# Chapter 2
# Greedy Algorithms

## Algorithm Theory
## WS 2013/14

Fabian Kuhn

# Matroids

- Same, but more abstract…

**Matroid: pair $(\boldsymbol{E}, \boldsymbol{I})$**

- $\boldsymbol{E}$: set, called the **ground set**
- $\boldsymbol{I}$: finite family of finite subsets of $E$ (i.e., $I \subseteq 2^E$), called **independent sets**

$(E, I)$ needs to satisfy 3 properties:

1. Empty set is independent, i.e., $\emptyset \in \boldsymbol{I}$ (implies that $I \neq \emptyset$)
2. **Hereditary property**: For all $A \subseteq I$ and all $A' \subseteq A$,

   **if $\boldsymbol{A} \in \boldsymbol{I}$, then also $\boldsymbol{A'} \in \boldsymbol{I}$**

3. **Augmentation / Independent set exchange property:**
   If $A, B \in I$ and $|A| > |B|$, there exists $x \in A \setminus B$ such that

   $$\mathbf{B}' := \boldsymbol{B} \cup \{\boldsymbol{x}\} \in \boldsymbol{I}$$

# Matroids and Greedy Algorithms

**Weighted matroid**: each $e \in E$ has a weight $w(e) > 0$

**Goal:** find maximum weight independent set

**Greedy algorithm:**

1. Start with $S = \emptyset$
2. Add max. weight $e \in E \setminus S$ to $S$ such that $S \cup \{e\} \in I$

**Claim:** greedy algorithm computes optimal solution

# Greedy is Optimal

if $\forall i \; w(x_i) \geq w(y_i)$
then $S$ has at least the weight of $A$

- $S$: greedy solution $\qquad$ $A$: any other solution

$S \subseteq E$, $S$: ind. set $\qquad$ $A \subseteq E$, $A$ ind. set

$S = \{x_1, x_2, \dots, x_{|S|}\}$ $\qquad$ $w(x_1) \geq w(x_2) \geq \dots \geq w(x_{|S|})$

$A = \{y_1, y_2, \dots, y_{|A|}\}$ $\qquad$ $w(y_1) \geq w(y_2) \geq \dots \geq w(x_{|A|})$

$|S| \geq |A|$ : assume $|S| < |A|$ $\quad$ exch. prop. $\exists y_i \in A \setminus S$ s.t. $S \cup \{y_i\}$ is in $I$

every subset is also in $I$

for contradiction, assume that $S$ has smaller weight than $A$

$k$ : smallest index s.t. $w(x_k) < w(y_k)$ $\quad$ ($k$ exists)

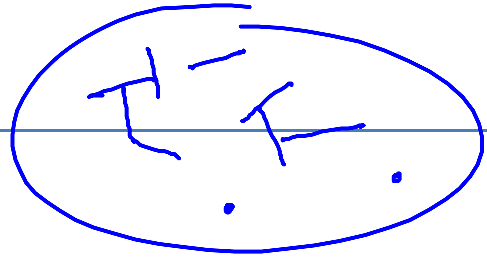
$S' = \{x_1, \dots, x_{k-1}\}$, $\quad A' = \{y_1, \dots, y_k\}$

$y_j$

augm. prop. $\exists j \in \{1, \dots, k\}$ s.t. $S' \cup \{y_j\} \in I$ (and $y_j \notin S'$)

$w(y_j) \geq w(y_k) > w(x_k)$ $\qquad$ greedy adds $y_j$ $\Rightarrow$ $S$ is opt.

# Matroids: Examples



**Forests of a graph $G = (V, E)$:**

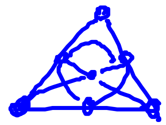
- forest $F$: subgraph with no cycles (i.e., $F \subseteq E$)
- $\mathcal{F}$: set of all forests → $(E, \mathcal{F})$ is a matroid
- Greedy algorithm gives maximum weight forest (equivalent to MST problem)

**Bicircular matroid of a graph $G = (V, E)$:**

- $\mathcal{B}$: set of edges such that every connected subset has $\leq 1$ cycle
- $(E, \mathcal{B})$ is a matroid → greedy gives max. weight such subgraph

**Linearly independent vectors:**

- Vector space $V$, $E$: finite set of vectors, $I$: sets of lin. indep. vect.
- Fano matroid can be defined like that



# Greedoid

- Matroids can be generalized even more
- Relax hereditary property:

  Replace $A' \subseteq A \subseteq I \quad \Longrightarrow \quad A' \in I$

  by $\emptyset \neq A \subseteq I \quad \Longrightarrow \quad \exists a \in A, \text{ s.t. } A \setminus \{a\} \in I$

- Exchange property holds as before
- Under certain conditions on the weights, greedy is optimal for computing the max. weight $A \in I$ of a greedoid.
  - Additional conditions automatically satisfied by hereditary property
- More general than matroids