Incremental SCC Maintenance in Sparse Graphs

Aaron Bernstein
Rutgers University

Aditi Dudeja
Rutgers University

Seth Pettie
University of Michigan

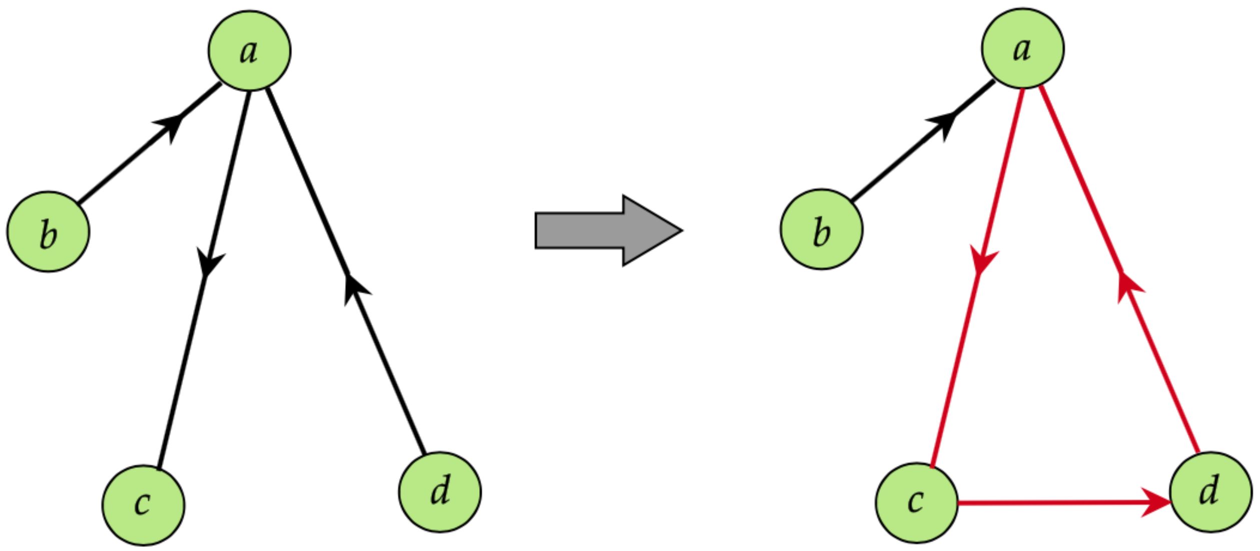
Background

Cycle Detection:

