Faster Matroid Partition Algorithms

Tatsuya Terao

Kyoto University

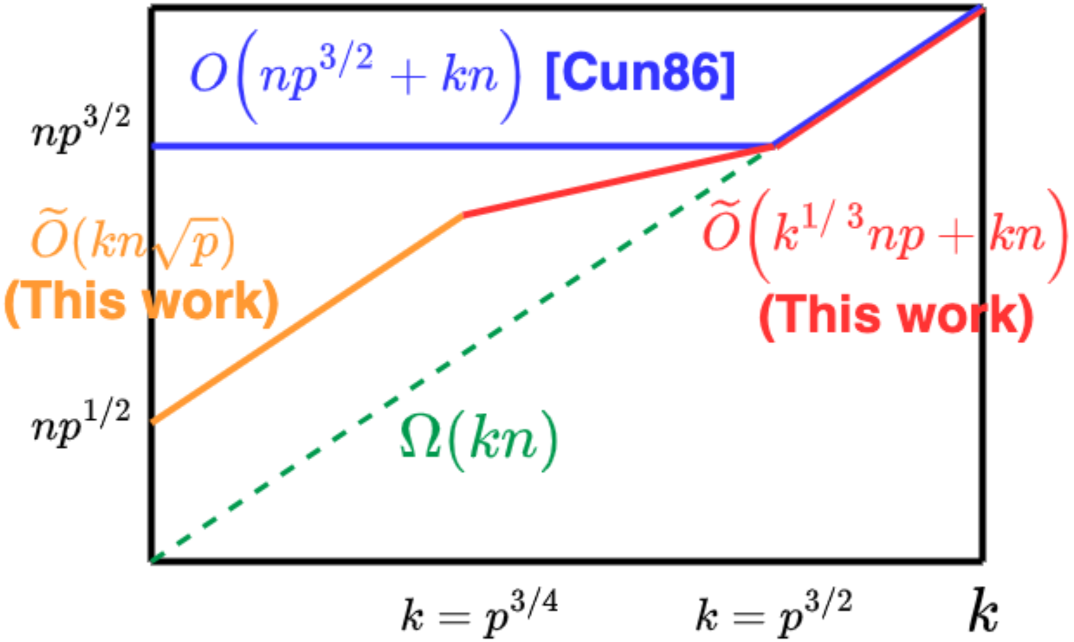
ICALP 2023 @Paderborn July 14, 2023

Summary

Result

Three fast algorithms for **matroid partition**

- Algorithm 1.
  $\tilde{O}(kn\sqrt{p})$ **independence** queries
- Algorithm 2.
  $\tilde{O}(k^{1/3}np + kn)$ **independence** queries
- Algorithm 3.
  $\tilde{O}((n+k)\sqrt{p})$ **rank** queries



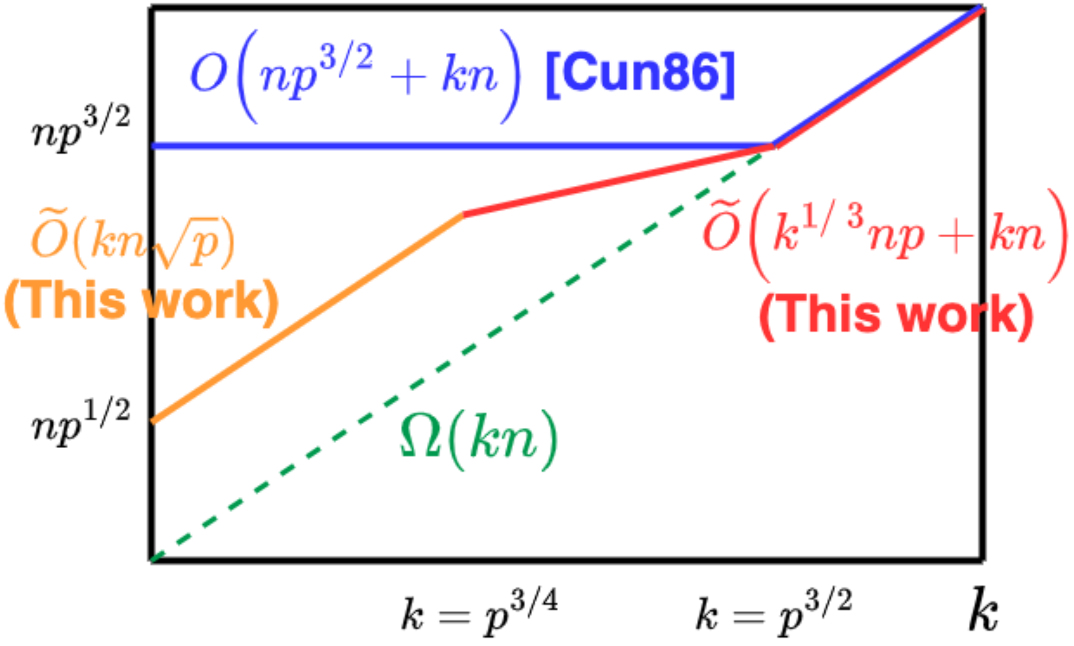
n = #elements, k = #matroids
p = solution size

