Greedy & Matroids

Goal: Find a large "class of problems" where greedy works!

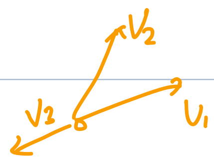
Def. A set system $(U, \mathcal{L})$ is a matroid if — collection of subsets of U.

1. U is finite.
2. If $A \subseteq B$ and $B \in \mathcal{L}$, then $A \in \mathcal{L}$. (hereditary)
3. If $A, B \in \mathcal{L}$ and $|A| < |B|$, $\exists e \in B \setminus A$ s.t. $A \cup \{e\} \in \mathcal{L}$. (exchange).

If $S \in \mathcal{L}$, say S is "independent"

Examples. ① $U = \{v_1, \dots, v_n\}$ where each $v_i \in \mathbb{R}^d$. (linear matroid)

$S \in \mathcal{L} \iff S$ is linearly independent.



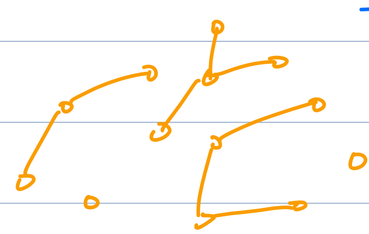
$\{v_1, v_3\} \notin \mathcal{L}$.

$\{v_1, v_2\} \in \mathcal{L}$.

② Fix an undirected graph $G = (V, E)$ (graphic matroid)

$U = E$.

$S \in \mathcal{L}$ iff subgraph (V, S) is a "forest" — no cycle.



How is $\mathcal{L}$ given? ($|\mathcal{L}|$ can be large)

: will assume that $\exists$ algo Oracle(S) that takes $S \subseteq U$ and

say $\begin{cases} \text{YES if } S \in \mathcal{L}, \\ \text{NO o.w.} \end{cases}$

Max-weight Independent Set in Matroids.

e_i has weight w_i.

Input: Matroid $(U, \mathcal{L})$ where $U = \{e_1 \dots e_n\}$ and, weights. $w_1 \dots w_n$.

Output: Find $S \in \mathcal{L}$ that maximizes $w(S)$ $\left(= \sum_{e_i \in S} w_i\right)$.

Greedy for Matroids

Greedy-MISM$(U, \mathcal{I})$

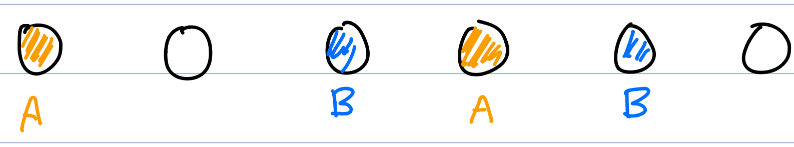
Sort U s.t. $w_1 \geq \cdots \geq w_n$.
Let $S = \phi$.
For $i=1$ to n
  If Oracle$(S \cup \{e_i\})$ = YES
    $S = S \cup \{e_i\}$
Return S

Runtime: $O(n \log n) + n \cdot$ (time of Oracle).

Optimality: Let ALG $= S$ at the end
OPT = optimal set.

$ALG_i = ALG \cap \{e_1 \ldots e_i\}$, $OPT_i = OPT \cap \{e_1 \ldots e_i\}$

Given $A, B \subseteq U$ s.t. $A = \{e_{a_1}, \ldots, e_{a_k}\}$ and
$B = \{e_{b_1}, \ldots, e_{b_\ell}\}$ $(a_1 \leq \cdots \leq a_k$ and $b_1 \leq \cdots \leq b_\ell)$,
say A "dominates" B if $a_i \leq b_i$ $\forall 1 \leq i \leq \ell$ (implies $k \geq \ell$).
Then since $w(e_{a_i}) \geq w(e_{b_i})$, $w(A) \geq w(B)$.



Claim $\forall 1 \leq i \leq n$, ALG_i dominates OPT_i.

Proof: Induction on i.

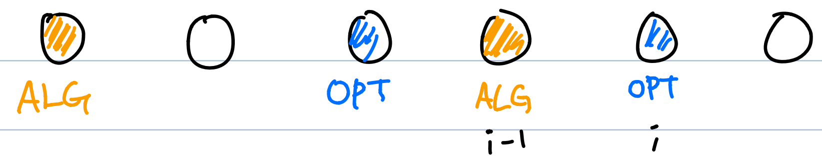
When $i=1$: if $OPT_1 = \{e_1\}$, then $\{e_1\}$ is ind. so alg. picks e_1 too.

