Deterministic $\left(\frac{2}{3}-\varepsilon\right)$-approximation of Matroid Intersection Using Nearly-Linear Independence-Oracle Queries

Tatsuya Terao (Kyoto University)

WADS 2025 @ Toronto

What is Matroid?

Def.

A finite set V and non-empty collection $\mathcal{I}$ of subset of V

s.t.

① $S' \subseteq S \in \mathcal{I} \Rightarrow S' \in \mathcal{I}$

② $S, T \in \mathcal{I}, |S| > |T| \Rightarrow \exists e \in S \setminus T$ s.t. $T \cup \{e\} \in \mathcal{I}$

What is Matroid? - Generalization of Linear Independence

Def.

A finite set V and non-empty collection $\mathcal{I}$ of subset of V

s.t. ① $S' \subseteq S \in \mathcal{I} \Rightarrow S' \in \mathcal{I}$

② $S, T \in \mathcal{I}, |S| > |T| \Rightarrow \exists e \in S \setminus T$ s.t. $T \cup \{e\} \in \mathcal{I}$

e.g.

- Linear matroid

$$\begin{pmatrix} \boxed{\begin{matrix} 1 \\ 2 \end{matrix}} & \boxed{\begin{matrix} 1 \\ 1 \end{matrix}} & \boxed{\begin{matrix} 2 \\ 2 \end{matrix}} \end{pmatrix}$$

$V = \{①, ②, ③\}$

$\mathcal{I} = \{\phi, \{①\}, \{②\}, \{③\}, \{①, ②\}, \{①, ③\}\}$

What is Matroid? - Generalization of Linear Independence

Def.

A finite set V and non-empty collection $\mathcal{I}$ of subset of V

s.t. ① $S' \subseteq S \in \mathcal{I} \Rightarrow S' \in \mathcal{I}$

② $S, T \in \mathcal{I}, |S| > |T| \Rightarrow \exists e \in S \setminus T$ s.t. $T \cup \{e\} \in \mathcal{I}$

$S \in \mathcal{I}$: S is independent

e.g.

- Linear matroid

$$\begin{pmatrix} 1 & 1 & 2 \\ 2 & 1 & 2 \end{pmatrix}$$

$V = \{①, ②, ③\}$

$\mathcal{I} = \{\phi, \{①\}, \{②\}, \{③\}, \{①, ②\}, \{①, ③\}\}$

What is Matroid? - Generalization of Linear Independence

Def.

A finite set V and non-empty collection $\mathcal{I}$ of subset of V

s.t.

① $S' \subseteq S \in \mathcal{I} \Rightarrow S' \in \mathcal{I}$

② $S, T \in \mathcal{I}, |S| > |T| \Rightarrow \exists e \in S \setminus T$ s.t. $T \cup \{e\} \in \mathcal{I}$

e.g.

- Linear matroid

$$\begin{pmatrix} 1 & 1 & 2 \\ 2 & 1 & 2 \end{pmatrix}$$

$V = \{①, ②, ③\}$

$\mathcal{I} = \{\phi, \{①\}, \{②\}, \{③\}, \{①, ②\}, \{①, ③\}\}$

- partition matroid
- laminar matroid
- graphic matroid

etc...

What is Matroid Intersection?

input: two matroids $M_1 = (V, \mathcal{I}_1)$, $M_2 = (V, \mathcal{I}_2)$

output: maximum common independent set $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

What is Matroid Intersection?

— Generalization of Bipartite Matching

input: two matroids $M_1 = (V, \mathcal{I}_1)$, $M_2 = (V, \mathcal{I}_2)$

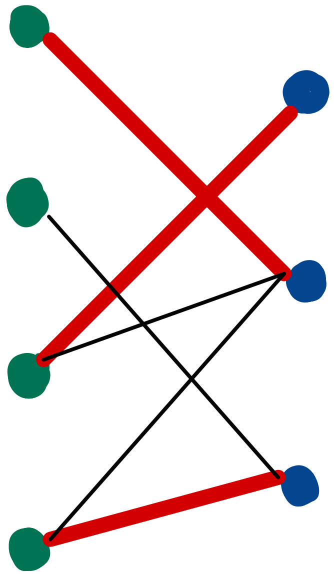
output: maximum common independent set $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

e.g. Maximum Bipartite Matching

V = edges

$\mathcal{I}_1$ = each left vertex has at most 1 edge

$\mathcal{I}_2$ = each right vertex has at most 1 edge



What is Matroid Intersection?

— Generalization of Bipartite Matching

input: two matroids $M_1 = (V, \mathcal{I}_1)$, $M_2 = (V, \mathcal{I}_2)$
output: maximum common independent set $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

Why this problem matters?

What is Matroid Intersection?

— Generalization of Bipartite Matching

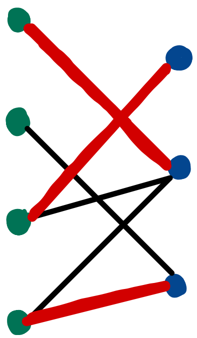
input: two matroids $M_1 = (V, \mathcal{I}_1)$, $M_2 = (V, \mathcal{I}_2)$

output: maximum common independent set $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

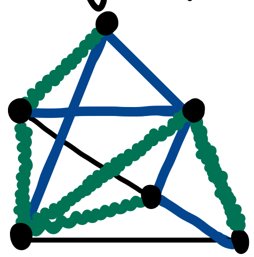
Why this problem matters?

1 Generalize many problems

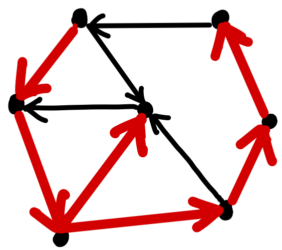
Bipartite Matching



Packing Spanning Tree



Arborescences



What is Matroid Intersection?

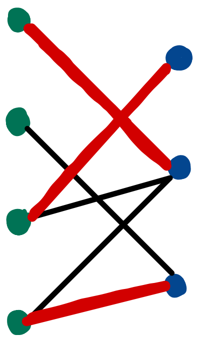
— Generalization of Bipartite Matching

input: two matroids $M_1 = (V, \mathcal{I}_1)$, $M_2 = (V, \mathcal{I}_2)$
output: maximum common independent set $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

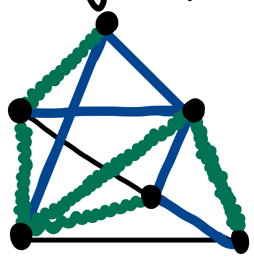
Why this problem matters?

