準線形回の独立性オラクルへのクエリを使用するマトロイド交叉問題の決定的な$(\frac{2}{3}-\varepsilon)$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
@ 筑波大学　5月31日

arXiv: 2410.18820 (To appear in WADS'25)

マトロイドとは?

Def.

有限集合Vとその空でない部分集合族$\mathcal{I} \subseteq 2^V$ 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. $T \cup \{e\} \in \mathcal{I}$

マトロイドとは？ ー 線形独立性の一般化

Def.

有限集合 V とその空でない部分集合族 $\mathcal{I} \subseteq 2^V$ 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.

- 線形マトロイド

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

$V = \{①, ②, ③\}$

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

マトロイドとは？ー線形独立性の一般化

Def.

有限集合 V とその空でない部分集合族 $\mathcal{I} \subseteq 2^V$ 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 は独立

e.g.

- 線形マトロイド

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

$V = \{①, ②, ③\}$

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

マトロイドとは？ ― 線形独立性の一般化

Def.

有限集合Vとその空でない部分集合族 $\mathcal{I} \subseteq 2^V$ 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.

- 線形マトロイド

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

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

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

- 分割マトロイド
- ラミナーマトロイド
- グラフ的マトロイド
- sparse paving マトロイド

etc...

マトロイド交叉問題とは？

入力: 2つのマトロイド $M_1 = (V, \mathcal{I}_1)$, $M_2 = (V, \mathcal{I}_2)$

出力: 最大サイズの共通独立集合 $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

マ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つのマトロイド $M_1=(V,\mathcal{I}_1)$, $M_2=(V,\mathcal{I}_2)$

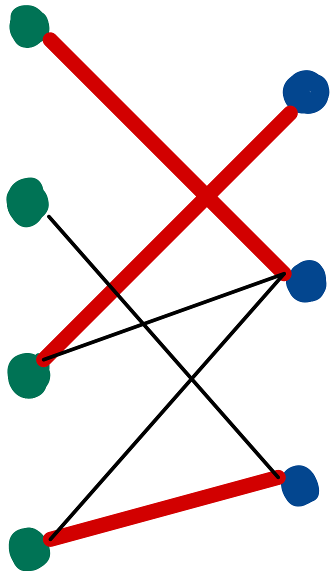
出力: 最大サイズの共通独立集合 $S\in\mathcal{I}_1\cap\mathcal{I}_2$

e.g. 二部グラフの最大マッチング

V = 辺集合

$\mathcal{I}_1$ = 左側の各頂点に接する辺は高々1本

$\mathcal{I}_2$ = 右側の各頂点に接する辺は高々1本



マ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つのマトロイド $M_1 = (V, \mathcal{I}_1)$, $M_2 = (V, \mathcal{I}_2)$

出力：最大サイズの共通独立集合 $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

なぜ、本問題は重要か？

マ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

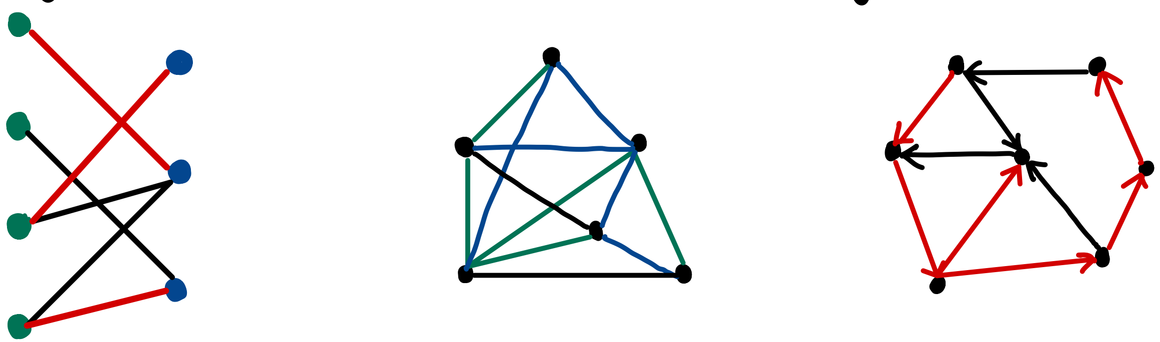
入力: 2つのマトロイド $M_1 = (V, \mathcal{I}_1)$, $M_2 = (V, \mathcal{I}_2)$

出力: 最大サイズの共通独立集合 $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

なぜ、本問題は重要か？

① 多くの問題の拡張

二部グラフのマッチング, 全域木の詰め込み, 有向全域木



マトロイド交叉問題とは？

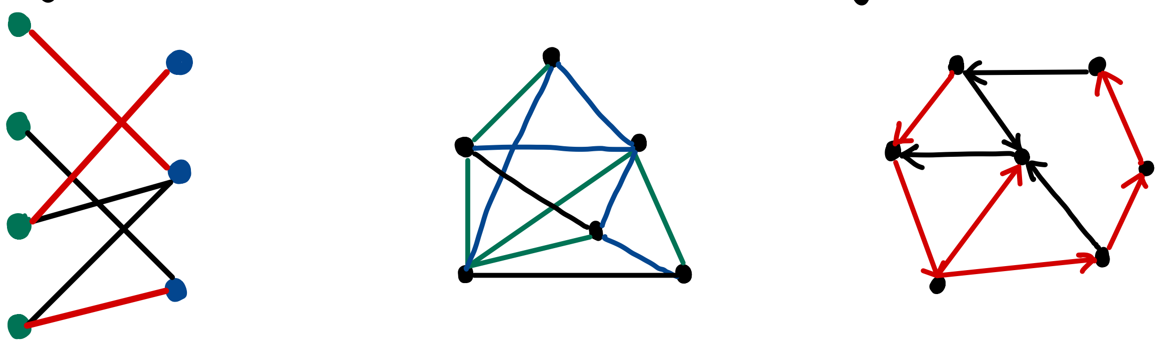
— 二部グラフの最大マッチング問題の一般化

入力: 2つのマトロイド $M_1=(V,\mathcal{I}_1)$, $M_2=(V,\mathcal{I}_2)$
出力: 最大サイズの共通独立集合 $S\in\mathcal{I}_1\cap\mathcal{I}_2$

