

準線形回の独立性オラクルへのツエイを  
使用するマトロイド交叉問題の  
決定的な $(\frac{2}{3} - \epsilon)$ 近似アルゴリズム

寺尾 樹哉 (京都大学数理解析研究所 D2)

最適化の理論とアルゴリズム：未来を担う研究者の集い'2025

① 筑波大学 5月31日

arXiv: 2410.18820 (To appear in WADS'25)

# マトロイト"とは？

Def.

有限集合 $V$ とその上でない部分集合族 $I \subseteq 2^V$  s.t.

$$\textcircled{1} \quad S' \subseteq S \in I \Rightarrow S' \in I$$

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

# マトロイド"とは? - 線形独立性の一般化

Def.

有限集合  $V$  とその上でない部分集合族  $\mathcal{I} \subseteq 2^V$  s.t.

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

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

e.g. • 線形マトロイド

$$\left( \begin{array}{c|cc} 1 & 1 & 2 \\ \hline 2 & 1 & 2 \end{array} \right)$$

$$V = \{ \textcircled{1}, \textcircled{2}, \textcircled{3} \}$$

$$\mathcal{I} = \{ \emptyset, \{ \textcircled{1} \}, \{ \textcircled{2} \}, \{ \textcircled{3} \}, \{ \textcircled{1}, \textcircled{2} \}, \{ \textcircled{1}, \textcircled{3} \} \}$$

# マトロイド"とは？ - 線形独立性の一般化

Def.

有限集合  $V$  とその上でない部分集合族  $\mathcal{I} \subseteq 2^V$  s.t.

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

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

e.g. • 線形マトロイド

$S \in \mathcal{I}$  :  $S$  は独立

1	1	2
2	1	2

$$V = \{ \textcircled{1}, \textcircled{2}, \textcircled{3} \}$$

$$\mathcal{I} = \{ \emptyset, \{ \textcircled{1} \}, \{ \textcircled{2} \}, \{ \textcircled{3} \}, \{ \textcircled{1}, \textcircled{2} \}, \{ \textcircled{1}, \textcircled{3} \} \}$$

# マトロイド"とは？ - 線形独立性の一般化

Def.

有限集合  $V$  とその上でない部分集合族  $\mathcal{I} \subseteq 2^V$  s.t.

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

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

e.g. • 線形マトロイド

1	1	2
2	1	2

$$V = \{ \textcircled{1}, \textcircled{2}, \textcircled{3} \}$$

$$\mathcal{I} = \{ \emptyset, \{ \textcircled{1} \}, \{ \textcircled{2} \}, \{ \textcircled{3} \}, \{ \textcircled{1}, \textcircled{2} \}, \{ \textcircled{1}, \textcircled{3} \} \}$$

- (n)割マトロイド
  - ラミー-マトロイド
  - グラフ的マトロイド
- etc...

# 2トロイド交叉問題とは？

入力: 2つの2トロイド  $M_1 = (V, I_1)$ ,  $M_2 = (V, I_2)$

出力: 最大サイズの赤字独立集合  $S \in I_1 \cup I_2$

# クロイド交叉問題とは？

— 二部グラフの最大スナップ問題の一般化

入力：2つのクロイド  $M_1 = (V, I_1)$ ,  $M_2 = (V, I_2)$

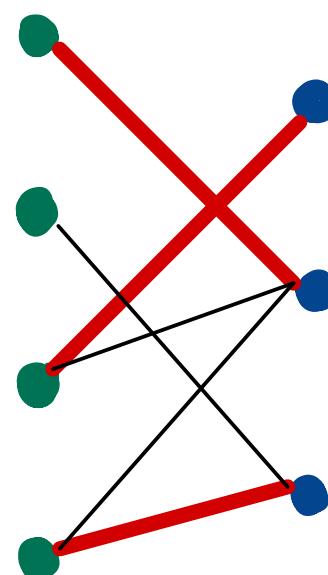
出力：最大サイズの共通独立集合  $S \in I_1 \cap I_2$

例：二部グラフの最大スナップ

$V$  = 頂点集合

$I_1$  = 左側の各頂点に接する辺は高々1本

$I_2$  = 右側の各頂点に接する辺は高々1本



# 2トロイド"交叉問題"とは?

—二部グラフの最大マッチング問題の一般化

入力: 2つの2トロイド  $M_1 = (V, I_1)$ ,  $M_2 = (V, I_2)$

出力: 最大サイズの半道独立集合  $S \in I_1 \cup I_2$

## なぜ、本問題が重要な?

# マトリイド交叉問題とは？

—二部グラフの最大マッチング問題の一般化

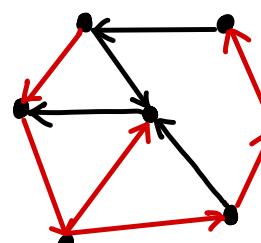
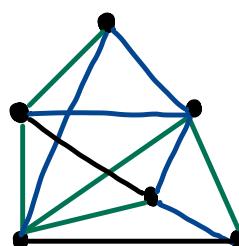
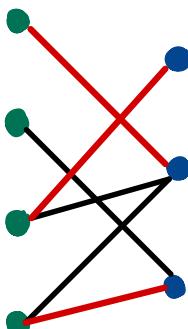
入力：2つのマトリイド  $M_1 = (V, I_1)$ ,  $M_2 = (V, I_2)$

出力：最大サイズの共通独立集合  $S \in I_1 \cap I_2$

## なぜ、本問題は重要な？

①多くの問題の応用

二部グラフのマッチング、全城木の構成、有向全城木



# クロイド交叉問題とは？

—二部グラフの最大マッチング問題の一般化

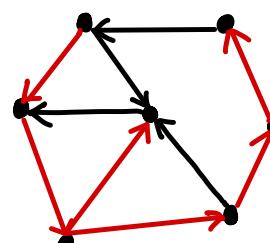
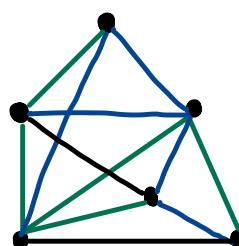
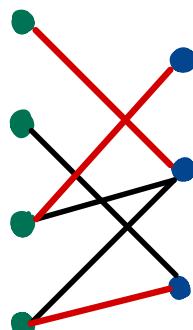
入力：2つのクロイド  $M_1 = (V, I_1)$ ,  $M_2 = (V, I_2)$

出力：最大サイズの両邊独立集合  $S \in I_1 \cup I_2$

## なぜ、本問題は重要な？

①多くの問題の基盤

二部グラフのマッチング、全城木の諸問題、有向全城木



②組合せ最適化への応用

- 電気回路解析
- ネットワーカ化

$\lambda(t) : (U, I_1), (V, I_2)$

$\max |S| \text{ s.t. } S \in I_1 \cap I_2$

クロイド交叉問題は

多项式時間で解ける

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

$\lambda(t) : (U, I_1), (V, I_2)$

$$\max |S| \text{ s.t. } S \in I_1 \cap I_2$$

クロイド交叉問題は

多項式時間で解ける

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

クロイドは独立性オブジェクトをもつ

リエリ(質問)

回答

$S \in I \leftrightarrow ?$

独立性  
オブジェクト

Yes or No

マトリイド交叉問題は

多項式回数の問題で解ける

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

マトリイドは独立性オラクルで与えられる

問題(質問)

回答

$S \in I$  か?