① Generalize many problems

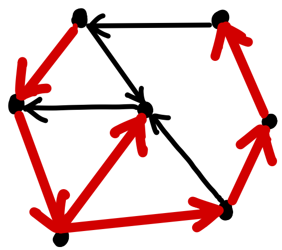
Bipartite Matching



Packing Spanning Tree



Arborescences



② Applications extend beyond Combinatorial Optimization

- electrical engineering
- network coding

input: $(V,\mathcal{I}_1),(V,\mathcal{I}_2)$

$\max |S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

Matroid Intersection can be solved in Polynomial time

[Edmonds'70, Aigner-Dowling'71, Lawler'75]

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$\max |S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

Matroid Intersection can be solved in Polynomial time

[Edmonds'70, Aigner-Dowling'71, Lawler'75]

Matroid is given via an independence oracle

Query — Answer

Is $S \in \mathcal{I}$? → Independence Oracle → Yes or No

Time Complexity = # of Queries

Time Complexity of Matroid Intersection

input : $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$\max |S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

\# of queries
to independence oracle

Time Complexity of Matroid Intersection

input : $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$
$\max |S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

		# of queries to independence oracle
1970s	Edmonds, Lawler, Aigner-Dowling	$O(nr^2)$
1986	Cunningham	$O(nr^{3/2})$

input : $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$
$\max |S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

Time Complexity of Matroid Intersection

of queries to independence oracle

1970s	Edmonds, Lawler, Aigner-Dowling	$O(nr^2)$
1986	Cunningham	$O(nr^{3/2})$
2015	Lee-Sidford-Wong	$\tilde{O}(n^2)$
2019	Nguyễn, Chakrabarty-Lee-Sidford-Singla-Wong	$\tilde{O}(nr)$
2021	Blikstad-v.d.Brand-Mukhopadhyay-Nanongkai	$\tilde{O}(n^{9/5})$
2021	Blikstad ※	$\tilde{O}(nr^{3/4})$

※ [Blikstad21] : $\tilde{O}(nr^{3/4})$ if randomized, $\tilde{O}(nr^{5/6})$ if deterministic

Today's Focus: Nearly Linear-Time Algo.

$O(n \cdot \mathrm{polylog}(n))$-time

Today's Focus: Nearly Linear-Time Algo.

$O(n \cdot \mathrm{polylog}(n))$-time

★ Targeting Approximation Rather than Exact Solution

α-approx.: $S \in I_1 \cap I_2$

s.t. $|S| \geq \alpha \cdot OPT$

Today's Focus: Nearly Linear-Time Algo.

$O(n \cdot \mathrm{polylog}(n))$-time

★ Targeting Approximation Rather than Exact Solution

α-approx.: $S \in I_1 \cap I_2$
s.t. $|S| \geq \alpha \cdot OPT$

e.g.

sparse cut : Khandekar-Rao-Vazirani'09, Madry'10

k-cut : Quanrud'19

k-ECSS : Chalermsook-Huang-Nanongkai-Saranurak-Sukprasert-Yingchareonthawornchai'22

weighted matching : Preis'99, Vinkemeier-Hougardy'03, Duan-Pettie'14, Zheng-Henzinger'23

Nearly-Linear Time Algo. for Matroid Intersection

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$
α-approx.: find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq \alpha \cdot r$

Folklore	$\frac{1}{2}$-approx.	$O(n)$ queries

input: $(V,\mathcal{I}_1),(V,\mathcal{I}_2)$
$n := |V|$, $r := \max_{S\in\mathcal{I}_1\cap\mathcal{I}_2} |S|$
α-approx.: find $S\in\mathcal{I}_1\cap\mathcal{I}_2$ s.t. $|S|\geq\alpha\cdot r$

Nearly-Linear Time Algo. for Matroid Intersection

Folklore	$\frac{1}{2}$-approx.	$O(n)$ queries
Guruganesh-Singla'17	$\left(\frac{1}{2}+10^{-4}\right)$-approx.	$O(n)$ queries

input : $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$
α-approx. : find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq \alpha \cdot r$

Nearly-Linear Time Algo. for Matroid Intersection

Folklore	$\frac{1}{2}$-approx.	$O(n)$ queries
Guruganesh-Singla '17	$(\frac{1}{2} + 10^{-4})$-approx.	$O(n)$ queries
Quanrud '24	$(1-\varepsilon)$-approx.	$\tilde{O}_\varepsilon(n + r^{3/2})$ queries
Blikstad-Tu '25	$(1-\varepsilon)$-approx.	$\tilde{O}_\varepsilon(n)$ queries

Nearly-Linear Time Algo. for Matroid Intersection

input: $(V,\mathcal{I}_1),(V,\mathcal{I}_2)$

$n:=|V|$, $r:=\max_{S\in\mathcal{I}_1\cap\mathcal{I}_2}|S|$

α-approx.: find $S\in\mathcal{I}_1\cap\mathcal{I}_2$ s.t. $|S|\geq\alpha\cdot r$

Folklore	$\frac{1}{2}$-approx.	$O(n)$ queries
Guruganesh-Singla '17	$(\frac{1}{2}+10^{-4})$-approx.	$O(n)$ queries
Quanrud '24	$(1-\varepsilon)$-approx.	$\tilde{O}_\varepsilon($[illegible]$)$ queries
Blikstad-Tu '25	$(1-\varepsilon)$-approx.	$\tilde{O}_\varepsilon(n)$ queries

All recent algo. are **randomized**!

Q. What about deterministic algo.?

Q. What about deterministic algo.?

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

α-approx. : find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq \alpha \cdot r$

Only the trivial $O(n)$-query $\frac{1}{2}$-approx. algo. is known.

Q. What about deterministic algo.?

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$
α-approx. : find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq \alpha \cdot r$

Only the trivial $O(n)$-query $\frac{1}{2}$-approx. algo. is known.

Greedy algo.

```
S ← ∅
for v ∈ V do
  if S ∪ {v} ∈ I1 ∩ I2 then
    S ← S ∪ {v}
return S
```

Q. What about deterministic algo?

input: $(V,\mathcal{I}_1),(V,\mathcal{I}_2)$
$n:=|V|$, $r:=\max_{S\in\mathcal{I}_1\cap\mathcal{I}_2}|S|$
α-approx.: find $S\in\mathcal{I}_1\cap\mathcal{I}_2$ s.t. $|S|\geq\alpha\cdot r$

Only the trivial $O(n)$-query $\frac{1}{2}$-approx. algo. is known.

Main Thm.