なぜ、本問題は重要か？

① 多くの問題の拡張

二部グラフのマッチング, 全域木の詰め込み, 有向全域木



② 組合せ最適化外への応用

- 電気回路解析
- ネットワーク符号化

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

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

マトロイド交叉問題は

多項式時間で解ける

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

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

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

マトロイド交叉問題は多項式時間で解ける

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

マトロイドは独立性オラクルとして与えられる

クエリ(質問)　　回答

$S \in \mathcal{I}$ か? → $\boxed{\text{独立性オラクル}}$ → Yes or No

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

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

マトロイド交叉問題は

多項式回のクエリで解ける

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

マトロイドは独立性オラクルとして与えられる

クエリ(質問) 回答

$S \in \mathcal{I}$か? $\rightarrow$ $\boxed{\text{独立性オラクル}}$ $\rightarrow$ Yes or No

計算時間 = オラクルへのクエリ回数

マトロイド交叉問題の時間計算量

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

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

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

独立性オラクルへのクエリ回数

マトロイド交叉問題の時間計算量

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

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

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

独立性オラクルへのクエリ回数

1970s	Edmonds, Lawler, Aigner-Dowling	$O(nr^2)$
1986	Cunningham	$O(nr^{3/2})$

マトロイド交叉問題の時間計算量

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

		独立性オラクルへのクエリ回数
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] は乱択を許せば $\tilde{O}(nr^{3/4})$, 決定的な場合は $\tilde{O}(nr^{5/6})$

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 $O(n \cdot \mathrm{polylog}(n))$ 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 $O(n \cdot \mathrm{polylog}(n))$
のアルゴリズム

★ 厳密解ではなく、近似解を目指す

α 近似解 $S \in \mathcal{I}_1 \cap \mathcal{I}_2$

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

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 $O(n \cdot \mathrm{polylog}(n))$ のアルゴリズム

★ 厳密解ではなく、近似解を目指す

α 近似解 $S \in \mathcal{I}_1 \cap \mathcal{I}_2$ s.t. $|S| \geq \alpha \cdot \mathrm{OPT}$

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

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

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

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

マトロイド交叉問題の準線形時間アルゴリズム

Folklore	$\frac{1}{2}$ 近似	$O(n)$ クエリ

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

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

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

マトロイド交叉問題の準線形時間アルゴリズム

Folklore	$\frac{1}{2}$ 近似	$O(n)$ クエリ
Guruganesh-Singla '17	$\frac{1}{2} + 10^{-4}$ 近似	$O(n)$ クエリ

マトロイド交叉問題の準線形時間アルゴリズム

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

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

※ arXiv に投稿したのは本研究とほぼ同時

マトロイド交叉問題の準線形時間アルゴリズム

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

Folklore	$\frac{1}{2}$ 近似	$O(n)$ クエリ
Guruganesh-Singla '17	$\frac{1}{2} + 10^{-4}$ 近似	$O(n)$ クエリ
Blikstad-Tu '25 ※	$1-\varepsilon$ 近似	$\tilde{O}_\varepsilon(n)$ クエリ

最近の研究は全て乱択アルゴリズム!

Q. 決定的には？

乱数  を使わないアルゴリズム

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

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

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

Q. 決定的には？

乱数  を使わないアルゴリズム

自明な $O(n)$ クエリの $\frac{1}{2}$ 近似アルゴリズムしか知られていない。

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

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

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

Q. 決定的には？

乱数  を使わないアルゴリズム

自明な $O(n)$ クエリの $\frac{1}{2}$ 近似アルゴリズムしか知られていない。

貪欲アルゴリズム

$S \leftarrow \emptyset$

for $v \in V$ do

　if $S\cup\{v\}\in\mathcal{I}_1\cap\mathcal{I}_2$ then

　　$S \leftarrow S\cup\{v\}$

return S

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

Q. 決定的には?

乱数  を使わないアルゴリズム

自明な $O(n)$ クエリの $\frac{1}{2}$ 近似アルゴリズムしか知られていない。

主結果

$O\left(\frac{n}{\varepsilon} + r \log r\right)$ クエリの $\left(\frac{2}{3} - \varepsilon\right)$ 近似アルゴリズム

アイデア: (二部)マッチングからの類推

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

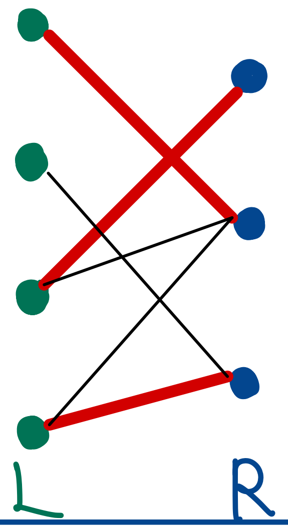
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$