1. Initially: empty graph, adversary adds directed edges.

2. The algorithm stops when a cycle appears.

3. Related problem: maintaining a topological sort.

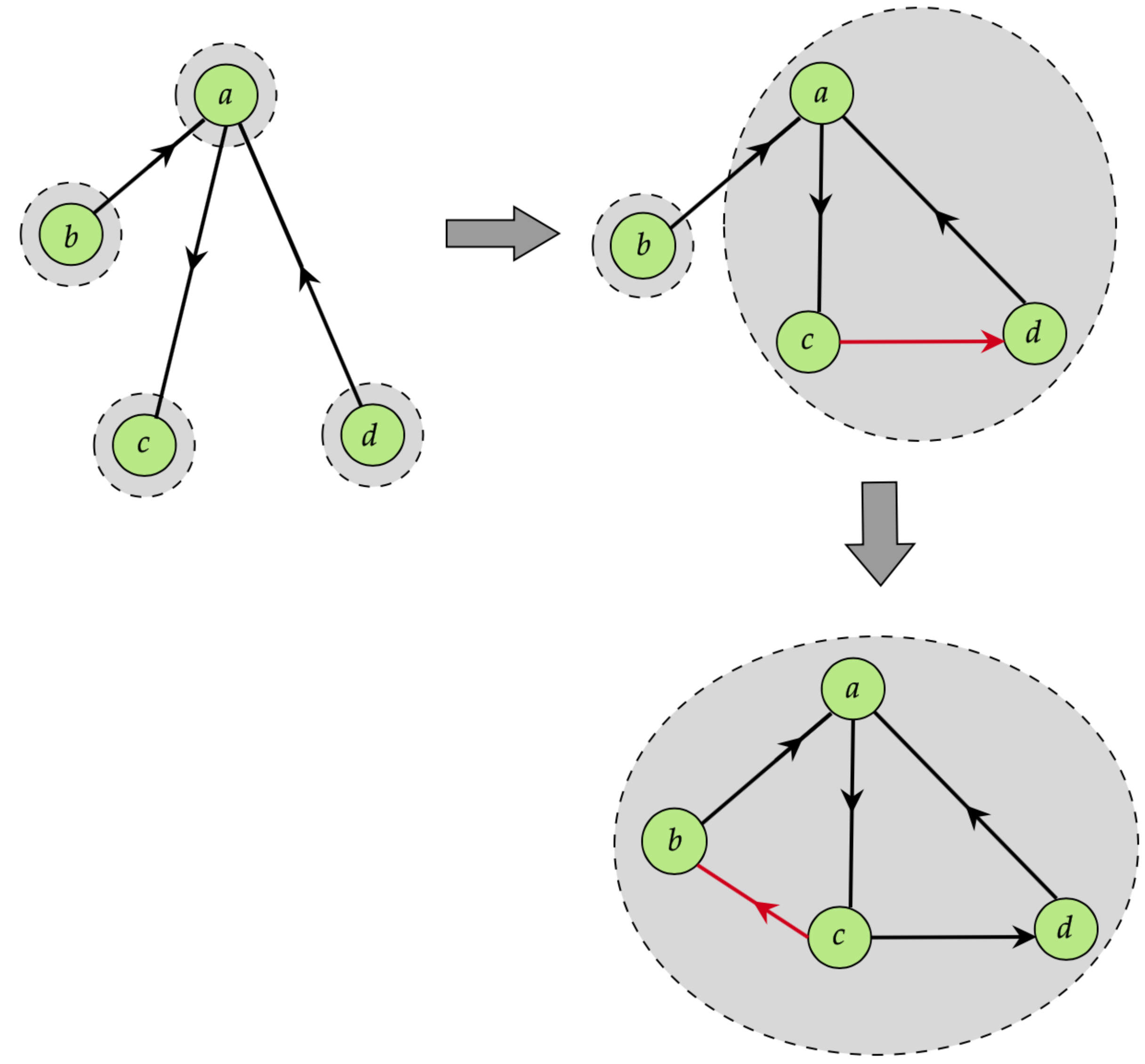


Background

SCC Maintenance:

1. The algorithm doesn't terminate on first appearance of cycle. Instead, update strongly connected components.

2. Related problem: maintaining a topological sort.



Known Results

Reference	Update Time	Incremental SCC
[KB06]	$O(\min\{m^{3/2}\log n, m^{3/2}+n^2\log n\})$	No
[LC07]	$O(m^{3/2}+m\sqrt{n}\log n)$	No
[AFM08]	$O(n^{2.75})$	No
[BFG09]	$\tilde{O}(n^2)$	No
[HKM+12]	$O(m^{3/2})$	Yes
[BFGT16]	$O(m\cdot\min\{m^{1/2}, n^{2/3}\})$, $\tilde{O}(n^2)$	Yes
[BC18]	$\tilde{O}(m\sqrt{n})$	No
[BK20]	$\tilde{O}(m^{4/3})$	No

