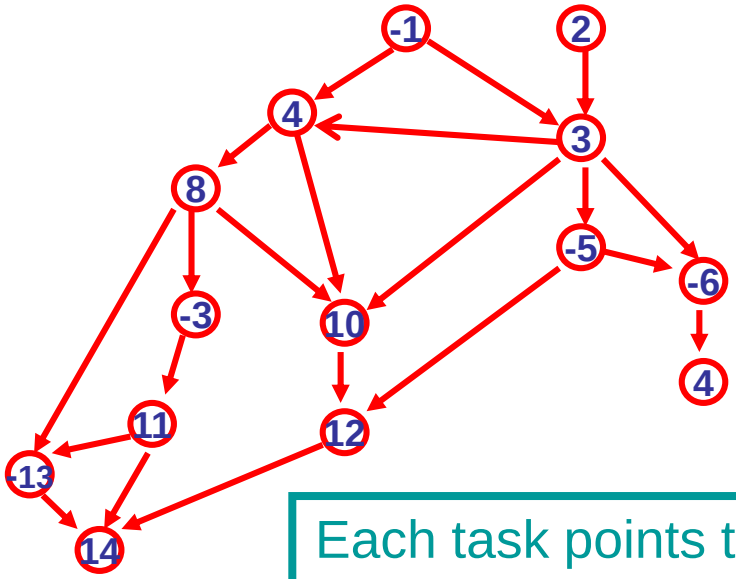
CSE 421

Applications

Yin Tat Lee

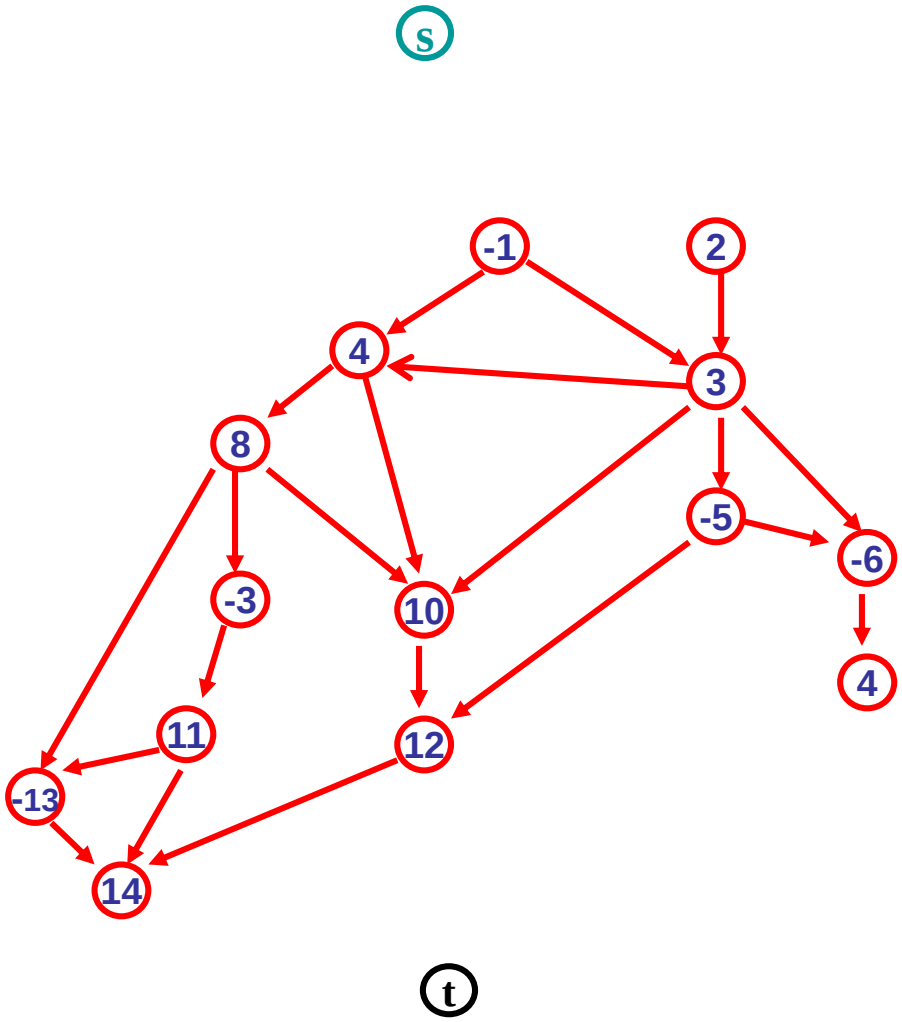
Project Selection

- Given a DAG **G=(V,E)** representing
  precedence constraints on tasks
  (a task points to its **predecessors**)
  a profit value **p(v)** for each task **v∈V**
  (may be positive or negative)
- Find a set **A⊆V** of tasks that is closed under predecessors,