When ALG_j dominates OPT_j for $j=1,\dots,i-1$:

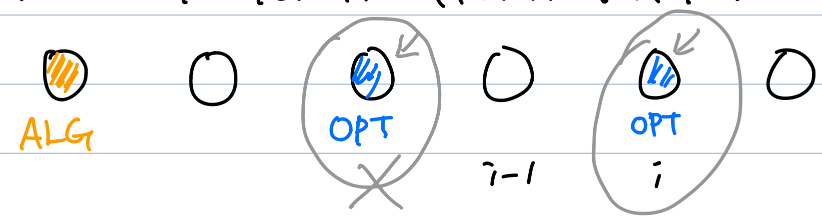
① $OPT_i \not\ni e_i$: $OPT_i = OPT_{i-1}$, which is dominated by ALG_{i-1}.

② $OPT_i \ni e_i$: $OPT_i = OPT_{i-1} \cup \{e_i\}$

(i) $|ALG_{i-1}| \geq |OPT_i|$: ALG_{i-1} dominates OPT_i.



(ii) $|ALG_{i-1}| < |OPT_i|$: $(|ALG_{i-1}| = |OPT_i| - 1)$



By exchange property, $\exists e_j \in OPT_i \setminus ALG_{i-1}$ s.t.
$ALG_{i-1} \cup \{e_j\} \in \mathcal{I}$.

But if $j < i$, then it means that $ALG_j \cup \{e_j\} \in \mathcal{I}$,
which contradicts the algorithm. (hereditary)

So, $j = i$ and $ALG_i = ALG_{i-1} \cup \{e_i\}$ so that
ALG_i dominates OPT_i □.

Minimum Spanning Tree

Minimum Spanning Tree

Input: Graph $G=(V,E)$ with $w: E \to \mathbb{R}^+$.

Output: Spanning tree $T \subseteq E$ s.t. $w(T) = \sum_{e \in T} w(e)$ is minimized.

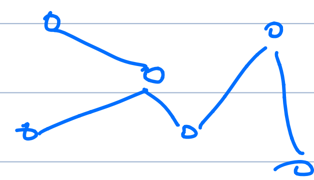
Spanning tree: Tree that connects every vertex.

① No cycle

② Contains exactly $n-1$ edges. $(n=|V|)$

③ Between any $u, v \in V$,

exactly one simple path between u and v.



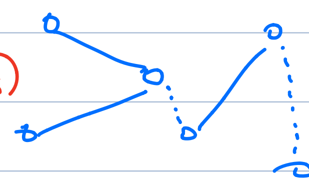
Forest: $F \subseteq E$ that doesn't have a cycle.

① No cycle

② (# edges) = $n-$ (# connected components)

③ Between any $u, v \in V$,

at most one simple path between u and v.



Definition. A set system $(U, \mathcal{I})$ is a "matroid" if

1. U is finite.
2. If $A \subseteq B$ and $B \in \mathcal{I}$, then $A \in \mathcal{I}$ (hereditary)
3. If $A, B \in \mathcal{I}$ and $|A| < |B|$, $\exists e_i \in B \setminus A$ s.t. $A \cup \{e_i\} \in \mathcal{I}$. (exchange)

Given $G=(V,E)$ and $\mathcal{L}=\{F\subseteq E : F \text{ is a forest}\}$

Lemma: $(E,\mathcal{L})$ is a matroid.

Pf. 1. E is finite

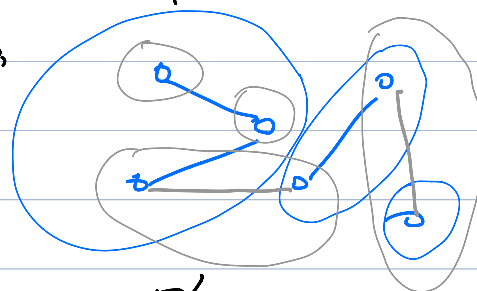
2. If $F\in\mathcal{L}$ and $F'\subseteq F$, $F'\in\mathcal{L}$.

3. Suppose $F', F\in\mathcal{L}$ with $|F'|<|F|$.

Let c', c be (# com. components) of F', F resp.

By prop. ② of forests,

$c' > c$.



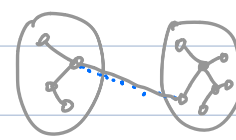
Then, $\exists (u,v)\in F$ s.t.

u, v are in different c.c.s in F'.

(otherwise, $c' \le c$)

(1) $(u,v)\notin F'$

(2) $F'\cup\{(u,v)\}$ doesn't have a cycle.



$\square$

Then first algo. for MST (Kruskal)

Let $M=\left(\max_{e\in E} w(e)\right)+1$, and $w'(e)=M-w(e)$.

Use the previous lecture's algo. to find max-weight independent set on the matroid $(E,\mathcal{L})$ with weight w'.

(For any spanning tree T, $w(T)=(n-1)M - w'(T)$)