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

(二部)マッチング



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

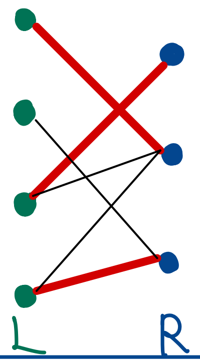
find: $M \subseteq E$

s.t. Mのどの2辺も共通の端点をもたない

マトロイド交叉

アイデア: (二部)マッチングからの類推

(二部)マッチング

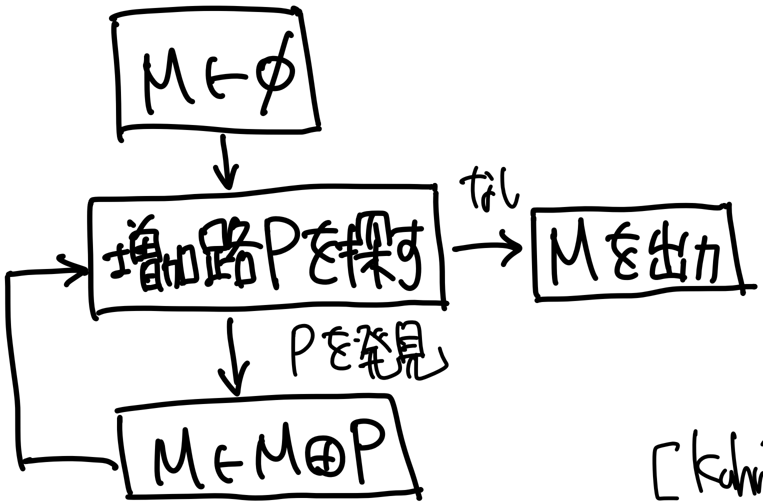


given: $G=(L\cup R, E)$

find: $M\subseteq E$

s.t. Mのどの2辺も共通の端点をもたない

アルゴリズム



[Kuhn'55]
[Edmonds'65]

マトロイド交叉

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

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

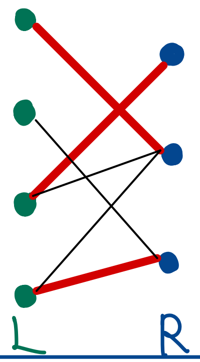
find $S\in\mathcal{I}_1\cap\mathcal{I}_2$

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

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

アイデア：(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のどの2辺も共通の端点をもたない

アルゴリズム

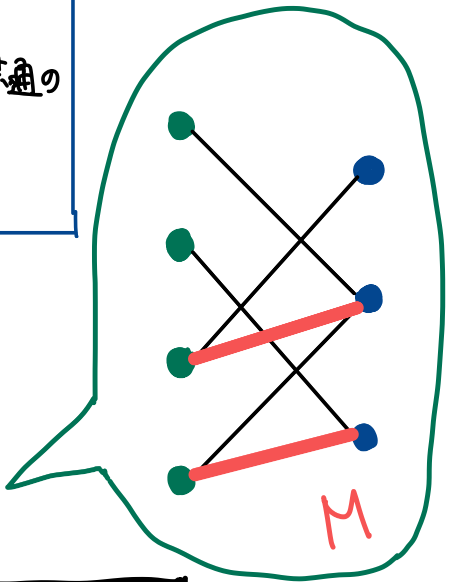
$M \leftarrow \emptyset$

↓

増加路Pを探す $\xrightarrow{\text{なし}}$ Mを出力

↓ Pを発見

$M \leftarrow M \oplus P$



[Kuhn'55]

[Edmonds'65]

マトロイド交叉

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

アイデア: (二部)マッチングからの類推

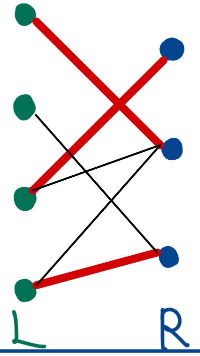
input: $(V,\mathcal{I}_1),(V,\mathcal{I}_2)$

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

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

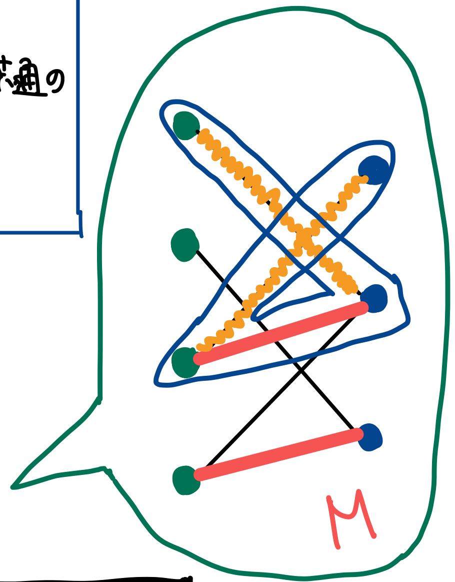
(二部)マッチング



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

find: $M\subseteq E$

s.t. Mのどの2辺も共通の端点をもたない



アルゴリズム

$M\leftarrow\emptyset$

↓

増加路Pを探す → (なし) → Mを出力

↓ Pを発見

$M\leftarrow M\oplus P$ (→ 増加路Pを探す へ戻る)

[Kuhn'55]
[Edmonds'65]

マトロイド交叉

アイデア：(二部)マッチングからの類推

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

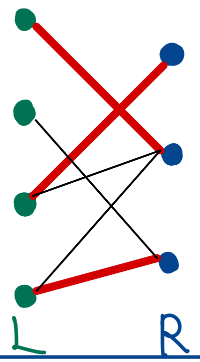
$n := |V|,\ r := \max_{S\in\mathcal{I}_1\cap\mathcal{I}_2} |S|$