  (i.e. if **(u,v)∈E** and **u∈A** then **v∈A**)
  that maximizes Profit(**A**)=$\Sigma_{v\in A}$ **p(v)**



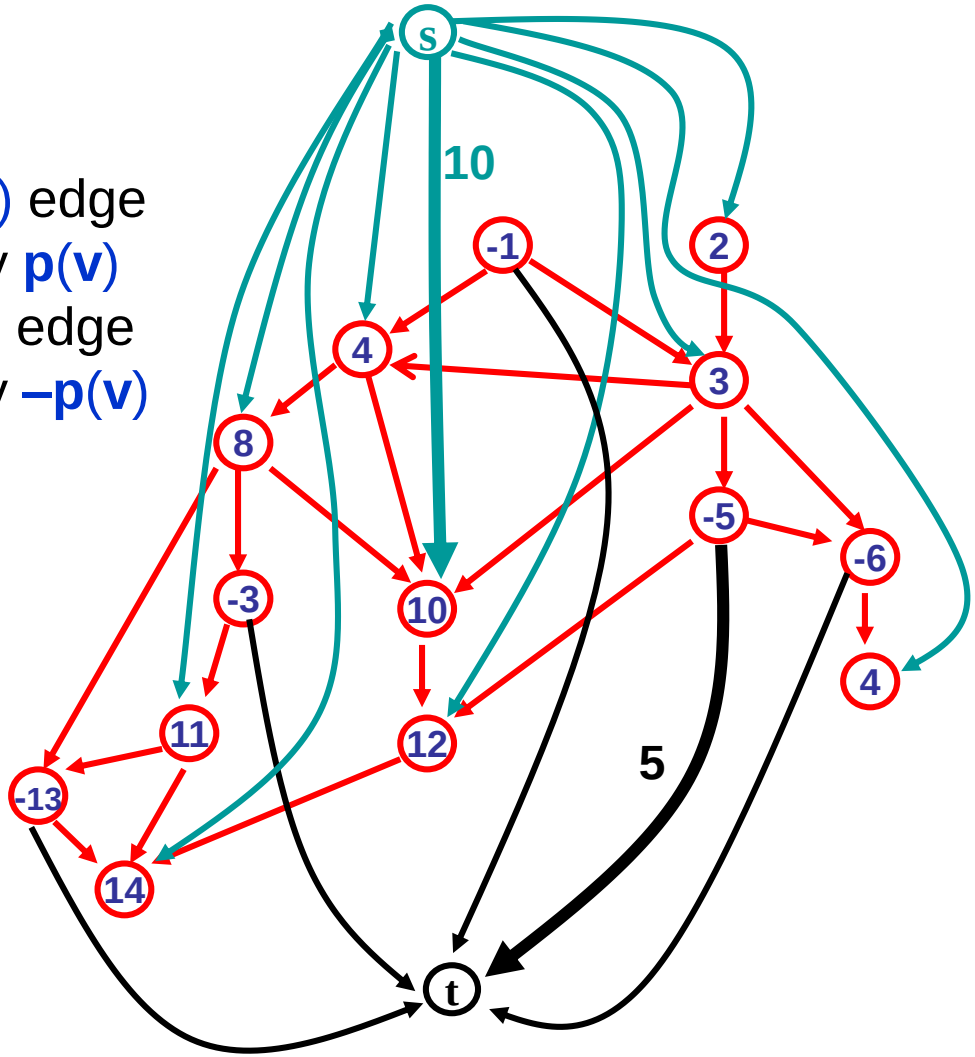
Each task points to its predecessor tasks

Extended Graph



Extended Graph G'

For each vertex **v**
If **p(v)≥0** add **(s,v)** edge
with capacity **p(v)**
If **p(v)<0** add **(v,t)** edge
with capacity **–p(v)**