独立性  
オラクル

Yes or No

計算時間 = オラクルへの問題回数

出入り:  $(U, I_1), (V, I_2)$   
 $\max |S|$  s.t.  $S \subseteq I_1 \cap I_2$

# マトリクス交叉問題の時間計算量

$\lambda(t) : (V, I_1), (V, I_2)$   
 $\max |S| \text{ s.t. } S \in I_1 \cap I_2$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$

独立性オフсет  
への繰り回数

# マトリクス交叉問題 の 時間計算量

1970s

Edmonds, Lawler, Aigner-Dantzig

1986

Cunningham

独立性オウル  
への ウエイ回数

$O(nr^2)$

$O(nr^{3/2})$

$$\lambda I : (V, I_1), (V, I_2)$$

$$\max |S| \text{ s.t. } S \in I_1 \cap I_2$$

$$n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$$

# マトリクス交叉問題 の 時間計算量

1970s

Edmonds, Lawler, Aigner-Dantzig

$O(nr^2)$

1986

Cunningham

$O(nr^{3/2})$

2015

Lee - Sidford - Wong

$\tilde{O}(n^2)$

2019

Nguyen, Chakrabarty-Lee-Sidford-Singla-Wong

$\tilde{O}(nr)$

2021

Blikstad - v.d. Brand - Mukhopadhyay - Namngkai

$\tilde{O}(n^{9/5})$

2021

Blikstad ✘

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

✗: [Blikstad21] は乱数を許せば  $\tilde{O}(nr^{3/4})$ , 実験的の場合で  $\tilde{O}(nr^{5/6})$

$$\lambda I := (V, I_1), (V, I_2) \\ \max |S| \text{ s.t. } S \in I_1 \cap I_2 \\ n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$$

独立性オフсет  
への ウエイト回数

# 今日のお話：準線形時間 アルゴリズム

入力のサイズ  $n$  の ほぼ 線形時間  $O(n \cdot \text{polylog}(n))$   
の アルゴリズム

# 今日のお話：準線形時間 アルゴリズム

入力のサイズ  $n$  のほぼ線形時間  $O(n \cdot \text{polylog}(n))$  のアルゴリズム

\* 厳密解ではなく、近似解を目指す

$\alpha$  近似解  $S \in I_1 \cup I_2$

s.t.  $|S| \geq \alpha \cdot \text{OPT}$

# 今日のお話：準線形時間 アルゴリズム

入力のサイズ  $n$  のほぼ線形時間  $O(n \cdot \text{polylog}(n))$  のアルゴリズム

\* 厳密解ではなく、近似解を目指す

$\alpha$  近似解  $S \in I_1 \cup I_2$

$$\text{s.t. } |S| \geq \alpha \cdot \text{OPT}$$

e.g.

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

$k$ -cut : Quanrud'19

$k$ -ECSS : Chalermsook-Huang-Nansgkai-Saranak-Sukprasert-Yingcharoenwanichai'22

Weighted matching : Preis'99, Vinkemeier-Haugard'03, Duan-Pettie'14, Zhang-Henzinger'23

# マトリオド交叉問題 の

準線形時間 アルゴリズム

Folklore

$\frac{1}{2}$  近似

$O(n \log')$

$\lambda I : (V, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 $\alpha\text{-approx.} : \text{find } S \in I_1 \cup I_2$   
s.t.  $|S| \geq \alpha \cdot r$

# マトリオド交叉問題 の

準線形時間 アルゴリズム

Folklore

Guruganesh-Singla'17

$\frac{1}{2}$  近似

$\frac{1}{2} + 10^{-4}$  近似

$O(n \log')$

$O(n \log)$

$\lambda \in (V, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 $\alpha\text{-approx.} : \text{find } S \in I_1 \cup I_2$   
s.t.  $|S| \geq \alpha \cdot r$

# マトリオド交叉問題 の

## 準線形時間 アルゴリズム

$\lambda \in \{U, I_1, I_2\}$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 $\alpha\text{-approx.} : \text{find } S \in I_1 \cup I_2 \text{ s.t. } |S| \geq \alpha \cdot r$

Folklore

Guruganesh-Singla '17

Quanrud '24

Blikstad-Tu '25 \*

$\frac{1}{2}$  近似

$\frac{1}{2} + \tilde{O}^4$  近似

$1 - \varepsilon$  近似

$1 - \varepsilon$  近似

$O(n) \cap I'$

$O(n) \cap S$

$\tilde{O}_\varepsilon(n + r^{3/2}) \cap I'$

$\tilde{O}_\varepsilon(n) \cap I'$

\* arXivに投稿したのが本研究とほぼ同時

# マトリオド交叉問題 の

準線形時間

アルゴリズム

$\lambda \in (V, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 $\alpha\text{-approx.} : \text{find } S \in I_1 \cup I_2 \text{ s.t. } |S| \geq \alpha \cdot r$

Folklore

$\frac{1}{2}$  近似

$O(n) \text{ ジエ'}$

Guruganesh-Singh '17

$\frac{1}{2} + 10^{-4}$  近似

$O(n) \text{ ジエ'}$

最近の研究は全て乱択アルゴリズム！

Blikstad-Tu '25

$1 - \varepsilon$  近似

$\tilde{O}_\varepsilon(n) \text{ ジエ'}$

Q. 決定的には？

乱数  を使わないアルゴリズム

Q. 決定的には？

$\lambda \in (V, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 $\alpha\text{-approx.} : \text{find } S \in I_1 \cup I_2 \text{ s.t. } |S| \geq \alpha \cdot r$

乱数  を使わないとアルゴリズム

自明な  $O(n) \text{ ウエイ}$  の  $\frac{1}{2}$  近似アルゴリズム (かくほうがくいは)。

Q. 決定的には？

$\lambda \in (V, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 $\alpha\text{-approx.} : \text{find } S \in I_1 \cap I_2$   
s.t.  $|S| \geq \alpha \cdot r$

乱数  を使わばいアルゴリズム

自明な  $O(n|I|)$  の  $\frac{1}{2}$  近似アルゴリズム（かくひくわん）。

貪欲アルゴリズム

```
S ← φ
for  $v \in V$  do
    if  $S \cup \{v\} \in I_1 \cap I_2$  then
        S ←  $S \cup \{v\}$ 
return S
```

Q. 決定的には？

$\lambda \in (V, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 $\alpha\text{-approx.} : \text{find } S \in I_1 \cup I_2 \text{ s.t. } |S| \geq \alpha \cdot r$

乱数  を使わないとどうぞ

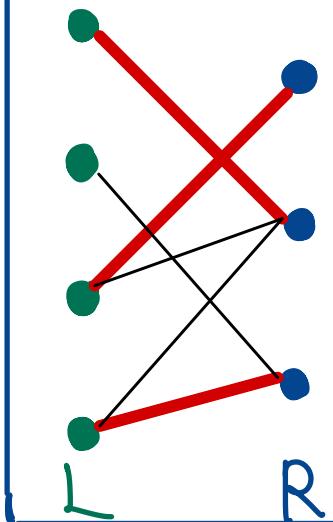
自明な  $O(n) \text{ ケイ}$  の  $\frac{1}{2}$  近似アルゴリズム（かくひくわざ）。

主結果

$O(\frac{n}{\epsilon} + r \log r) \text{ ケイ}$  の  $(\frac{2}{3} - \epsilon)$  近似アルゴリズム

