Chapter 2
Greedy Algorithms

Algorithm Theory
WS 2017/18

Fabian Kuhn

Greedy Algorithms

- No clear definition, but essentially:

In each step make the choice that looks best at the moment!

- Depending on problem, greedy algorithms can give
  - Optimal solutions
  - Close to optimal solutions
  - No (reasonable) solutions at all
- If it works, very interesting approach!
  - And we might even learn something about the structure of the problem

Goal: Improve understanding where it works (mostly by examples)

Exchange Argument

- General approach that often works to analyze greedy algorithms

- Start with any solution
- Define basic exchange step that allows to transform solution into a new solution that is not worse
- Show that exchange step move solution closer to the solution produced by the greedy algorithm
- Number of exchange steps to reach greedy solution should be finite…

Another Exchange Argument Example

- **Minimum spanning tree (MST)** problem
  - Classic graph-theoretic optimization problem

- **Given**: weighted graph
- **Goal**: spanning tree with min. total weight

- Several greedy algorithms work

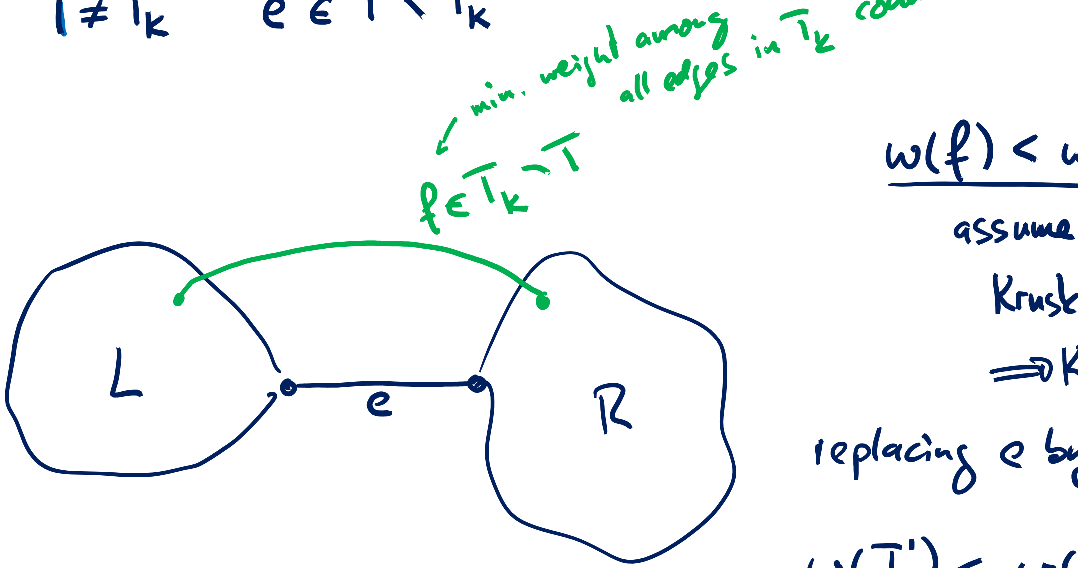
- Kruskal's algorithm:
  - Start with empty edge set
  - As long as we do not have a spanning tree:
    add minimum weight edge that doesn't close a cycle

Kruskal is Optimal

- Basic exchange step: swap to edges to get from tree T to tree T^* T_K
  - Swap out edge not in Kruskal tree, swap in edge in Kruskal tree
  - Swapping does not increase total weight
- For simplicity, assume, weights are unique:

T : any spanning tree T_K : Kruskal tree

$T \neq T_K$ $e \in T \setminus T_K$



$w(f) < w(e)$:

assume otherwise:

Kruskal considers e before f

$\Rightarrow$ Kruskal would add e

replacing e by f $\Rightarrow$ new spanning tree T'

