Pieter Kleer
Max Planck Institute for Informatics (D1)
Saarland Informatics Campus
January 27, 2020

Lecture 10
Matroid Secretary Problems
Matroids (recap)
Matroids

Generalization of linear independence of vectors in, e.g., $\mathbb{R}^n$. 

Let $E = \{v_1, \ldots, v_k\}$ be collection of vectors $v_i \in \mathbb{R}^n$ for all $i$. 

Assume that $k > n$ and $\text{span}(E) = \mathbb{R}^n$. 

Subset of vectors $X \subseteq E$ is called linearly independent if, for $\gamma_i \in \mathbb{R}$,

$$\sum_{v_i \in X} \gamma_i \cdot v_i = 0 \Rightarrow \gamma_i = 0 \quad \forall i.$$ 

No $v_i \in X$ can be written as linear combination of other vectors. 

Example $E = \{(3, 2), (2, 7), (17, 34), (-4, -2)\}$ 

Is $X = \{v_1, v_2, v_3\}$ independent?

NO, because $v_3 = 3v_1 + 4v_2$. 

Maximal independent sets are bases (of $\mathbb{R}^n$).
Matroids

*Generalization of linear independence of vectors in, e.g., \( \mathbb{R}^n \).*

Let \( E = \{ v_1, \ldots, v_k \} \) be collection of vectors \( v_i \in \mathbb{R}^n \) for all \( i \).
Matroids

*Generalization of linear independence of vectors in, e.g., \( \mathbb{R}^n \).*

Let \( E = \{ v_1, \ldots, v_k \} \) be collection of vectors \( v_i \in \mathbb{R}^n \) for all \( i \).

- Assume that \( k > n \) and \( \text{span}(E) = \mathbb{R}^n \).
Matroids

*Generalization of linear independence of vectors in, e.g., \( \mathbb{R}^n \).*

Let \( E = \{ v_1, \ldots, v_k \} \) be collection of vectors \( v_i \in \mathbb{R}^n \) for all \( i \).

- Assume that \( k > n \) and \( \text{span}(E) = \mathbb{R}^n \).

Subset of vectors \( X \subseteq E \) is called **linearly independent** if, for \( \gamma_i \in \mathbb{R} \),

\[
\sum_{v_i \in X} \gamma_i \cdot v_i = 0 \quad \Rightarrow \quad \gamma_i = 0 \ \forall i.
\]
Matroids

Generalization of linear independence of vectors in, e.g., $\mathbb{R}^n$.

Let $E = \{v_1, \ldots, v_k\}$ be collection of vectors $v_i \in \mathbb{R}^n$ for all $i$.

- Assume that $k > n$ and $\text{span}(E) = \mathbb{R}^n$.

Subset of vectors $X \subseteq E$ is called linearly independent if, for $\gamma_i \in \mathbb{R}$,

$$\sum_{v_i \in X} \gamma_i \cdot v_i = 0 \implies \gamma_i = 0 \ \forall i.$$

- No $v_i \in X$ can be written as linear combination of other vectors.
Matroids

*Generalization of linear independence of vectors in, e.g., $\mathbb{R}^n$.*

Let $E = \{v_1, \ldots, v_k\}$ be collection of vectors $v_i \in \mathbb{R}^n$ for all $i$.
- Assume that $k > n$ and $\text{span}(E) = \mathbb{R}^n$.

Subset of vectors $X \subseteq E$ is called **linearly independent** if, for $\gamma_i \in \mathbb{R}$,

$$\sum_{v_i \in X} \gamma_i \cdot v_i = 0 \implies \gamma_i = 0 \ \forall \ i.$$ 

- No $v_i \in X$ can be written as linear combination of other vectors.

**Example**

\[
E = \{v_1, v_2, v_3, v_2\} = \left\{ \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 7 \end{pmatrix}, \begin{pmatrix} 17 \\ 34 \end{pmatrix}, \begin{pmatrix} -4 \\ -2 \end{pmatrix} \right\}
\]

Is $X = \{v_1, v_2, v_3\}$ independent?
Matroids

Generalization of linear independence of vectors in, e.g., $\mathbb{R}^n$.

Let $E = \{v_1, \ldots, v_k\}$ be collection of vectors $v_i \in \mathbb{R}^n$ for all $i$.

- Assume that $k > n$ and $\text{span}(E) = \mathbb{R}^n$.

Subset of vectors $X \subseteq E$ is called linearly independent if, for $\gamma_i \in \mathbb{R}$,

$$\sum_{v_i \in X} \gamma_i \cdot v_i = 0 \Rightarrow \gamma_i = 0 \ \forall i.$$ 

- No $v_i \in X$ can be written as linear combination of other vectors.

Example

$E = \{v_1, v_2, v_3, v_2\} = \left\{ \begin{pmatrix} 3 \\ 2 \\ 17 \\ -4 \end{pmatrix}, \begin{pmatrix} 2 \\ 7 \\ 34 \\ -2 \end{pmatrix} \right\}$

Is $X = \{v_1, v_2, v_3\}$ independent? NO, because $v_3 = 3v_1 + 4v_2$. 
Matroids

Generalization of linear independence of vectors in, e.g., $\mathbb{R}^n$.

Let $E = \{v_1, \ldots, v_k\}$ be collection of vectors $v_i \in \mathbb{R}^n$ for all $i$.
- Assume that $k > n$ and $\text{span}(E) = \mathbb{R}^n$.

Subset of vectors $X \subseteq E$ is called **linearly independent** if, for $\gamma_i \in \mathbb{R}$,

$$\sum_{v_i \in X} \gamma_i \cdot v_i = 0 \Rightarrow \gamma_i = 0 \ \forall \ i.$$  

- No $v_i \in X$ can be written as linear combination of other vectors.

**Example**

$$E = \{v_1, v_2, v_3, v_2\} = \left\{ \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 7 \end{pmatrix}, \begin{pmatrix} 17 \\ 34 \end{pmatrix}, \begin{pmatrix} -4 \\ -2 \end{pmatrix} \right\}$$

Is $X = \{v_1, v_2, v_3\}$ independent? NO, because $v_3 = 3v_1 + 4v_2$.

- Maximal independent sets are **bases** (of $\mathbb{R}^n$).
Matroid

**Definition (Matroid)**

Set system \( \mathcal{M} = (E, \mathcal{I}) \) with non-empty \( \mathcal{I} \subseteq 2^E = \{ X : X \subseteq E \} \) is **matroid** if it satisfies the following:

1. **Downward-closed:** If \( A \in \mathcal{I} \) and \( B \subseteq A \), then \( B \in \mathcal{I} \).
2. **Augmentation property:** If \( A, C \in \mathcal{I} \) and \( |C| > |A| \), then there exists an element \( e \in C \setminus A \) such that \( A \cup \{ e \} \in \mathcal{I} \).

Sets in \( \mathcal{I} \) are called independent sets.

**Example (Linear matroid)**

Let \( E = \{ v_i : i = 1, \ldots, k \} \subseteq \mathbb{R}^n \) and take \( W \in \mathcal{I} \iff \) vectors in \( W \) are linearly independent.

**Augmentation property:** Note that if \( |C| \geq |A| + 1 \) and every \( v_i \in C \) is a linear combination of vectors in \( A \), then \( \text{span}(C) \subseteq \text{span}(A) \), and hence \( |C| = \dim(\text{span}(C)) \leq \dim(\text{span}(A)) = |A| \), which gives a contradiction.
Definition (Matroid)

Set system $\mathcal{M} = (E, \mathcal{I})$ with non-empty $\mathcal{I} \subseteq 2^E = \{X : X \subseteq E\}$ is matroid if it satisfies the following:

- **Downward-closed**: $A \in \mathcal{I}$ and $B \subseteq A \Rightarrow B \in \mathcal{I}$,
Matroid

**Definition (Matroid)**

Set system $\mathcal{M} = (E, \mathcal{I})$ with non-empty $\mathcal{I} \subseteq 2^E = \{X : X \subseteq E\}$ is a matroid if it satisfies the following:

- **Downward-closed**: $A \in \mathcal{I}$ and $B \subseteq A \Rightarrow B \in \mathcal{I}$,

- **Augmentation property**: $A, C \in \mathcal{I}$ and $|C| > |A| \Rightarrow \exists e \in C \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.
Matroid

**Definition (Matroid)**

Set system \( \mathcal{M} = (E, \mathcal{I}) \) with non-empty \( \mathcal{I} \subseteq 2^E = \{ X : X \subseteq E \} \) is matroid if it satisfies the following:

- **Downward-closed:** \( A \in \mathcal{I} \) and \( B \subseteq A \Rightarrow B \in \mathcal{I} \),
- **Augmentation property:**
  \[ A, C \in \mathcal{I} \text{ and } |C| > |A| \Rightarrow \exists e \in C \setminus A \text{ such that } A \cup \{e\} \in \mathcal{I}. \]
Set system $\mathcal{M} = (E, \mathcal{I})$ with non-empty $\mathcal{I} \subseteq 2^E = \{X : X \subseteq E\}$ is a matroid if it satisfies the following:

- **Downward-closed**: $A \in \mathcal{I}$ and $B \subseteq A \Rightarrow B \in \mathcal{I}$,
- **Augmentation property**: $A, C \in \mathcal{I}$ and $|C| > |A| \Rightarrow \exists e \in C \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.

Sets in $\mathcal{I}$ are called independent sets.
Matroid

**Definition (Matroid)**

Set system $\mathcal{M} = (E, \mathcal{I})$ with non-empty $\mathcal{I} \subseteq 2^E = \{X : X \subseteq E\}$ is **matroid** if it satisfies the following:

- **Downward-closed**: $A \in \mathcal{I}$ and $B \subseteq A \Rightarrow B \in \mathcal{I}$,
- **Augmentation property**: $A, C \in \mathcal{I}$ and $|C| > |A| \Rightarrow \exists e \in C \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.

Sets in $\mathcal{I}$ are called **independent sets**.

**Example (Linear matroid)**

Let $E = \{v_i : i = 1, \ldots, k\} \subseteq \mathbb{R}^n$ and take $W \in \mathcal{I} \iff$ vectors in $W$ are linearly independent.
Matroid

**Definition (Matroid)**

Set system $\mathcal{M} = (E, \mathcal{I})$ with non-empty $\mathcal{I} \subseteq 2^E = \{X : X \subseteq E\}$ is **matroid** if it satisfies the following:

- **Downward-closed**: $A \in \mathcal{I}$ and $B \subseteq A \Rightarrow B \in \mathcal{I}$,
- **Augmentation property**: 
  
  $A, C \in \mathcal{I}$ and $|C| > |A| \Rightarrow \exists e \in C \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.

Sets in $\mathcal{I}$ are called **independent sets**.

**Example (Linear matroid)**

Let $E = \{v_i : i = 1, \ldots, k\} \subseteq \mathbb{R}^n$ and take

$W \in \mathcal{I} \iff$ vectors in $W$ are linearly independent.

- **Augmentation property**: Note that if $|C| \geq |A| + 1$ and every $v_i \in C$ is a linear combination of vectors in $A$, 

---

4 / 31
Matroid

Definition (Matroid)

Set system $\mathcal{M} = (E, \mathcal{I})$ with non-empty $\mathcal{I} \subseteq 2^E = \{X : X \subseteq E\}$ is a matroid if it satisfies the following:

- **Downward-closed**: $A \in \mathcal{I}$ and $B \subseteq A \Rightarrow B \in \mathcal{I}$,
- **Augmentation property**: $A, C \in \mathcal{I}$ and $|C| > |A| \Rightarrow \exists e \in C \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.

Sets in $\mathcal{I}$ are called independent sets.

Example (Linear matroid)

Let $E = \{v_i : i = 1, \ldots, k\} \subseteq \mathbb{R}^n$ and take

$W \in \mathcal{I} \iff$ vectors in $W$ are linearly independent.

- **Augmentation property**: Note that if $|C| \geq |A| + 1$ and every $v_i \in C$ is a linear combination of vectors in $A$, then $\text{span}(C) \subseteq \text{span}(A)$,
Matroid

Definition (Matroid)

Set system $\mathcal{M} = (E, \mathcal{I})$ with non-empty $\mathcal{I} \subseteq 2^E = \{ X : X \subseteq E \}$ is matroid if it satisfies the following:

- **Downward-closed**: $A \in \mathcal{I}$ and $B \subseteq A \Rightarrow B \in \mathcal{I}$,
- **Augmentation property**: $A, C \in \mathcal{I}$ and $|C| > |A| \Rightarrow \exists e \in C \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.

Sets in $\mathcal{I}$ are called independent sets.