Extended Graph G'

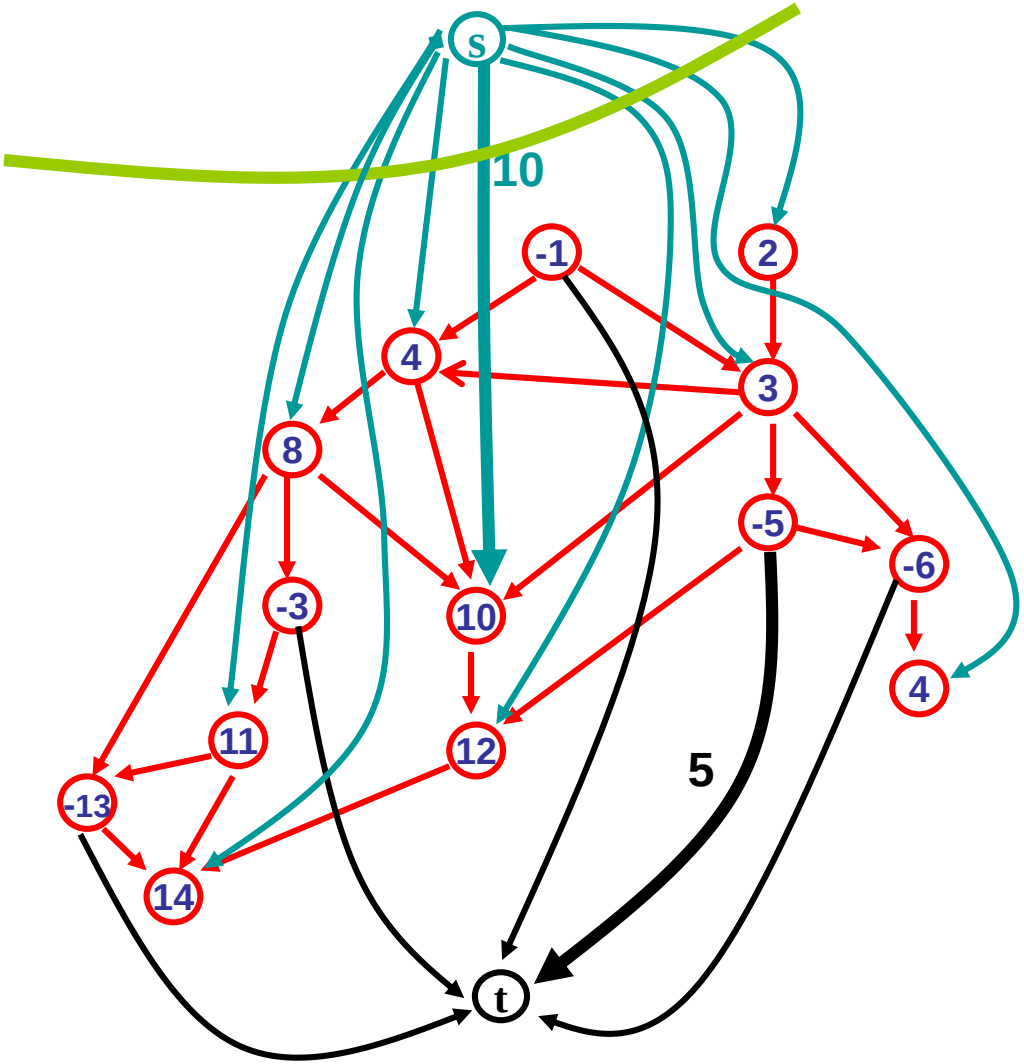
- Want: Set capacities on edges of **G** so that for minimum **s**-**t**-cut $(S, \overline{S})$ in **G'**, the set **A**=**S**-{**s**}
  - satisfies precedence constraints
  - has maximum possible profit in **G**
- Cut capacity with **S**={**s**} is just **C**=$\sum_{\mathbf{v:\ p(v)\geq 0}} \mathbf{p(v)}$

  Profit(**A**) $\leq$ **C** for any set **A**
- To satisfy constraints, don't want any original edges of **G** going forward across the minimum cut

  That would correspond to a task in **A**=**S**-{**s**} that had a predecessor not in **A**=**S**-{**s**}
- Set capacity of each of the edges of **G** to $+\infty$**.**