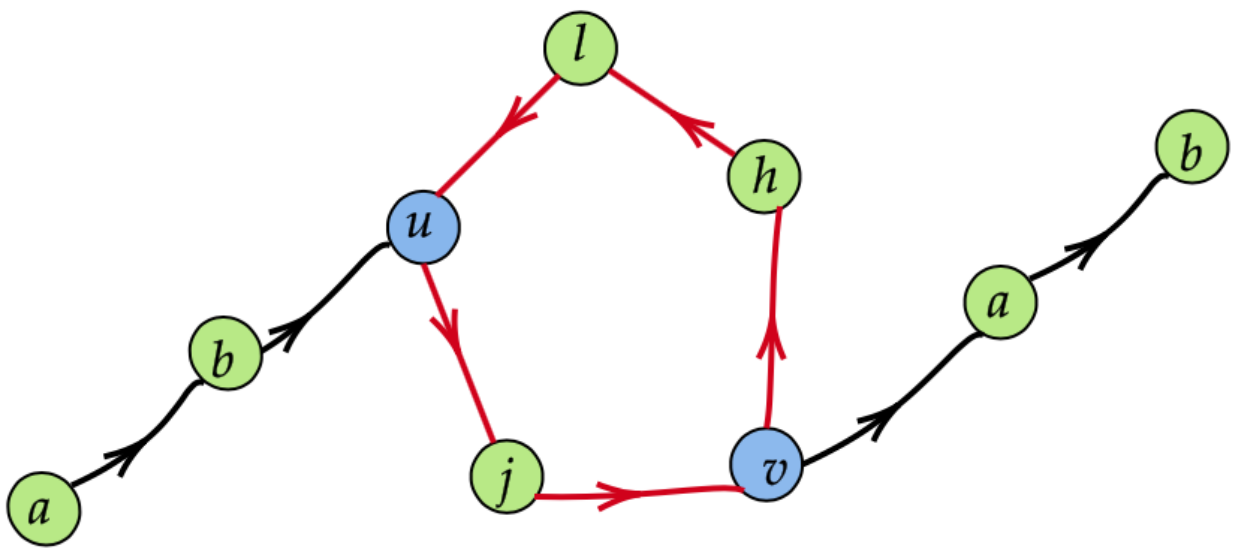
Our results: Incremental SCC maintenance in $\tilde{O}\left(m^{\frac{4}{3}}\right)$ total time.

Techniques

Observation:

u and v are on a cycle iff $A(u) = A(v), D(u) = D(v)$.

Too expensive to maintain for all vertices.



Techniques

Framework of [BC18]:

1. Sample a set of hubs S.

2.Let $V_{i,j} = \{v$ **with** $|A(v) \cap S| = i, |D(v) \cap S| = j\}$.

3. A cycle, if present is contained in a fixed $G[V_{i,j}]$.

4. If edge (u, v) is inserted then do a forward search from v
Exploring nodes reachable from v but lying in $G[V_{i,j}]$.

Runtime

τ **- related nodes:** There is a path from u to v and $|A(v) \backslash A(u)| \leq \tau$, $|D(u) \backslash D(v)| \leq \tau$.

Intuition for Relatedness

1. Let $|S| = \dfrac{100 n \log n}{\tau}$.

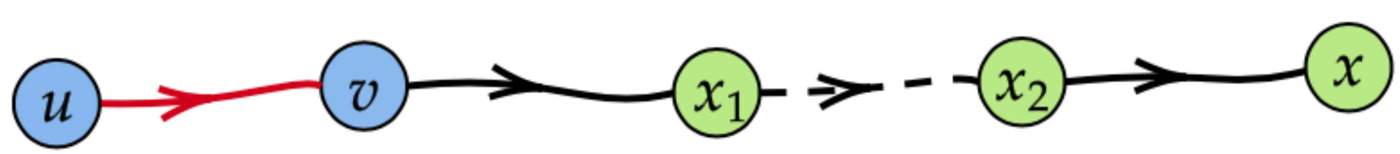
2. Suppose there is a path from u to v and they lie in the same partition, then they are τ- related with high probability.

Why? $A(u) \cap S \subseteq A(v) \cap S, D(v) \cap S \subseteq D(u) \cap S$.

Runtime

1. For each node x explored during forward search, we get a τ-related pair (u, x).

2. Total number of τ-related pairs at any time = $\tilde{O}(n\tau)$



(u, x_1), (u, x_2), $\cdots$, (u, x) are new τ − related pairs.

Techniques

Framework of [BK20]:

1. Combines [BC18] with balanced search: if an edge (u, v) is inserted then, alternately do forward search from v and backward search from u .

2. Now, if total number of explored vertices is λ, then the $O(\lambda^2)$ τ - related pairs are discovered. This gives a better bound.

$(i+1)^2$ new τ – related pairs formed.