Example (Linear matroid)

Let $E = \{ v_i : i = 1, \ldots, k \} \subseteq \mathbb{R}^n$ and take $W \in \mathcal{I} \iff$ vectors in $W$ are linearly independent.

- **Augmentation property**: Note that if $|C| \geq |A| + 1$ and every $v_i \in C$ is a linear combination of vectors in $A$, then $\text{span}(C) \subseteq \text{span}(A)$, and hence $|C| = \text{dim}(\text{span}(C)) \leq \text{dim}(\text{span}(A)) = |A|$,
Matroid

Definition (Matroid)

Set system $\mathcal{M} = (E, \mathcal{I})$ with non-empty $\mathcal{I} \subseteq 2^E = \{X : X \subseteq E\}$ is matroid if it satisfies the following:

- **Downward-closed**: $A \in \mathcal{I}$ and $B \subseteq A \Rightarrow B \in \mathcal{I}$,
- **Augmentation property**:
  \[ A, C \in \mathcal{I} \text{ and } |C| > |A| \Rightarrow \exists e \in C \setminus A \text{ such that } A \cup \{e\} \in \mathcal{I}. \]

Sets in $\mathcal{I}$ are called independent sets.

Example (Linear matroid)

Let $E = \{v_i : i = 1, \ldots, k\} \subseteq \mathbb{R}^n$ and take $W \in \mathcal{I} \iff$ vectors in $W$ are linearly independent.

- **Augmentation property**: Note that if $|C| \geq |A| + 1$ and every $v_i \in C$ is a linear combination of vectors in $A$, then $\text{span}(C) \subseteq \text{span}(A)$, and hence $|C| = \dim(\text{span}(C)) \leq \dim(\text{span}(A)) = |A|$, which gives a contradiction.
Example (Graphic matroid)

Let $G = (V, E)$ be an undirected graph and consider matroid $M = (E, I)$, with ground the edges $E$ of $G$, given by

$$W \in I \iff \text{subgraph with edges of } W \text{ has no cycle}.$$
Example (Graphic matroid)

Let $G = (V, E)$ be undirected graph and consider matroid $\mathcal{M} = (E, \mathcal{I})$, with ground the edges $E$ of $G$, given by

$W \in \mathcal{I} \iff$ subgraph with edges of $W$ has no cycle.
Example (Graphic matroid)

Let $G = (V, E)$ be undirected graph and consider matroid $\mathcal{M} = (E, \mathcal{I})$, with ground the edges $E$ of $G$, given by

$$W \in \mathcal{I} \iff \text{subgraph with edges of } W \text{ has no cycle}.$$
Let $G = (V, E)$ be undirected graph and consider matroid $\mathcal{M} = (E, \mathcal{I})$, with ground the edges $E$ of $G$, given by

$$W \in \mathcal{I} \iff \text{subgraph with edges of } W \text{ has no cycle.}$$
Example (Graphic matroid)

Let $G = (V, E)$ be an undirected graph and consider matroid $\mathcal{M} = (E, \mathcal{I})$, with ground the edges $E$ of $G$, given by

$W \in \mathcal{I} \iff$ subgraph with edges of $W$ has no cycle.
Let $G = (V, E)$ be an undirected graph and consider matroid $M = (E, \mathcal{I})$, with ground the edges $E$ of $G$, given by

$$W \in \mathcal{I} \iff \text{subgraph with edges of } W \text{ has no cycle}.$$
Maximal independents set of a matroid $\mathcal{M} = (E, \mathcal{I})$ are called bases.
Maximal independents set of a matroid $\mathcal{M} = (E, \mathcal{I})$ are called bases.

**Definition (Base)**

An independent set $X \in \mathcal{I}$ is a base if for every $e \in E \setminus X$ it holds that $X + e \notin \mathcal{I}$.
Bases of a matroid

Maximal independents set of a matroid $\mathcal{M} = (E, \mathcal{I})$ are called bases.

**Definition (Base)**

An independent set $X \in \mathcal{I}$ is a base if for every $e \in E \setminus X$ it holds that $X + e \notin \mathcal{I}$, i.e., no element can be added to $X$ while preserving independence.
Bases of a matroid

Maximal independents set of a matroid $\mathcal{M} = (E, \mathcal{I})$ are called bases.

**Definition (Base)**

An independent set $X \in \mathcal{I}$ is a base if for every $e \in E \setminus X$ it holds that $X + e \notin \mathcal{I}$, i.e., no element can be added to $X$ while preserving independence.

**Lemma**

All bases of a given matroid $\mathcal{M}$ have the same cardinality.
Maximal independents set of a matroid $\mathcal{M} = (E, \mathcal{I})$ are called bases.

**Definition (Base)**

An independent set $X \in \mathcal{I}$ is a base if for every $e \in E \setminus X$ it holds that $X + e \notin \mathcal{I}$, i.e., no element can be added to $X$ while preserving independence.

**Lemma**

All bases of a given matroid $\mathcal{M}$ have the same cardinality. This common cardinality $r$ is called the rank of the matroid.
Bases of a matroid

Maximal independents set of a matroid $\mathcal{M} = (E, \mathcal{I})$ are called bases.

**Definition (Base)**
An independent set $X \in \mathcal{I}$ is a base if for every $e \in E \setminus X$ it holds that $X + e \notin \mathcal{I}$, i.e., no element can be added to $X$ while preserving independence.

**Lemma**
All bases of a given matroid $\mathcal{M}$ have the same cardinality. This common cardinality $r$ is called the rank of the matroid.

**Example**
- Bases of graphic matroid on $G = (V, E)$, with $|V| = n$, are spanning trees (when $G$ is connected).
Bases of a matroid

Maximal independents set of a matroid $\mathcal{M} = (E, \mathcal{I})$ are called bases.

**Definition (Base)**

An independent set $X \in \mathcal{I}$ is a base if for every $e \in E \setminus X$ it holds that $X + e \notin \mathcal{I}$, i.e., no element can be added to $X$ while preserving independence.

**Lemma**

All bases of a given matroid $\mathcal{M}$ have the same cardinality. This common cardinality $r$ is called the rank of the matroid.

**Example**

- Bases of graphic matroid on $G = (V, E)$, with $|V| = n$, are spanning trees (when $G$ is connected). Rank is $n - 1$. 
Bases of a matroid

Maximal independents set of a matroid $\mathcal{M} = (E, \mathcal{I})$ are called bases.

**Definition (Base)**

An independent set $X \in \mathcal{I}$ is a base if for every $e \in E \setminus X$ it holds that $X + e \notin \mathcal{I}$, i.e., no element can be added to $X$ while preserving independence.

**Lemma**

All bases of a given matroid $\mathcal{M}$ have the same cardinality. This common cardinality $r$ is called the **rank** of the matroid.

**Example**

- Bases of graphic matroid on $G = (V, E)$, with $|V| = n$, are spanning trees (when $G$ is connected). Rank is $n - 1$. 

![Diagram](6/31)
Consider matroid \( M = (E, I) \) with \( E = \{ e_1, \ldots, e_m \} \).

Rename elements such that \( w_1 \geq w_2 \geq \cdots \geq w_m \geq 0 \).

Greedy algorithm

Set \( X = \emptyset \).

For \( i = 1, \ldots, m \):

If \( X + e_i \in I \), then set \( X \leftarrow X + e_i \).

In other words, greedily add elements while preserving independence.

Example (Graphic matroid)
Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$. rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

Greedy algorithm:

Set $X = \emptyset$.

For $i = 1, \ldots, m$:

- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.

Example (Graphic matroid):

```
a b c d e f  
1 3 2 4 5 6  
7 8 9      
```
Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.

- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$. 

**Greedy algorithm**

Set $X = \emptyset$.

For $i = 1, \ldots, m$:

- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.

**Example (Graphic matroid)**

```
  a   b   c   d   e   f
  1   3   2   4   5   6
  7   8   9
```
Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.

- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$. 

**Greedy algorithm**

Set $X = \emptyset$.

For $i = 1, \ldots, m$:

- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.
Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.

- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

### Greedy algorithm

Set $X = \emptyset$. 

---
Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.

- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

**Greedy algorithm**

Set $X = \emptyset$. For $i = 1, \ldots, m$:

- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.
(Offline) maximum weight independent set

Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.

- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

**Greedy algorithm**

Set $X = \emptyset$. For $i = 1, \ldots, m$:

- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$. 
Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.
- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

**Greedy algorithm**

Set $X = \emptyset$. For $i = 1, \ldots, m$:
- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.

**Example (Graphic matroid)**
(Offline) maximum weight independent set

Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.

- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

**Greedy algorithm**

Set $X = \emptyset$. For $i = 1, \ldots, m$:

- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.

**Example (Graphic matroid)**

![Diagram](attachment:image.png)
Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.
- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

**Greedy algorithm**

Set $X = \emptyset$. For $i = 1, \ldots, m$:
- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.

**Example (Graphic matroid)**

![Diagram of a graphic matroid with weights on edges]
(Offline) maximum weight independent set

Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.

- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

**Greedy algorithm**

Set $X = \emptyset$. For $i = 1, \ldots, m$:

- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.

**Example (Graphic matroid)**

![Graphic matroid diagram]
(Offline) maximum weight independent set

Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.
- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

**Greedy algorithm**

Set $X = \emptyset$. For $i = 1, \ldots, m$:
- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.

**Example (Graphic matroid)**

![Graphic matroid diagram]
Consider matroid $\mathcal{M} = (E, \mathcal{I})$ with $E = \{e_1, \ldots, e_m\}$.

- Rename elements such that $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$.

**Greedy algorithm**

Set $X = \emptyset$. For $i = 1, \ldots, m$:

- If $X + e_i \in \mathcal{I}$, then set $X \leftarrow X + e_i$.

In other words, greedily add elements while preserving independence.

**Example (Graphic matroid)**

![Graphic matroid diagram](image-url)
Matroid secretary problem
Matroid secretary problem

Selecting maximum weight independent set online.

Given is matroid $M = (E, I)$. Set $X = \emptyset$.

Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.

Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.

Decide irrevocably whether to accept or reject it.

Acceptance is only allowed if $X + e$ is independent, i.e., $X + e \in I$.

Matroid secretary problem: Select (online) independent set $X \in I$ of maximum weight.

In the offline setting, $X$ is maximum weight base of the matroid.

Generalization of the secretary problem.

Corresponds to the so-called 1-uniform matroid.

In $k$-uniform matroid, $X \in I$ if and only if $|X| \leq k$. 
Matroid secretary problem

Selecting maximum weight independent set online.
Matroid secretary problem

*Selecting maximum weight independent set online.*

Given is matroid $\mathcal{M} = (E, \mathcal{I})$. Set $X = \emptyset$. 

In the offline setting, $X$ is maximum weight base of the matroid.

Generalization of the secretary problem.

Corresponds to the so-called 1-uniform matroid.

In $k$-uniform matroid, $X \in \mathcal{I}$ if and only if $|X| \leq k$. 
Matroid secretary problem

Selecting maximum weight independent set online.

Given is matroid $\mathcal{M} = (E, \mathcal{I})$. Set $X = \emptyset$.

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
Selecting maximum weight independent set online.

Given is matroid $\mathcal{M} = (E, \mathcal{I})$. Set $X = \emptyset$.

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
Matroid secretary problem

*Selecting maximum weight independent set online.*

Given is matroid $\mathcal{M} = (E, \mathcal{I})$. Set $X = \emptyset$.

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
Matroid secretary problem

**Selecting maximum weight independent set online.**

Given is matroid $\mathcal{M} = (E, \mathcal{I})$. Set $X = \emptyset$.

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is independent, i.e., $X + e \in \mathcal{I}$. 
Selecting maximum weight independent set online.

Given is matroid $\mathcal{M} = (E, \mathcal{I})$. Set $X = \emptyset$.

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is independent, i.e., $X + e \in \mathcal{I}$.

**Matroid secretary problem**: Select (online) independent set $X \in \mathcal{I}$ of maximum weight.

- In the offline setting, $X$ is maximum weight base of the matroid.
Matroid secretary problem

Selecting maximum weight independent set online.

Given is matroid $\mathcal{M} = (E, \mathcal{I})$. Set $X = \emptyset$.

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is independent, i.e., $X + e \in \mathcal{I}$.

**Matroid secretary problem:** Select (online) independent set $X \in \mathcal{I}$ of maximum weight.

- In the offline setting, $X$ is maximum weight base of the matroid.
- Generalization of the secretary problem.
Selecting maximum weight independent set online.

Given is matroid $\mathcal{M} = (E, \mathcal{I})$. Set $X = \emptyset$.