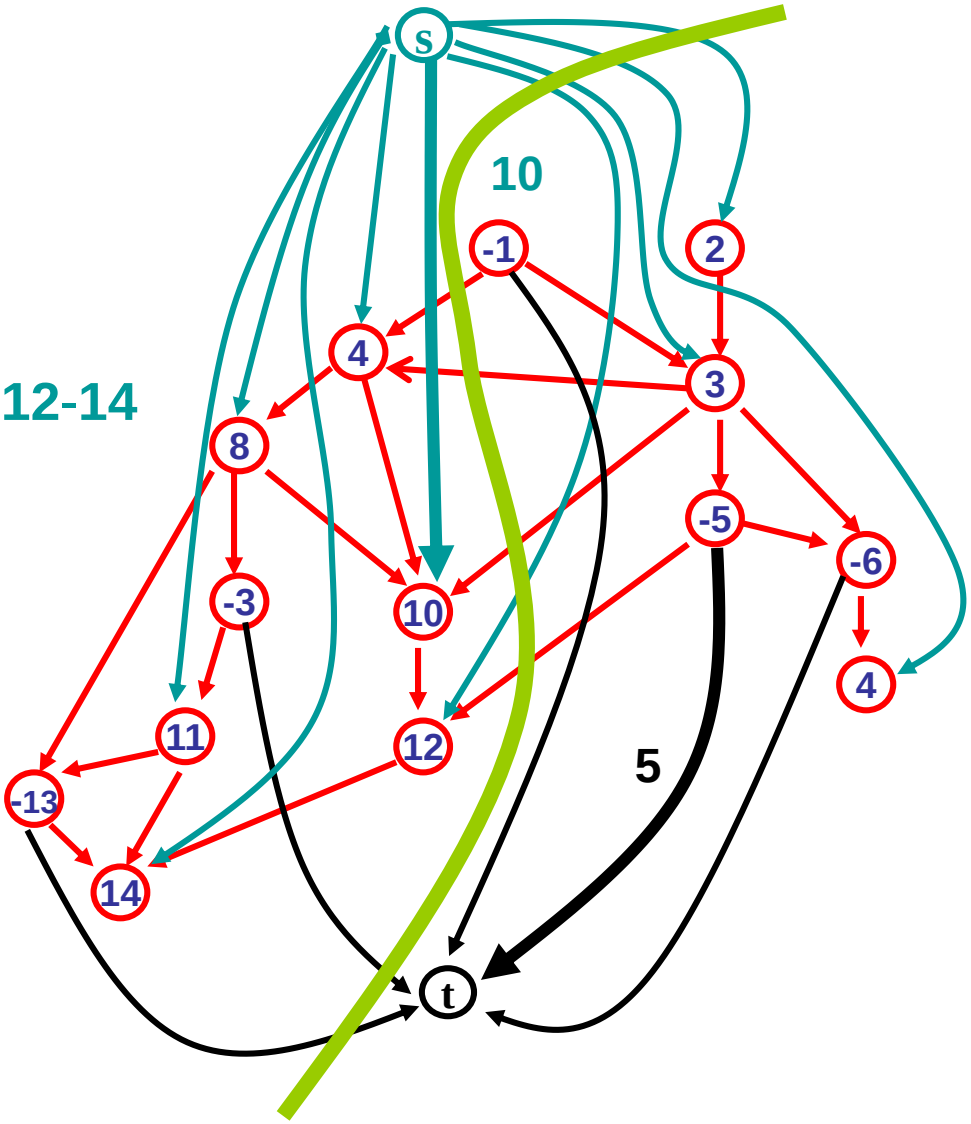
Extended Graph G'

Capacity $C=\Sigma_{v:\ p(v)\geq 0}\ p(v)$



Extended Graph G'

Cut value
=**13**+**3**+**2**+**3**+**4**
=**13**+**3**
+**C-4-8-10-11-12-14**



Project Selection

- **Claim** Any **s**-**t**-cut $(S, \overline{S})$ in **G'** such that **A**=**S**-{**s**} satisfies
  - precedence constraints and
  - has capacity c(**S**,**T**)=**C** - $\Sigma_{\mathbf{v} \in \mathbf{A}}$ **p**(**v**) = **C** - Profit(**A**)