Summary

Result

Three fast algorithms for **matroid partition**

- Algorithm 1.
  $\tilde{O}(kn\sqrt{p})$ **independence** queries
- Algorithm 2.
  $\tilde{O}(k^{1/3}np + kn)$ **independence** queries
- Algorithm 3.
  $\tilde{O}((n+k)\sqrt{p})$ **rank** queries



A new approach

Edge Recycling Augmentation

Outline

- **Summary**
- **Preliminaries**
  - -Matroid
  - -Matroid Intersection
  - -Matroid Partition
- **Result**
  - -Faster Matroid Partition Algorithms
- **Idea**
  - -Blocking Flow
  - -Edge Recycling Augmentation
- **Conclusion**

Matroid $\mathcal{M} = (V, \mathcal{I})$

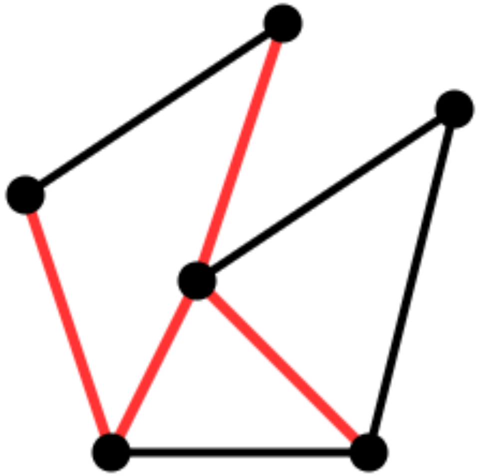
Def

A finite set V and non-empty family of **independent** sets $\mathcal{I} \subseteq 2^V$ such that

- $S' \subseteq S \in \mathcal{I} \implies S' \in \mathcal{I}$
- $S, T \in \mathcal{I}$, $|S| > |T| \implies \exists\, e \in S - T$ s.t. $T \cup \{e\} \in \mathcal{I}$

Eg.

- **Graphic Matroid**



V = edges
$\mathcal{I}$ = forests

- **Linear Matroid**

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

V = row vectors
$\mathcal{I}$ = linearly independent

Matroid $\mathcal{M} = (V, \mathcal{I})$

Def

A finite set V and non-empty family of **independent** sets $\mathcal{I} \subseteq 2^V$ such that

- $S' \subseteq S \in \mathcal{I} \implies S' \in \mathcal{I}$
- $S, T \in \mathcal{I}, |S| > |T| \implies \exists\, e \in S - T$ s.t. $\mathrm{T} \cup \{e\} \in \mathcal{I}$

Algorithm accesses a matroid through an **oracle**

Matroid $\mathcal{M} = (V, \mathcal{I})$

Def

A finite set V and non-empty family of **independent** sets $\mathcal{I} \subseteq 2^V$ such that

- $S' \subseteq S \in \mathcal{I} \implies S' \in \mathcal{I}$
- $S, T \in \mathcal{I}$, $|S| > |T| \implies \exists\, e \in S - T$ s.t. $\mathrm{T} \cup \{e\} \in \mathcal{I}$

Algorithm accesses a matroid through an **oracle**

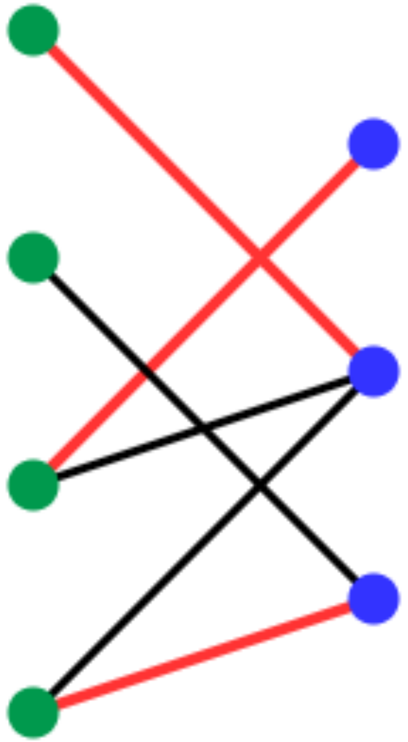
- **Independence** oracle query: Is $S \in \mathcal{I}$?

