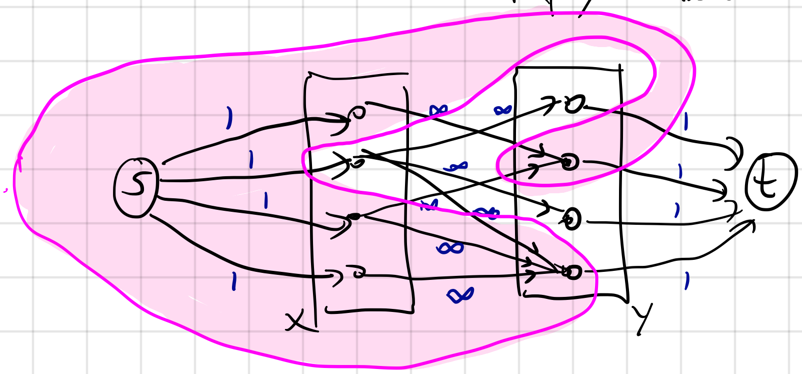
13 Mar 2019

Minimum Cuts

Recap from last time...

What does max-flow min-cut imply about bipartite matchings?



IF (A,B) is a min s-t cut then $c(A,B) \leq C < \infty$.

Write

$$A = \{s\} \cup A_X \cup A_Y \qquad A_X := A \cap X$$
$$\qquad\qquad\qquad\qquad\qquad A_Y := A \cap Y.$$
$$B = \{t\} \cup B_X \cup B_Y$$

Edges from A_X to B_Y have capacity ∞ so a min-cut cannot have any such edges.

$\Longrightarrow$ every edge leaving A_X goes to A_Y.

$\Longrightarrow$ every edge has an endpoint in A_Y or B_X or both.

$\boxed{A_Y \cup B_X \text{ is a "vertex cover" of } G.}$

2 types of edges crossing the cut: $\{s\}$ to B_X, A_Y to $\{t\}$.

pink nodes on right — white nodes on left.

$$c(A,B) = |B_X| + |A_Y|.$$

$$= |B_X| + |A_X| - |A_X| + |A_Y|$$

a set that contains all neighbors of A_X.

$$= |X| - (|A_X| - |A_Y|)$$

nodes on left side — ("how constricted A_X is")

Conclusion. IF the max matching in G leaves exactly k nodes of X unmatched, we can always find $A_X \subseteq X$ with only $|A_X| - k$ neighbors on the opposite side of the graph.

Corollary (Hall's Marriage Theorem) IF every $A_X \subseteq X$ has at least $|A_X|$ neighbors in Y then G has a matching that covers every node of X.

$A_Y \cup B_X$ is a vertex cover. (Every edge has an endpoint in $A_Y \cup B_X$.)

<u>Note.</u> If M is a matching and S is a vertex cover, $|M| \leq |S|$.

We proved, for G bipartite,

$$\begin{aligned} \text{size of max matching} &= v(\text{max flow}) \\ &= c(\text{min cut}) \\ &= |A_Y \cup B_X| \end{aligned}$$

$\therefore$ $A_Y \cup B_X$ is a vertex cover of min cardinality and its size <u>equals</u> the max matching size.

König–Egervary Theorem: In any bipartite graph max matching = min vertex cover.

How to find a min cut efficiently?
Compute a max flow and then....

(A) $A = \{$all vertices that don't have a path to t with positive residual capacity left over$\}$

(B) $A = \{$ all vertices reachable from s in $G_f\}$

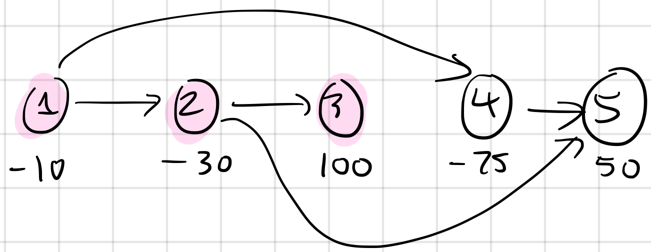
(c) $A = \{$all vertices with no outgoing edges in $G_f\}$

<u>Project Selection</u> (§7.11)

Given: a set of n projects $\{1, 2, \ldots, n\}$.
value of project i is $v_i \in \mathbb{Z}$ could be positive or negative

dependency graph H: a directed graph with vertex set $\{1, \ldots, n\}$. Edge (i,j) is that project i is a prerequisite for project j.

Goal: Pick a set of compatible projects with maximum net value.



Def. A set of projects, P, is compatible if $\forall\, j \in P$ and $\forall$ dependency $(i,j) \in E(H)$ P must contain i.

Idea: Selecting a compatible set of proj is like selecting a cut in a graph, so try reducing this problem to min-cut.

Issues:
[1.] How to model prerequisites?
[2.] Values are on nodes, not edges.
[3.] No source, sink nodes.
[4.] Values can be positive or negative, capacities in a flow network are only positive.
[5.] Objective of proj. selection is to <u>maximize</u> a sum.

$$\text{maximize}\left(\sum_{j \in P} v_j\right) \Longleftrightarrow \text{minimize}\left(\sum_{j \notin P} v_j\right).$$

Next step: Set up a graph where $\sum_{j \notin P} v_j$ is represented as a cut capacity.

NOT ACHIEVABLE!

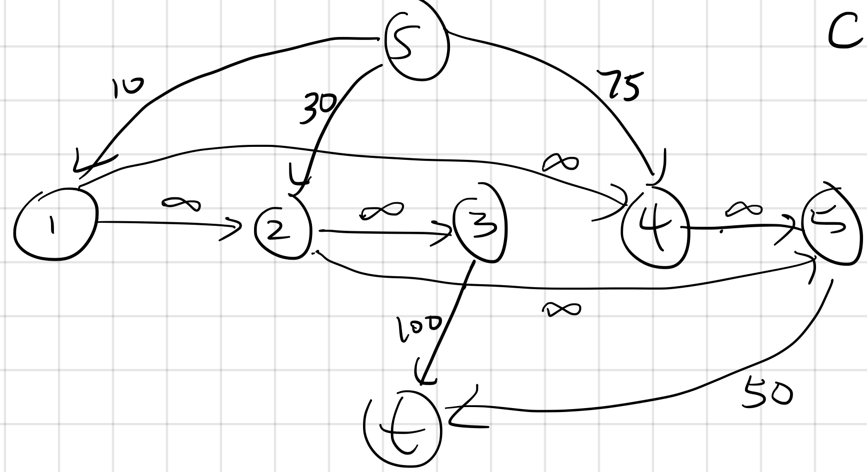
(No dependencies)

$\sum_{j \notin P} v_j = -115$

Let $B = \sum_{j=1}^{n} \max\{0, v_j\}$.

Minimize $B - \sum_{j \in P} v_j$.

∞ is a notation for $C+1$.



Selected set of projects $\equiv$ sink side of min cut

<u>Lemma:</u> If A, B is a finite capacity cut, $P = \{ j \mid j \in B \}$ is compatible.

Capacity of cut in this graph is

$$\sum_{\substack{j \in P \\ v_j < 0}} -v_j + \sum_{\substack{j \notin P \\ v_j > 0}} v_j$$

$$= \sum_{\substack{j \in P \\ v_j < 0}} (-v_j) + \sum_{\substack{j \in P \\ v_j \geq 0}} (-v_j) + \underbrace{\left[\sum_{\substack{j \in P \\ v_j \geq 0}} v_j + \sum_{\substack{j \notin P \\ v_j > 0}} v_j \right]}_{= B}$$

$$= B - \sum_{j \in P} v_j$$