Breaking the Quadratic Barrier for Matroid Intersection

Joakim Blikstad[1] (me) Jan van den Brand[1]
Sagnik Mukhopadhyay[1] Danupon Nanongkai[1,2]

STOC 2021

[1] KTH Royal Institute of Technology, Sweden
[2] University of Copenhagen, Denmark

Summary

Result:

First subquadratic independence-query matroid intersection algorithm.

- Previous best: $\tilde{O}(n^2)$ queries.
- **Ours:** $\tilde{O}(n^{9/5})$ randomized and $\tilde{O}(n^{11/6})$ deterministic.

Summary

Result:

First subquadratic independence-query matroid intersection algorithm.

- Previous best: $\tilde{O}(n^2)$ queries.
- **Ours:** $\tilde{O}(n^{9/5})$ randomized and $\tilde{O}(n^{11/6})$ deterministic.

Technique:

Previous work + a **new** simple subquadratic reachability algorithm.

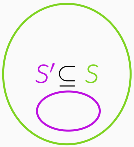
- Previous best: $O(n^2)$ queries.
- **Ours:** $\tilde{O}(n^{3/2})$ randomized and $\tilde{O}(n^{5/3})$ deterministic.

What is a Matroid?

- Set of elements V. $n = |V|$.
- Notion of independence $\mathcal{I} \subseteq 2^V$.

Downward Closure

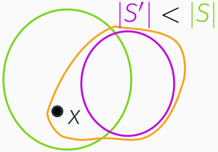
$S \in \mathcal{I}$



$\Rightarrow S' \in \mathcal{I}$

Exchange Property

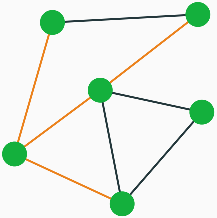
$S \in \mathcal{I}$ $S' \in \mathcal{I}$



$\Rightarrow \exists x \in S \setminus S'$ such that $S' \cup \{x\} \in \mathcal{I}$

Matroids — Examples

Graphic Matroid



$V = \text{edges}$

$\mathcal{I} = \text{forests}$

Linear Matroid

$$\begin{bmatrix} 0 & 1 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 & 0 \\ 2 & 0 & 2 & 4 & 0 \\ 1 & 1 & 3 & 2 & 1 \\ 0 & 0 & 1 & 0 & 5 \end{bmatrix}$$

$V = \text{row vectors}$

$\mathcal{I} = \text{linearly independent}$

Matroid Intersection

Given: two matroid $\mathcal{M}_1 = (V, \mathcal{I}_1)$ and $\mathcal{M}_2 = (V, \mathcal{I}_2)$

Goal: find a *common independent set* $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ of maximum size.

Matroid Intersection

Matroid Intersection

Given: two matroid $\mathcal{M}_1 = (V, \mathcal{I}_1)$ and $\mathcal{M}_2 = (V, \mathcal{I}_2)$

Goal: find a *common independent set* $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ of maximum size.

How do we access the matroids?

Matroid Intersection

Given: two matroid $\mathcal{M}_1 = (V, \mathcal{I}_1)$ and $\mathcal{M}_2 = (V, \mathcal{I}_2)$

Goal: find a *common independent set* $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ of maximum size.

How do we access the matroids?

Independence oracle queries: *Is* $X \in \mathcal{I}_1$*?* *Is* $X \in \mathcal{I}_2$*?*

Matroid Intersection

Matroid Intersection

Given: two matroid $\mathcal{M}_1 = (V, \mathcal{I}_1)$ and $\mathcal{M}_2 = (V, \mathcal{I}_2)$

Goal: find a *common independent set* $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ of maximum size.

How do we access the matroids?

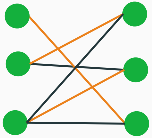
Independence oracle queries: *Is* $X \in \mathcal{I}_1$? *Is* $X \in \mathcal{I}_2$?

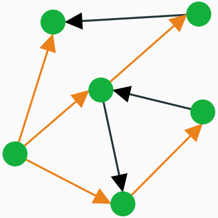
Intersection of **three** matroids is NP-hard.

Matroid Intersection — Examples

Models many combinatorial optimization problems

- Bipartite matching
  - $\mathcal{M}_1 =$ "≤ 1 edge per vertex on the left"
  - $\mathcal{M}_2 =$ "≤ 1 edge per vertex on the right"
- Arborescence (directed spanning tree)
- Colorful spanning trees
- Tree packing
- Graph orientation problems
- ...