Matroid Intersection

Input : two matroids $\mathcal{M}_1 = (V, \mathcal{I}_1), \mathcal{M}_2 = (V, \mathcal{I}_2)$
Find : maximum **common independent set** $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

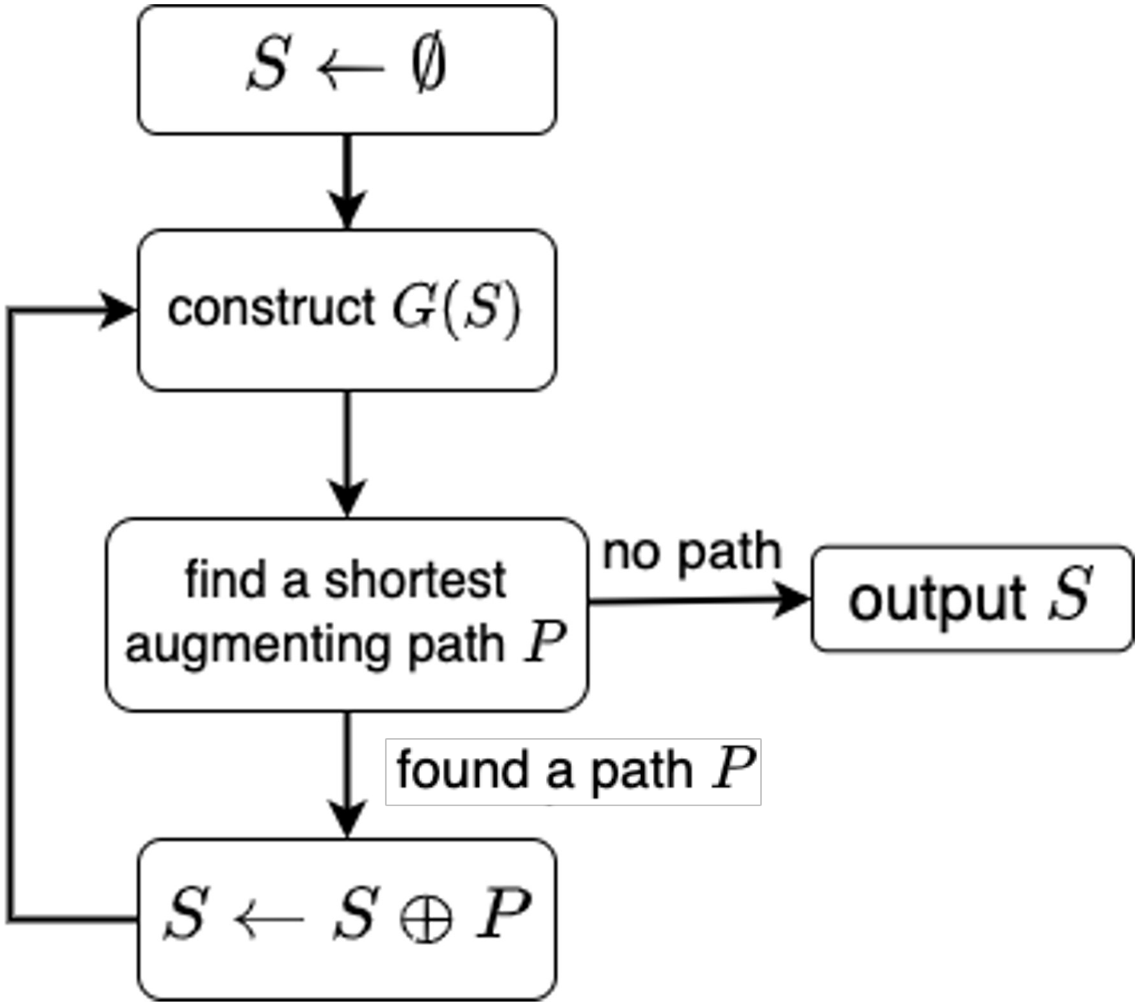
E.g. Bipartite Matching