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

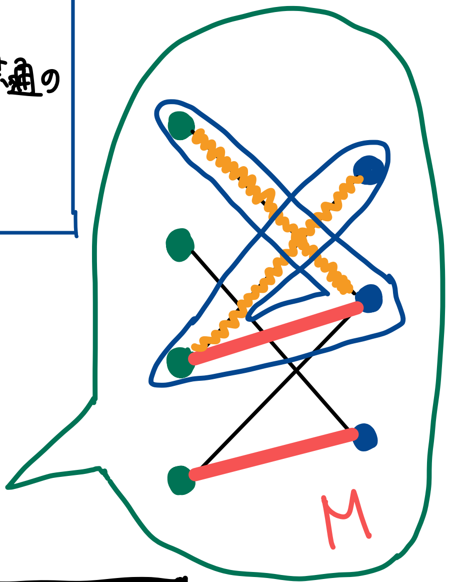
(二部)マッチング



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

find: $M \subseteq E$

s.t. Mのどの2辺も共通の端点をもたない



アルゴリズム

$M \leftarrow \phi$

↓

増加路Pを探す → なし → Mを出力

↓ Pを発見

$M \leftarrow M \oplus P$

[Kuhn '55]

[Edmonds '65]

マトロイド交叉

アルゴリズム

$S \leftarrow \phi$

↓

交換グラフ$G(S)$を構成

↓

最短の増加路Pを探す → なし → Sを出力

↓ Pを発見

$S \leftarrow S \oplus P$

[Edmonds '70]

アイデア: (二部)マッチングからの類推

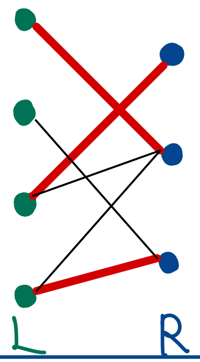
input: $(V,\mathcal{I}_1),(V,\mathcal{I}_2)$