アイト": (二部)マッチングからの類推

(二部)マッチング



given:  $G = (L \cup R, E)$

find:  $M \subseteq E$

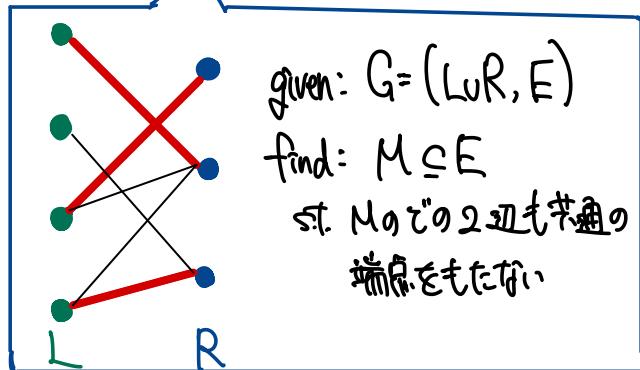
s.t.  $M$  のでの 2 端も共通の  
端点をもたない

short" 実現

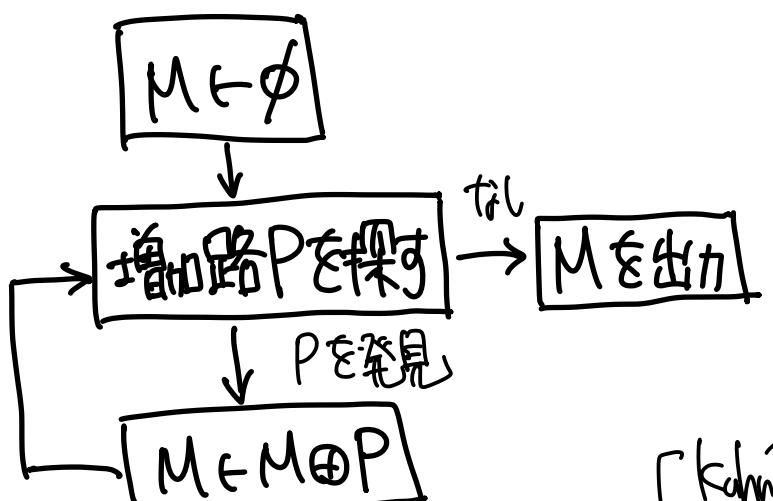
input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 find  $S \in I_1 \cup I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

アイテム：(二部)マッチングの類推

(二部)マッチング



アルゴリズム



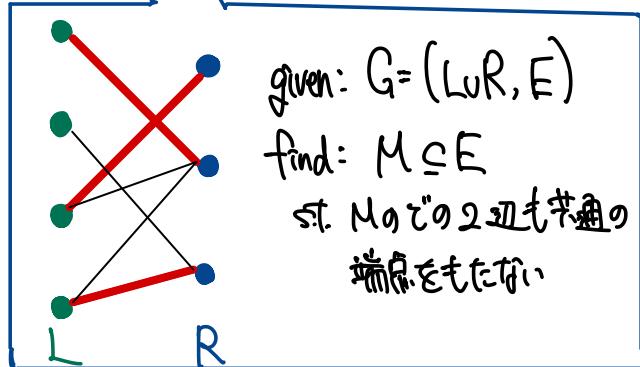
[Kuhn '55]  
 [Edmonds '65]

shortest

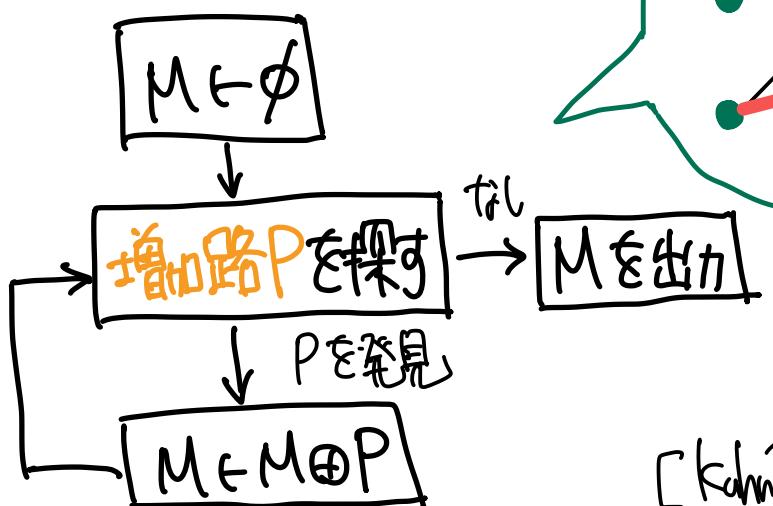
input:  $(V, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 find  $S \in I_1 \cup I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

# アイテム：(二部)マッチングから類推

## (二部)マッチング



アルゴリズム



[Kuhn '55]  
 [Edmonds '65]

## short 実装

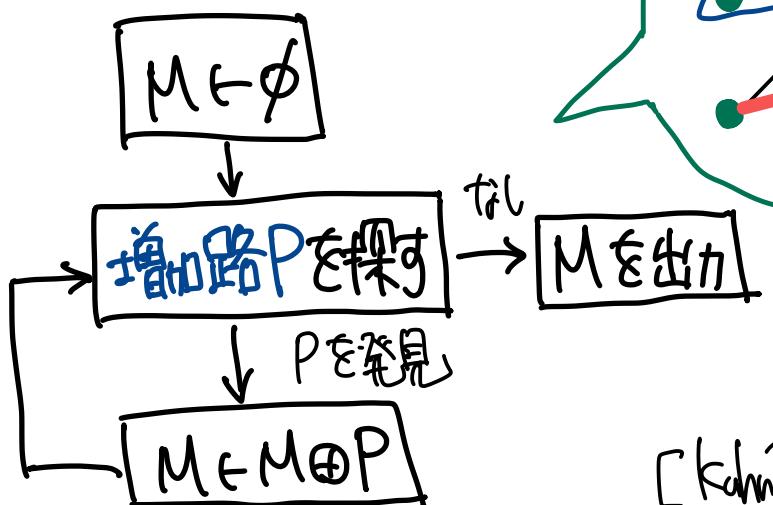
input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 find  $S \in I_1 \cap I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

# アイテム：(二部)マッチングから類推

## (二部)マッチング



アルゴリズム



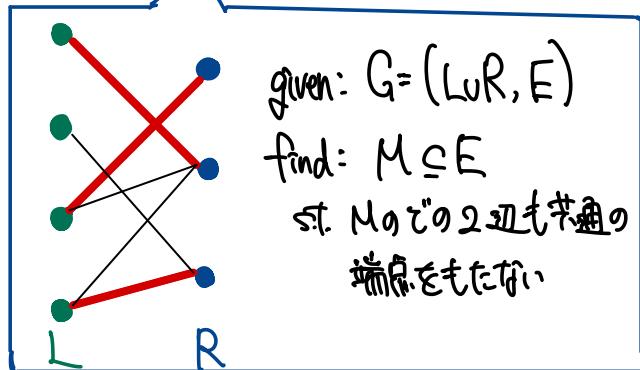
[Kuhn '55]  
 [Edmonds '65]