$\boxed{O\left(\frac{n}{\varepsilon}+r\log r\right)\text{-query } \left(\frac{2}{3}-\varepsilon\right)\text{-approx. algo.}}$

Idea: Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

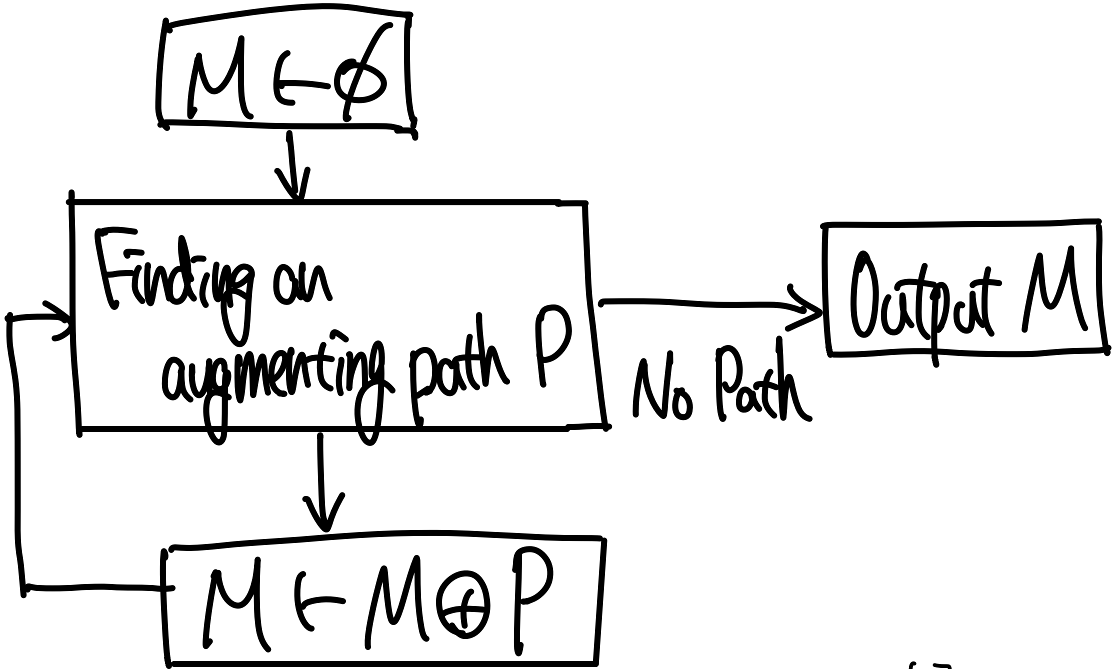
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

Algo.



[Kuhn'55]
[Edmonds'65]

Matroid Intersection

Idea: Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

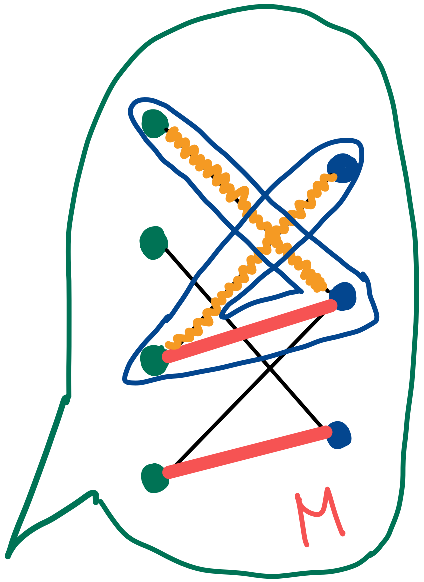
Algo.

$M \leftarrow \emptyset$

Finding an augmenting path P

No Path → Output M

$M \leftarrow M \oplus P$



[Kuhn '55]

[Edmonds '65]

Matroid Intersection

Idea : Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

Algo.

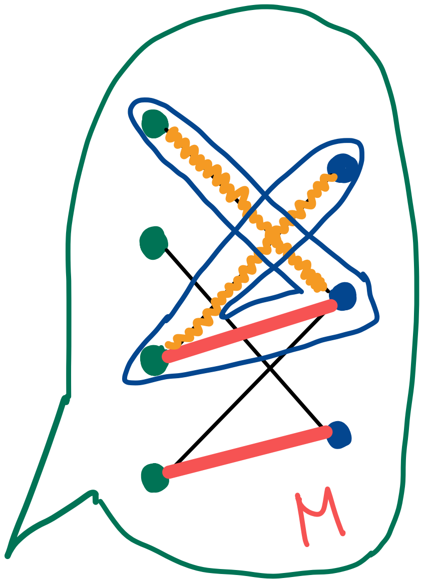
$M \leftarrow \emptyset$

Finding an augmenting path P

$M \leftarrow M \oplus P$

No Path

Output M



[Kuhn '55]
[Edmonds '65]

Matroid Intersection

Algo.

$S \leftarrow \emptyset$

Constructing the exchange graph $G(S)$

Finding a shortest augmenting path P

$S \leftarrow S \oplus P$

No Path

Output S

[Edmonds '70]

Idea : Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

Algo.

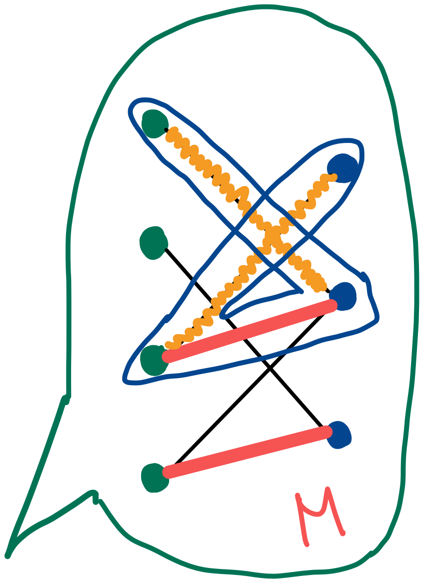
$M \leftarrow \phi$

Finding an augmenting path P

$M \leftarrow M \oplus P$

No Path

Output M



[Kuhn'55]

[Edmonds'65]

Matroid Intersection

Algo.