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

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

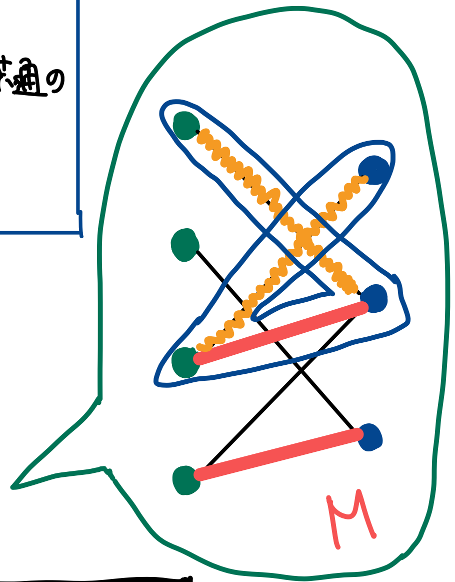
(二部)マッチング



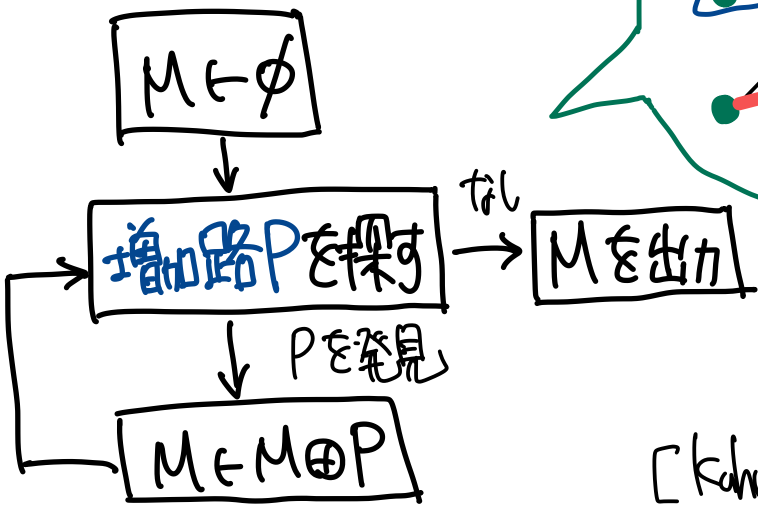
given: $G=(L\cup R, E)$

find: $M\subseteq E$

s.t. Mのどの2辺も共通の端点をもたない



アルゴリズム

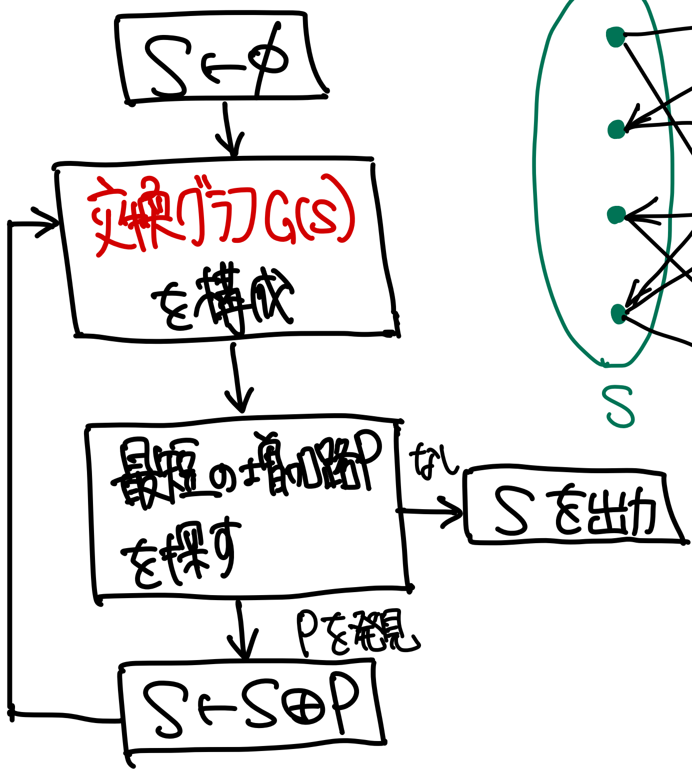


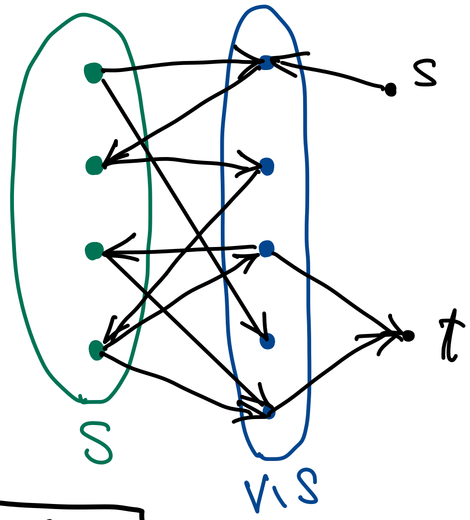
[Kuhn'55]

[Edmonds'65]

マトロイド交叉

アルゴリズム





[Edmonds'70]

アイデア: (二部)マッチングからの類推

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

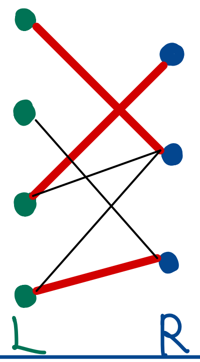
$n := |V|$, $r := \max_{S\in\mathcal{I}_1\cap\mathcal{I}_2} |S|$

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

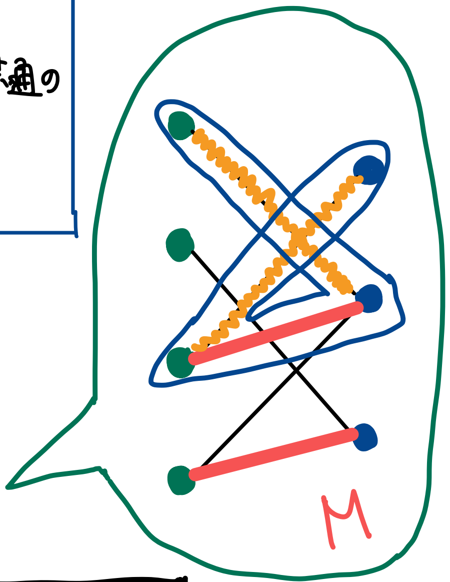
(二部)マッチング



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

find: $M\subseteq E$

s.t. Mのどの2辺も共通の端点をもたない



アルゴリズム

$M \leftarrow \emptyset$

↓

増加路Pを探す → なし → Mを出力

↓ Pを発見

$M \leftarrow M \oplus P$ (→ 増加路Pを探す へ戻る)

[Kuhn'55]

[Edmonds'65]

マトロイド交叉

アルゴリズム

$S \leftarrow \emptyset$

↓

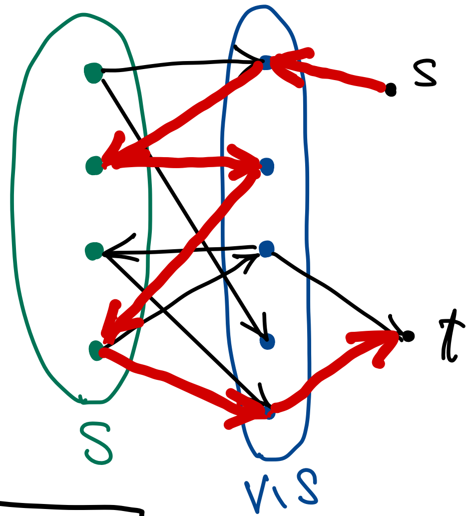
交換グラフ$G(S)$を構成

↓

最短の増加路Pを探す → なし → Sを出力

↓ Pを発見

$S \leftarrow S \oplus P$ (→ 交換グラフ$G(S)$を構成 へ戻る)



[Edmonds'70]

アイデア：(二部)マッチングからの類推

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

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

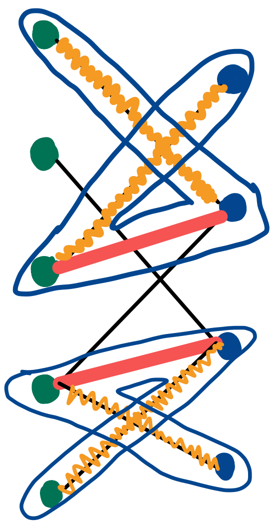
(二部)マッチング

高速なアルゴリズム

★ 最短の増加路をまとめてみつける

- ブロッキングフロー

[Hopcroft-Karp '73]



マトロイド交叉

アイデア：(二部)マッチングからの類推

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

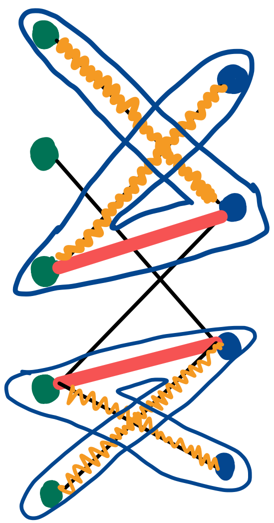
(二部)マッチング

高速なアルゴリズム

★ 最短の増加路をまとめてみつける

- ブロッキングフロー

[Hopcroft-Karp '73]