## short 実装

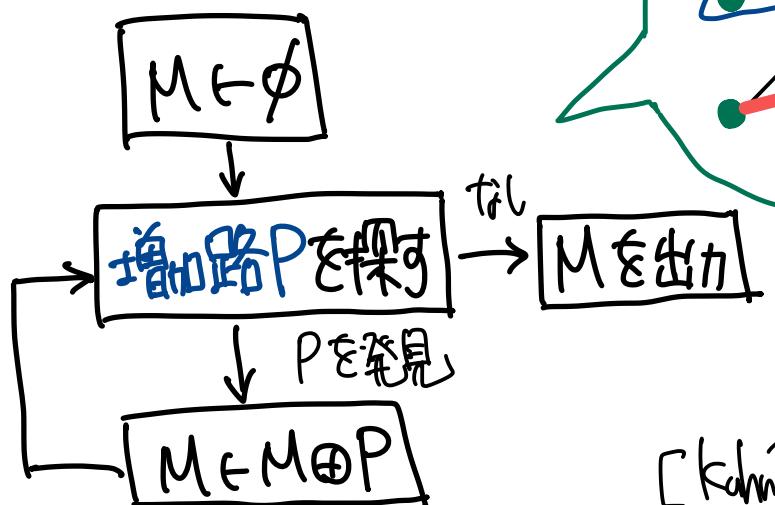
input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 find  $S \in I_1 \cap I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

# アイト": (二部)マッチングからの類推

## (二部)マッチング

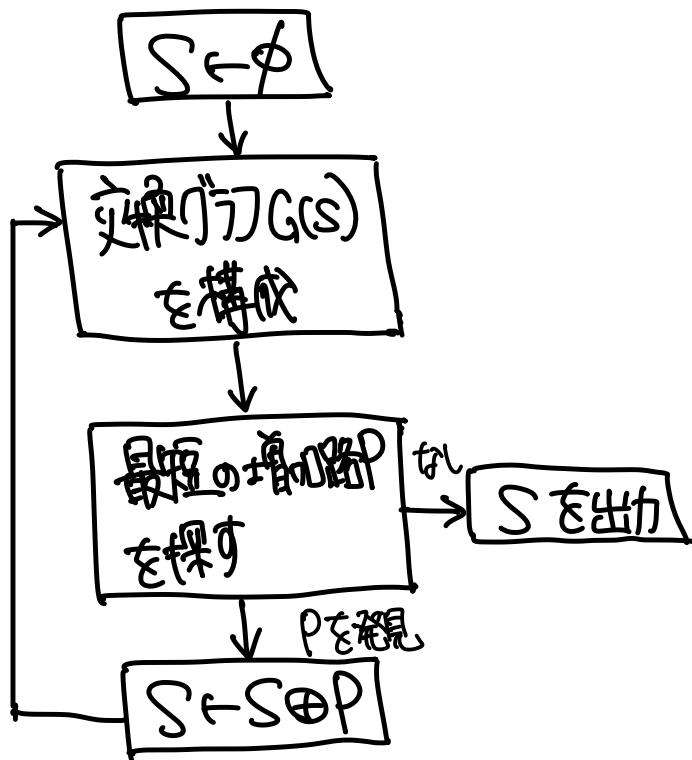


アルゴリズム



## short 実装

アルゴリズム



[Edmonds '70]

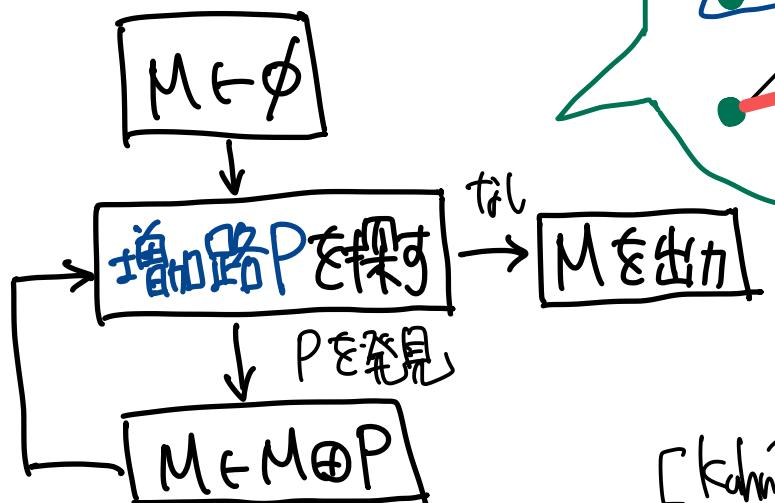
input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 find  $S \in I_1 \cap I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

# アイト": (二部)マッチングの類推

## (二部)マッチング



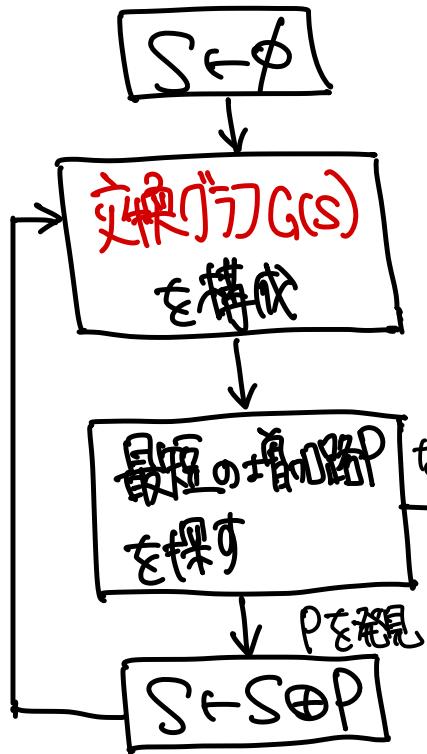
アルゴリズム



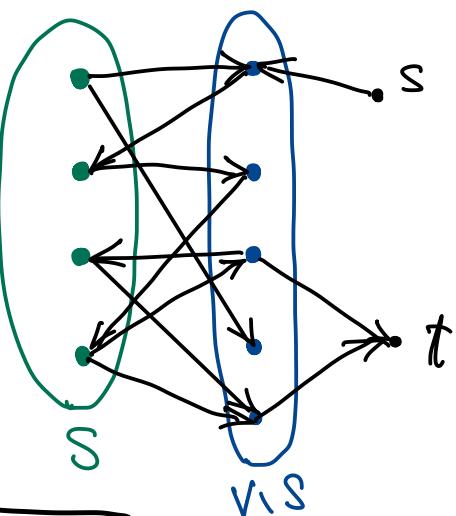
[Kahn '55]  
 [Edmonds '65]

## short 実装

アルゴリズム



input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 find  $S \in I_1 \cap I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries