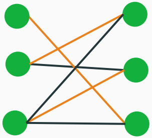


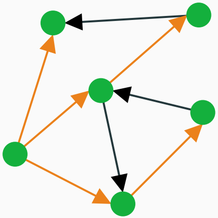


Matroid Intersection — Examples

Models many combinatorial optimization problems

- Bipartite matching
  - $\mathcal{M}_1 =$ "≤ 1 edge per vertex on the left"
  - $\mathcal{M}_2 =$ "≤ 1 edge per vertex on the right"
- Arborescence (directed spanning tree)
- Colorful spanning trees
- Tree packing
- Graph orientation problems
- . . .





Also connections to Submodular Function Minimization

Previous work

1960s-70s Edmonds, Lawler and Aigner-Downling: $O(n^3)$ queries

- Finding augmenting paths in the *exchange graph*.

Previous work

1960s-70s Edmonds, Lawler and Aigner-Downling: $O(n^3)$ queries

- Finding augmenting paths in the *exchange graph*.

1986 Cunningham: $O(n^{2.5})$ queries

- Blocking-flow ideas from Hopcroft-Karp algorithm.

Previous work

1960s-70s Edmonds, Lawler and Aigner-Downling: $O(n^3)$ queries

- Finding augmenting paths in the *exchange graph*.

1986 Cunningham: $O(n^{2.5})$ queries

- Blocking-flow ideas from Hopcroft-Karp algorithm.

2015 Lee-Sidford-Wong: $\tilde{O}(n^2)$ queries

- Cutting plane method.

Previous work

1960s-70s Edmonds, Lawler and Aigner-Downling: $O(n^3)$ queries

- Finding augmenting paths in the *exchange graph*.

1986 Cunningham: $O(n^{2.5})$ queries

- Blocking-flow ideas from Hopcroft-Karp algorithm.

2015 Lee-Sidford-Wong: $\tilde{O}(n^2)$ queries

- Cutting plane method.

2019 Chakrabarty-Lee-Sidford-Singla-Wong and Nguyễn: $\tilde{O}(n^2)$

- Efficient implementations of Cunningham's algorithm.

Major open problem:

Can we break this Quadratic Barrier?

Can we break this quadratic barrier?

Can we break this quadratic barrier?

YES, with a more powerful rank-oracle.

- Algorithm using $\tilde{O}(n^{1.5})$ rank-queries. [CLSSW 2019]

Queries

Independence: Is $X \in \mathcal{I}$?

Rank: What is $\max_{Y \subseteq X,\ Y \in \mathcal{I}} |Y|$?

Can we break this quadratic barrier?

YES, with a more powerful rank-oracle.

- Algorithm using $\tilde{O}(n^{1.5})$ rank-queries. [CLSSW 2019]

YES, for a $(1-\varepsilon)$-approximate solution.

- Algorithm using $\tilde{O}(n^{1.5}/\varepsilon^{1.5})$ independence-queries. [CLSSW 2019]

Can we break this quadratic barrier?

YES, with a more powerful rank-oracle.

- Algorithm using $\tilde{O}(n^{1.5})$ rank-queries. [CLSSW 2019]

YES, for a $(1-\varepsilon)$-approximate solution.

- Algorithm using $\tilde{O}(n^{1.5}/\varepsilon^{1.5})$ independence-queries. [CLSSW 2019]

Our contribution: YES!

For the classic independence-query and with exact solution.

Can we break this quadratic barrier?

YES, with a more powerful rank-oracle.

- Algorithm using $\tilde{O}(n^{1.5})$ rank-queries. [CLSSW 2019]

YES, for a $(1-\varepsilon)$-approximate solution.

- Algorithm using $\tilde{O}(n^{1.5}/\varepsilon^{1.5})$ independence-queries. [CLSSW 2019]

Our contribution: YES!

For the classic independence-query and with exact solution.

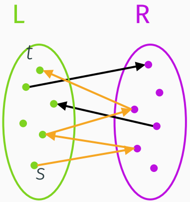
- Randomized: $\tilde{O}(n^{9/5})$ independence-queries.
- Deterministic: $\tilde{O}(n^{11/6})$ independence-queries.