- 点素かつ極大な増加路の集合をとればOK
- BFS+DFS

マトロイド交叉

アイデア：(二部)マッチングからの類推

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

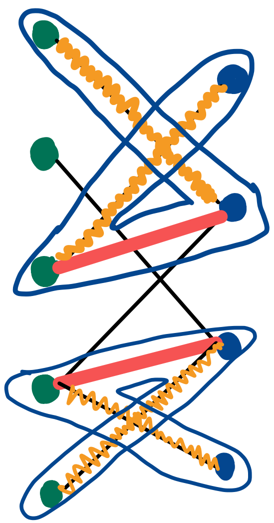
(二部)マッチング

高速なアルゴリズム

★ 最短の増加路をまとめてみつける

- ブロッキングフロー

[Hopcroft-Karp '73]



- 点素かつ極大な増加路の集合をとればOK
- BFS+DFS

マトロイド交叉

高速なアルゴリズム

★ 最短の増加路をまとめてみつける

- ブロッキングフロー

[Cunningham '86]

- 点素かつ極大な増加路の集合をとればOK？

アイデア: (二部)マッチングからの類推

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

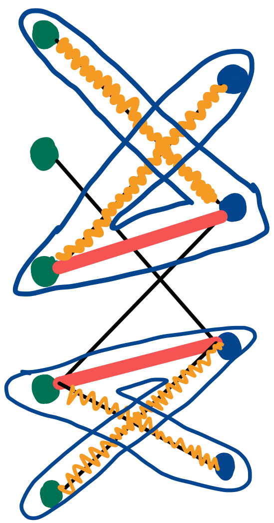
(二部)マッチング

高速なアルゴリズム

★ 最短の増加路をまとめてみつける

- ブロッキングフロー

[Hopcroft-Karp '73]



- 点素かつ極大な増加路の集合をとればOK
- BFS+DFS

マトロイド交叉

高速なアルゴリズム

★ 最短の増加路をまとめてみつける

- ブロッキングフロー

[Cunningham '86]

- 点素 ~~かつ極大な増加路の集合をとれば~~ OK
- BFS+DFS

アイデア：(二部)マッチングからの類推

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

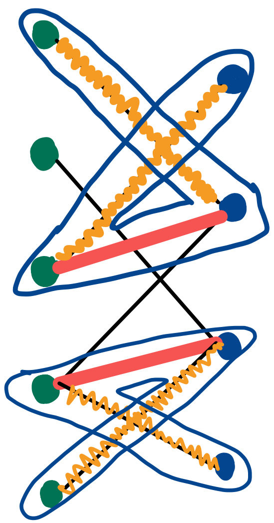
(二部)マッチング

高速なアルゴリズム

★ 最短の増加路をまとめてみつける

- ブロッキングフロー

[Hopcroft-Karp '73]



- 点素かつ極大な増加路の集合をとればOK
- BFS+DFS

マトロイド交叉

高速なアルゴリズム

★ 最短の増加路をまとめてみつける

- ブロッキングフロー

[Cunningham '86]

- 点素かつ ~~極大な増加路の集合をとれ~~ばOK

増加集合をみつければOK [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 $|B_1| = |A_1| = \cdots = |B_{\ell+1}|$

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 $S - A_k + B_k \in \mathcal{I}_2$

アイデア：(二部)マッチングからの類推

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

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

(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

マトロイド交叉

アイデア: (二部)マッチングからの類推

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

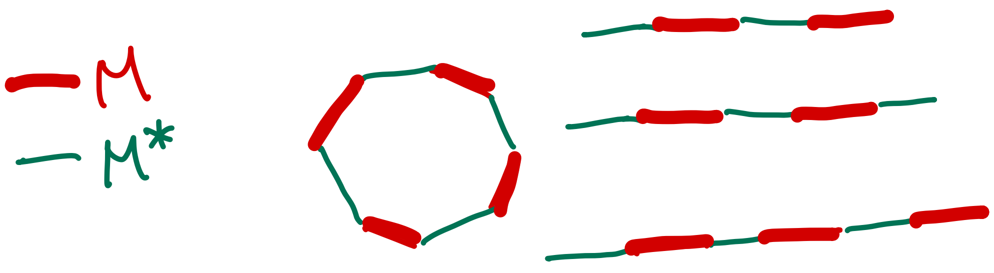
using $O(n/\varepsilon + r \log r)$ queries

(二部)マッチング

Mに長さ3の増加路が存在しない

⇒ Mは $\frac{2}{3}$ 近似

∵ 最大マッチングM^*をもってきたとき



$M \triangle M^*$ は長さ4以上の交互道 or 交互サイクル に分解できる

マトロイド交叉

アイデア：(二部)マッチングからの類推

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

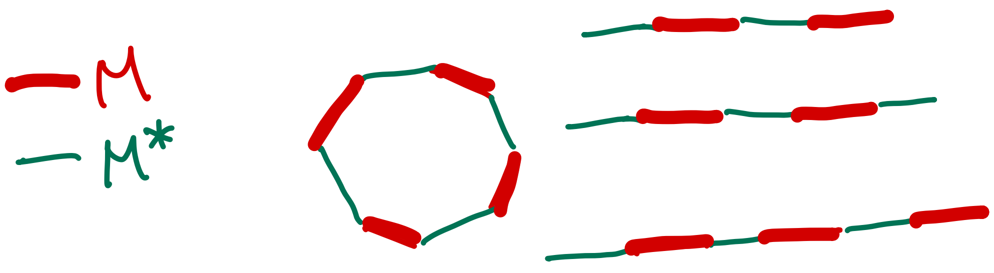
using $O(n/\varepsilon + r \log r)$ queries