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is independent, i.e., $X + e \in \mathcal{I}$.

**Matroid secretary problem:** Select (online) independent set $X \in \mathcal{I}$ of maximum weight.

- In the offline setting, $X$ is maximum weight base of the matroid.
- Generalization of the secretary problem.
  - Corresponds to the so-called 1-uniform matroid.
Matroid secretary problem

*Selecting maximum weight independent set online.*

Given is matroid $\mathcal{M} = (E, \mathcal{I})$. Set $X = \emptyset$.

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is independent, i.e., $X + e \in \mathcal{I}$.

**Matroid secretary problem:** Select (online) independent set $X \in \mathcal{I}$ of maximum weight.

- In the offline setting, $X$ is maximum weight base of the matroid.
- Generalization of the secretary problem.
  - Corresponds to the so-called 1-uniform matroid.
  - In $k$-uniform matroid, $X \in \mathcal{I}$ if and only if $|X| \leq k$. 
Some literature

They gave $\Omega \left( \frac{1}{\log r} \right)$-approximation.
Remember that $r$ is rank of the matroid.

State of the art: $\Omega \left( \frac{1}{\log \log r} \right)$-approximation.
First by Lachish (2014).
Constant factor approximations known for various special cases
Graphic matroids, $k$-uniform matroids, laminar matroids, transversal matroids, and more.

Open question: Does there exist, for an arbitrary matroid, a constant factor approximation?
Stronger question: Does there exist a $\frac{1}{e}$-approximation?
Would yield (another) generalization of secretary problem.
Some literature

About the matroid secretary problem:

Problem introduced by Babaioff, Immorlica and Kleinberg (2007). They gave $\Omega\left(1 \log r\right)$-approximation. Remember that $r$ is rank of the matroid.

State of the art: $\Omega\left(1 \log \log r\right)$-approximation. First by Lachish (2014). Simpler algorithm by Feldman, Svensson and Zenklusen (2015).

Constant factor approximations known for various special cases: graphic matroids, $k$-uniform matroids, laminar matroids, transversal matroids, and more.

Open question: Does there exist, for an arbitrary matroid, a constant factor approximation? Stronger question: Does there exist a $1/e$-approximation? Would yield (another) generalization of secretary problem.
Some literature

About the matroid secretary problem:
  • Problem introduced by Babaioff, Immorlica and Kleinberg (2007).
Some literature

About the matroid secretary problem:

  - They gave $\Omega \left( \frac{1}{\log(r)} \right)$-approximation.

State of the art:

$\Omega \left( \frac{1}{\log \log(r)} \right)$-approximation.

First by Lachish (2014).


Constant factor approximations known for various special cases: graphic matroids, $k$-uniform matroids, laminar matroids, transversal matroids, and more.

Open question:

Does there exist, for an arbitrary matroid, a constant factor approximation?

Stronger question: Does there exist a $\frac{1}{e}$-approximation?

Would yield (another) generalization of secretary problem.
Some literature

About the matroid secretary problem:
  - They gave $\Omega \left( \frac{1}{\log(r)} \right)$-approximation.
  - Remember that $r$ is rank of the matroid.

State of the art: $\Omega \left( \frac{1}{\log \log(r)} \right)$-approximation. First by Lachish (2014). Simpler algorithm by Feldman, Svensson and Zenklusen (2015).

Constant factor approximations known for various special cases: Graphic matroids, $k$-uniform matroids, laminar matroids, transversal matroids, and more.

Open question: Does there exist, for an arbitrary matroid, a constant factor approximation? Stronger question: Does there exist a $\frac{1}{e}$-approximation? Would yield (another) generalization of secretary problem.
About the matroid secretary problem:

  - They gave $\Omega \left( \frac{1}{\log(r)} \right)$-approximation.
  - Remember that $r$ is rank of the matroid.
- State of the art: $\Omega \left( \frac{1}{\log \log(r)} \right)$-approximation.
About the matroid secretary problem:

  - They gave $\Omega \left( \frac{1}{\log(r)} \right)$-approximation.
  - Remember that $r$ is rank of the matroid.
- State of the art: $\Omega \left( \frac{1}{\log \log(r)} \right)$-approximation.
  - First by Lachish (2014).
Some literature

About the matroid secretary problem:

  - They gave $\Omega \left( \frac{1}{\log(r)} \right)$-approximation.
  - Remember that $r$ is rank of the matroid.
- State of the art: $\Omega \left( \frac{1}{\log \log(r)} \right)$-approximation.
  - First by Lachish (2014).

Constant factor approximations known for various special cases: Graphic matroids, $k$-uniform matroids, laminar matroids, transversal matroids, and more.

Open question: Does there exist, for an arbitrary matroid, a constant factor approximation? Stronger question: Does there exist a $\frac{1}{e}$-approximation? Would yield (another) generalization of secretary problem.
Some literature

About the matroid secretary problem:

  - They gave $\Omega \left( \frac{1}{\log(r)} \right)$-approximation.
  - Remember that $r$ is rank of the matroid.
- State of the art: $\Omega \left( \frac{1}{\log \log(r)} \right)$-approximation.
  - First by Lachish (2014).
- Constant factor approximations known for various special cases
About the matroid secretary problem:
  - They gave $\Omega \left( \frac{1}{\log(r)} \right)$-approximation.
  - Remember that $r$ is rank of the matroid.
- State of the art: $\Omega \left( \frac{1}{\log \log(r)} \right)$-approximation.
  - First by Lachish (2014).
- Constant factor approximations known for various special cases
  - Graphic matroids, $k$-uniform matroids, laminar matroids, transversal matroids, and more.
About the matroid secretary problem:

  - They gave $\Omega\left(\frac{1}{\log(r)}\right)$-approximation.
  - Remember that $r$ is rank of the matroid.

- State of the art: $\Omega\left(\frac{1}{\log \log(r)}\right)$-approximation.
  - First by Lachish (2014).

- Constant factor approximations known for various special cases
  - Graphic matroids, $k$-uniform matroids, laminar matroids, transversal matroids, and more.

Open question: Does there exist, for an arbitrary matroid, a constant factor approximation?
Some literature

About the matroid secretary problem:
  - They gave $\Omega \left( \frac{1}{\log(r)} \right)$-approximation.
  - Remember that $r$ is rank of the matroid.
- State of the art: $\Omega \left( \frac{1}{\log \log(r)} \right)$-approximation.
  - First by Lachish (2014).
- Constant factor approximations known for various special cases
  - Graphic matroids, $k$-uniform matroids, laminar matroids, transversal matroids, and more.

Open question: Does there exist, for an arbitrary matroid, a constant factor approximation?

- Stronger question: Does there exist a $\frac{1}{e}$-approximation?
Some literature

About the matroid secretary problem:
  - They gave $\Omega\left(\frac{1}{\log(r)}\right)$-approximation.
  - Remember that $r$ is rank of the matroid.
- State of the art: $\Omega\left(\frac{1}{\log \log(r)}\right)$-approximation.
  - First by Lachish (2014).
- Constant factor approximations known for various special cases
  - Graphic matroids, $k$-uniform matroids, laminar matroids, transversal matroids, and more.

Open question: Does there exist, for an arbitrary matroid, a constant factor approximation?

- Stronger question: Does there exist a $\frac{1}{e}$-approximation?
- Would yield (another) generalization of secretary problem.
Matroid secretary problem
\[ \Omega \left( \frac{1}{\log(r)} \right) \)-approximation
Random threshold algorithm

Consider (given) matroid $\mathcal{M} = (E, \mathcal{I})$ of rank $r$ with $|E| = m$. 

Phase I (Observation).

For $i = 1, \ldots, m/2$:

- Reject $\sigma(i)$.

Phase II (Selection).

Let $w = \max_{i = 1, \ldots, m/2} w_\sigma(i)$, and choose $j \in \{0, 1, \ldots, \lceil \log(r) \rceil \}$ uniformly at random.

Set threshold $t = w_2^j$.

For $i = m/2 + 1, \ldots, m$:

- Select $\sigma(i)$ if $w_\sigma(i) \geq t$ and $X + \sigma(i) \in \mathcal{I}$. 


Random threshold algorithm

Consider (given) matroid $\mathcal{M} = (E, I)$ of rank $r$ with $|E| = m$.

Random threshold algorithm for arrival order $\sigma$

Set $X = \emptyset$. 

Phase I (Observation).

For $i = 1, \ldots, m/2$:
- Reject $\sigma(i)$.

Phase II (Selection).

Let $w = \max_{i = 1, \ldots, m/2} w(\sigma(i))$,
and choose $j \in \{0, 1, \ldots, \lceil \log(r) \rceil \}$ uniformly at random.
- Set threshold $t = w(j)$.

For $i = m/2 + 1, \ldots, m$:
- Select $\sigma(i)$ if $w(\sigma(i)) \geq t$ and $X + \sigma(i) \in I$. 


Consider (given) matroid $\mathcal{M} = (E, \mathcal{I})$ of rank $r$ with $|E| = m$.

### Random threshold algorithm for arrival order $\sigma$

Set $X = \emptyset$.

**Phase I (Observation).**

- For $i = 1, \ldots, \frac{m}{2}$:
Random threshold algorithm

Consider (given) matroid $\mathcal{M} = (E, \mathcal{I})$ of rank $r$ with $|E| = m$.

**Random threshold algorithm for arrival order $\sigma$**

Set $X = \emptyset$.

**Phase I (Observation).**

- For $i = 1, \ldots, \frac{m}{2}$: Reject $\sigma(i)$.
Random threshold algorithm

Consider (given) matroid $\mathcal{M} = (E, \mathcal{I})$ of rank $r$ with $|E| = m$.

Random threshold algorithm for arrival order $\sigma$

Set $X = \emptyset$.

**Phase I (Observation).**

- For $i = 1, \ldots, \frac{m}{2}$: Reject $\sigma(i)$.

**Phase II (Selection).**
Random threshold algorithm

Consider (given) matroid $\mathcal{M} = (E, \mathcal{I})$ of rank $r$ with $|E| = m$.

Random threshold algorithm for arrival order $\sigma$

Set $X = \emptyset$.

**Phase I (Observation).**
- For $i = 1, \ldots, \frac{m}{2}$: Reject $\sigma(i)$.

**Phase II (Selection).**
- Let $w = \max_{i=1,\ldots,m/2} w_{\sigma}(i)$,
Consider (given) matroid $\mathcal{M} = (E, I)$ of rank $r$ with $|E| = m$.

**Random threshold algorithm for arrival order $\sigma$**

Set $X = \emptyset$.

**Phase I (Observation).**
- For $i = 1, \ldots, \frac{m}{2}$: Reject $\sigma(i)$.

**Phase II (Selection).**
- Let $w = \max_{i=1,\ldots,m/2} w_{\sigma(i)}$, and choose $j \in \{0, 1, \ldots, \lceil \log(r) \rceil \}$ uniformly at random.
Consider (given) matroid $\mathcal{M} = (E, \mathcal{I})$ of rank $r$ with $|E| = m$.

**Random threshold algorithm for arrival order $\sigma$**

Set $X = \emptyset$.

**Phase I (Observation).**

- For $i = 1, \ldots, \frac{m}{2}$: Reject $\sigma(i)$.

**Phase II (Selection).**

- Let $w = \max_{i = 1, \ldots, m/2} w_{\sigma(i)}$, and choose $j \in \{0, 1 \ldots, \lceil \log(r) \rceil \}$ uniformly at random.
- Set threshold

$$t = \frac{w}{2^j}.$$
Consider (given) matroid $\mathcal{M} = (E, \mathcal{I})$ of rank $r$ with $|E| = m$.

**Random threshold algorithm for arrival order $\sigma$**

Set $X = \emptyset$.

**Phase I (Observation).**
- For $i = 1, \ldots, \frac{m}{2}$: Reject $\sigma(i)$.

**Phase II (Selection).**
- Let $w = \max_{i=1,\ldots,m/2} w_{\sigma(i)}$, and choose $j \in \{0, 1 \ldots, \lceil \log(r) \rceil \}$ uniformly at random.
- Set threshold
  $$t = \frac{w}{2^j}.$$  

- For $i = \frac{m}{2} + 1, \ldots, m$: Select $\sigma(i)$ if $w_{\sigma(i)} \geq t$ and $X + \sigma(i) \in \mathcal{I}$. 

