Network Flow: Min Faculty Hire 11/09

Also "airplane scheduling" problem.

- VIU has n classes scheduled
- ∀ professor can teach any class
- classes have a start time and an end time and location.

$C[i] \rightarrow start$ $\quad C[i] \rightarrow end$ $\quad C[i] \rightarrow loc$

Also it takes $T[i, j]$ time to walk from location u to location v.

What is min # professors needed to teach all the classes?

Reduce to a disjoint-path-cover problem:

1. Construct the DAG $G = (V, E)$

   V = classes $\quad i = 1..n$

   $E = (i, j) \in E$ iff

   $$C[i] \rightarrow end + \underbrace{T[C[i] \rightarrow loc, C[j] \rightarrow loc]}_{\text{travel time between class } i \text{ and class } j} \leq C[j] \rightarrow start$$

2. Find disjoint path cover of G.

Each disjoint path corresponds to the class assignment that a single professor can do. $\boxtimes$

11.6 in Jeff Erickson's text is Baseball Elimination

- until Network Flow algorithms were invented (and computer scientists/mathematicians applied them to this problem) the only way to determine Elimination was brute force, or ad hoc arguments!

11.7 Project Selection

- Projects are related to one another by dependencies

eg $\boxed{x} \longrightarrow \boxed{y}$

"x must be done before y (if y is executed at all)"

Note: Erickson's text puts edge direction in opposite direction.

Each project has a value: positive = net profit

negative = net loss (but might allow

ensuing projects to make profit.)

Our goal: determine which projects to

Take and which ones to

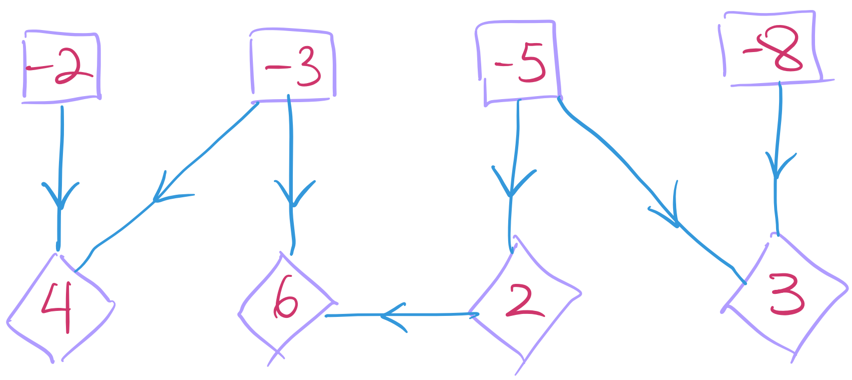
Skip

so as to maximize total value (profit)

Constraint

if $x \in T$ then so is every y

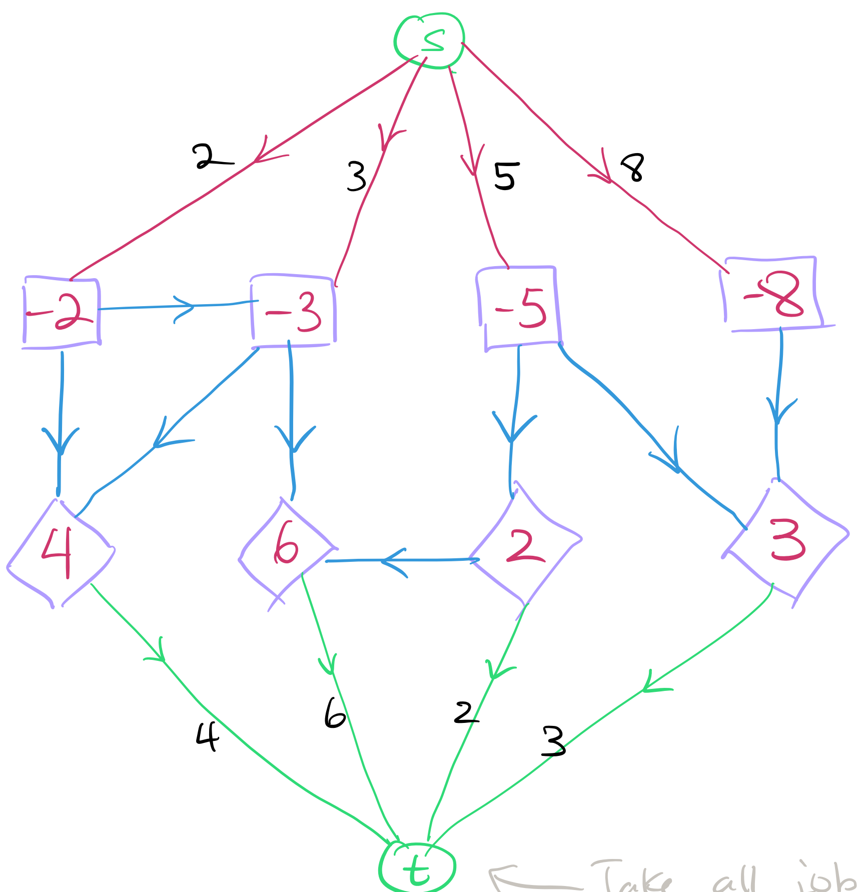
where y has a directed path to x.



ensuing projects to make profit.)

Our goal: determine which projects to
Take and which ones to
Skip
so as to maximize total value (profit)

Constraint $x \in T$ then so are all y where $y \to x$



Take all jobs in same partition as t.

Recall: Max Flow = Min Cut!

Claim: Let S, T be a min cut, T contains t (take)

Then taking all the projects in T has the following **cost** and **value** implications:

Cutting across a red edge — bear the cost on this (project) edge

Cutting across a green edge — forego the profit of this edge (project)

$$\text{Total profit} = \underbrace{4+6+2+3}_{\substack{\text{always sum of} \\ \text{all positive} \\ \text{project values}}} - \text{Cost of cut}$$

$$3 + 2 + 3 + 5$$

See how it guarantees that if you "take" project with value 4 you must also "take" projects with value -2, -3 ... !