$S \leftarrow \phi$

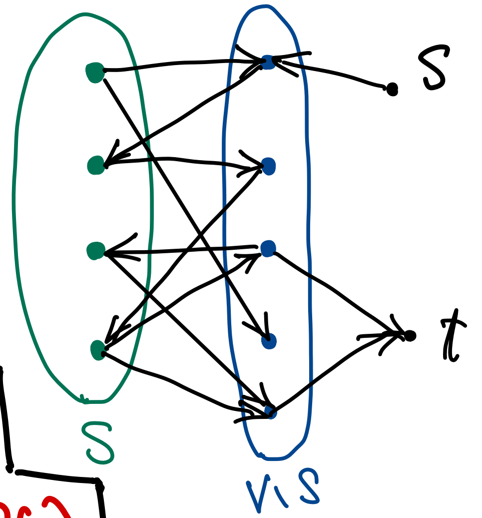
Constructing the exchange graph $G(S)$

Finding a shortest augmenting path P

$S \leftarrow S \oplus P$

No Path

Output S



[Edmonds'70]

Idea : Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

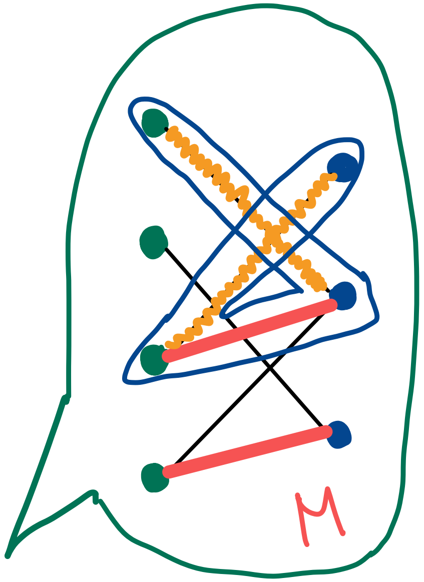
Algo.

$M \leftarrow \emptyset$

Finding an augmenting path P

No Path → Output M

$M \leftarrow M \oplus P$



[Kuhn '55]
[Edmonds '65]

Matroid Intersection

Algo.

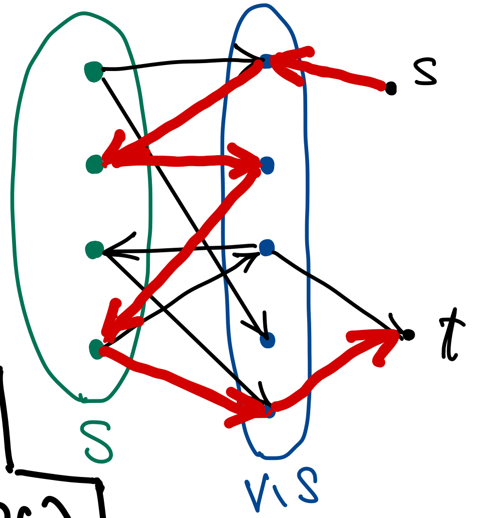
$S \leftarrow \emptyset$

Constructing the exchange graph $G(S)$

Finding a shortest augmenting path P

No Path → Output S

$S \leftarrow S \oplus P$



[Edmonds '70]

Idea : Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

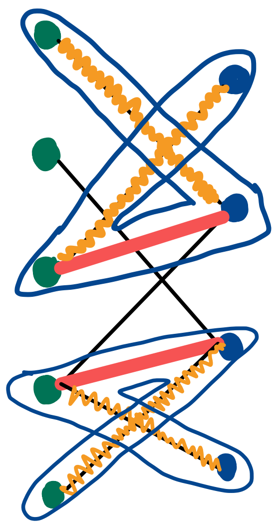
using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

Fast Algo.

★ Blocking Flow

- Find multiple shortest augmenting paths in one phase [Hopcroft - Karp '73]



Matroid Intersection

Idea : Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

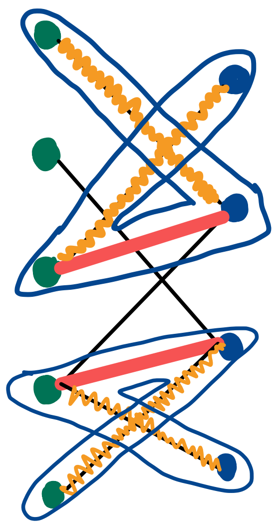
using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

Fast Algo.

★ Blocking Flow

- Find multiple shortest augmenting paths in one phase [Hopcroft - Karp '73]



- Suffices to find a set of vertex-disjoint maximal augmenting paths
- BFS + DFS

Matroid Intersection

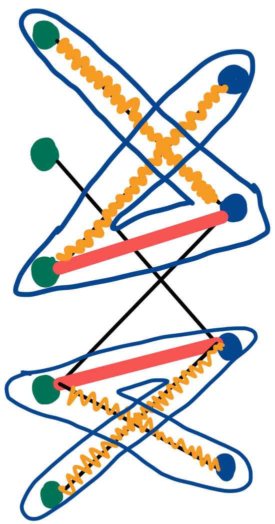
Idea : Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$
find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$
s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$
using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

Fast Algo.

* Blocking Flow
  - Find multiple shortest augmenting paths in one phase
    [Hopcroft - Karp '73]



- Suffices to find a set of vertex-disjoint maximal augmenting paths
- BFS + DFS

Matroid Intersection

Fast Algo.

* Blocking Flow
  - Find multiple shortest augmenting paths in one phase
    [Cunningham '86]

- Suffices to find a set of vertex-disjoint maximal augmenting paths ?

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

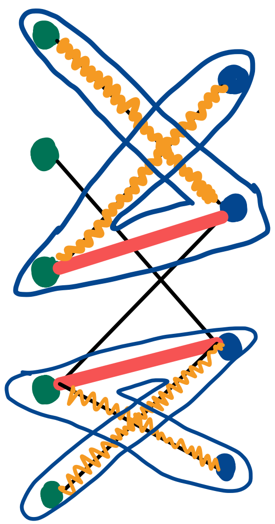
using $O(n/\varepsilon + r \log r)$ queries

Idea : Analogy from (Bipartite) Matching

(Bipartite) Matching

Fast Algo.

* Blocking Flow
  - Find multiple shortest augmenting paths in one phase
    [Hopcroft - Karp '73]



- Suffices to find a set of vertex-disjoint maximal augmenting paths
- BFS + DFS

Matroid Intersection

Fast Algo.

* Blocking Flow
  - Find multiple shortest augmenting paths in one phase
    [Cunningham '86]

- ~~Suffices to find a set of vertex-disjoint maximal augmenting paths~~
- BFS + DFS

Idea : Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

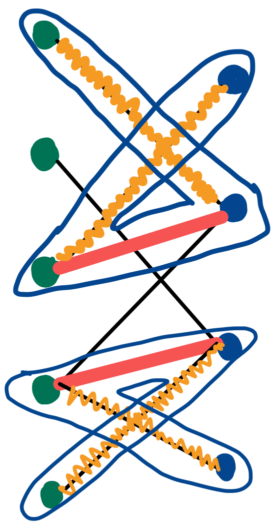
using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

Fast Algo.