Maintaining Components

As before:

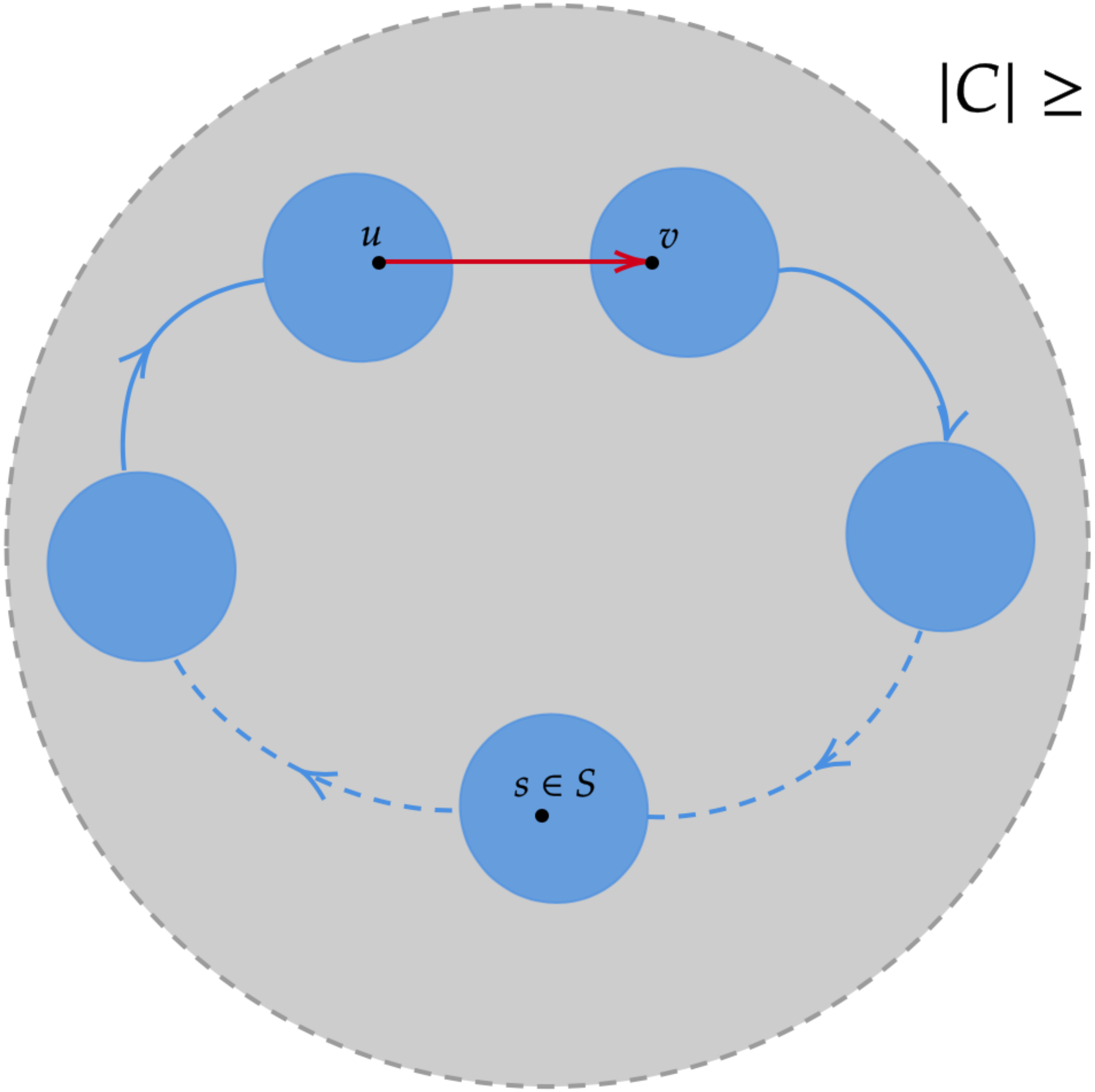
1. Sample a set of hubs S.

2. Let $V_{i,j} = \{v$ **with** $|A(v) \cap S| = i, |D(v) \cap S| = j\}$.

3. A new SCC, if present is contained in a fixed $G[V_{i,j}]$.

4. If edge (u, v) is inserted then do a forward search from v and backward search from u. Exploring nodes in $G[V_{i,j}]$.