$w(T') < w(T)$

Matroids

$E = \{1, 2, 3, 4\}$

$I = \{\emptyset, \{1\}, \dots, \{4\}, \{1,2\}, \dots, \{3,4\}\}$

- Same, but more abstract…

set system

Matroid: **pair (E, I)**

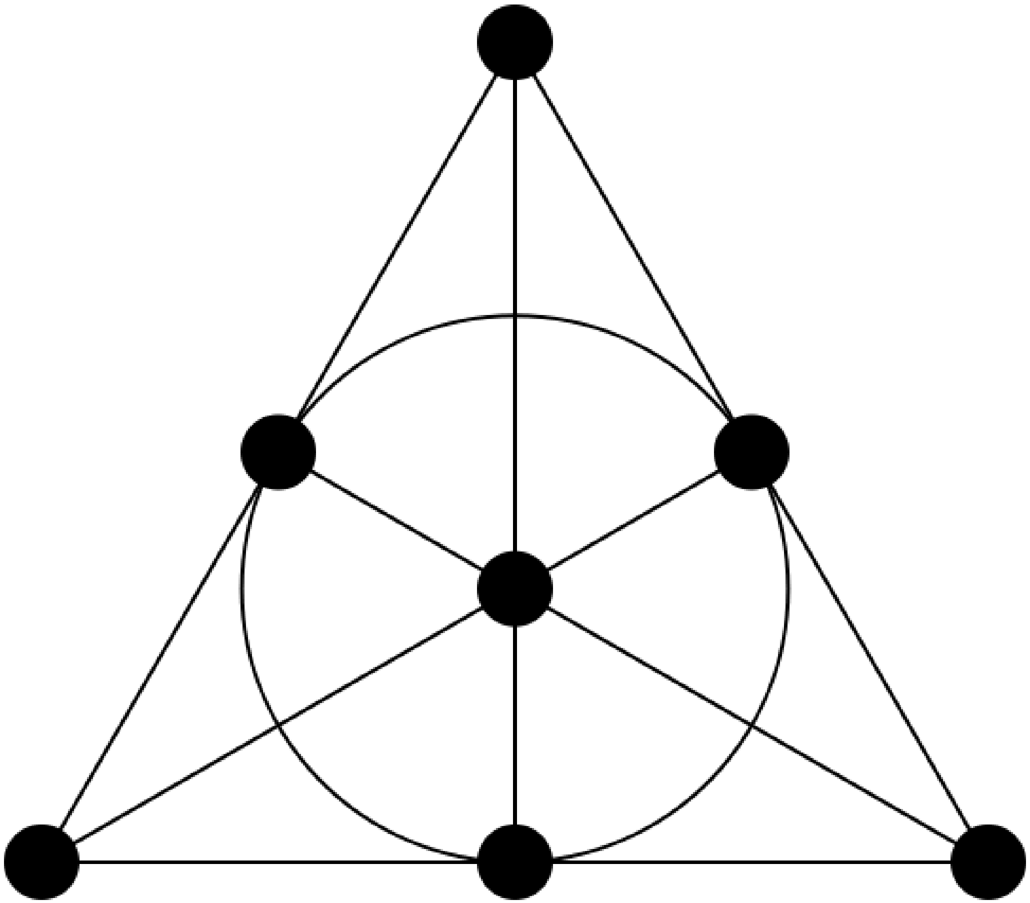
- $\boldsymbol{E}$: set, called the **ground set**
- $\boldsymbol{I}$: finite family of finite subsets of E (i.e., $I \subseteq 2^E$),
  called **independent sets**

(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 $A \subseteq E$ and all $A' \subseteq A$,
   if $A \in I$, then also $A' \in I$
3. **Augmentation / Independent set exchange property:**
   If $A, B \in I$ and $|A| > |B|$, there exists $x \in A \setminus B$ such that
   $$\mathbf{B}' := \boldsymbol{B} \cup \{\boldsymbol{x}\} \in \boldsymbol{I}$$

Example

- Fano matroid:
  - Smallest finite projective plane of order 2…



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 $(E, \mathcal{I})$

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

  $S \subseteq E,\ S \in \mathcal{I}$ $\quad$ $A \subseteq E,\ A \in \mathcal{I}$

$|S| \geq |A|$:

for contradiction, assume $|A| > |S|$: exch. prop. : $\exists x \in A \setminus S$ s.t. $S \cup \{x\} \in \mathcal{I}$

greedy would add x

$w(S) \geq w(A)$:

for contradiction, assume $w(S) < w(A)$

$S = \{x_1, x_2, \dots, x_s\}$ $\quad w(x_1) \geq w(x_2) \geq \dots \geq w(x_s)$

$A = \{y_1, y_2, \dots, y_a\}$ $\quad w(y_1) \geq w(y_2) \geq \dots \geq w(y_a)$

will show that (*)