Proof outline

Reachability Problem

Given: Directed bipartite graph G with bipartition (L, R);
Two vertices $s, t \in \mathsf{L}$.

Goal: Find an (s, t)-path, or determine none exist.



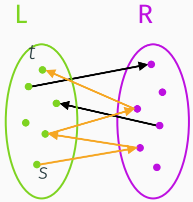
Reachability Problem

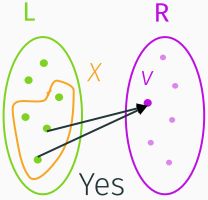
Given: Directed bipartite graph G with bipartition (L, R);
Two vertices $s, t \in$ L.

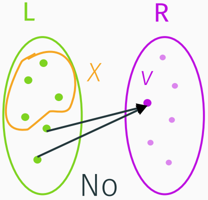
Goal: Find an (s, t)-path, or determine none exist.

Queries: Specify $v \in$ R and $X \subseteq$ L and ask:

- Does v have an in-neighbor from X?
- Does v have an out-neighbor to X?







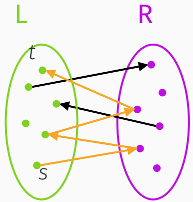
Reachability Problem

Given: Directed bipartite graph G with bipartition (L, R);
Two vertices $s, t \in \mathsf{L}$.

Goal: Find an (s, t)-path, or determine none exist.

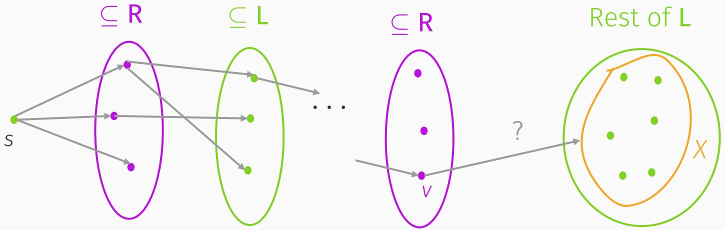
Queries: Specify $v \in \mathsf{R}$ and $X \subseteq \mathsf{L}$ and ask:

- Does v have an in-neighbor from X?
- Does v have an out-neighbor to X?



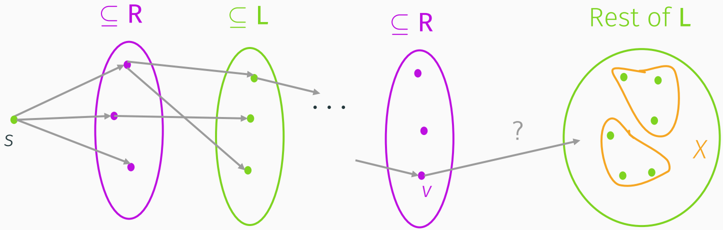
Theorem: Subquadratic **Reachability Problem**
$\implies$ Subquadratic **Matroid Intersection**.

Idea: Many short paths — CLSSW approximation algorithm
Few long paths — Reachability problem.



Allowed Queries: Does $v \in$ **R** have an {out/in}-neighbor from $X \in$ **L**?

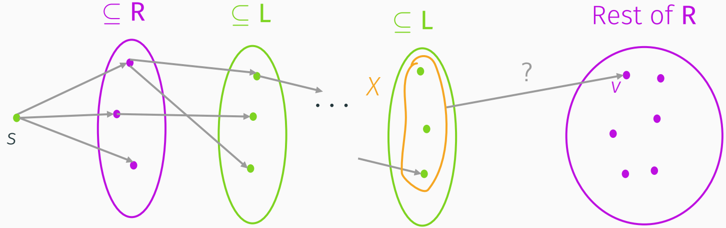
First try: Breadth First Search — from R to L



Allowed Queries: Does $v \in$ R have an {out/in}-neighbor from $X \in$ L?

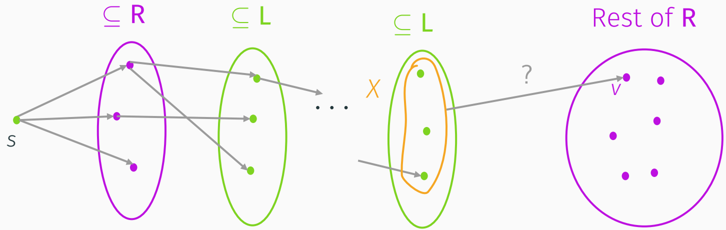
Binary-search: $O(\log n)$

First try: Breadth First Search — from L to R



Allowed Queries: Does $v \in$ R have an {out/in}-neighbor from $X \in$ L?