- **Corollary** A minimum cut $(S, \overline{S})$ in **G'** yields an optimal solution **A**=**S**-{**s**} to the project selection problem

- **Algorithm** Compute maximum flow **f** in **G'**, find the set **S** of nodes reachable from **s** in $\mathbf{G'_f}$ and return **S**-{**s**}

Proof of Claim

- **A=S-{s}** satisfies precedence constraints

  No edge of **G** crosses forward out of **A** since those edges have capacity $+\infty$

  Only forward edges cut are of the form

  $(\mathbf{v},\mathbf{t})$ for $\mathbf{v}\in\mathbf{A}$ or $(\mathbf{s},\mathbf{v})$ for $\mathbf{v}\notin\mathbf{A}$

  The $(\mathbf{v},\mathbf{t})$ edges for $\mathbf{v}\in\mathbf{A}$ contribute

  $$\sum_{\mathbf{v}\in\mathbf{A}:\mathbf{p}(\mathbf{v})<0} -\mathbf{p}(\mathbf{v}) = -\sum_{\mathbf{v}\in\mathbf{A}:\mathbf{p}(\mathbf{v})<0} \mathbf{p}(\mathbf{v})$$

  The $(\mathbf{s},\mathbf{v})$ edges for $\mathbf{v}\notin\mathbf{A}$ contribute

  $$\sum_{\mathbf{v}\notin\mathbf{A}:\ \mathbf{p}(\mathbf{v})\geq 0} \mathbf{p}(\mathbf{v}) = \mathbf{C} - \sum_{\mathbf{v}\in\mathbf{A}:\ \mathbf{p}(\mathbf{v})\geq 0} \mathbf{p}(\mathbf{v})$$

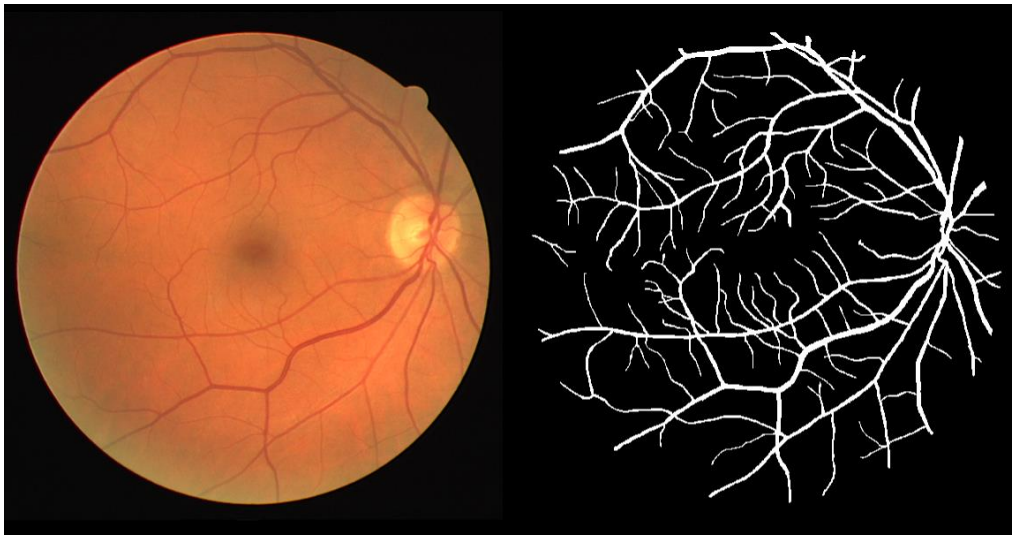
  Therefore the total capacity of the cut is

  $$\mathbf{c}(\mathbf{S},\mathbf{T}) = \mathbf{C} - \sum_{\mathbf{v}\in\mathbf{A}} \mathbf{p}(\mathbf{v}) = \mathbf{C} - \text{Profit}(\mathbf{A})$$

Image Segmentation

Given an image we want to separate foreground from background

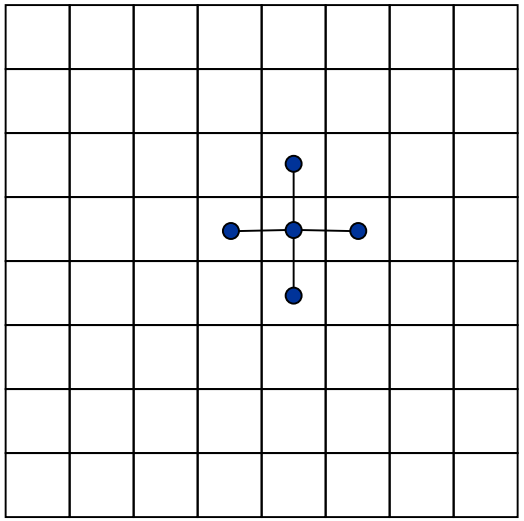
- Important problem in image processing.
- Divide image into coherent regions.