$\forall i \in \{1, \dots, a\} : w(x_i) \geq w(y_i)$

$\hookrightarrow w(S) \geq w(A)$

$\neg(*) \Rightarrow$ there is a smallest $k \leq a$ s.t. $w(x_k) < w(y_k)$

$S' = \{x_1, \dots, x_{k-1}\}$ $\quad$ augm. prop. : $\exists y \in A' \setminus S' : S' \cup \{y\} \in \mathcal{I}$

$A' = \{y_1, \dots, y_k\}$

$w(y) \geq w(y_k) > w(x_k)$

greedy considers y before x_k

greedy would add y $\quad$ ↯

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 ≤ 1 cycle
- $(E, \mathcal{B})$ is a matroid → greedy gives max. weight such subgraph

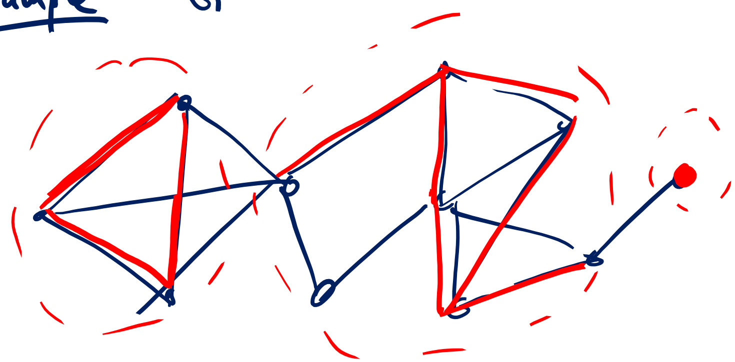
Linearly independent vectors:

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

Bicircular Matroid

$G=(V,E)$, matroid $(E,\mathcal{B})$ $S\subseteq E$: $S\in\mathcal{B}$ iff all comp. of (V,S) have ≤ 1 cycle

example G



Claim: $(E,\mathcal{B})$ is a matroid

Proof: We need to show that $(E,\mathcal{B})$ satisfies properties 1, 2, and 3

Prop 1: $\emptyset\in\mathcal{B}$ $(V,\emptyset)$ has no cycles ✓

Prop 2: $A\in\mathcal{B}\Rightarrow A'\subseteq A: A'\in\mathcal{B}$ ✓

Exch. prop (3) edge set $A\in\mathcal{B}$, $C\in\mathcal{B}$

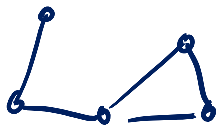
$|C|>|A|\Rightarrow\exists e\in C\setminus A$ s.t. $A\cup\{e\}\in\mathcal{B}$

(V,A) (V,C) ← every comp. ≤ 1 cycle

Bicircular Matroid

Components with ≤ 1 cycle

Comp. has $k \geq 1$ nodes



$\Longrightarrow$ #edges:

no cycle: $= k-1$

1 cycle: $= k$

(V,S) : $S \in \mathcal{B}$

$|S| \leq n$

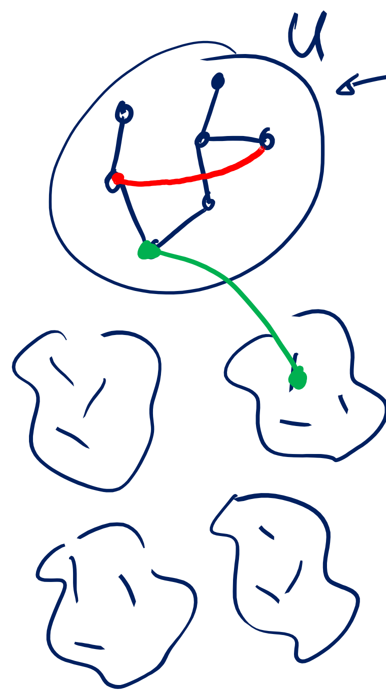
$|S| = n \Longleftrightarrow$ all comp. have a cycle

(V,A), (V,C) $\quad (A, C \in \mathcal{B})$

$|A| < |C| \Longrightarrow |A| \leq n-1$

(V,A)

↳ there is a component $U \subseteq V$ with no cycle



show: can add an edge of $C - A$ to A

Case 1: C has an edge connecting U to $V \setminus U$

case 2: $C \setminus A$ contains an edge between two nodes in U

case 3: consider matroid defined by $V \setminus U$

Greedoid

- Matroids can be generalized even more

- Relax hereditary property:

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

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

- Augmentation 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