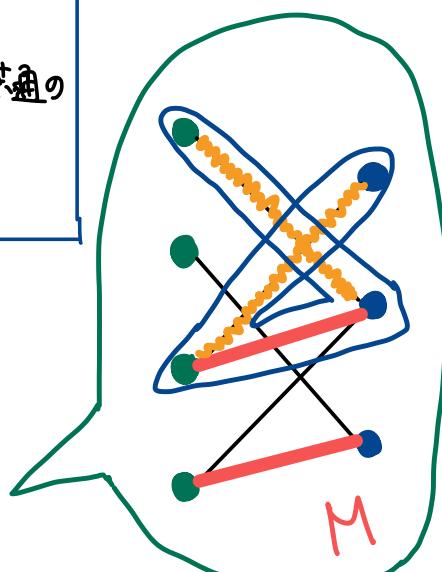
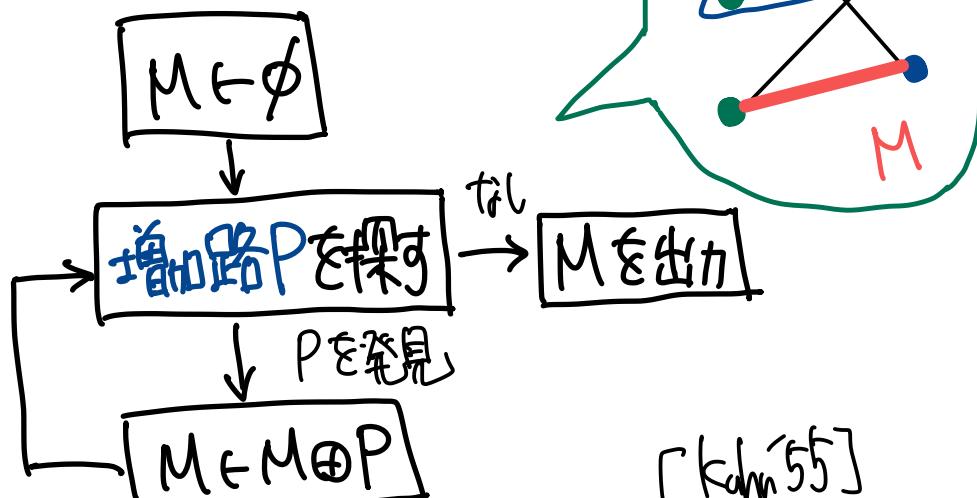
[Edmonds '70]

# アイテム：(二部)マッチングから類推

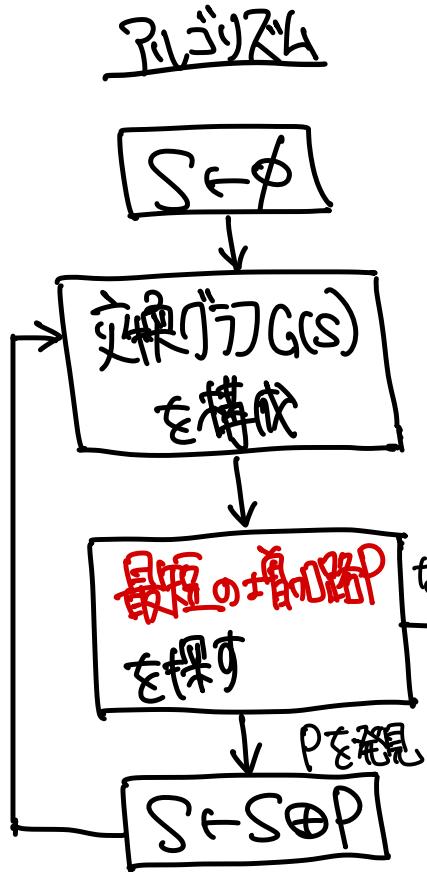
## (二部)マッチング



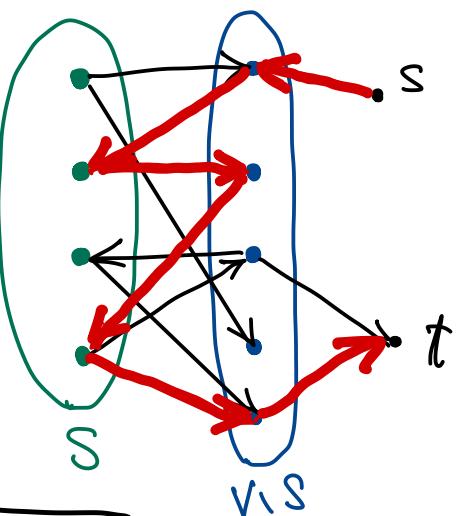
アルゴリズム



## short 実装



input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 find  $S \in I_1 \cap I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries



アイト": (二部)マッチングから類推

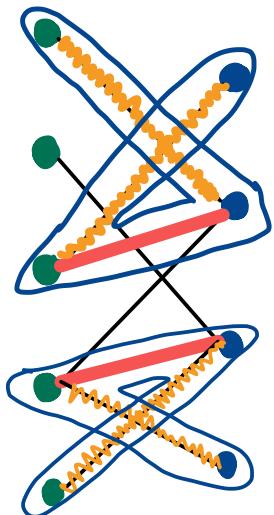
(二部)マッチング

高速なO(n log n)

\* 最短の増加路を求めるアルゴリズム

- ジロッキンゴロー

[Hopcroft-Karp'73]



shortest

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
find  $S \in I_1 \cup I_2$   
s.t.  $|S| \geq \frac{2}{3}r$   
using  $O(n/\epsilon + r \log r)$  queries

アイト": (二部)マッチングから類推

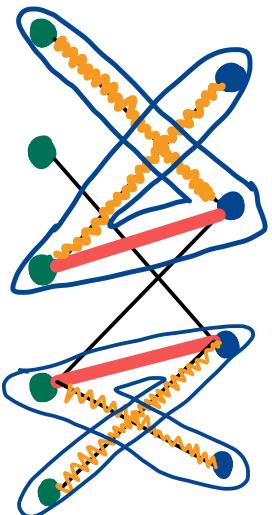
(二部)マッチング

高速なアルゴリズム

\* 最短の増加路をまとめておく

- ジロッキンゴロー

[Hopcroft-Karp'73]



- 素朴な極大な増加路の集合をとめるとO( $n^2$ )
- BFS+DFS

shortest

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
find  $S \in I_1 \cup I_2$   
s.t.  $|S| \geq \frac{2}{3}r$   
using  $O(n/\epsilon + r \log r)$  queries

# アイト": (二部)マッチングから類推

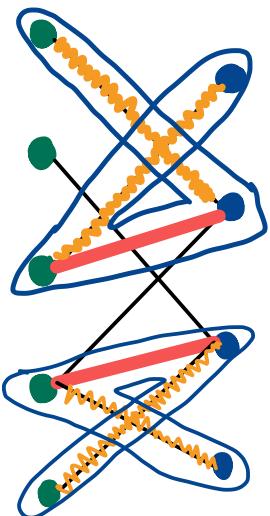
## (二部)マッチング

高速な3(1)アルゴリズム

\* 最短の増加路をまとめてみる

- ジロッキングロー

[Hopcroft-Karp'73]



- 点素から極大な増加路の集合をとれるか?
- BFS+DFS

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 find  $S \in I_1 \cup I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

## shortest"交叉

高速な3(1)アルゴリズム

\* 最短の増加路をまとめてみる

- ジロッキングロー

[Cunningham'86]

- 点素から極大な増加路の集合をとれるか?