Foreground / background segmentation

Label each pixel as foreground/background.

- V = set of pixels, E = pairs of neighboring pixels.
- a_i is the original image.
- $a_i \gg 0$ means we prefer to label i in foreground.
- $p_{i,j} \geq 0$ is separation penalty for labeling one of i and j as foreground, and the other as background.



Goals:

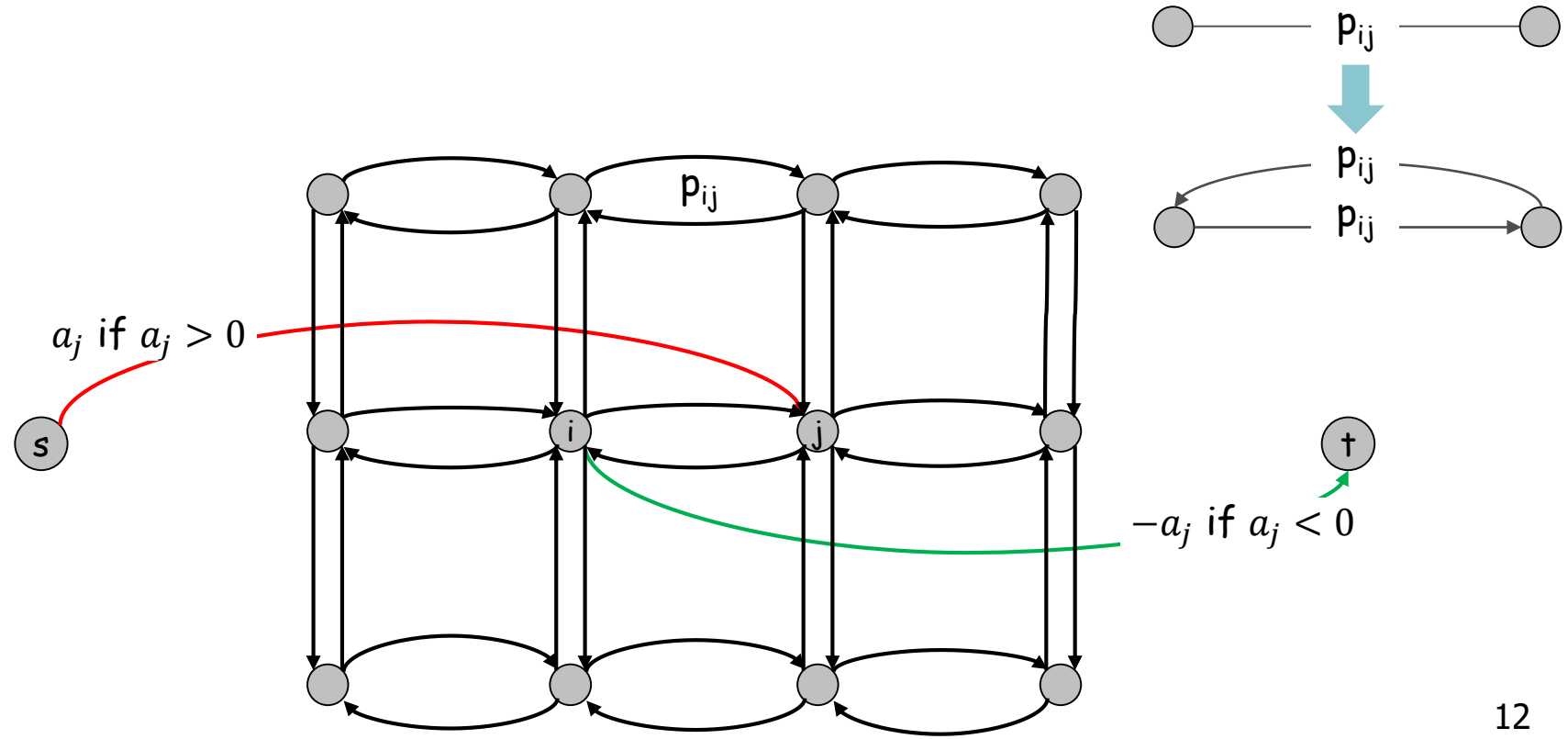
Find partition $(S, \overline{S})$ that minimizes:

$$-\sum_{i \in S} a_i + \sum_{\substack{(i,j) \in E \\ i \in S, j \in \overline{S}}} p_{i,j}$$

where S is the foreground.

Min cut Formulation

$G' = (V', E')$.
Add s to correspond to foreground;
Add t to correspond to background;
Use two anti-parallel edges instead of undirected edge.



G'

Min cut Formulation (cont'd)

- Consider min cut $(S, \overline{S})$ in G'. (S = foreground.)

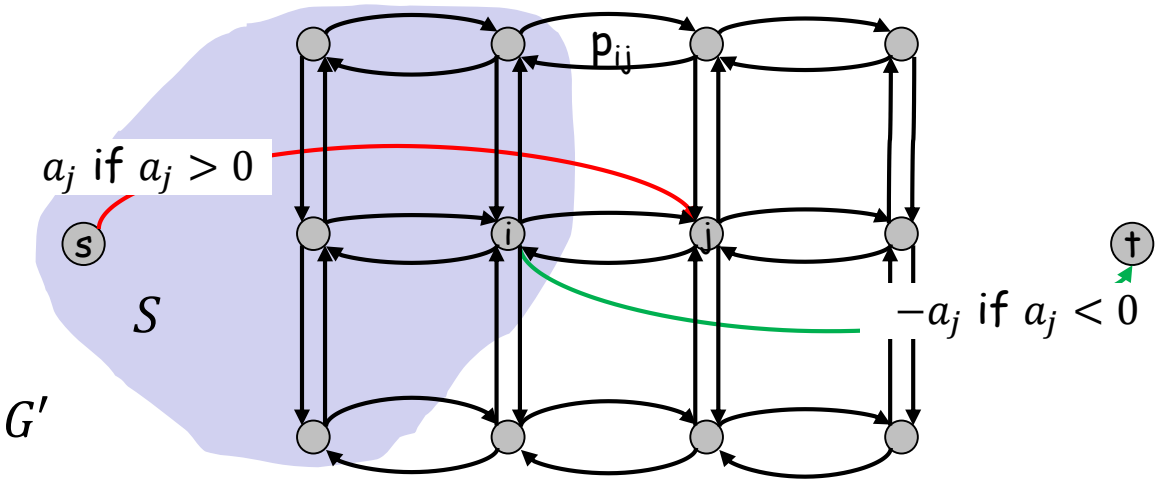
$$cap(S, \overline{S}) = \sum_{i \in S} -a_i \, 1_{a_i<0} + \sum_{i \in \overline{S}} a_i \, 1_{a_i>0} + \sum_{\substack{(i,j) \in E \\ i \in S, j \in \overline{S}}} p_{i,j}$$

$$= -\sum_{i \in S} a_i + \sum_{i \in S} a_i \, 1_{a_i>0} + \sum_{i \in \overline{S}} a_i \, 1_{a_i>0} + \sum\nolimits_{\dots} p_{i,j}$$

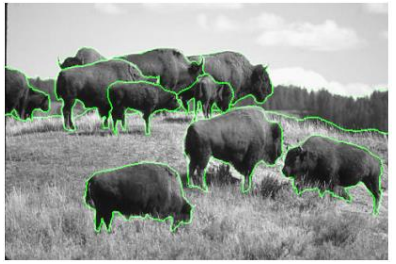
$$= -\sum_{i \in S} a_i + \sum_i a_i + \sum\nolimits_{\dots} p_{i,j}$$

$$= -\sum_{i \in S} a_i + \sum\nolimits_{\dots} p_{i,j} + \text{constant}$$

Precisely, what we want to minimize.



Reality



- The main difficulty is to come up with a good model.
- May want to have human interaction.

- Segmentation may be real-valued instead of {0,1}.
- There are many more than 1 objects.
- May need labeling.
- Augmenting path is not great for GPU.