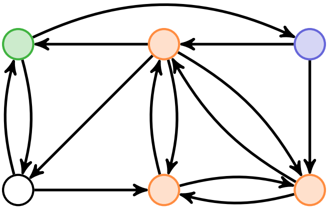


Graph is 2-vertex-connected if all pairs of vertices are and it has ≥ 3 vertices

Analyzing 2-Connectivity in Digraphs

1. **2-Connected Blocks/Components:**
   Which pairs of vertices are 2-connected?
   - Paths can use vertices not in same block
   - $O(m)$ time Georgiadis et al. SODA'15 & ICALP'15
2. **Maximal 2-Connected Subgraphs:**
   All vertex pairs 2-connected within subgraph
   This work. Open: Linear time algorithm?

Coincide for undirected graphs: $O(m)$ time Tarjan '72



Results

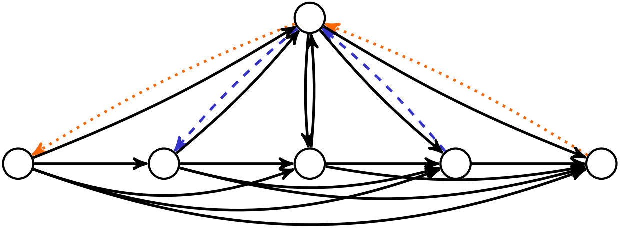
Baseline: $O(mn)$ time algorithm (Tarjan ’76, Georgiadis ’10)
m edges, n vertices

- $O(n^2)$ time algorithm
  - M. Henzinger, S. Krinninger, V. Loitzenbauer ICALP’15
- $O(m^{3/2})$ time algorithm
  - S. Chechik, T. D. Hansen, G. F. Italiano, V. Loitzenbauer, N. Parotsidis SODA’17
- extends to improvements for k-connected subgraphs for const. k, even for undirected graphs

This talk: 2-edge-connected subgraphs

Basic Algorithm

- As long as there is an edge whose removal increases number of SCCs, remove it
- Output remaining SCCs



$\Theta(mn)$ worst case

Beating $O(mn)$...

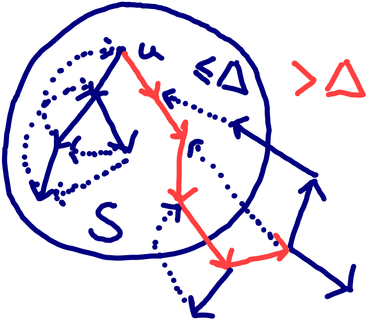
- Still $\Theta(n)$ iterations
- How to refine partition of vertices in $o(m)$?
- Find directed edge cut of size ≤ 1 in each iteration
  - 1-edge-out set: vertex set S with ≤ 1 edge to $V \setminus S$
- In proper subgraph in time proportional to size of S
  - $O(n^2)$: in time $O(n \cdot |S|)$
    consider i outgoing edges per vertex to find $|S| \leq 2^i$
  - $O(m^{3/2})$: in time $O(|E(S)|)$
    "local" depth-first search from vertices that lost edges

1-Ege-Out Set S of u of Size $\leq \Delta$

Idea: Send one unit of flow from u to vertex outside of S

$\Rightarrow$ No path out of S in residual graph

$\Rightarrow$ Second search from u explores S



- Run DFS from u for $2\Delta + 1$ edges
- Path of vertices whose subtraversal had $> \Delta$ edges

Thank you!