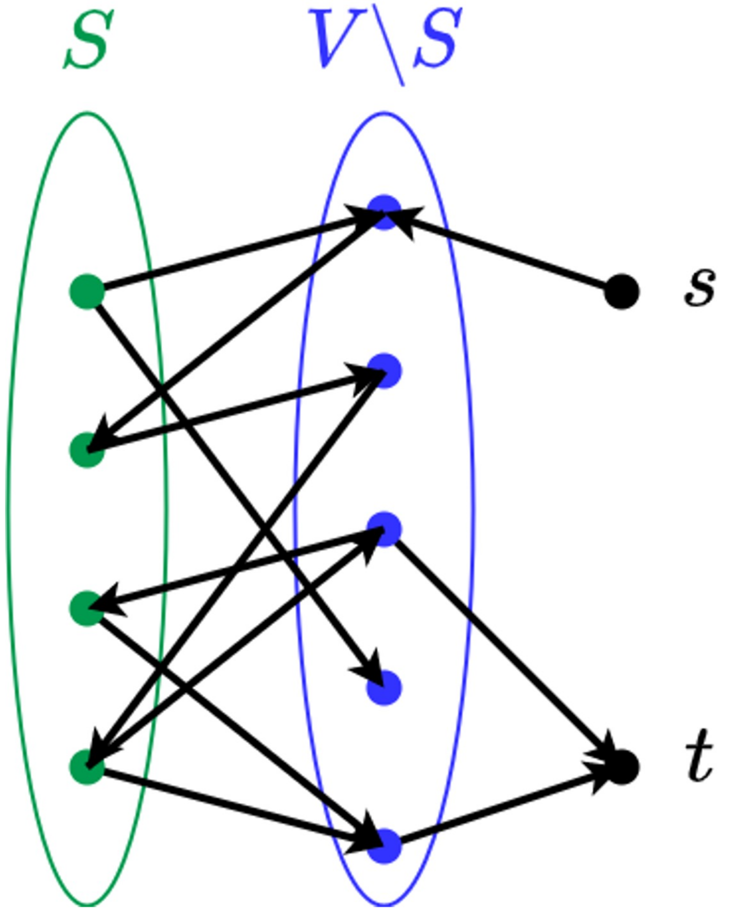
$\boldsymbol{V} =$ **edges**
$\boldsymbol{\mathcal{I}_1} =$ **each left vertex has at most 1 edge**
$\boldsymbol{\mathcal{I}_2} =$ **each right vertex has at most 1 edge**



Edmonds' Algorithm for Matroid Intersection

[Edmonds 1970, Aigner-Dowling 1971, Lawler 1975]