★ Blocking Flow

- Find multiple shortest augmenting paths in one phase [Hopcroft-Karp '73]



- Suffices to find a set of vertex-disjoint maximal augmenting paths
- BFS + DFS

Matroid Intersection

Fast Algo.

★ Blocking Flow

- Find multiple shortest augmenting paths in one phase [Cunningham '86]

- ~~Suffices to find a set of vertex-disjoint maximal augmenting paths~~ ⇒ **augmenting sets** [CLSSW19]
- BFS + DFS

What is "Augmenting Sets"?

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

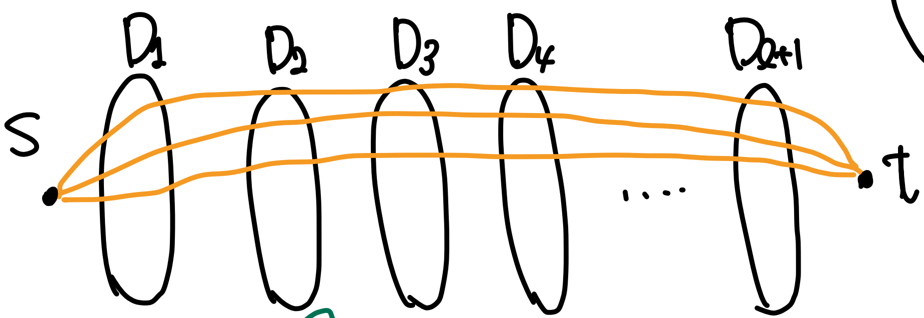
find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

using $O(n/\varepsilon + r \log r)$ queries

Matroid Intersection

- ★ Blocking Flow [Cunningham '86]
  - Find multiple shortest augmenting paths in one phase

distance layers in the exchange graph



Not vertex-disjoint paths

What is "Augmenting Sets"?

input: $(V,\mathcal{I}_1),(V,\mathcal{I}_2)$

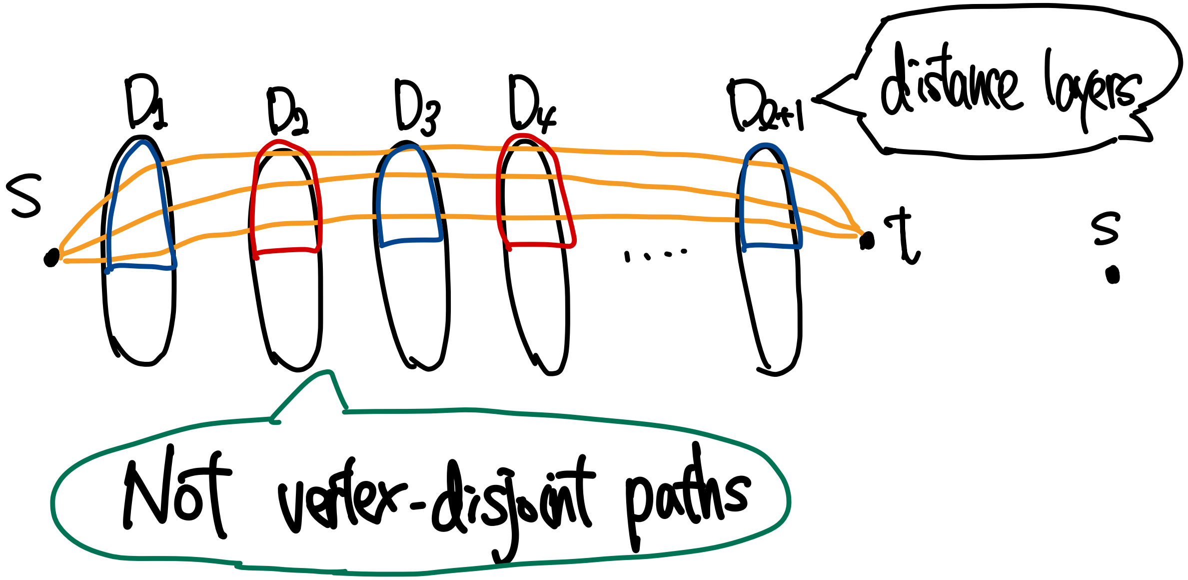
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3}-\varepsilon)\cdot r$

using $O(n/\varepsilon + r\log r)$ queries

Matroid Intersection

* Blocking Flow [Cunningham'86]
  - Find multiple shortest augmenting paths in one phase



* Augmenting Sets [CLSSW'19]

$\Pi_\ell := (B_1, A_1, B_2, \cdots, B_{\ell+1})$

① $A_k \subseteq D_{2k}$, $B_k \subseteq D_{2k-1}$

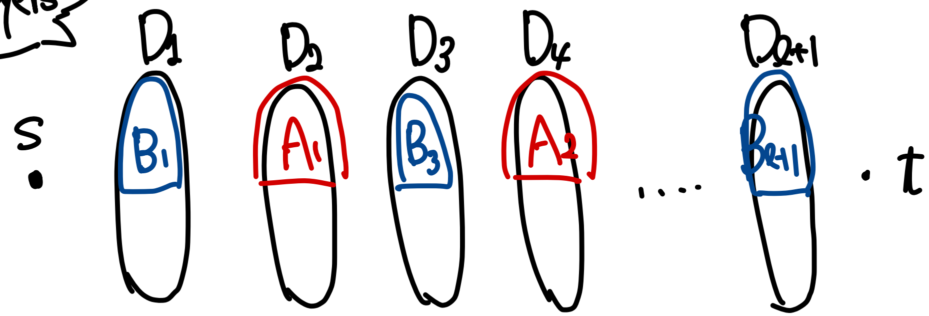
② $|B_1| = |A_1| = \cdots = |B_{k+1}|$

③ $S + B_1 \in \mathcal{I}_1$

④ $S + B_{\ell+1} \in \mathcal{I}_2$

⑤ $S - A_k + B_{k+1} \in \mathcal{I}_1$

⑥ $S - A_k + B_k \in \mathcal{I}_2$



What is "Augmenting Sets"?