# アイト": (二部)マッチングから類推

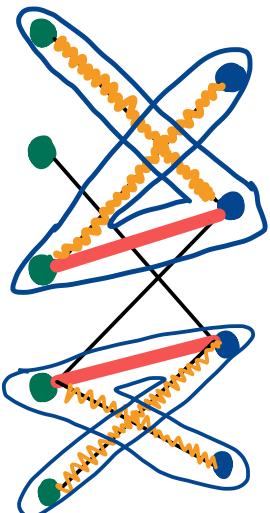
## (二部)マッチング

高速な3(1)アルゴリズム

\* 最短の増加路をまとめてみる

- ジロッキンゴロー

[Hopcroft-Karp'73]



- 点素のみ、極大な増加路の集合をとれるOK
- BFS+DFS

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 find  $S \in I_1 \cup I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

## short" 実装

高速な3(1)アルゴリズム

\* 最短の増加路をまとめてみる

- ジロッキンゴロー

[Cunningham'86]

- 点素のみ、極大な増加路の集合をとれるOK
- BFS+DFS

# アイト": (二部)マッチングから類推

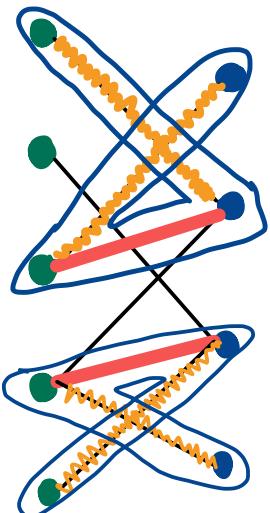
## (二部)マッチング

高速な3(1)アルゴリズム

\* 最短の増加路をまとめておく

- ジロッキンゴロー

[Hopcroft-Karp'73]



- 点素み、極大な増加路の集合をみてOK
- BFS + DFS

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 find  $S \in I_1 \cup I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

## short" 実装

高速な3(1)アルゴリズム

\* 最短の増加路をまとめておく

- ジロッキンゴロー

[Cunningham'86]

- 点素み、極大な増加路の集合をみてOK



増加集合をみてOK [CLSSW'19]

$\Pi_0 := (B_1, A_1, B_2, \dots, B_{k+1})$

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

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

③  $S + B_j \in I_1$

④  $S + B_{k+1} \in I_2$

⑤  $S - A_k + B_{k+1} \in I_1$

⑥  $S - A_k + B_k \in I_2$

アイト":  $(\frac{1}{2})$  マチ= ジ の類推

$(\frac{1}{2})$  マチ= ジ

Mに長さ3の増加路が存在しない

$\Rightarrow M$  は  $\frac{2}{3}$  近似

short" 実

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
find  $S \in I_1 \cap I_2$   
s.t.  $|S| \geq \frac{2}{3}r$   
using  $O(n/\epsilon + r \log r)$  queries

アイト": (二部)マッチングから類推

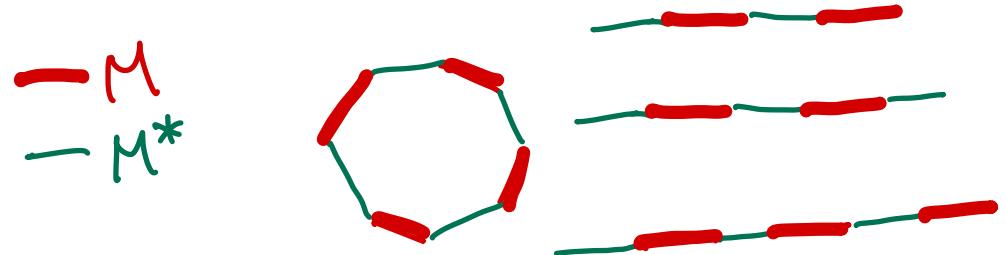
(二部)マッチング

short"交叉

$M$ に長さ3の増加路が存在しあい

$\Rightarrow M$ は  $\frac{2}{3}$  近似

∴ 最大マッチング  $M^*$  をもってきたとき、



$M \cup M^*$  は長さ4以上の交替道 or 交替サインに分解できる

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
find  $S \in I_1 \cup I_2$   
s.t.  $|S| \geq \frac{2}{3}r$   
using  $O(n/\epsilon + r \log r)$  queries

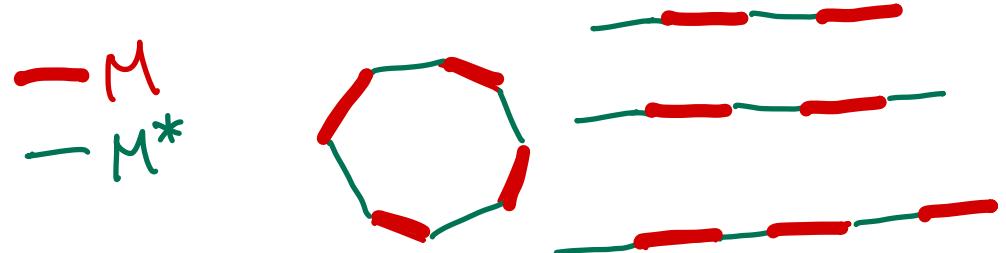
アイト": (二部)マッチングの類推

(二部)マッチング

$M \in E^3$  の増加路が存在しない

$\Rightarrow M$  は  $\frac{2}{3}$  近似

∴ 最大マッチング  $M^*$  をもってきたとき、



$M \Delta M^*$  は長さ4以上の交替道 or 交替サインに分解できる

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
find  $S \in I_1 \cup I_2$   
s.t.  $|S| \geq \frac{2}{3}r$   
using  $O(n/\epsilon + r \log r)$  queries

short 実験

$G(S) \in E^4$  の増加路が存在しない

$\Rightarrow S$  は  $\frac{2}{3}$  近似

アイト": (二部)マッチングの類推

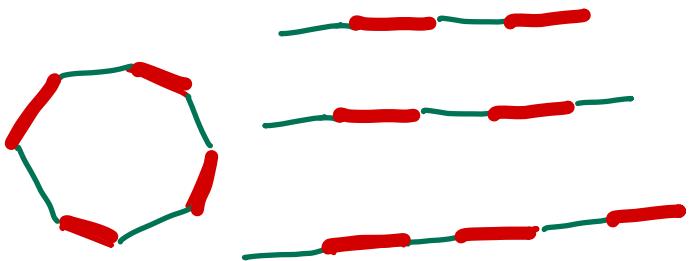
(二部)マッチング

$M$ に長さ3の増加路が存在しない

$\Rightarrow M$ は  $\frac{2}{3}$  近似

∴ 最大マッチング  $M^*$  をもってきたとき、

-  $M$   
-  $M^*$



$M \Delta M^*$  は長さ4以上の交互道 or 交互サインに分解できる

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 find  $S \in I_1 \cup I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

short 実験

$G(S)$ に長さ4の増加路が存在しない

$\Rightarrow S$ は  $\frac{2}{3}$  近似

本研究で示す証明!

[Blickstad'21] で、長さ4の増加路に対するある  
増加集合をもつて3ループを途中で止める



ほぼ長さ4の増加路が存在しないようにでき、  
 $(\frac{2}{3} - \epsilon)$  近似

アイト": (二部)マッチングからの類推

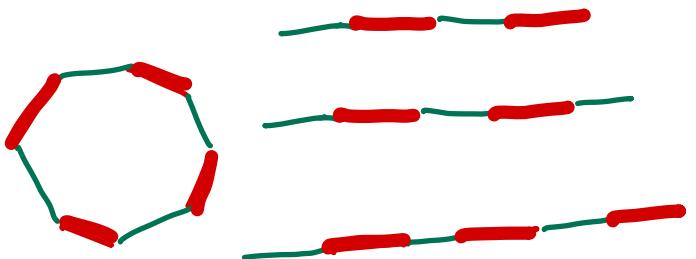
(二部)マッチング

$M$ に長さ3の増加路が存在しない

$\Rightarrow M$ は  $\frac{2}{3}$  近似

∴ 最大マッチング  $M^*$  をもってきたとき、

-  $M$   
-  $M^*$



$M \Delta M^*$  は長さ4以上の交互道 or 交互サインに分解できる

input:  $(U, I_1), (V, I_2)$   
 $n := |V|, r := \max_{S \in I_1 \cup I_2} |S|$   
 find  $S \in I_1 \cup I_2$   
 s.t.  $|S| \geq \frac{2}{3}r$   
 using  $O(n/\epsilon + r \log r)$  queries

short文

$G(S)$ に長さ4の増加路が存在しない

$\Rightarrow S$ は  $\frac{2}{3}$  近似

本研究で示す証明!