Consider graphic matroid as example:
Consider graphic matroid as example:
Weight

Consider graphic matroid as example:
Consider graphic matroid as example:
Consider graphic matroid as example:
Consider graphic matroid as example:
Consider graphic matroid as example:
Consider graphic matroid as example:
Consider graphic matroid as example:
Consider graphic matroid as example:
Consider graphic matroid as example:
Consider graphic matroid as example:
Theorem

The random threshold algorithm is a $1/32$ $(\lceil \log(r) \rceil + 1)$-approximation, where $r$ is the rank of the matroid.

Proof: Consider an optimal base $B^* = \{x_1, \ldots, x_r\}$.

Assume that $w(x_1) > w(x_2) > \cdots > w(x_r)$.

Let $1 \leq q \leq r$ be the largest number for which $w(x_q) \geq w(x_1) / r$.

Let $w = (35, 14, 8, 6, 3, 2, 1)$, so that $r = 7$. Then $w(x_1) r = 5$ and $q = 4$.

Then it holds that $\sum_{i=1}^{q} w(x_i) \geq \frac{1}{2} \cdot w(B^*)$.

Why?

$r \sum_{i=1}^{q} w(x_i) \leq r \sum_{i=1}^{q+1} w(x_i) \leq w(x_1)$.
Theorem

The random threshold algorithm is a $\frac{1}{32(\lfloor \log(r) \rfloor + 1)}$-approximation, where $r$ is the rank of the matroid.

Proof: Consider an optimal base $B^* = \{x_1, \ldots, x_r\}$. Assume that $w(x_1) > w(x_2) > \cdots > w(x_r)$. Let $1 \leq q \leq r$ be the largest number for which $w(x_q) \geq w(x_1)/r$. Let $w = (35, 14, 8, 6, 3, 2, 1)$, so that $r = 7$. Then $w(x_1)r = 5$ and $q = 4$. Then it holds that $q \sum_{i=1}^{q} w(x_i) \geq \frac{1}{2} \cdot w(B^*)$. Why?

$r \sum_{i=1}^{r} w(x_i) \leq r \sum_{i=q+1}^{r} w(x_i) \leq w(x_1)$.
Theorem

The random threshold algorithm is a $\frac{1}{32(\lceil \log (r) \rceil + 1)}$-approximation, where $r$ is the rank of the matroid.

Proof: Consider an optimal base $B^* = \{x_1, \ldots, x_r\}$. 
The random threshold algorithm is a $\frac{1}{32(\lceil \log(r) \rceil + 1)}$-approximation, where $r$ is the rank of the matroid.

Proof: Consider an optimal base $B^* = \{x_1, \ldots, x_r\}$.

- Assume that $w(x_1) > w(x_2) > \cdots > w(x_r)$.
Analysis (sketch)

**Theorem**

The random threshold algorithm is a \( \frac{1}{32(\lceil \log(r) \rceil + 1)} \)-approximation, where \( r \) is the rank of the matroid.

**Proof:** Consider an optimal base \( B^* = \{x_1, \ldots, x_r\} \).

- Assume that \( w(x_1) > w(x_2) > \cdots > w(x_r) \).
- Let \( 1 \leq q \leq r \) be the largest number for which \( w(x_q) \geq w(x_1)/r \).
Analysis (sketch)

**Theorem**

The random threshold algorithm is a \( \frac{1}{32(|\log(r)|+1)} \)-approximation, where \( r \) is the rank of the matroid.

**Proof:** Consider an optimal base \( B^* = \{x_1, \ldots, x_r\} \).
- Assume that \( w(x_1) > w(x_2) > \cdots > w(x_r) \).
- Let \( 1 \leq q \leq r \) be the largest number for which \( w(x_q) \geq w(x_1)/r \).

Let \( w = (35, 14, 8, 6, 3, 2, 1) \), so that \( r = 7 \). Then \( \frac{w(x_1)}{r} = 5 \) and \( q = 4 \).
Analysis (sketch)

Theorem

The random threshold algorithm is a \( \frac{1}{32(\lceil \log(r) \rceil + 1)} \)-approximation, where \( r \) is the rank of the matroid.

Proof: Consider an optimal base \( B^* = \{x_1, \ldots, x_r\} \).

- Assume that \( w(x_1) > w(x_2) > \cdots > w(x_r) \).
- Let \( 1 \leq q \leq r \) be the largest number for which \( w(x_q) \geq w(x_1)/r \).

Let \( w = (35, 14, 8, 6, 3, 2, 1) \), so that \( r = 7 \). Then \( \frac{w(x_1)}{r} = 5 \) and \( q = 4 \).

Then it holds that

\[
\sum_{i=1}^{q} w(x_i) \geq \frac{1}{2} \cdot w(B^*).
\]
Theorem

The random threshold algorithm is a $\frac{1}{32(|\log(r)|+1)}$-approximation, where $r$ is the rank of the matroid.

Proof: Consider an optimal base $B^* = \{x_1, \ldots, x_r\}$.

- Assume that $w(x_1) > w(x_2) > \cdots > w(x_r)$.
- Let $1 \leq q \leq r$ be the largest number for which $w(x_q) \geq w(x_1)/r$.

Let $w = (35, 14, 8, 6, 3, 2, 1)$, so that $r = 7$. Then $\frac{w(x_1)}{r} = 5$ and $q = 4$.

- Then it holds that
  \[ \sum_{i=1}^{q} w(x_i) \geq \frac{1}{2} \cdot w(B^*). \]
- Why?
Theorem

*The random threshold algorithm is a* \( \frac{1}{32(\lceil \log(r) \rceil + 1)} \)-approximation, where \( r \) is the rank of the matroid.*

Proof: Consider an optimal base \( B^* = \{x_1, \ldots, x_r\} \).

- Assume that \( w(x_1) > w(x_2) > \cdots > w(x_r) \).
- Let \( 1 \leq q \leq r \) be the largest number for which \( w(x_q) \geq \frac{w(x_1)}{r} \).

Let \( w = (35, 14, 8, 6, 3, 2, 1) \), so that \( r = 7 \). Then \( \frac{w(x_1)}{r} = 5 \) and \( q = 4 \).

- Then it holds that
  \[
  \sum_{i=1}^{q} w(x_i) \geq \frac{1}{2} \cdot w(B^*).
  \]
- Why?
  \[
  \sum_{i=q+1}^{r} w(x_i) \leq \sum_{i=q+1}^{r} \frac{w(x_1)}{r} \leq w(x_1).
  \]
Remember we may focus on q largest elements in optimal base \( B^* = \{x_1, \ldots, x_r\} \) with \( w(x_1) \geq \cdots \geq w(x_q) \geq \cdots \geq w(x_r) \).
Remember we may focus on q largest elements in optimal base \( B^* = \{x_1, \ldots, x_r\} \) with \( w(x_1) \geq \cdots \geq w(x_q) \geq \cdots \geq w(x_r) \).

Some notation for (random) set \( T \):

### Lemma

Let \( X \) be the set outputted by the random threshold algorithm. For \( i = 1, \ldots, q \), we have (remember \( n_i(B^*) = i \))

\[
E_{\sigma} [m_i(X)] \geq \frac{1}{8} (\lceil \log(r) \rceil + 1) \cdot i.
\]
Remember we may focus on q largest elements in optimal base $B^* = \{x_1, \ldots, x_r\}$ with $w(x_1) \geq \cdots \geq w(x_q) \geq \cdots \geq w(x_r)$.

Some notation for (random) set $T$:

- Let $n_i(T)$ be the number of elements whose weight is at least $w(x_i)$. 

Lemma: Let $X$ be the set outputted by the random threshold algorithm. For $i = 1, \ldots, q$, we have (remember $n_i(B^*) = i$) $E[\sigma \cdot m_i(X)] \geq 1/8 \left( \lceil \log(r) \rceil + 1 \right) \cdot i$. 

We first show how lemma leads to desired approximation guarantee.
Remember we may focus on q largest elements in optimal base $B^* = \{x_1, \ldots, x_r\}$ with $w(x_1) \geq \cdots \geq w(x_q) \geq \cdots \geq w(x_r)$.

Some notation for (random) set $T$:

- Let $n_i(T)$ be the number of elements whose weight is at least $w(x_i)$.
  - Note that $n_i(B^*) = i$. 


Remember we may focus on \( q \) largest elements in optimal base \( B^* = \{x_1, \ldots, x_r\} \) with \( w(x_1) \geq \cdots \geq w(x_q) \geq \cdots \geq w(x_r) \).

Some notation for (random) set \( T \):

- Let \( n_i(T) \) be the number of elements whose weight is at least \( w(x_i) \).
  - Note that \( n_i(B^*) = i \).
- Let \( m_i(T) \) be the number of elements whose weight is at least \( w(x_i)/2 \).
Remember we may focus on \( q \) largest elements in optimal base \( B^* = \{x_1, \ldots, x_r\} \) with \( w(x_1) \geq \cdots \geq w(x_q) \geq \cdots \geq w(x_r) \).

Some notation for (random) set \( T \):

- Let \( n_i(T) \) be the number of elements whose weight is at least \( w(x_i) \).
  - Note that \( n_i(B^*) = i \).
- Let \( m_i(T) \) be the number of elements whose weight is at least \( w(x_i)/2 \).

Lemma

Let \( X \) be the set outputted by the random threshold algorithm.
Remember we may focus on $q$ largest elements in optimal base $B^* = \{x_1, \ldots, x_r\}$ with $w(x_1) \geq \cdots \geq w(x_q) \geq \cdots \geq w(x_r)$. 

Some notation for (random) set $T$:

- Let $n_i(T)$ be the number of elements whose weight is at least $w(x_i)$.
  - Note that $n_i(B^*) = i$.
- Let $m_i(T)$ be the number of elements whose weight is at least $w(x_i)/2$.

**Lemma**

Let $X$ be the set outputted by the random threshold algorithm. For $i = 1, \ldots, q$, we have (remember $n_i(B^*) = i$)
Remember we may focus on $q$ largest elements in optimal base $B^* = \{x_1, \ldots, x_r\}$ with $w(x_1) \geq \cdots \geq w(x_q) \geq \cdots \geq w(x_r)$.

Some notation for (random) set $T$:

- Let $n_i(T)$ be the number of elements whose weight is at least $w(x_i)$.
  - Note that $n_i(B^*) = i$.
- Let $m_i(T)$ be the number of elements whose weight is at least $w(x_i)/2$.

**Lemma**

Let $X$ be the set outputted by the random threshold algorithm. For $i = 1, \ldots, q$, we have (remember $n_i(B^*) = i$)

$$\mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i.$$
Remember we may focus on \( q \) largest elements in optimal base \( B^* = \{x_1, \ldots, x_r\} \) with \( w(x_1) \geq \cdots \geq w(x_q) \geq \cdots \geq w(x_r) \).

Some notation for (random) set \( T \):

- Let \( n_i(T) \) be the number of elements whose weight is at least \( w(x_i) \).
  - Note that \( n_i(B^*) = i \).
- Let \( m_i(T) \) be the number of elements whose weight is at least \( w(x_i)/2 \).

**Lemma**

Let \( X \) be the set outputted by the random threshold algorithm. For \( i = 1, \ldots, q \), we have (remember \( n_i(B^*) = i \))

\[
\mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i.
\]

We first show how lemma leads to desired approximation guarantee.
\[ \mathbb{E}_\sigma [m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i. \]
\[ E_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i. \]

Remember \( m_i(X) \) is number of elements with weight at least \( w(x_i)/2 \) in \( X \).
\[ \mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8([\log(r)] + 1)} \cdot i. \]

**Remember** \( m_i(X) \) **is number of elements with weight at least** \( w(x_i)/2 \) **in** \( X \).  

First note that (remember \( n_i(B^*) = i \))

\[
\sum_{i=1}^{q} w(x_i) = \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1}))n_i(B^*) \right] + w(x_q)n_q(B^*)
\]
\[ \mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i. \]

Remember \( m_i(X) \) is number of elements with weight at least \( w(x_i)/2 \) in \( X \).

First note that (remember \( n_i(B^*) = i \))

\[
\sum_{i=1}^{q} w(x_i) = \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1}))n_i(B^*) \right] + w(x_q)n_q(B^*)
\]

\[
w(X) \geq \frac{1}{2} \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1}))m_i(X) \right] + \frac{1}{2} w(x_q)m_q(X)
\]
\[ \mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i. \]

Remember \( m_i(X) \) is number of elements with weight at least \( w(x_i)/2 \) in \( X \).

First note that (remember \( n_i(B^*) = i \))

\[
\sum_{i=1}^{q} w(x_i) = \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) n_i(B^*) \right] + w(x_q) n_q(B^*)
\]

\[
w(X) \geq \frac{1}{2} \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) m_i(X) \right] + \frac{1}{2} w(x_q) m_q(X)
\]

The approximation guarantee then follows as

\[ \mathbb{E}_\sigma[w(X)] \geq \]
\[ \mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i. \]

Remember \( m_i(X) \) is number of elements with weight at least \( w(x_i)/2 \) in \( X \).