Maintaining Components

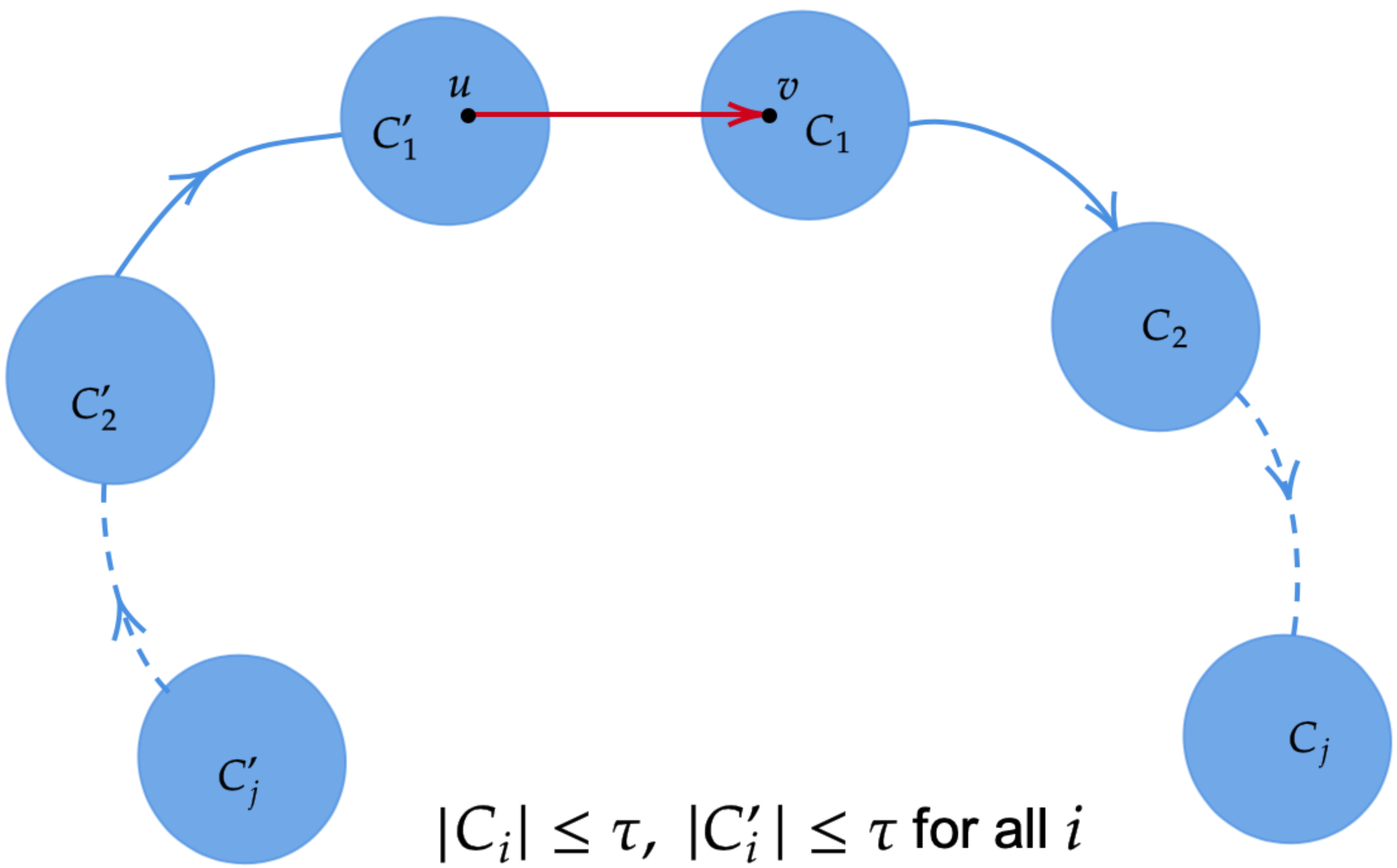


$|C| \geq \tau + 1$

We update C while updating $A(s), D(s)$.

Maintaining Components

So we only do search when there is no new component, or the new component is small.



$|C_i| \leq \tau,\ |C'_i| \leq \tau$ for all i

Maintaining Components

τ- related nodes: There is a path from u to v, $|A(u)\backslash A(v)| \leq \tau$, $|D(v)\backslash D(u)| \leq \tau$, and the SCCs containing u and v are smaller than τ.

Total number of τ- related pairs is $\tilde{O}(n\tau)$.