First try: Breadth First Search — from L to R

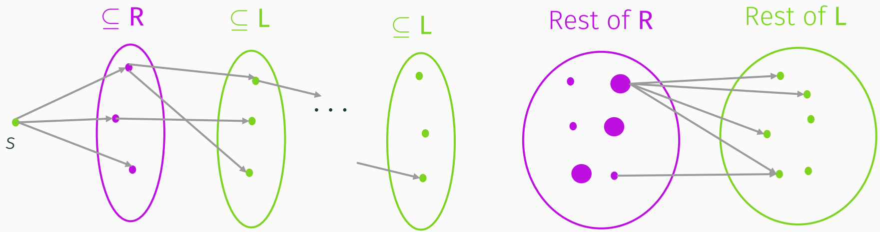


Allowed Queries: Does $v \in$ **R** have an {out/in}-neighbor from $X \in$ **L**?

Need $\Omega(n)$ queries
Total: $\Theta(n^2)$ queries

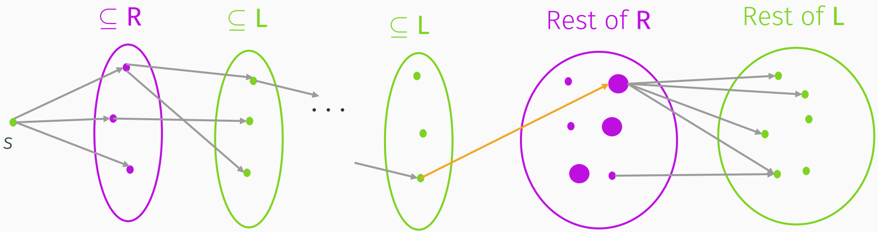
Can we do better?

Heavy and Light Vertices



- Heavy: $v \in$ R has large out-degree $(> \sqrt{n})$
- Light: $v \in$ R has small out-degree$(\leq \sqrt{n})$

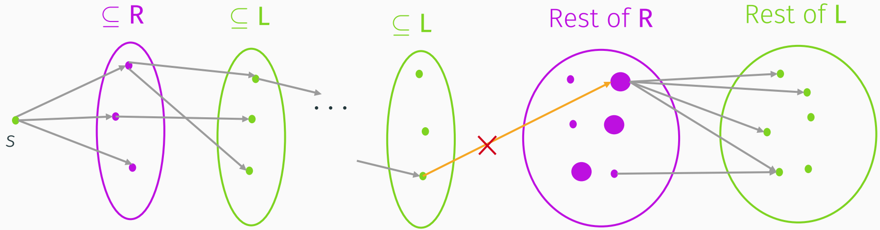
Heavy and Light Vertices



- **Heavy:** $v \in$ R has large out-degree $(> \sqrt{n})$
- **Light:** $v \in$ R has small out-degree$(\leq \sqrt{n})$

Discovering an heavy vertex is good! Only happens $\frac{n}{\sqrt{n}} = \sqrt{n}$ times.

Heavy and Light Vertices



- Heavy: $v \in$ R has large out-degree $(> \sqrt{n})$
- Light: $v \in$ R has small out-degree$(\leq \sqrt{n})$

Discovering an heavy vertex is good! Only happens $\frac{n}{\sqrt{n}} = \sqrt{n}$ times.

But what if next layer consists of only light vertices?

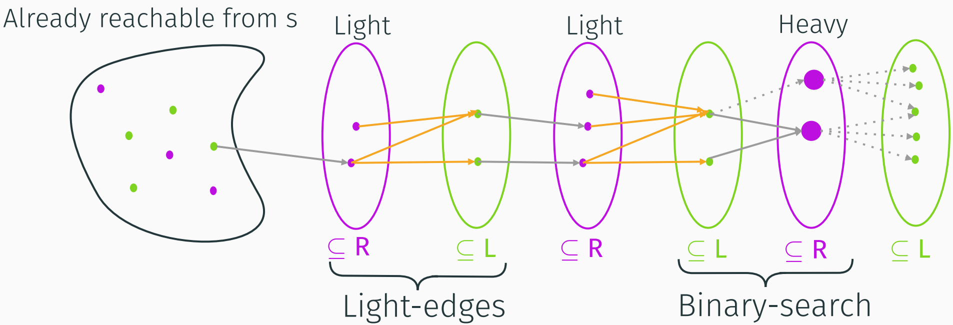
Our main insight: We can still efficiently find a *heavy* vertex!