First note that (remember \( n_i(B^*) = i \))

\[
\sum_{i=1}^{q} w(x_i) = \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) n_i(B^*) \right] + w(x_q) n_q(B^*)
\]

\[
w(X) \geq \frac{1}{2} \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) m_i(X) \right] + \frac{1}{2} w(x_q) m_q(X)
\]

The approximation guarantee then follows as

\[
\mathbb{E}_\sigma[w(X)] \geq \frac{1}{2} \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) \mathbb{E}_\sigma[m_i(X)] \right] + \frac{1}{2} w(x_q) \mathbb{E}_\sigma[m_q(X)]
\]
\[ \mathbb{E}_\sigma [m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i. \]

Remember \( m_i(X) \) is number of elements with weight at least \( w(x_i)/2 \) in \( X \).

First note that (remember \( n_i(B^*) = i \))

\[
\sum_{i=1}^{q} w(x_i) = \left[ \sum_{i=1}^{q-1} \left( w(x_i) - w(x_{i+1}) \right) n_i(B^*) \right] + w(x_q) n_q(B^*) \]

\[
w(X) \geq \frac{1}{2} \left[ \sum_{i=1}^{q-1} \left( w(x_i) - w(x_{i+1}) \right) m_i(X) \right] + \frac{1}{2} w(x_q) m_q(X) \]

The approximation guarantee then follows as

\[
\mathbb{E}_\sigma [w(X)] \geq \frac{1}{2} \left[ \sum_{i=1}^{q-1} \left( w(x_i) - w(x_{i+1}) \right) \mathbb{E}_\sigma [m_i(X)] \right] + \frac{1}{2} w(x_q) \mathbb{E}_\sigma [m_q(X)] \]

\[
\geq \frac{1}{16(\lceil \log(r) \rceil + 1)} \left( \left[ \sum_{i=1}^{q-1} \left( w(x_i) - w(x_{i+1}) \right) i \right] + w(x_q) q \right) \]
\[ \mathbb{E}_\sigma [m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i. \]

Remember \( m_i(X) \) is number of elements with weight at least \( w(x_i)/2 \) in \( X \).

First note that (remember \( n_i(B^*) = i \))

\[
\sum_{i=1}^{q} w(x_i) = \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) n_i(B^*) \right] + w(x_q) n_q(B^*)
\]

\[ w(X) \geq \frac{1}{2} \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) m_i(X) \right] + \frac{1}{2} w(x_q) m_q(X) \]

The approximation guarantee then follows as

\[
\mathbb{E}_\sigma [w(X)] \geq \frac{1}{2} \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) \mathbb{E}_\sigma [m_i(X)] \right] + \frac{1}{2} w(x_q) \mathbb{E}_\sigma [m_q(X)]
\]

\[
\geq \frac{1}{16(\lceil \log(r) \rceil + 1)} \left( \left[ \sum_{i=1}^{q-1} (w(x_i) - w(x_{i+1})) i \right] + w(x_q) q \right)
\]

\[ = \frac{1}{16(\lceil \log(r) \rceil + 1)} \sum_{i=1}^{q} w(x_i) \geq \frac{1}{32(\lceil \log(r) \rceil + 1)} w(B^*). \]
Lemma

Let $X$ be set outputted by algorithm. For $i = 1, \ldots, q$,

$$\mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i$$

with $m_i(X)$ number of elements selected with weight at least $w(x_i)/2$. 
Lemma

Let \( X \) be set outputted by algorithm. For \( i = 1, \ldots, q \),

\[
\mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i
\]

with \( m_i(X) \) number of elements selected with weight at least \( w(x_i)/2 \).

Proof: Fix \( i \) and let \( A \) be the event that (both)
Lemma

Let $X$ be set outputted by algorithm. For $i = 1, \ldots, q$,

$$\mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i$$

with $m_i(X)$ number of elements selected with weight at least $w(x_i)/2$.

Proof: Fix $i$ and let $A$ be the event that (both)
- The max. weight element $x_1$ in $B^*$ appears in Phase I, and
Lemma

Let $X$ be set outputted by algorithm. For $i = 1, \ldots, q$,

$$\mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i$$

with $m_i(X)$ number of elements selected with weight at least $w(x_i)/2$.

Proof: Fix $i$ and let $A$ be the event that (both)
- The max. weight element $x_1$ in $B^*$ appears in Phase I, and
- The chosen $j \in \{0, \ldots, \lceil \log(r) \rceil \}$ has the property that

$$w(x_i) \geq t := \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}. \quad (1)$$
Lemma

Let $X$ be set outputted by algorithm. For $i = 1, \ldots, q,$

$$\mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i$$

with $m_i(X)$ number of elements selected with weight at least $w(x_i)/2$.

Proof: Fix $i$ and let $A$ be the event that (both)

- The max. weight element $x_1$ in $B^*$ appears in Phase I, and
- The chosen $j \in \{0, \ldots, \lceil \log(r) \rceil\}$ has the property that

$$w(x_i) \geq t := \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.$$  \hspace{1cm} (1)
Lemma

Let $X$ be set outputted by algorithm. For $i = 1, \ldots, q$, 

$$\mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i$$ 

with $m_i(X)$ number of elements selected with weight at least $w(x_i)/2$.

Proof: Fix $i$ and let $A$ be the event that (both)
- The max. weight element $x_1$ in $B^*$ appears in Phase I, and
- The chosen $j \in \{0, \ldots, \lceil \log(r) \rceil \}$ has the property that

$$w(x_i) \geq t := \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.$$  \hspace{1cm} (1)
Lemma

Let \( X \) be set outputted by algorithm. For \( i = 1, \ldots, q \),

\[
\mathbb{E}_\sigma[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i
\]

with \( m_i(X) \) number of elements selected with weight at least \( w(x_i)/2 \).

Proof: Fix \( i \) and let \( A \) be the event that (both)

- The max. weight element \( x_1 \) in \( B^* \) appears in Phase I, and
- The chosen \( j \in \{0, \ldots, \lceil \log(r) \rceil \} \) has the property that

\[
w(x_i) \geq t := \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.
\]

(1)

(\text{In the example, it could also be that } w(x_1)/r \geq w(x_i)/2.)
Lemma

Let $X$ be set outputted by algorithm. For $i = 1, \ldots, q$,

$$
\mathbb{E}_\sigma [m_i(X)] \geq \frac{1}{8(\lceil \log (r) \rceil + 1)} \cdot i
$$

with $m_i(X)$ number of elements selected with weight at least $w(x_i)/2$.

Proof: Fix $i$ and let $A$ be the event that (both)

- The max. weight element $x_1$ in $B^*$ appears in Phase I, and
- The chosen $j \in \{0, \ldots, \lceil \log (r) \rceil \}$ has the property that

$$w(x_i) \geq t := \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}. \tag{1}$$

- (In the example, it could also be that $w(x_1)/r \geq w(x_i)/2$.)

Choice of $q$ guarantees $w(x_i) \geq w(x_1)/r$, so at least one $j$ satisfies (1):
Lemma

Let $X$ be set outputted by algorithm. For $i = 1, \ldots, q$,

$$\mathbb{E}_{\sigma}[m_i(X)] \geq \frac{1}{8(\lceil \log(r) \rceil + 1)} \cdot i$$

with $m_i(X)$ number of elements selected with weight at least $w(x_i)/2$.

Proof: Fix $i$ and let $A$ be the event that (both)

- The max. weight element $x_1$ in $B^*$ appears in Phase I, and
- The chosen $j \in \{0, \ldots, \lceil \log(r) \rceil\}$ has the property that

$$w(x_i) \geq t := \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.$$  \hspace{1cm} (1)

- (In the example, it could also be that $w(x_1)/r \geq w(x_i)/2$.)

Choice of $q$ guarantees $w(x_i) \geq w(x_1)/r$, so at least one $j$ satisfies (1):

$$\mathbb{P}(A) \geq \frac{1}{2(\lceil \log(r) \rceil + 1)}.$$
With probability $\mathbb{P}(A) \geq 1/(2(\lceil \log(r) \rceil + 1))$, chosen $j$ is such that

$$w(x_i) \geq t = \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.$$
With probability $\mathbb{P}(A) \geq 1/(2(\lceil \log(r) \rceil + 1))$, chosen $j$ is such that

$$w(x_i) \geq t = \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.$$ 

For any $1 \leq j < i$, it holds that $w(x_j) \geq w(x_i) \geq t$. 
With probability $\mathbb{P}(A) \geq 1/(2(\lceil \log(r) \rceil + 1))$, chosen $j$ is such that

$$w(x_i) \geq t = \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.$$ 

For any $1 \leq j < i$, it holds that $w(x_j) \geq w(x_i) \geq t$.

- Every such $x_j$ can potentially be chosen in Phase II as it exceeds the threshold.
With probability $\mathbb{P}(A) \geq 1/(2(\lceil \log(r) \rceil + 1))$, chosen $j$ is such that

$$w(x_i) \geq t = \frac{w(x_1)}{2j} \geq \frac{w(x_i)}{2}. $$

For any $1 \leq j < i$, it holds that $w(x_j) \geq w(x_i) \geq t$.

- Every such $x_j$ can potentially be chosen in Phase II as it exceeds the threshold.
  - It might be rejected still based on the independence criterium.
With probability $\mathbb{P}(A) \geq 1/(2(\lceil \log(r) \rceil + 1))$, chosen $j$ is such that

$$w(x_i) \geq t = \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.$$ 

For any $1 \leq j < i$, it holds that $w(x_j) \geq w(x_i) \geq t$.
- Every such $x_j$ can potentially be chosen in Phase II as it exceeds the threshold.
  - It might be rejected still based on the independence criterium.

For given ordering $\sigma$, let $Y$ be cardinality of maximal size independent set of threshold-exceeding elements that appear in Phase II.
With probability \( \mathbb{P}(A) \geq 1/(2(\lceil \log(r) \rceil + 1)) \), chosen \( j \) is such that

\[
w(x_i) \geq t = \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.
\]

For any \( 1 \leq j < i \), it holds that \( w(x_j) \geq w(x_i) \geq t \).

- Every such \( x_j \) can potentially be chosen in Phase II as it exceeds the threshold.
  - It might be rejected still based on the independence criterium.

For given ordering \( \sigma \), let \( Y \) be cardinality of maximal size independent set of threshold-exceeding elements that appear in Phase II.

- Because the set \( \{x_2, \ldots, x_i\} \) is independent, it follows that

\[
\mathbb{E}_\sigma[Y \mid A] \geq \frac{i - 1}{2} \geq \frac{i}{4}
\]

as every \( x_j \) appears in Phase II with prob. 1/2.
With probability \( \mathbb{P}(A) \geq 1/(2(\lceil \log(r) \rceil + 1)) \), chosen \( j \) is such that

\[
w(x_i) \geq t = \frac{w(x_1)}{2^j} \geq \frac{w(x_i)}{2}.
\]

For any \( 1 \leq j < i \), it holds that \( w(x_j) \geq w(x_i) \geq t \).

- Every such \( x_j \) can potentially be chosen in Phase II as it exceeds the threshold.
  - It might be rejected still based on the independence criterium.

For given ordering \( \sigma \), let \( Y \) be cardinality of maximal size independent set of threshold-exceeding elements that appear in Phase II.

- Because the set \( \{x_2, \ldots, x_i\} \) is independent, it follows that

\[
\mathbb{E}_\sigma[Y | A] \geq \frac{i - 1}{2} \geq \frac{i}{4}
\]

as every \( x_j \) appears in Phase II with prob. \( 1/2 \).

- Here we use the fact that we are considering a matroid!
One might interpret Phase II as just greedily selecting elements that exceed the threshold $t$. Greedy algorithm (with weights equal to 1 for every element) implies that the size of the set chosen is at least $Y$. (Might also argue directly through the augmentation property.)

To conclude, $E_{\sigma}[Y \mid A] \geq \frac{i-1}{2} \geq \frac{i}{4}$. 
One might interpret Phase II as just greedily selecting elements that exceed the threshold $t$. 

$$\mathbb{E}_\sigma [Y \mid A] \geq \frac{i-1}{2} \geq \frac{i}{4}$$
One might interpret Phase II as just greedily selecting elements that exceed the threshold $t$.