[Blickstad'21] で、長さ4の増加路に対するある  
増加集合をもつて、アルゴリズムを途中で止める



ほぼ長さ4の増加路が存在しないようにでき、  
 $(\frac{2}{3} - \epsilon)$  近似

長さ4の増加路はない  
もうない

本結果の予想外に面白いポイント

本アレゴリズムはストリーミングアレゴリズム  
に挑戦できる！

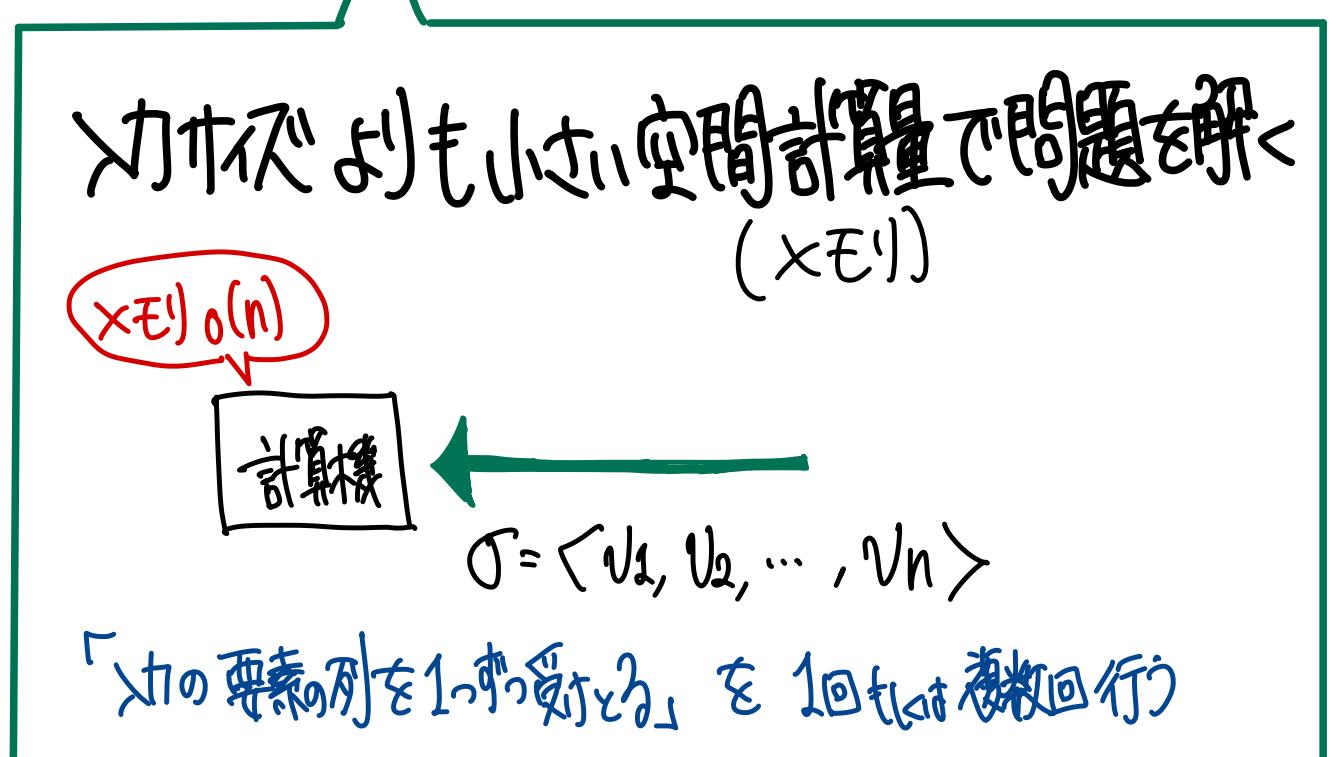
# 本結果の予想外に面白いポイント

本アルゴリズムはストリーミングアルゴリズム  
に挑戦できる！

メモリよりも小さい空間計算量で問題を解く  
(メモリ)