input: $(V,\mathcal{I}_1),(V,\mathcal{I}_2)$

$n:=|V|$, $r:=\max_{S\in\mathcal{I}_1\cap\mathcal{I}_2}|S|$

find $S\in\mathcal{I}_1\cap\mathcal{I}_2$ s.t. $|S|\geq(\frac{2}{3}-\varepsilon)\cdot r$

using $O(n/\varepsilon+r\log r)$ queries

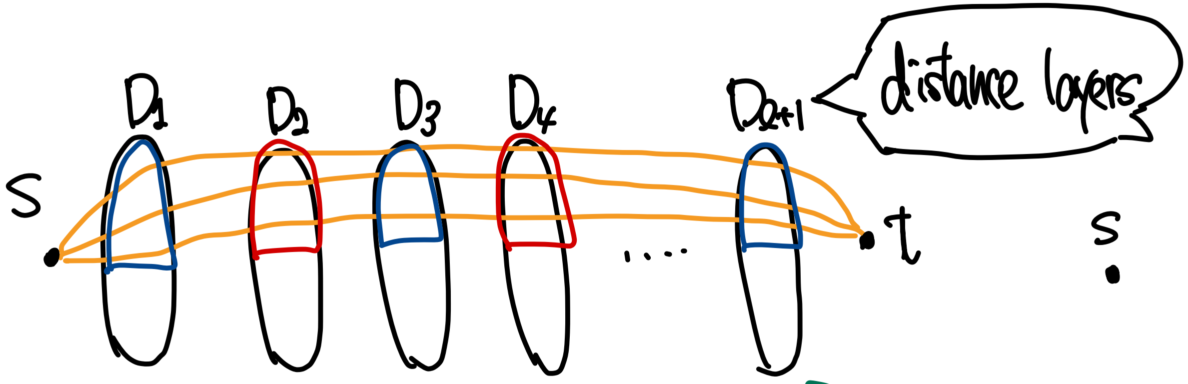
Matroid Intersection

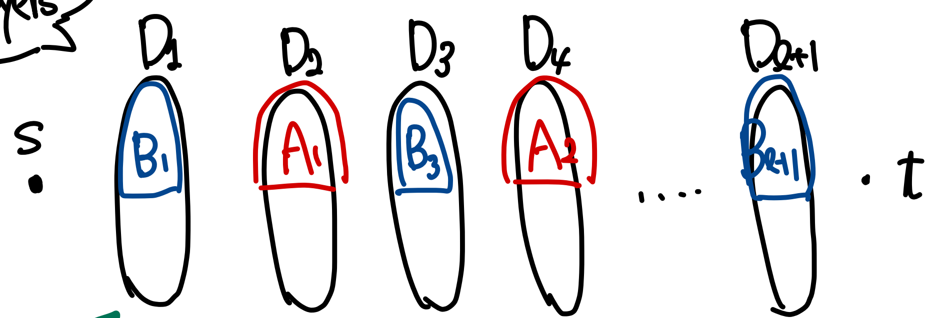
* Blocking Flow [Cunningham'86]
  - Find multiple shortest augmenting paths in one phase

* Augmenting Sets [CLSSW'19]

$\Pi_\ell:=(B_1,A_1,B_2,\cdots,B_{\ell+1})$

① $A_\ell\subseteq D_{2\ell}$, $B_\ell\subseteq D_{2\ell-1}$
② $|B_1|=|A_1|=\cdots=|B_{\ell+1}|$
③ $S+B_1\in\mathcal{I}_1$
④ $S+B_{\ell+1}\in\mathcal{I}_2$
⑤ $S-A_\ell+B_{\ell+1}\in\mathcal{I}_1$
⑥ $S-A_\ell+B_\ell\in\mathcal{I}_2$





Each set corresponds to another set!

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

using $O(n/\varepsilon + r \log r)$ queries

Idea: Analogy from (Bipartite) Matching

(Bipartite) Matching

There is no augmenting path of length 3 in M

$\Rightarrow$ M is a $2/3$-approx.

Matroid Intersection

Idea: Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

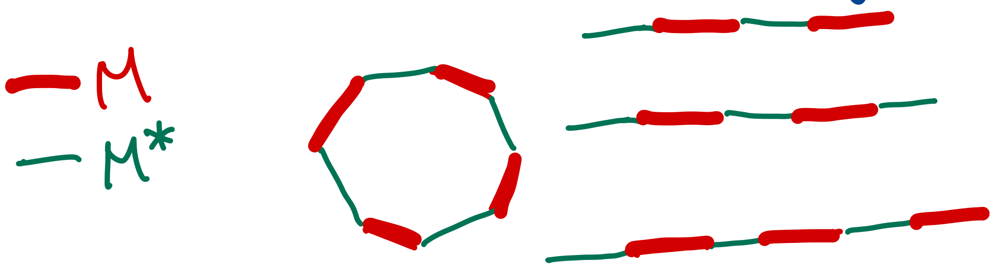
using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

There is no augmenting path of length 3 in M

$\Rightarrow$ M is a $2/3$-approx.

Let M^* be a maximum matching.



$M \triangle M^*$ can be decomposed into alternating paths of length ≥ 4 and alternating cycles

Matroid Intersection

input: $(V,\mathcal{I}_1),(V,\mathcal{I}_2)$

$n:=|V|$, $r:=\max_{S\in\mathcal{I}_1\cap\mathcal{I}_2}|S|$

find $S\in\mathcal{I}_1\cap\mathcal{I}_2$ s.t. $|S|\geq(\frac{2}{3}-\varepsilon)\cdot r$

using $O(n/\varepsilon+r\log r)$ queries

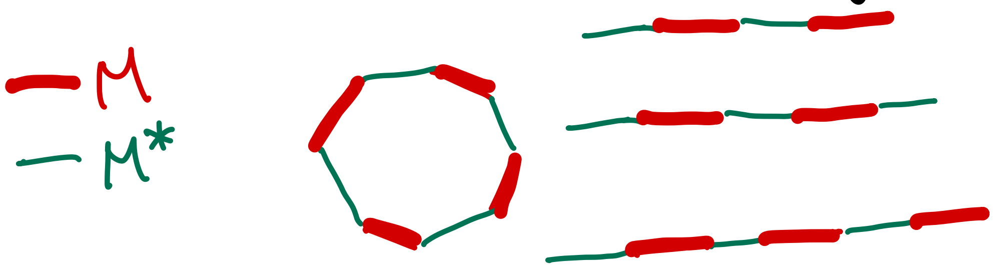
Idea : Analogy from (Bipartite) Matching

(Bipartite) Matching

There is no augmenting path of length 3 in M

$\Rightarrow$ M is a $2/3$-approx.

Let M^* be a maximum matching.



$M\triangle M^*$ can be decomposed into alternating paths of length ≥ 4 and alternating cycles

Matroid Intersection

There is no augmenting path of length 4 in $G(S)$

$\Rightarrow$ S is $2/3$-approx.