- Greedy algorithm (with weights equal to 1 for every element) implies that the size of the set chosen is at least $Y$. 

\[ \mathbb{E}_\sigma [Y \mid A] \geq \frac{i-1}{2} \geq \frac{i}{4} \]
$$\mathbb{E}_\sigma [Y \mid A] \geq \frac{i-1}{2} \geq \frac{i}{4}$$

One might interpret Phase II as just greedily selecting elements that exceed the threshold $t$.

- Greedy algorithm (with weights equal to 1 for every element) implies that the size of the set chosen is at least $Y$.
- ( Might also argue directly through the augmentation property.)
One might interpret Phase II as just greedily selecting elements that exceed the threshold $t$.

- Greedy algorithm (with weights equal to 1 for every element) implies that the size of the set chosen is at least $Y$.
- (Might also argue directly through the augmentation property.)

To conclude,
One might interpret Phase II as just greedily selecting elements that exceed the threshold $t$.

- Greedy algorithm (with weights equal to 1 for every element) implies that the size of the set chosen is at least $Y$.
- (Might also argue directly through the augmentation property.)

To conclude,

$$\mathbb{E}_\sigma[Y \mid A] \geq \frac{i-1}{2} \geq \frac{i}{4}$$
The random threshold algorithm is $\frac{1}{32} \left( \lceil \log(r) \rceil + 1 \right)$-approximation, where $r$ is the rank of the matroid $M = (E, I)$.

Algorithm can be adjusted to the setting where the rank of the matroid is unknown. This makes analysis more complicated.

"Single-threshold" algorithms can never give constant-factor approximation. As shown by Babaioff et al. (2018).

Problem can be turned into a randomized strategyproof mechanism. Elements are bidders that each can receive one "unit of stuff". Matroid constraint on which combination of bidders can be allocated a unit.
The random threshold algorithm is $\frac{1}{32(\lceil \log(r) \rceil + 1)}$-approximation, where $r$ is the rank of the matroid $\mathcal{M} = (E, \mathcal{I})$. 
The random threshold algorithm is $\frac{1}{32(\lceil \log(r) \rceil + 1)}$-approximation, where $r$ is the rank of the matroid $\mathcal{M} = (E, \mathcal{I})$.

- Algorithm can be adjusted to the setting where the rank of the matroid is unknown.
The random threshold algorithm is \(\frac{1}{32(\lceil \log(r) \rceil + 1)}\)-approximation, where \(r\) is the rank of the matroid \(M = (E, I)\).

- Algorithm can be adjusted to the setting where the rank of the matroid is unknown.
  - This makes analysis more complicated.
The random threshold algorithm is \( \frac{1}{32(\lceil \log(r) \rceil + 1)} \)-approximation, where \( r \) is the rank of the matroid \( \mathcal{M} = (E, \mathcal{I}) \).

- Algorithm can be adjusted to the setting where the rank of the matroid is unknown.
  - This makes analysis more complicated.
- “Single-threshold” algorithms can never give constant-factor approximation.
Theorem

The random threshold algorithm is \( \frac{1}{32(\lceil \log(r) \rceil + 1)} \)-approximation, where \( r \) is the rank of the matroid \( M = (E, I) \).

- Algorithm can be adjusted to the setting where the rank of the matroid is unknown.
  - This makes analysis more complicated.
- “Single-threshold” algorithms can never give constant-factor approximation.
  - As shown by Babaioff et al. (2018).
Theorem

The random threshold algorithm is \(\frac{1}{32(\lceil \log(r) \rceil + 1)}\)-approximation, where \(r\) is the rank of the matroid \(\mathcal{M} = (E, I)\).

- Algorithm can be adjusted to the setting where the rank of the matroid is unknown.
  - This makes analysis more complicated.
- “Single-threshold” algorithms can never give constant-factor approximation.
  - As shown by Babaioff et al. (2018).
- Problem can be turned into a randomized strategyproof mechanism.
Theorem

The random threshold algorithm is \( \frac{1}{32(\lceil \log(r) \rceil + 1)} \)-approximation, where \( r \) is the rank of the matroid \( \mathcal{M} = (E, I) \).

- Algorithm can be adjusted to the setting where the rank of the matroid is unknown.
  - This makes analysis more complicated.
- "Single-threshold" algorithms can never give constant-factor approximation.
  - As shown by Babaioff et al. (2018).
- Problem can be turned into a randomized strategyproof mechanism.
  - Elements are bidders that each can receive one "unit of stuff".
The random threshold algorithm is \( \frac{1}{32(\lceil \log(r) \rceil + 1)} \)-approximation, where \( r \) is the rank of the matroid \( M = (E, I) \).

- Algorithm can be adjusted to the setting where the rank of the matroid is unknown.
  - This makes analysis more complicated.
- “Single-threshold” algorithms can never give constant-factor approximation.
  - As shown by Babaioff et al. (2018).
- Problem can be turned into a randomized strategyproof mechanism.
  - Elements are bidders that each can receive one "unit of stuff".
  - Matroid constraint on which combination of bidders can be allocated a unit.
Beyond matroids
Online selection problems

Consider

- Finite set of elements $E = \{e_1, \ldots, e_m\}$.
- Weight function $w : E \rightarrow \mathbb{R}_{\geq 0}$.
- Downward-closed collection $\mathcal{F} \subseteq 2^E = \{X : X \subseteq E\}$.
  - Matroid set system (possibly) without augmentation property.
Online selection problems

Consider
- Finite set of elements $E = \{e_1, \ldots, e_m\}$.
- Weight function $w : E \to \mathbb{R}_{\geq 0}$.
- Downward-closed collection $\mathcal{F} \subseteq 2^E = \{X : X \subseteq E\}$.
  - Matroid set system (possibly) without augmentation property.

Online selection:

Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
Upon arrival of $e \in E$, its weight $w(e) \geq 0$ is revealed.
Decide irrevocably whether to accept or reject it.
Acceptance is only allowed if $X + e \in \mathcal{F}$.
Goal: Select (online) independent set $X \in \mathcal{F}$ of max. weight.

In general, for arbitrary downward-closed set systems, no constant-factor approximation exists.
Online selection problems

Consider

- Finite set of elements $E = \{e_1, \ldots, e_m\}$.
- Weight function $w : E \to \mathbb{R}_{\geq 0}$.
- Downward-closed collection $\mathcal{F} \subseteq 2^E = \{X : X \subseteq E\}$.
  - Matroid set system (possibly) without augmentation property.

Online selection:

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
Online selection problems

Consider
- Finite set of elements $E = \{e_1, \ldots, e_m\}$.
- Weight function $w : E \to \mathbb{R}_{\geq 0}$.
- Downward-closed collection $\mathcal{F} \subseteq 2^E = \{X : X \subseteq E\}$.
  - Matroid set system (possibly) without augmentation property.

Online selection:
- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
Online selection problems

Consider

- Finite set of elements $E = \{e_1, \ldots, e_m\}$.
- Weight function $w : E \to \mathbb{R}_{\geq 0}$.
- Downward-closed collection $\mathcal{F} \subseteq 2^E = \{X : X \subseteq E\}$.
  - Matroid set system (possibly) without augmentation property.

Online selection:

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
Online selection problems

Consider

- Finite set of elements $E = \{e_1, \ldots, e_m\}$.
- Weight function $w : E \rightarrow \mathbb{R}_{\geq 0}$.
- Downward-closed collection $\mathcal{F} \subseteq 2^E = \{X : X \subseteq E\}$.
  - Matroid set system (possibly) without augmentation property.

**Online selection:**

- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e \in \mathcal{F}$.
Online selection problems

Consider

- Finite set of elements \( E = \{ e_1, \ldots, e_m \} \).
- Weight function \( w : E \to \mathbb{R}_{\geq 0} \).
- Downward-closed collection \( \mathcal{F} \subseteq 2^E = \{ X : X \subseteq E \} \).
  - Matroid set system (possibly) without augmentation property.

Online selection:

- Elements in \( E \) arrive in unknown uniform random arrival order \( \sigma \).
- Upon arrival of \( e \in E \), its weight \( w_e \geq 0 \) is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if \( X + e \in \mathcal{F} \).

**Goal:** Select (online) independent set \( X \in \mathcal{F} \) of max. weight.
Online selection problems

Consider
- Finite set of elements $E = \{e_1, \ldots, e_m\}$.
- Weight function $w : E \to \mathbb{R}_{\geq 0}$.
- Downward-closed collection $F \subseteq 2^E = \{X : X \subseteq E\}$.
  - Matroid set system (possibly) without augmentation property.

Online selection:
- Elements in $E$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e \in F$.

**Goal:** Select (online) independent set $X \in F$ of max. weight.

*In general, for arbitrary downward-closed set systems, no constant-factor approximation exists.*
Theorem (Babaioff et al. (2007))

There is no randomized algorithm that, for every downward-closed set system $F = (E, I)$ with $m$ elements and (random) weights in $\{0, 1\}$, obtains an approximation guarantee better than $O(\ln \ln(n) / \ln(n))$.

Proof (very informal): Let $n \geq 0$ be an integer and set $r = \ln(n)$. Let $E = S_1 \cup S_2 \cup \cdots \cup S_k$ be the disjoint union of $k = \lceil n^{r} \rceil$ sets $S_i$.

Every $S_i$ either has $r$ or $r - 1$ elements.

$X \subseteq E$ in independent (i.e., $X \in F$) if and only if $X \subseteq S_i$ for some $i = 1, \ldots, k$.

This set system is (structurally) very “far away” from a matroid.
Online selection for general systems

Theorem (Babaioff et al. (2007))

There is no randomized algorithm that, for every downward-closed set system $\mathcal{F} = (E, \mathcal{I})$ with $m$ elements and (random) weights in $\{0, 1\}$, obtains an approximation guarantee better than $O(\ln \ln(n) / \ln(n))$. 

Proof (very informal): Let $n \geq 0$ be an integer and set $r = \ln(n)$. 

$E = S_1 \cup S_2 \cup \ldots \cup S_k$ is disjoint union of sets $S_i$ with $k = \lceil n^r \rceil$. 

Every $S_i$ either has $r$ or $r-1$ elements. 

$X \subseteq E$ independent (i.e., $X \in \mathcal{F}$) $\iff X \subseteq S_i$ for some $i = 1, \ldots, k$. 

This set system is (structurally) very "far away" from a matroid.
Online selection for general systems

Theorem (Babaioff et al. (2007))

There is no randomized algorithm that, for every downward-closed set system $\mathcal{F} = (E, I)$ with $m$ elements and (random) weights in $\{0, 1\}$, obtains an approximation guarantee better than $O(\ln \ln(n) / \ln(n))$.

Proof (very informal): Let $n \geq 0$ be an integer and set $r = \ln(n)$. 
Online selection for general systems

**Theorem (Babaioff et al. (2007))**

There is no randomized algorithm that, for every downward-closed set system \( \mathcal{F} = (E, \mathcal{I}) \) with \( m \) elements and (random) weights in \( \{0, 1\} \), obtains an approximation guarantee better than \( O(\ln \ln(n)/\ln(n)) \).

**Proof (very informal):** Let \( n \geq 0 \) be an integer and set \( r = \ln(n) \).

- \( E = S_1 \cup S_2 \cup \cdots \cup S_k \) is disjoint union of sets \( S_i \) with \( k = \lceil \frac{n}{r} \rceil \).
Online selection for general systems

**Theorem (Babaioff et al. (2007))**

There is no randomized algorithm that, for every downward-closed set system \( \mathcal{F} = (E, \mathcal{I}) \) with \( m \) elements and (random) weights in \( \{0, 1\} \), obtains an approximation guarantee better than \( O(\ln \ln(n) / \ln(n)) \).

**Proof (very informal):** Let \( n \geq 0 \) be an integer and set \( r = \ln(n) \).

- \( E = S_1 \cup S_2 \cup \cdots \cup S_k \) is disjoint union of sets \( S_i \) with \( k = \left\lceil \frac{n}{r} \right\rceil \).
- Every \( S_i \) either has \( r \) or \( r - 1 \) elements.
Online selection for general systems

**Theorem (Babaioff et al. (2007))**

There is no randomized algorithm that, for every downward-closed set system $\mathcal{F} = (E, I)$ with $m$ elements and (random) weights in $\{0, 1\}$, obtains an approximation guarantee better than $O(\ln \ln(n)/\ln(n))$.

**Proof (very informal):** Let $n \geq 0$ be an integer and set $r = \ln(n)$.

- $E = S_1 \cup S_2 \cup \cdots \cup S_k$ is disjoint union of sets $S_i$ with $k = \lceil \frac{n}{r} \rceil$.
- Every $S_i$ either has $r$ or $r - 1$ elements.