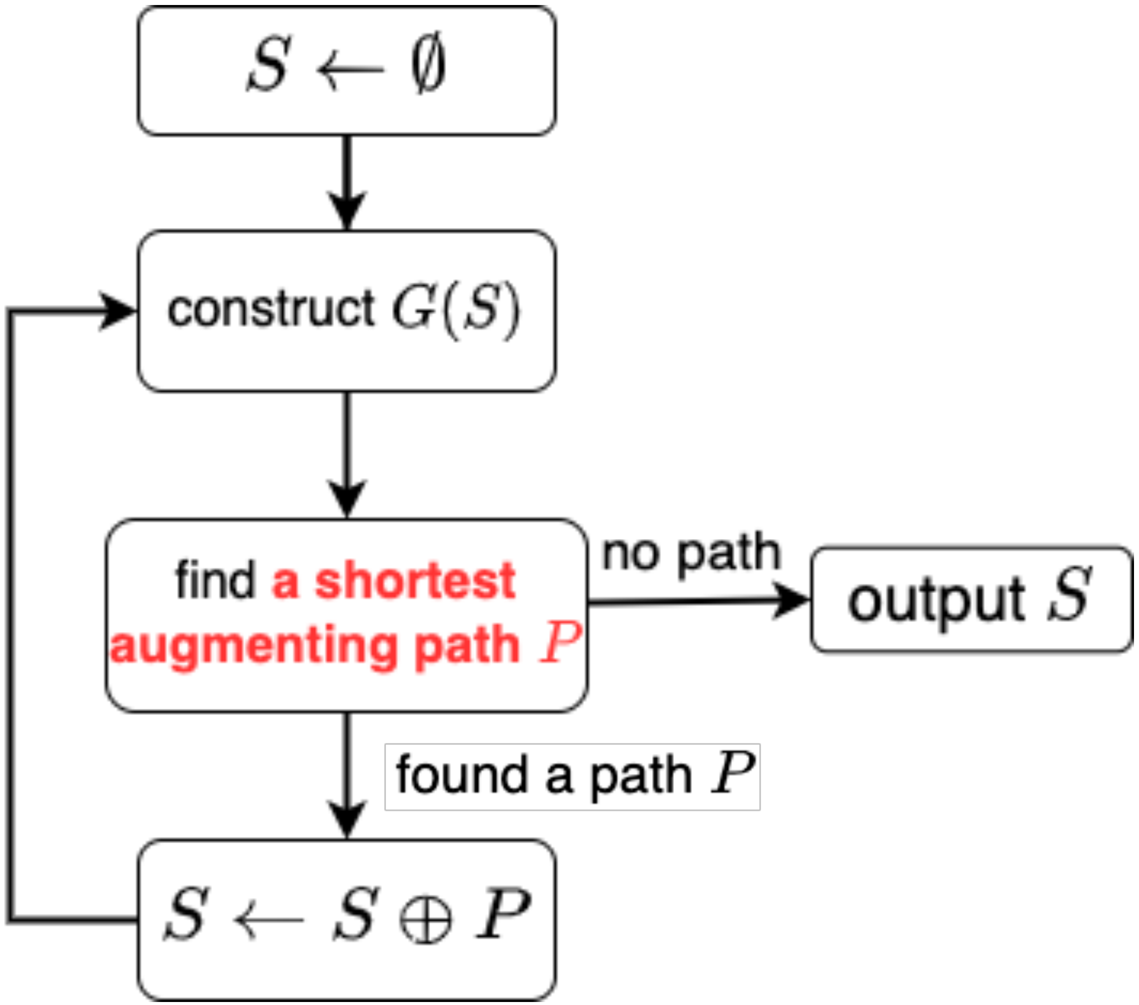


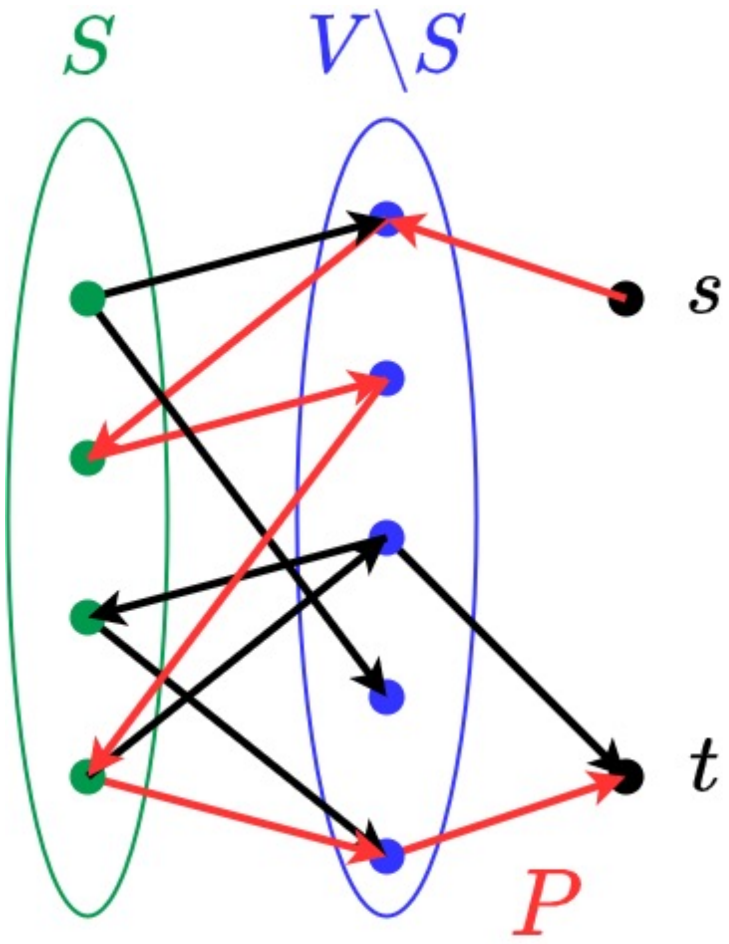


Exchange graph $G(S)$

Algorithm for Matroid Intersection

[Edmonds 1970, Aigner-Dowling 1971, Lawler 1975]





Exchange graph $G(S)$

Prior Work on Matroid Intersection

Independence query complexity

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	$\tilde{O}(nr^{3/4})$

$n = |V|$, $r =$ solution size