Idea: Analogy from (Bipartite) Matching

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

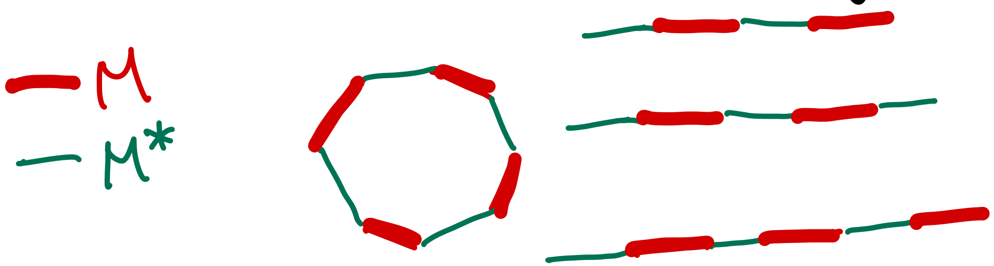
using $O(n/\varepsilon + r \log r)$ queries

(Bipartite) Matching

There is no augmenting path of length 3 in M

$\Rightarrow$ M is a $2/3$-approx.

☺ Let M^* be a maximum matching.



$M \Delta M^*$ can be decomposed into alternating paths of length ≥ 4 and alternating cycles

Matroid Intersection

There is no augmenting path of length 4 in $G(S)$

$\Rightarrow$ S is $2/3$-approx.

Key Observation

Terminate the algorithm of [Blikstad] for finding augmenting sets of length 4 early, after only $O(1/\varepsilon)$ phases

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$

using $O(n/\varepsilon + r \log r)$ queries

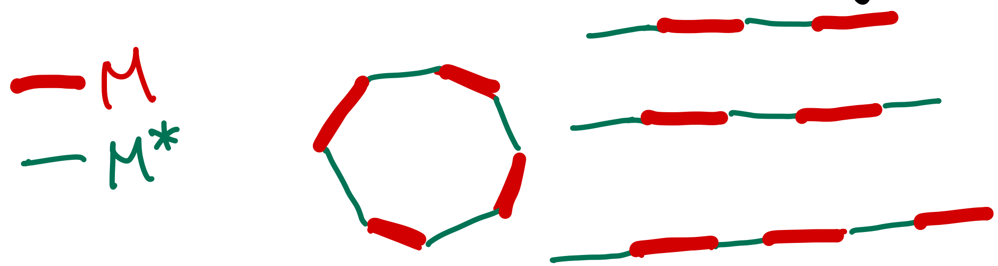
Idea : Analogy from (Bipartite) Matching

(Bipartite) Matching

There is no augmenting path of length 3 in M

$\Rightarrow$ M is a 2/3-approx.

☺ Let M^* be a maximum matching.



$M \triangle M^*$ can be decomposed into alternating paths of length ≥ 4 and alternating cycles

Matroid Intersection

There is no augmenting path of length 4 in $G(S)$

$\Rightarrow$ S is 2/3-approx.

Key Observation

Terminate the algorithm of [Blikstad] for finding augmenting sets of length 4 early, after only $O(1/\varepsilon)$ phases

➡ We can find almost all augmenting paths of length 4

input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$
$n := \lvert V \rvert$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} \lvert S \rvert$
find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $\lvert S \rvert \geq (\frac{2}{3} - \varepsilon) \cdot r$
using $O(n/\varepsilon + r \log r)$ queries

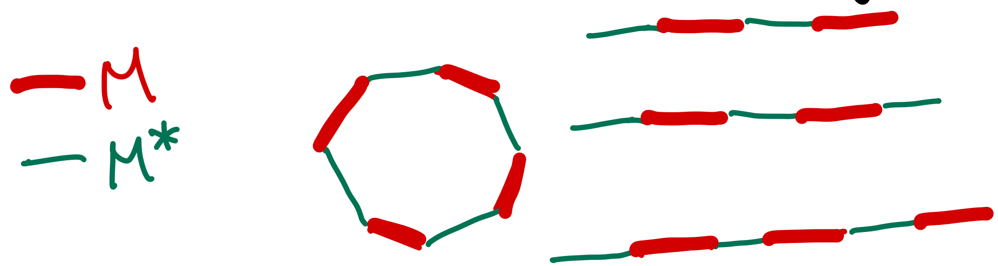
Idea : Analogy from (Bipartite) Matching

(Bipartite) Matching

There is no augmenting path of length 3 in M

$\Rightarrow$ M is a $2/3$-approx.

:(Let M^* be a maximum matching.



$M \triangle M^*$ can be decomposed into alternating paths of length ≥ 4 and alternating cycles

Matroid Intersection

There is no augmenting path of length 4 in $G(S)$

$\Rightarrow$ S is $2/3$-approx.

Key Observation

Terminate the algorithm of [Blikstad] for finding augmenting sets of length 4 early, after only $O(1/\varepsilon)$ phases

➡ We can find almost all augmenting paths of length 4

➡ $(2/3 - \varepsilon)$-approx.

Unexpectedly Interesting Aspect of Our Result

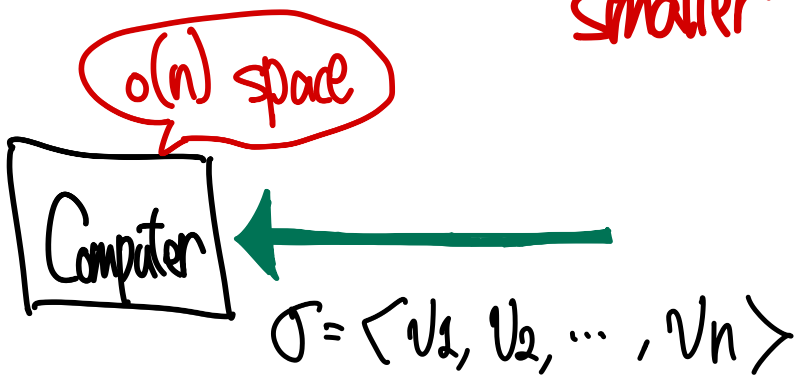
Our algorithm can be extended to

a Streaming Algo. !

Unexpectedly Interesting Aspect of Our Result

Our algorithm can be extended to a Streaming Algo.!