$E = S_1 \cup S_2 \cup \cdots \cup S_k$ is disjoint union of sets $S_i$ with $k = \lceil \frac{n}{r} \rceil$. Every $S_i$ either has $r$ or $r - 1$ elements.
Online selection for general systems

**Theorem (Babaioff et al. (2007))**

There is no randomized algorithm that, for every downward-closed set system $\mathcal{F} = (E, \mathcal{I})$ with $m$ elements and (random) weights in $\{0, 1\}$, obtains an approximation guarantee better than $O(\frac{\ln \ln(n)}{\ln(n)})$.

**Proof (very informal):** Let $n \geq 0$ be an integer and set $r = \ln(n)$.

- $E = S_1 \cup S_2 \cup \cdots \cup S_k$ is disjoint union of sets $S_i$ with $k = \lceil \frac{n}{r} \rceil$.
- Every $S_i$ either has $r$ or $r - 1$ elements.
- $X \subseteq E$ independent (i.e., $X \in \mathcal{F}$) $\iff$ $X \subseteq S_i$ for some $i = 1, \ldots, k$. 
Online selection for general systems

Theorem (Babaioff et al. (2007))

There is no randomized algorithm that, for every downward-closed set system $\mathcal{F} = (E, \mathcal{I})$ with $m$ elements and (random) weights in $\{0, 1\}$, obtains an approximation guarantee better than $O(\ln \ln(n)/\ln(n))$.

Proof (very informal): Let $n \geq 0$ be an integer and set $r = \ln(n)$.

- $E = S_1 \cup S_2 \cup \cdots \cup S_k$ is disjoint union of sets $S_i$ with $k = \lceil \frac{n}{r} \rceil$.
- Every $S_i$ either has $r$ or $r - 1$ elements.
- $X \subseteq E$ independent (i.e., $X \in \mathcal{F}$) $\iff X \subseteq S_i$ for some $i = 1, \ldots, k$.

This set system is (structurally) very “far away” from a matroid.
$X \subseteq E$ independent (i.e., $X \in \mathcal{F}$) $\iff X \subseteq S_i$ for some $i = 1, \ldots, k$. 
$X \subseteq E$ independent (i.e., $X \in \mathcal{F}$) $\iff$ $X \subseteq S_i$ for some $i = 1, \ldots, k$.

\[ |S_i| \in \{r, r - 1\} \]
X \subseteq E \text{ independent (i.e., } X \in \mathcal{F}) \iff X \subseteq S_i \text{ for some } i = 1, \ldots, k.

\begin{center}
\begin{tabular}{ccc}
\hline
$S_1$ & $S_2$ & $S_3$ \\
\hline
\hline
\end{tabular}
\end{center}

$|S_i| \in \{r, r-1\}$

The weights are generated independently for every $e \in E$:
$X \subseteq E$ independent (i.e., $X \in \mathcal{F}$) $\iff$ $X \subseteq S_i$ for some $i = 1, \ldots, k$.

\[
\begin{array}{c|c|c}
S_1 & \vdots & S_3 \\
\hline
\end{array}
\]

$|S_i| \in \{r, r-1\}$

The weights are generated independently for every $e \in E$:

\[
w_e = \begin{cases} 
1 & \text{with probability } \frac{1}{r} \\
0 & \text{with probability } 1 - \frac{1}{r}
\end{cases}
\]
$X \subseteq E$ independent (i.e., $X \in \mathcal{F}$) $\iff X \subseteq S_i$ for some $i = 1, \ldots, k$.

\[
\begin{array}{c|c|c}
S_1 & S_2 & S_3 \\
\hline
\square & \square & \square \ \square \ \square \ \square \\
\end{array}
\]

$|S_i| \in \{r, r - 1\}$

The weights are generated independently for every $e \in E$:

\[
w_e = \begin{cases} 
1 & \text{with probability } \frac{1}{r} \\
0 & \text{with probability } 1 - \frac{1}{r} 
\end{cases}
\]

No (randomized) algorithm $A$ can give constant-factor approximation.
\( X \subseteq E \) independent (i.e., \( X \in \mathcal{F} \)) \iff \( X \subseteq S_i \) for some \( i = 1, \ldots, k \).

\[
\begin{array}{cccc}
& & & \\
S_1 & S_2 & S_3 \\
& & & \square \square \square \square \square \square \square \square \square \\
|S_i| & \in & \{r, r - 1\}
\end{array}
\]

The weights are generated independently for every \( e \in E \):

\[
w_e = \begin{cases} 
1 & \text{with probability } \frac{1}{r} \\
0 & \text{with probability } 1 - \frac{1}{r}
\end{cases}.
\]

No (randomized) algorithm \( A \) can give constant-factor approximation.

What can we achieve online (sketch):
\[ X \subseteq E \text{ independent (i.e., } X \in \mathcal{F}) \iff X \subseteq S_i \text{ for some } i = 1, \ldots, k. \]

The weights are generated independently for every \( e \in E \):
\[
    w_e = \begin{cases} 
        1 & \text{with probability } \frac{1}{r} \\
        0 & \text{with probability } 1 - \frac{1}{r} 
    \end{cases}.
\]

No (randomized) algorithm \( A \) can give constant-factor approximation.

What can we achieve online (sketch):
- As soon as \( A \) selects an element \( e \in S_{i^*} \) (for some \( i^* \)), it can only pick subsequent elements from the same \( S_{i^*} \).
$X \subseteq E$ independent (i.e., $X \in \mathcal{F}$) $\iff X \subseteq S_i$ for some $i = 1, \ldots, k$.

The weights are generated independently for every $e \in E$:

$$w_e = \begin{cases} 
1 & \text{with probability } \frac{1}{r} \\
0 & \text{with probability } 1 - \frac{1}{r} 
\end{cases}.$$

No (randomized) algorithm $\mathcal{A}$ can give constant-factor approximation.

What can we achieve online (sketch):

- As soon as $\mathcal{A}$ selects an element $e \in S_{i^*}$ (for some $i^*$), it can only pick subsequent elements from the same $S_{i^*}$.
- Elements from $S_{i^*}$ that have not yet arrive, have total expected weight at most 1.
\( X \subseteq E \) independent (i.e., \( X \in \mathcal{F} \) ) \iff \( X \subseteq S_i \) for some \( i = 1, \ldots, k \).

\[ |S_i| \in \{ r, r - 1 \} \]

The weights are generated independently for every \( e \in E \):

\[
w_e = \begin{cases} 
1 & \text{with probability } \frac{1}{r} \\
0 & \text{with probability } 1 - \frac{1}{r} 
\end{cases}
\]

No (randomized) algorithm \( A \) can give constant-factor approximation.

What can we achieve online (sketch):

- As soon as \( A \) selects an element \( e \in S_{i^*} \) (for some \( i^* \)), it can only pick subsequent elements from the same \( S_{i^*} \).
- Elements from \( S_{i^*} \) that have not yet arrive, have total expected weight at most 1. (By definition of weights.)
$X \subseteq E$ independent (i.e., $X \in \mathcal{F}$) $\iff X \subseteq S_i$ for some $i = 1, \ldots, k$.

The weights are generated independently for every $e \in E$:

$$w_e = \begin{cases} 1 & \text{with probability } \frac{1}{r} \\ 0 & \text{with probability } 1 - \frac{1}{r} \end{cases}.$$  

No (randomized) algorithm $A$ can give constant-factor approximation.

What can we achieve online (sketch):

- As soon as $A$ selects an element $e \in S_{i^*}$ (for some $i^*$), it can only pick subsequent elements from the same $S_{i^*}$.
- Elements from $S_{i^*}$ that have not yet arrive, have total expected weight at most 1. (By definition of weights.)
- Therefore, set selected by $A$ has weight at most 2 in expectation.
What can we achieve offline (sketch):

Balls-in-bins calculation shows that, in expectation, there will be always at least one $S_i$ that has $\Omega(\ln(n)/\ln\ln(n))$ elements with weight 1.

Offline optimum $\text{OPT} = \Omega(\ln(n)/\ln\ln(n))$ in expectation.

Final remark:

Theorem (Rubinstein, 2016)

There exists an $\Omega(1/\log(n))$-approximation w.r.t. the offline optimum for general downward-closed set system with weights in $\{0, 1\}$. This is then tight up to a factor $\log\log(n)$. 

$|S_i| \in \{r, r - 1\}$
What can we achieve offline (sketch):

$|S_i| \in \{r, r - 1\}$

Theorem (Rubinstein, 2016)
There exists an $\Omega\left(\frac{1}{\log(n)}\right)$-approximation w.r.t. the offline optimum for general downward-closed set system with weights in $\{0, 1\}$. This is then tight up to a factor $\log\log(n)$. 

25 / 31
What can we achieve offline (sketch):

- **Balls-in-bins** calculation shows that, in expectation, there will be always at least one $S_i$ that has $\Omega(\ln(n)/\ln \ln(n))$ elements with weight 1.
What can we achieve offline (sketch):

- **Balls-in-bins** calculation shows that, in expectation, there will be always at least one $S_i$ that has $\Omega(\ln(n)/\ln \ln(n))$ elements with weight 1.

- Offline optimum $\text{OPT} = \Omega(\ln(n)/\ln \ln(n))$ in expectation.
What can we achieve offline (sketch):

- **Balls-in-bins** calculation shows that, in expectation, there will be always at least one $S_i$ that has $\Omega(\ln(n) / \ln \ln(n))$ elements with weight 1.

- Offline optimum $\text{OPT} = \Omega(\ln(n) / \ln \ln(n))$ in expectation.

Final remark:
What can we achieve offline (sketch):

- **Balls-in-bins** calculation shows that, in expectation, there will be always at least one $S_i$ that has $\Omega(\ln(n) / \ln \ln(n))$ elements with weight 1.

- Offline optimum $\text{OPT} = \Omega(\ln(n) / \ln \ln(n))$ in expectation.

Final remark:

**Theorem (Rubinstein, 2016)**

There exists an $\Omega(1 / \log(n))$-approximation w.r.t. the offline optimum for general downward-closed set system with weights in $\{0, 1\}$. 
What can we achieve offline (sketch):

- **Balls-in-bins** calculation shows that, in expectation, there will be always at least one $S_i$ that has $\Omega(\ln(n)/\ln\ln(n))$ elements with weight 1.

- Offline optimum $OPT = \Omega(\ln(n)/\ln\ln(n))$ in expectation.

Final remark:

**Theorem (Rubinstein, 2016)**

*There exists an $\Omega(1/\log(n))$-approximation w.r.t. the offline optimum for general downward-closed set system with weights in $\{0, 1\}$.**

- This is then tight up to a factor $\log\log(n)$. 
Graphic matroid

Korula-Pál algorithm
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$. 
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
Graphic matroid secretary problem

For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

**Online selection (with initially $X = \emptyset$)**

- Edges of (known) graph $G = (V, E)$ arrive in unknown **uniform random arrival order** $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$. 
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
Graphic matroid secretary problem

For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

**Online selection (with initially $X = \emptyset$)**

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown **uniform random arrival** order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
Graphic matroid secretary problem

For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

**Online selection (with initially \( X = \emptyset \))**

- Edges of (known) graph \( G = (V, E) \) arrive in unknown uniform random arrival order \( \sigma \).
- Upon arrival of \( e \in E \), its weight \( w_e \geq 0 \) is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if \( X + e \) is forest of \( G \).
  - That is, \( X + e \) does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
Graphic matroid secretary problem

For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

**Online selection (with initially $X = \emptyset$)**

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown **uniform random arrival order** $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

**Online selection (with initially $X = \emptyset$)**

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
For many special matroids, there exists a constant-factor approximation (often based on a reduction to secretary problems).

Online selection (with initially $X = \emptyset$)

- Edges of (known) graph $G = (V, E)$ arrive in unknown uniform random arrival order $\sigma$.
- Upon arrival of $e \in E$, its weight $w_e \geq 0$ is revealed.
- Decide irrevocably whether to accept or reject it.
  - Acceptance is only allowed if $X + e$ is forest of $G$.
  - That is, $X + e$ does not contain a cycle.
1/(2e)-approximation

Assume that $V = \{1, \ldots, n\}$. 
Assume that $V = \{1, \ldots, n\}$.

**Graphic matroid secretary algorithm for graph $G = (V, E)$**

*Before the edges arrive:*

- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and $A_z = \{(u, z) \in A : \{u, z\} \in E\}$ for $z \in V$.

*When the edges arrive:*

- Run (in parallel) the secretary algorithm on every $A_z$.
- We either orient every edge to its node with highest index, or every edge to its node with lowest index.

