Greedy alg for matroids:

Matroids:

$M = (S, \mathcal{I})$ is a matroid if it satisfies the

S: nonempty finite set

$\mathcal{I}$: collection of subsets of S, called independent sets

following properties:

(1) $A \in \mathcal{I}$, $B \subseteq A \Rightarrow B \in \mathcal{I}$

a subset of an indep set is indep. "hereditary property"

(2) $A \in \mathcal{I}$, $B \in \mathcal{I}$, $|A| > |B|$

$\Rightarrow \exists\, x \in A \setminus B$ s.t. $B \cup \{x\} \in \mathcal{I}$ "exchange property"

Examples

a) Matric Matroid

S: collection of vectors in $\mathbb{R}^n$

$\mathcal{I}$: all subsets of S that are linearly indep.

b) Graphic Matroid.

S: set of edges E in an undirected graph $G = (V, E)$

$\mathcal{I}$: collection of subsets of E that are acyclic

need to check 2 properties:

hereditary ✓

exchange property: let A, B acyclic subsets of edges s.t. $|A| > |B|$

Suppose all edges in A have both endpts in same connected component of B. For each component of B, say with k vertices, B has $k-1$ edges, A can't have more than this # of edges (internal to the component) $\Rightarrow |A| \le |B|$ contradiction

Other matroid terminology

A maximal indep set in a matroid is an indep set I s.t. $\forall e \in S \setminus I$, $I \cup \{e\}$ is not independent

A maximal indep set is also called a base (or basis)

rank(I) for $I \subseteq S$ is the size of maximal indep set in I.

Fact All bases of a matroid have same cardinality.

Pf by contradiction.

A, B bases, $|A| > |B|$.

by exchange property B not maximal

Weighted matroid $(S, \mathcal{I}, w)$

$\forall$ each $e \in S$ $w(e)$ is the weight of element e

Problem: find base of maximum total weight.

Greedy Alg: Order elements of matroid in order of nonincreasing weight.

$A := \phi$

Consider elements in order above.

if ~~element e~~ ~~AB~~ element e under consideration can be added to A, and A stays independent, add e to A

Thm Greedy alg outputs optimal (max weight) base.

Pf Suppose not.

Suppose greedy chooses elements

$$e_{i_1}, e_{i_2}, \ldots, e_{i_k}$$

Among OPT solns, consider one that agrees with Greedy on as large a prefix as possible, say first l elts

so $OPT = e_{i_1}, e_{i_2}, \ldots, e_{i_l}, e_{j_{l+1}}, \ldots e_{j_k}$ ordered in order of nonincreasing weight

Let $A = \{e_{i_1}, e_{i_2}, \ldots, e_{i_l}, e_{i_{l+1}}\}$

by defn of Greedy $w(e_{i_{l+1}}) \geq w(e_{j_{l+1}}) \geq \ldots \geq w(e_{j_k})$

apply exchange to A & OPT repeatedly until grow A to cardinality k

result has cost no more than OPT $\rightarrow\leftarrow$

Use this to solve MST

spanning trees are bases of graphic matroid.

Define $w(e) = -\text{cost}(e)$

One more example: task scheduling
optimally scheduling unit time tasks with deadlines & a penalty that must be paid if deadline missed

Input: set $S = \{1, \ldots, n\}$ of unit time tasks.
d_i deadline for task i
w_i penalty if $\underbrace{\text{deadline for job } i \text{ not met}}_{\text{job is late}}$

single resource

job	1	2	3	4	5	6	7
d_i	4	2	4	3	1	4	6
w_i	70	60	50	40	30	20	10

if scheduled in order

	0	1	2	3	4	5	6	7	8
job	1	2	3	4	5	6	7		
				late	late	late	late		

penalty 40 + 30 + 20 + 10

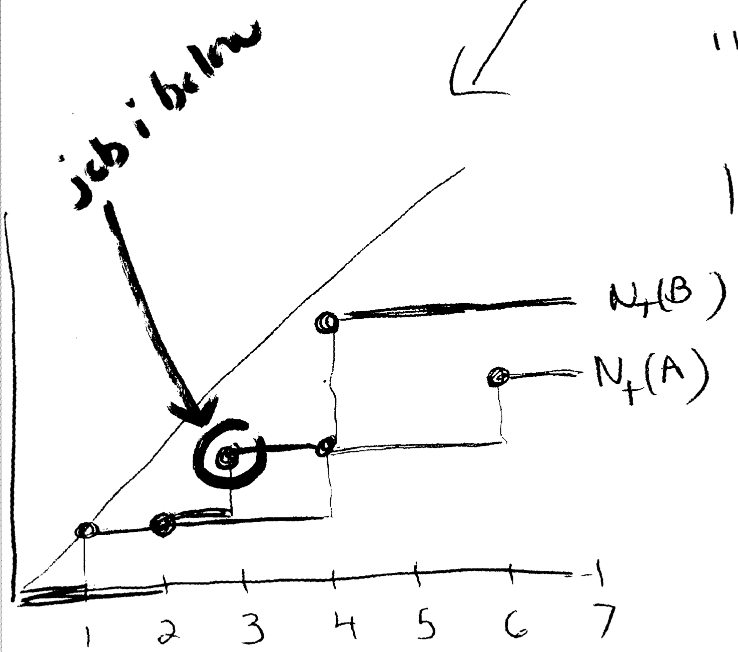
Claim: $(S, \mathcal{I})$ is matroid, where $I \subseteq S$ is indep. if $\exists$ schedule for I s.t. no jobs in I are late

Proof: next page

Proof: hereditary ✓

to prove exchange property
let A, B be indep sets s.t. $|A| < |B|$

e.g. in above example $A = \{5, 6, 7\}$
$B = \{1, 2, 3, 4\}$



in each set sort tasks by deadline

let $N_t(A)$ = # tasks in A with deadline $\le t$

note $N_n(A) = |A|$
$N_n(B) = |B|$

let k be largest t s.t. $N_t(B) \le N_t(A)$

(in this example $k=2$)

$\Rightarrow N_t(B) > N_t(A) \quad \forall\, t > k$

$\Rightarrow$ B has a job, say i s.t. i has deadline $k+1$ that is not in A

Claim: can add that job to A & result still independent.

$N_t(A \cup \{i\}) \le t \quad \forall\, t \Rightarrow$
jobs are scheduled in order of nondecreasing deadline, none of them will be late.

Use this to solve problem

2 observations:

(1) wlog optimal schedule schedules all early tasks first.

(2) minimizing sum of penalties of late task = maximizing sum of penalties of early tasks.

6