Solving the problem with space complexity smaller than the input size

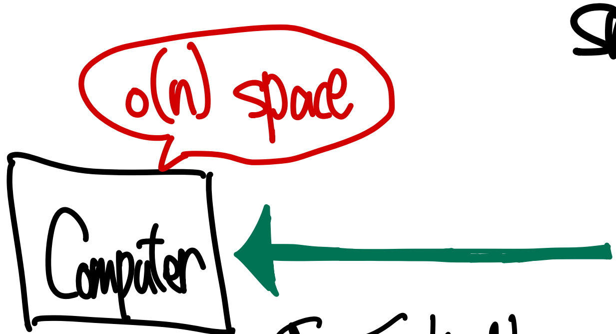


Unexpectedly Interesting Aspect of Our Result

Our algorithm can be extended to

a Streaming Algo.!

Solving the problem with space complexity smaller than the input size



$\sigma = \langle v_1, v_2, \cdots, v_n \rangle$

# of passes = How many times the algo. scans the entire input stream

Unexpectedly Interesting Aspect of Our Result

Our algorithm can be extended to a Streaming Algo. !

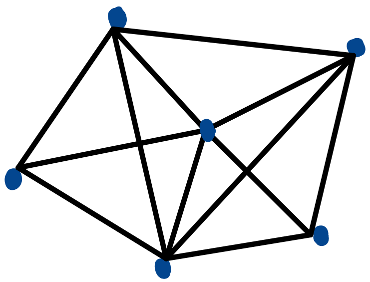
Solving the problem with space complexity smaller than the input size

o(n) space

Computer

$\sigma = \langle v_1, v_2, \cdots, v_n \rangle$

\# of passes = How many times the algo. scans the entire input stream



★ In particular, for graph problems, Semi-Streaming Algo. using O(# of vertices) space are well studied!

Streaming Algo. for Matroid Intersection

input : $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$\max |S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

$r_i := \max_{S \in \mathcal{I}_i} |S|$ $(i = 1, 2)$

Streaming Algo. for Matroid Intersection

input : $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$\max |S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

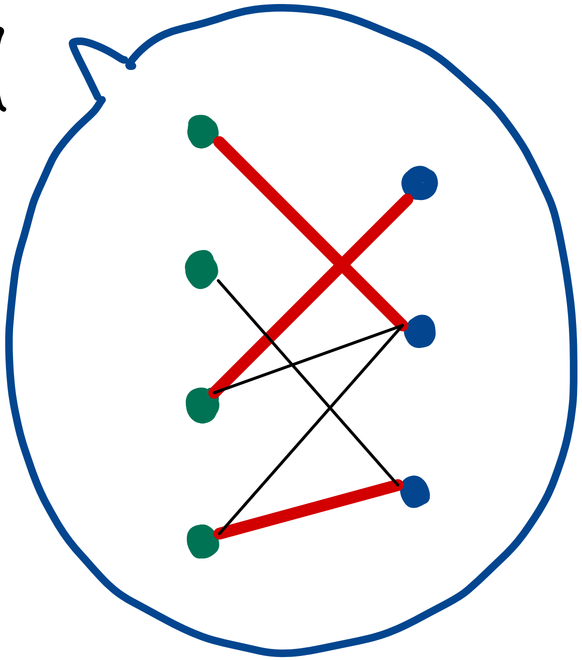
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

$r_i := \max_{S \in \mathcal{I}_i} |S| \ (i = 1, 2)$

Semi-streaming algo. for matching

Paz-Schwartzman '17

Bernstein '20



Streaming Algo. for Matroid Intersection

input : $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$
max $|S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$
$r_i := \max_{S \in \mathcal{I}_i} |S|$ $(i = 1, 2)$

Semi-streaming algo. for matching

Generalization

Paz-Schwartzman'17 $\Rightarrow$ Garg-Jordan-Svensson'21

1 pass, weighted, $(1/2 - \varepsilon)$-approx. $\tilde{O}_\varepsilon(r_1 + r_2)$ space

Bernstein'20 $\Rightarrow$ Huang-Sellier'24

1 pass, random-order, $(2/3 - \varepsilon)$-approx. $\tilde{O}_\varepsilon(r)$ space

Streaming Algo. for Matroid Intersection

input : $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$\max |S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

$r_i := \max_{S \in \mathcal{I}_i} |S|$ $(i = 1, 2)$

Semi-streaming algo. for matching

Generalization

Paz-Schwartzman'17 ➡ Garg-Jordan-Svensson'21

1 pass, weighted, $(1/2 - \varepsilon)$-approx. $\tilde{O}_\varepsilon(r_1 + r_2)$ space

Bernstein'20 ➡ Huang-Sellier'24

1 pass, random-order, $(2/3 - \varepsilon)$-approx. $\tilde{O}_\varepsilon(r)$ space

This work

$O(1/\varepsilon)$ passes, $(2/3 - \varepsilon)$-approx., $\tilde{O}(r_1 + r_2)$ space

Streaming Algo. for Matroid Intersection

input : $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$

$\max |S|$ s.t. $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

$r_i := \max_{S \in \mathcal{I}_i} |S| \ (i = 1, 2)$

Semi-streaming algo. for matching

Generalization

Paz-Schwartzman'17 → Garg-Jordan-Svensson'21

1 pass, weighted, $(1/2 - \varepsilon)$-approx. $\tilde{O}_\varepsilon(r_1 + r_2)$ space

Bernstein'20 → Huang-Sellier'24

1 pass, random-order, $(2/3 - \varepsilon)$-approx. $\tilde{O}_\varepsilon(r)$ space

Feigenbaum-Kannan-McGregor-Suri-Zhang'04 → **This work**

$O(1/\varepsilon)$ passes, $(2/3 - \varepsilon)$-approx., $\tilde{O}(r_1 + r_2)$ space

First streaming graph algo. paper

Conclusion

- Design
  a Deterministic $(2/3-\varepsilon)$-approx. $\tilde{O}_\varepsilon(n)$ queries Algo.
  for Matroid Intersection
- Extending this to
  a $O(1/\varepsilon)$ passes $(2/3-\varepsilon)$-approx. Semi-Streaming Algo.
  (Generalization of FKMZ'04)

Conclusion

- Design
  a Deterministic $(2/3-\varepsilon)$-approx. $\tilde{O}_\varepsilon(n)$ queries Algo.
  for Matroid Intersection
- Extending this to
  a $O(1/\varepsilon)$ passes $(2/3-\varepsilon)$-approx. Semi-Streaming Algo.
  (Generalization of FKMZ'04)

open questions

- Deterministic $(1-\varepsilon)$-approx. $\tilde{O}_\varepsilon(n)$ queries Algo.
- poly$(1/\varepsilon)$ passes $(1-\varepsilon)$-approx. Semi-Streaming Algo.