$A_z$ is the set of all arcs that are oriented into $z$.

For every $z \in V$ at most one arc from every $A_z$ is selected.
1/(2e)-approximation

Assume that $V = \{1, \ldots, n\}$.

### Graphic matroid secretary algorithm for graph $G = (V, E)$

**Before the edges arrive:**
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ $(i < j)$ with arc $(i, j)$, or
1/(2e)-approximation

Assume that $V = \{1, \ldots, n\}$.

### Graphic matroid secretary algorithm for graph $G = (V, E)$

**Before the edges arrive:**
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\} \ (i < j)$ with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\} \ (i < j)$ with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and $A_z = \{(u, z) \in A : \{u, z\} \in E\}$ for $z \in V$.

When the edges arrive:
- Run (in parallel) the secretary algorithm on every $A_z$.
- We either orient every edge to its node with highest index, or every edge to its node with lowest index.
- $A_z$ is set of all arcs that are oriented into $z$.
- For every $z \in V$ at most one arc from every $A_z$ is selected.
1/(2e)-approximation

Assume that $V = \{1, \ldots, n\}$.

**Graphic matroid secretary algorithm for graph $G = (V, E)$**

*Before the edges arrive:*
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and
1/(2e)-approximation

Assume that $V = \{1, \ldots, n\}$.

**Graphic matroid secretary algorithm for graph $G = (V, E)$**

*Before the edges arrive:*

- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$
Assume that $V = \{1, \ldots, n\}$.

**Graphic matroid secretary algorithm for graph $G = (V, E)$**

**Before the edges arrive:**
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$

**When the edges arrive:**
Assume that $V = \{1, \ldots, n\}$.

**Graphic matroid secretary algorithm for graph $G = (V, E)$**

**Before the edges arrive:**
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$

**When the edges arrive:**
- Run (in parallel) the secretary algorithm on every $A_z$. 
1/(2e)-approximation

Assume that \( V = \{1, \ldots, n\} \).

**Graphic matroid secretary algorithm for graph \( G = (V, E) \)**

*Before the edges arrive:*

- With prob. \( \frac{1}{2} \) replace every edge \( \{i, j\} \) \( (i < j) \) with arc \( (i, j) \), or
- with prob. \( \frac{1}{2} \) replace every edge \( \{i, j\} \) \( (i < j) \) with arc \( (j, i) \).

Let \( A \) be the resulting (random) set of directed arcs, and

\[
A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.
\]

*When the edges arrive:*

- Run (in parallel) the secretary algorithm on every \( A_z \).

- We either orient every edge to its node with highest index, or every edge to its node with lowest index.
1/(2e)-approximation

Assume that $V = \{1, \ldots, n\}$.

Graphic matroid secretary algorithm for graph $G = (V, E)$

**Before the edges arrive:**
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$A_z = \{(u, z) \in A : \{u, z\} \in E\}$ for $z \in V$.

**When the edges arrive:**
- Run (in parallel) the secretary algorithm on every $A_z$.
  - We either orient every edge to its node with highest index, or every edge to its node with lowest index.
  - $A_z$ is set of all arcs that are oriented into $z$. 
1/(2\(e\))-approximation

Assume that \(V = \{1, \ldots, n\}\).

**Graphic matroid secretary algorithm for graph \(G = (V, E)\)**

**Before the edges arrive:**
- With prob. \(\frac{1}{2}\) replace every edge \(\{i, j\} (i < j)\) with arc \((i, j)\), or
- with prob. \(\frac{1}{2}\) replace every edge \(\{i, j\} (i < j)\) with arc \((j, i)\).

Let \(A\) be the resulting (random) set of directed arcs, and

\[
A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.
\]

**When the edges arrive:**
- Run (in parallel) the secretary algorithm on every \(A_z\).
- We either orient every edge to its node with highest index, or every edge to its node with lowest index.
- \(A_z\) is set of all arcs that are oriented into \(z\).
- For every \(z \in V\) at most one arc from every \(A_z\) is selected.
Example (Every edge oriented to lowest index node)

Preprocessing.
Randomly orient every edge to highest index, or every edge to lowest index.

Resulting arcs $A$ are partitioned into sets $A_z$ for $z \in V$.

Running secretary algorithms on the $A_z$.

For all $z \in V$ (in parallel):

Phase I: First observe $\lfloor |A_z| \rfloor$ of edges contained in $A_z$.

Phase II: Select first edge whose weight exceeds best weight seen in Phase I.
Preprocessing.

- Randomly orient every edge to highest index, or every edge to lowest index.
Example (Every edge oriented to lowest index node)

\[
\begin{align*}
A_1 &= \{(6, 1), (5, 1), (2, 1)\} \\
A_2 &= \{(5, 2), (4, 2), (3, 2)\} \\
A_3 &= \{(4, 3)\} \\
A_4 &= \{(5, 4)\} \\
A_5 &= \{(6, 5)\} \\
A_6 &= \emptyset
\end{align*}
\]

Preprocessing.
- Randomly orient every edge to highest index, or every edge to lowest index.
- Resulting arcs $A$ are partitioned into sets $A_z$ for $z \in V$. 
Preprocessing.

- Randomly orient every edge to highest index, or every edge to lowest index.
- Resulting arcs $A$ are partitioned into sets $A_z$ for $z \in V$.

Running secretary algorithms on the $A_z$. 
Preprocessing.

- Randomly orient every edge to highest index, or every edge to lowest index.
- Resulting arcs $A$ are partitioned into sets $A_z$ for $z \in V$.

Running secretary algorithms on the $A_z$. For all $z \in V$ (in parallel):
Preprocessing.

- Randomly orient every edge to highest index, or every edge to lowest index.
- Resulting arcs $A$ are partitioned into sets $A_z$ for $z \in V$.

Running secretary algorithms on the $A_z$. For all $z \in V$ (in parallel):

- Phase I: First observe $\left\lfloor \frac{|A_z|}{e} \right\rfloor$ of edges contained in $A_z$. 

Example (Every edge oriented to lowest index node)

$$A_1 = \{(6, 1), (5, 1), (2, 1)\}$$  
$$A_2 = \{(5, 2), (4, 2), (3, 2)\}$$  
$$A_3 = \{(4, 3)\}$$  
$$A_4 = \{(5, 4)\}$$  
$$A_5 = \{(6, 5)\}$$  
$$A_6 = \emptyset$$
Preprocessing.

- Randomly orient every edge to highest index, or every edge to lowest index.
- Resulting arcs $A$ are partitioned into sets $A_z$ for $z \in V$.

Running secretary algorithms on the $A_z$. For all $z \in V$ (in parallel):

- Phase I: First observe $\left\lfloor \frac{|A_z|}{e} \right\rfloor$ of edges contained in $A_z$.
- Phase II: Select first edge whose weight exceeds best weight seen in Phase I.
Example (Every edge oriented to lowest index node)

\[
A_1 = \{(6, 1), (5, 1), (2, 1)\}
A_2 = \{(5, 2), (4, 2), (3, 2)\}
A_3 = \{(4, 3)\}
A_4 = \{(5, 4)\}
A_5 = \{(6, 5)\}
A_6 = \{\emptyset\}
\]

Preprocessing.
- Randomly orient every edge to highest index, or every edge to lowest index.
- Resulting arcs $A$ are partitioned into sets $A_z$ for $z \in V$.

Running secretary algorithms on the $A_z$. For all $z \in V$ (in parallel):
- Phase I: First observe $\left\lfloor \frac{|A_z|}{e} \right\rfloor$ of edges contained in $A_z$.
- Phase II: Select first edge whose weight exceeds best weight seen in Phase I.
Graphic matroid secretary algorithm for graph $G = (V, E)$

**Before the edges arrive:**
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$  

**When the edges arrive:**
- Run (in parallel) the secretary algorithm on every $A_z$. 

High-level steps to show it is $\frac{1}{2}e$-approximation:
- First show that indeed forest is outputted. That is, an independent set of the graphic matroid.
- Then compare to (oriented) offline max. weight spanning tree.
- Give bound on expected contribution per node:
  - Factor $\frac{1}{2}$ is result of (randomly) orienting edges.
  - Factor $\frac{1}{e}$ is result of running (parallel) secretary algorithms.
Graphic matroid secretary algorithm for graph \( G = (V, E) \)

Before the edges arrive:
- With prob. \( \frac{1}{2} \) replace every edge \( \{i, j\} \) \((i < j)\) with arc \((i, j)\), or
- with prob. \( \frac{1}{2} \) replace every edge \( \{i, j\} \) \((i < j)\) with arc \((j, i)\).

Let \( A \) be the resulting (random) set of directed arcs, and
\[
A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.
\]

When the edges arrive:
- Run (in parallel) the secretary algorithm on every \( A_z \).

High-level steps to show it is \( \frac{1}{2e} \)-approximation:
Graphic matroid secretary algorithm for graph $G = (V, E)$

Before the edges arrive:
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$  

When the edges arrive:
- Run (in parallel) the secretary algorithm on every $A_z$.

High-level steps to show it is $\frac{1}{2e}$-approximation:
- First show that indeed forest is outputted.
Before the edges arrive:

- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$  

When the edges arrive:

- Run (in parallel) the secretary algorithm on every $A_z$.

High-level steps to show it is $\frac{1}{2e}$-approximation:

- First show that indeed forest is outputted.
  - That is, an independent set of the graphic matroid.
Graphic matroid secretary algorithm for graph $G = (V, E)$

**Before the edges arrive:**
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$

**When the edges arrive:**
- Run (in parallel) the secretary algorithm on every $A_z$.

High-level steps to show it is $\frac{1}{2e}$-approximation:
- First show that indeed forest is outputted.
  - That is, an independent set of the graphic matroid.
- Then compare to (oriented) offline max. weight spanning tree.
Graphic matroid secretary algorithm for graph $G = (V, E)$

**Before the edges arrive:**

- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ $(i < j)$ with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ $(i < j)$ with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$

**When the edges arrive:**

- Run (in parallel) the secretary algorithm on every $A_z$.

**High-level steps to show it is $\frac{1}{2e}$-approximation:**

- First show that indeed forest is outputted.
  - That is, an independent set of the graphic matroid.
- Then compare to (oriented) offline max. weight spanning tree.
- Give bound on expected contribution per node:
Graphic matroid secretary algorithm for graph $G = (V, E)$

Before the edges arrive:
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$

When the edges arrive:
- Run (in parallel) the secretary algorithm on every $A_z$.

High-level steps to show it is $\frac{1}{2e}$-approximation:
- First show that indeed forest is outputted.
  - That is, an independent set of the graphic matroid.
- Then compare to (oriented) offline max. weight spanning tree.
- Give bound on expected contribution per node:
  - Factor $\frac{1}{2}$ is result of (randomly) orienting edges.
Graphic matroid secretary algorithm for graph $G = (V, E)$

Before the edges arrive:
- With prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(i, j)$, or
- with prob. $\frac{1}{2}$ replace every edge $\{i, j\}$ ($i < j$) with arc $(j, i)$.

Let $A$ be the resulting (random) set of directed arcs, and

$$A_z = \{(u, z) \in A : \{u, z\} \in E\} \text{ for } z \in V.$$

When the edges arrive:
- Run (in parallel) the secretary algorithm on every $A_z$.

High-level steps to show it is $\frac{1}{2e}$-approximation:
- First show that indeed forest is outputted.
  - That is, an independent set of the graphic matroid.
- Then compare to (oriented) offline max. weight spanning tree.
- Give bound on expected contribution per node:
  - Factor $\frac{1}{2}$ is result of (randomly) orienting edges.
  - Factor $\frac{1}{e}$ is result of running (parallel) secretary algorithms.
By now, $\frac{1}{4}$-approximation for graphic matroid secretary problem is known.
Final remarks

By now, $\frac{1}{4}$-approximation for graphic matroid secretary problem is known.

By now, $\frac{1}{4}$-approximation for graphic matroid secretary problem is known.

- Proof uses similar algorithm and analysis as that of Kesselheim et al. (2013) for online bipartite matching.
Final remarks

By now, $\frac{1}{4}$-approximation for graphic matroid secretary problem is known.

- Proof uses similar algorithm and analysis as that of Kesselheim et al. (2013) for online bipartite matching.
- Technique also applies to other special cases of matroids.
By now, $\frac{1}{4}$-approximation for graphic matroid secretary problem is known.

- Proof uses similar algorithm and analysis as that of Kesselheim et al. (2013) for online bipartite matching.
- Technique also applies to other special cases of matroids.

Is there $\frac{1}{e}$-approximation for graphic matroid secretary problem?