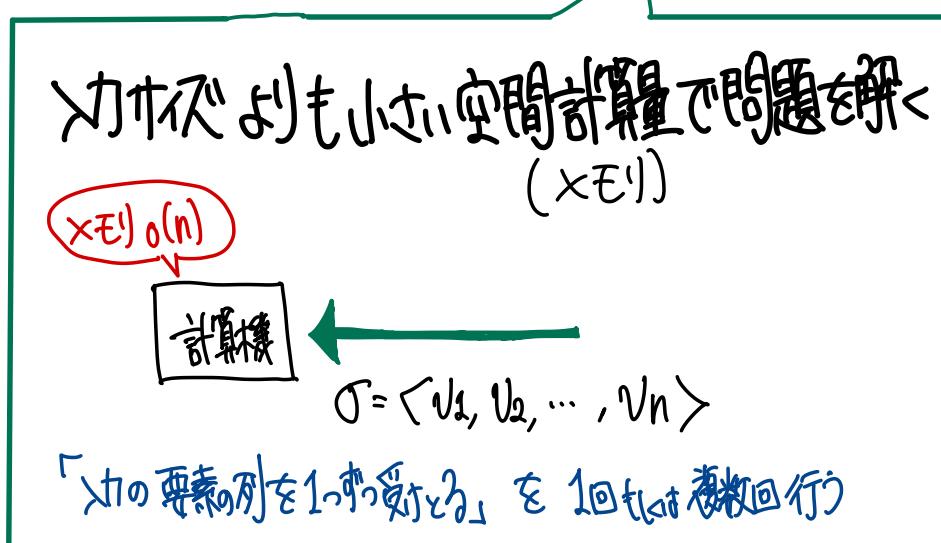
# 本結果の予想外に面白いポイント

本アルゴリズムは **ストリーミングアルゴリズム**  
に挑戦できる！

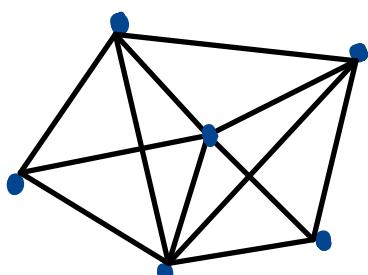


# 本結果の予想外に面白いポイント

本アルゴリズムは **ストリーミングアルゴリズム** に挑戦できる！



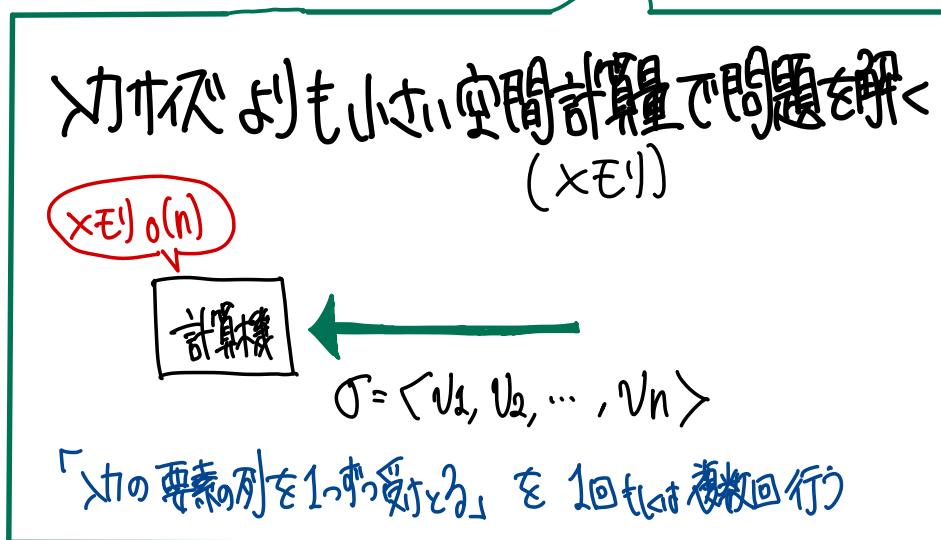
\* 特に、グラフの問題では、頂点数のオーダーのメモリで解く  
セミストリーミングアルゴリズムがよく研究



# 本結果の予想外に面白いポイント

本アルゴリズムはストリーミングアルゴリズム

に挑戦できる！



必要なメモリが少なくて済む

\* 特にグラフの問題では、頂点数のオーダーのメモリで解く

セミストリーミングアルゴリズムがよく研究

e.g. マッチング問題

$\lambda I : (U, I_1), (V, I_2)$

$\max |S| \text{ s.t. } S \in I_1 \cap I_2$

$n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$

# マトリイド交叉問題

の ストライニーフ 3レボリズム

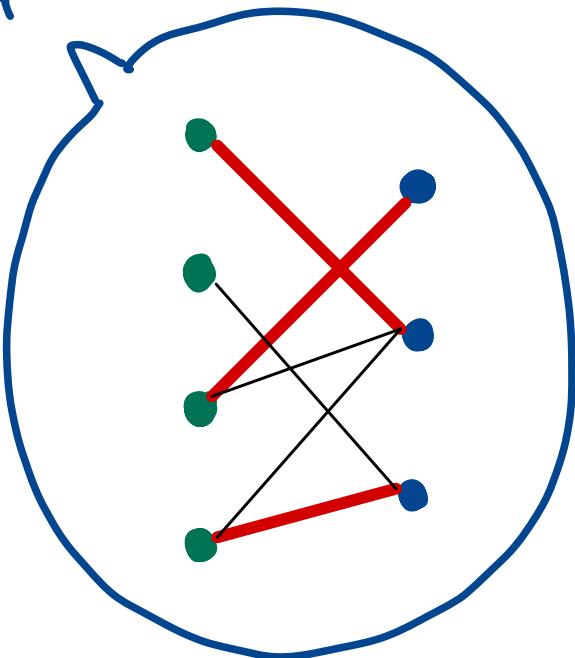
# マトロイド交叉問題

の 2トリ-ミニマム 3ルゴリズム

2トリ問題の  
セミストリーミングアルゴリズム

Paz-Schwartzman'17

Bernstein'20



$\lambda I = (U, I_1), (V, I_2)$   
 $\max |S| \text{ s.t. } S \in I_1 \cap I_2$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 $r_i := \max_{S \in I_i} |S| (i=1,2)$

# マトロイド交叉問題

$\lambda(I) := (U, I_1), (V, I_2)$   
 $\max |S| \text{ s.t. } S \in I_1 \cap I_2$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 $r_i := \max_{S \in I_i} |S| (i=1,2)$

の ストリーミングアルゴリズム

ストリーミング問題の  
セミストリーミングアルゴリズム

Paz-Schwartzman'17

概要



Garg-Jordan-Svensson'21

1回のパス, 重みあり,  $(\frac{1}{2} - \epsilon)$  精度, 空間  $\tilde{O}(r_1 + r_2)$

Bernstein'20



Huang-Sellier'24

1回のパス, ランダム走査,  $(\frac{2}{3} - \epsilon)$  精度, 空間  $\tilde{O}_e(r)$

# マトロイド交叉問題

$\lambda(t) : (U, I_1), (V, I_2)$   
 $\max |S| \text{ s.t. } S \in I_1 \cap I_2$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 $r_i := \max_{S \in I_i} |S| (i=1,2)$

の ストリーミングアルゴリズム

マトロイド問題の  
アリストリーミングアルゴリズム  
Paz-Schwartzman'17

拡張



Garg-Jordan-Svensson'21

1回のパス, 重みあり,  $(\frac{1}{2} - \epsilon)$  近似, 空間  $\tilde{O}_\epsilon(r_1 + r_2)$

Bernstein'20



Huang-Sellier'24

1回のパス, ランダム走査,  $(2/3 - \epsilon)$  近似, 空間  $\tilde{O}_\epsilon(r)$

本研究

$O(\frac{1}{\epsilon})$  回のパス,  $(2/3 - \epsilon)$  近似, 空間  $\tilde{O}_\epsilon(r_1 + r_2)$

# マトロイド交叉問題

の ストリーミングアルゴリズム

2,4-ツイグ問題の  
セミストリーミングアルゴリズム

Paz-Schwartzman'17

拡張



Garg-Jordan-Svensson'21

1回のパス, 重みあり,  $(\frac{1}{2} - \epsilon)$ 近似, 空間  $\tilde{O}_\epsilon(r_1 + r_2)$

Bernstein'20



Huang-Sellier'24

1回のパス, ランダム走査,  $(2/3 - \epsilon)$ 近似, 空間  $\tilde{O}_\epsilon(r)$

Feigenbaum-Kannan-McGregor  
- Suri-Zhang'04



本研究

$O(\frac{1}{\epsilon})$ 回のパス,  $(2/3 - \epsilon)$ 近似, 空間  $\tilde{O}_\epsilon(r_1 + r_2)$

グラフのストリーミングの最初の論文

$\lambda(t) : (U, I_1), (V, I_2)$   
 $\max |S| \text{ s.t. } S \in I_1 \cap I_2$   
 $n := |V|, r := \max_{S \in I_1 \cap I_2} |S|$   
 $r_i := \max_{S \in I_i} |S| (i=1,2)$

## 結論

- マトリクス交叉問題に対し、  
決定的な  $(2/3 - \varepsilon)$ -近似  $\tilde{O}_\varepsilon(n) \cdot \mathcal{I}^*$  アルゴリズム  
を設計
- これを  $O(1/\varepsilon)$  回以上  $(2/3 - \varepsilon)$  近似でミストリートするアルゴリズム  
(応用 : FKMSZ'04 の拡張)

# 結論

- マトリクス交叉問題に対し、  
決定的な  $(2/3 - \varepsilon)$ -近似  $\tilde{O}_\varepsilon(n) \text{ オイ}$  アルゴリズム  
を設計
- これを  $O(1/\varepsilon)$  回以上  $(2/3 - \varepsilon)$  近似 セミストリーディングアルゴリズム  
(応用：FKMSZ'04 の拡張)

## open questions

- 決定的な  $(1 - \varepsilon)$  近似  $\tilde{O}_\varepsilon(n) \text{ オイ} \text{アルゴリズム}$
- $\text{poly}(1/\varepsilon)$  回以上  $(1 - \varepsilon)$  近似 セミストリーディングアルゴリズム