(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、



$M \triangle M^*$ は長さ4以上の交互道 or 交互サイクルに分解できる

マトロイド交叉

$G(S)$ に長さ4の増加路が存在しない

$\Rightarrow$ S は $\frac{2}{3}$ 近似

アイデア: (二部)マッチングからの類推

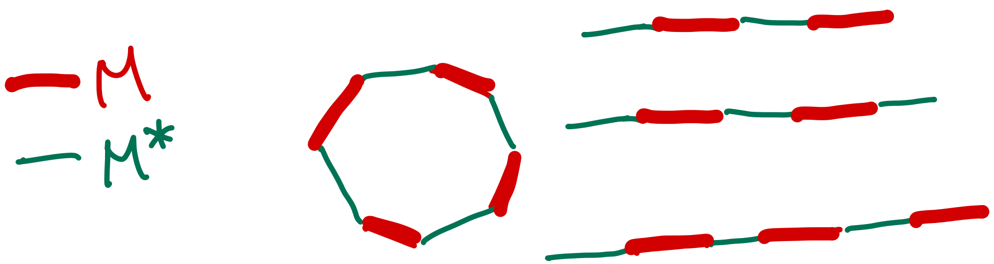
input: $(V, \mathcal{I}_1), (V, \mathcal{I}_2)$
$n := |V|$, $r := \max_{S \in \mathcal{I}_1 \cap \mathcal{I}_2} |S|$
find $S \in \mathcal{I}_1 \cap \mathcal{I}_2$
s.t. $|S| \geq (\frac{2}{3} - \varepsilon) \cdot r$
using $O(n/\varepsilon + r \log r)$ queries

(二部)マッチング

Mに長さ3の増加路が存在しない

⇒ Mは$\frac{2}{3}$近似

∵ 最大マッチングM^*をもってきたとき、



$M \triangle M^*$ は長さ4以上の交互道 or 交互サイクルに分解できる

マトロイド交叉

G(S)に長さ4の増加路が存在しない

⇒ Sは$\frac{2}{3}$近似

本研究における気付き①

[Blikstad'21]で、長さ4の増加路に対応する増加集合をみつける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て、$(\frac{2}{3} - \varepsilon)$近似

アイデア3: (二部)マッチングからの類推

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

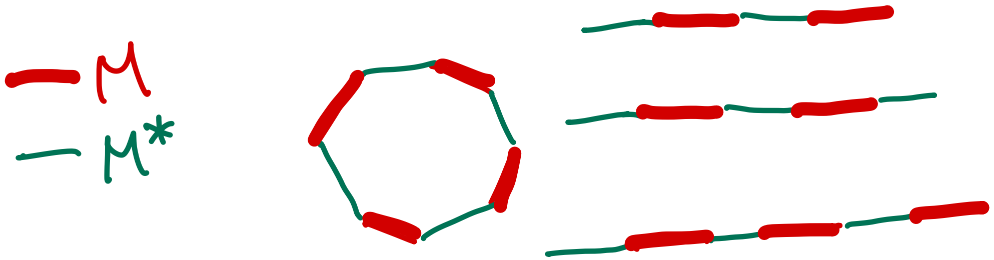
using $O(n/\varepsilon + r \log r)$ queries

(二部)マッチング

Mに長さ3の増加路が存在しない

⇒ Mは $\frac{2}{3}$ 近似

∵ 最大マッチングM^*をもってきたとき、



$M \triangle M^*$ は長さ4以上の交互道 or 交互サイクルに分解できる

マトロイド交叉

G(S)に長さ4の増加路が存在しない

⇒ Sは $\frac{2}{3}$ 近似

本研究における気付き!

[Blikstad'21]で、長さ4の増加路に対応する増加集合をみつける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て、$(\frac{2}{3} - \varepsilon)$近似

長さ4の増加路はεr本しかもうとれない

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 に拡張できる!

本結果の予想外に面白いポイント

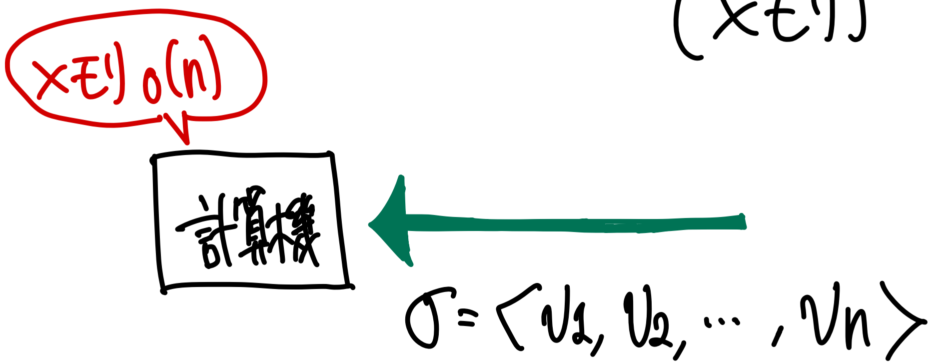
本アルゴリズムはストリーミングアルゴリズムに拡張できる！

入力サイズ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 に拡張できる！

入力サイズよりも小さい空間計算量で問題を解く
（メモリ）

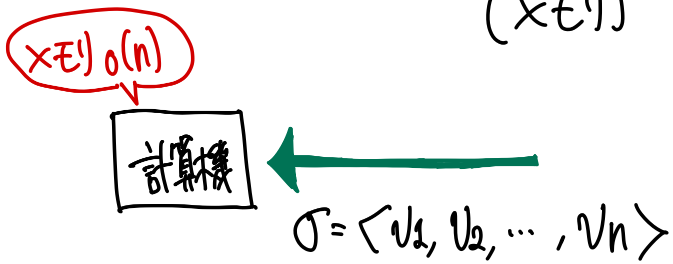


「入力の要素の列を1つずつ受けとる」を 1回もしくは複数回行う

本結果の予想外に面白いポイント

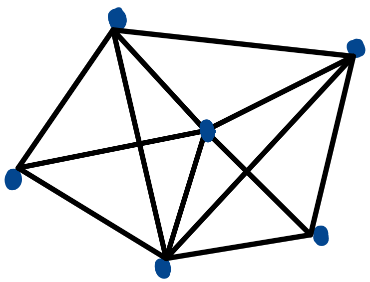
本アルゴリズムは ストリーミングアルゴリズム に拡張できる！

入力サイズよりも小さい空間計算量で問題を解く
(メモリ)