Reverse BFS

- Light vertices have only $O(\sqrt{n})$ outgoing edges. **Find all of them!**

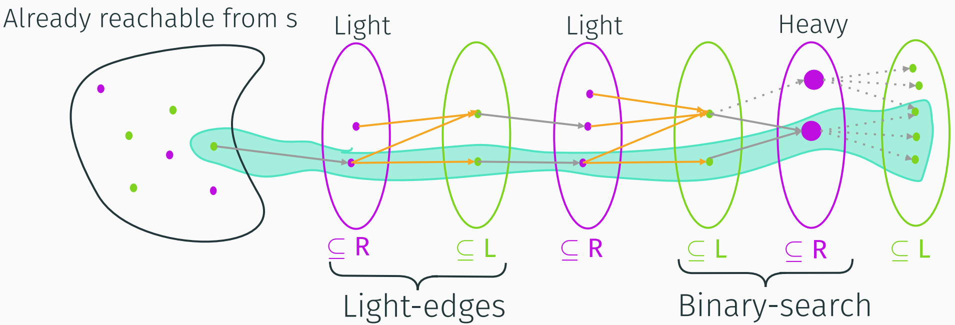
Reverse BFS

- Light vertices have only $O(\sqrt{n})$ outgoing edges. **Find all of them!**
- BFS starting from all heavy vertices in the reverse graph.



Reverse BFS

- Light vertices have only $O(\sqrt{n})$ outgoing edges. **Find all of them!**
- BFS starting from all heavy vertices in the reverse graph.



Reachability Problem — Algorithm

Run in $O(\sqrt{n})$ phases:

- Categorize heavy / light

- Find all outgoing edges of newly light vertices
- Use the reverse BFS to find a heavy vertex reachable from s

Reachability Problem — Algorithm

Run in $O(\sqrt{n})$ phases:

- Categorize heavy / light
  - Random sampling ← only place we use randomization
  - Carefully keeping track of small lists of neighbors ← less efficient
- Find all outgoing edges of newly light vertices
- Use the reverse BFS to find a heavy vertex reachable from s

Reachability Problem — Algorithm

Run in $O(\sqrt{n})$ phases:

- Categorize heavy / light
  - Random sampling ← only place we use randomization
  - Carefully keeping track of small lists of neighbors ← less efficient
- Find all outgoing edges of newly light vertices
- Use the reverse BFS to find a heavy vertex reachable from s

Total Query Complexity: $\tilde{O}(n\sqrt{n})$ randomized or $\tilde{O}(n^{5/3})$ deterministic.

Summary

- **Reachability Problem**: subquadratic number of queries.
  - Previous best: $O(n^2)$ queries.
  - **Ours:** $\tilde{O}(n^{3/2})$ randomized and $\tilde{O}(n^{5/3})$ deterministic.

Previous work + subquadratic Reachability Problem $\implies$

- **Matroid Intersection**: subquadratic number of *independence-queries*.
  - Previous best: $\tilde{O}(n^2)$ queries.
  - **Ours:** $\tilde{O}(n^{9/5})$ randomized and $\tilde{O}(n^{11/6})$ deterministic.

Open Problems

- Gap between lower and upper bounds for matroid intersection.
    - No $\Omega(n^{1+\epsilon})$ lower-bound is known for $\epsilon > 0$.
- Tight bounds for the reachability problem.
  We conjecture that our $\tilde{O}(n\sqrt{n})$ bound is tight.
- Can one also solve **weighted** matroid intersection with subquadratic number of queries?
- Investigating the gap between *independence* and *rank* oracle models.
    - Reachability Problem: $O(n\sqrt{n})$ vs $O(n)$.
    - Approximate Matroid Intersection: $O(n\sqrt{n}/\text{poly}(\varepsilon))$ vs $O(n/\varepsilon)$.

Thanks!

This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme under grant agreement No 715672. Jan van den Brand is partially supported by the Google PhD Fellowship Program. Danupon Nanongkai and Sagnik Mukhopadhyay are also partially supported by the Swedish Research Council (Reg. No. 2019-05622).