「入力の要素の列を1つずつ受けとる」を 1回もしくは複数回行う

★ 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

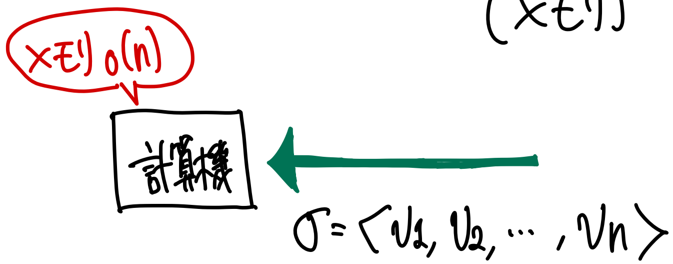


セミストリーミングアルゴリズムがよく研究

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拡張できる！

入力サイズよりも小さい空間計算量で問題を解く
(メモリ)



「入力の要素の列を1つずつ受けとる」を 1回もしくは複数回行う

辺数より小

★ 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

e.g. マッチング問題

セミストリーミングアルゴリズムがよく研究

マトロイド交叉問題 の ストリーミングアルゴリズム

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

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

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

マトロイド交叉問題

の ストリーミングアルゴリズム

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

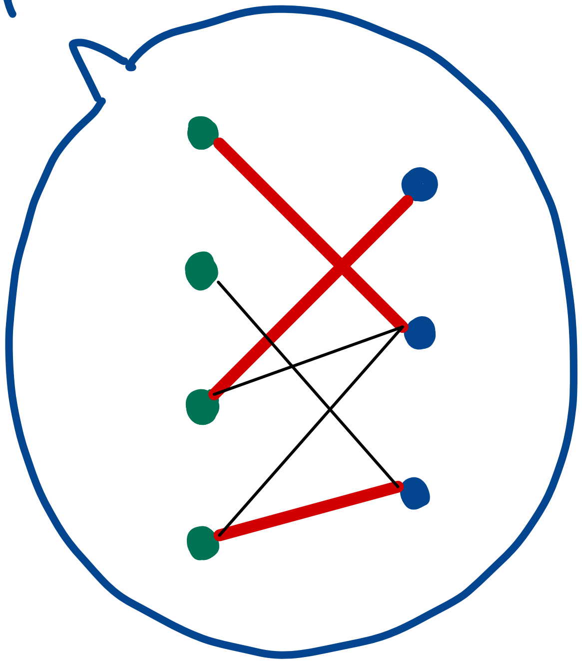
$\max |S| \;\text{ s.t. }\; S \in \mathcal{I}_1 \cap \mathcal{I}_2$

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

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

マッチング問題の
セミストリーミングアルゴリズム

Paz-Schwartzman'17



Bernstein'20

マトロイド交叉問題のストリーミングアルゴリズム

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

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

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

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

マッチング問題のセミストリーミングアルゴリズム

拡張

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

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

Bernstein'20 → Huang-Sellier'24

1回のパス, ランダムオーダー, $(2/3 - \varepsilon)$近似, 空間$\tilde{O}_\varepsilon(r)$

マトロイド交叉問題

の ストリーミングアルゴリズム

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

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

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

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

マッチング問題の
セミストリーミングアルゴリズム

拡張

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

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

Bernstein'20 → Huang-Sellier'24

1回のパス, ランダムオーダー, $(2/3-\varepsilon)$近似, 空間 $\tilde{O}_\varepsilon(r)$

本研究

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

マトロイド交叉問題 の ストリーミングアルゴリズム

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

マッチング問題の セミストリーミングアルゴリズム

拡張

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

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

Bernstein '20 ➡ Huang-Sellier '24

1回のパス, ランダムオーダー, $(2/3 - \varepsilon)$近似, 空間 $\tilde{O}_\varepsilon(r)$

Feigenbaum-Kannan-McGregor-Suri-Zhang '04 ➡ 本研究

グラフのストリーミングの最初の論文

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

結論

- マトロイド交叉問題に対して、
  決定的な $(2/3-\varepsilon)$-近似 $\tilde{O}_\varepsilon(n)$ クエリ アルゴリズム
  を設計
- これを $O(1/\varepsilon)$ 回パス $(2/3-\varepsilon)$ 近似 セミストリーミング アルゴリズム
  に応用：FKMSZ'04の拡張

結論

- マトロイド交叉問題に対して、
  決定的な $(2/3-\varepsilon)$-近似 $\tilde{O}_\varepsilon(n)$ クエリ アルゴリズム
  を設計
- これを $O(1/\varepsilon)$ 回パス $(2/3-\varepsilon)$ 近似 セミストリーミング アルゴリズム
  に応用：FKMSZ'04 の拡張

open questions

- 決定的な $(1-\varepsilon)$ 近似 $\tilde{O}_\varepsilon(n)$ クエリ アルゴリズム
- poly$(1/\varepsilon)$ 回 パス $(1-\varepsilon)$ 近似 セミストリーミング アルゴリズム