Algorithm Design and Analysis

CSE 565

LECTURE 22

Maximum Flow Applications

- Image segmentation
- Project selection

Extensions to Max Flow

Sofya Raskhodnikova

Review

1) We saw Menger's theorem in the last lecture.

- Try to recall the statement of Menger's theorem for directed graphs.
- Does a similar statement hold if the graph is undirected?

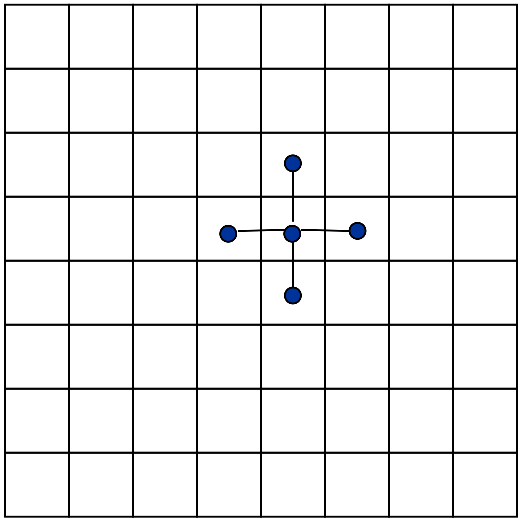
7.10 Image Segmentation

Image Segmentation

- Image segmentation.
  - Central problem in image processing.
  - Divide image into coherent regions.

- Ex: Pictures of people standing against a nature background. Identify each person as a coherent object, distinct from the background.

Image Segmentation



Foreground / background segmentation.

- Want to label each pixel in picture as belonging to foreground or background.

Inputs:

- V = set of pixels, E = pairs of neighboring pixels.
- $a_i \geq 0$ is likelihood pixel i in foreground.
- $b_i \geq 0$ is likelihood pixel i in background.
- p_{ij} : separation penalty for labeling one of i and j as foreground, and the other as background.

Goals.

- Accuracy: if $a_i > b_i$ in isolation, prefer to label i in foreground.
- Smoothness: if many neighbors of i are labeled foreground, we should be inclined to label i as foreground.
- Find partition (A, B) that **maximizes**: $$\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$$

foreground background

Image Segmentation

Let's try to formulate as min cut problem.
Obstacles:
- Maximization.
- No source or sink.
- Undirected graph.

Turn into minimization problem.

- Maximizing $\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$

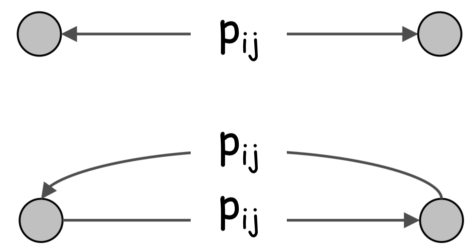
  is equivalent to minimizing $\underbrace{\left(\sum_{i \in V} a_i + \sum_{j \in V} b_j\right)}_{\text{a constant}} - \sum_{i \in A} a_i - \sum_{j \in B} b_j + \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$

- or alternatively $\sum_{j \in B} a_j + \sum_{i \in A} b_i + \sum_{\substack{(i,j) \in E \\ |A \cap \{i,j\}| = 1}} p_{ij}$

Image Segmentation

Formulate as min cut problem.

- $G' = (V', E')$.
- Add source to correspond to foreground; add sink to correspond to background
- Use two anti-parallel edges instead of undirected edge.



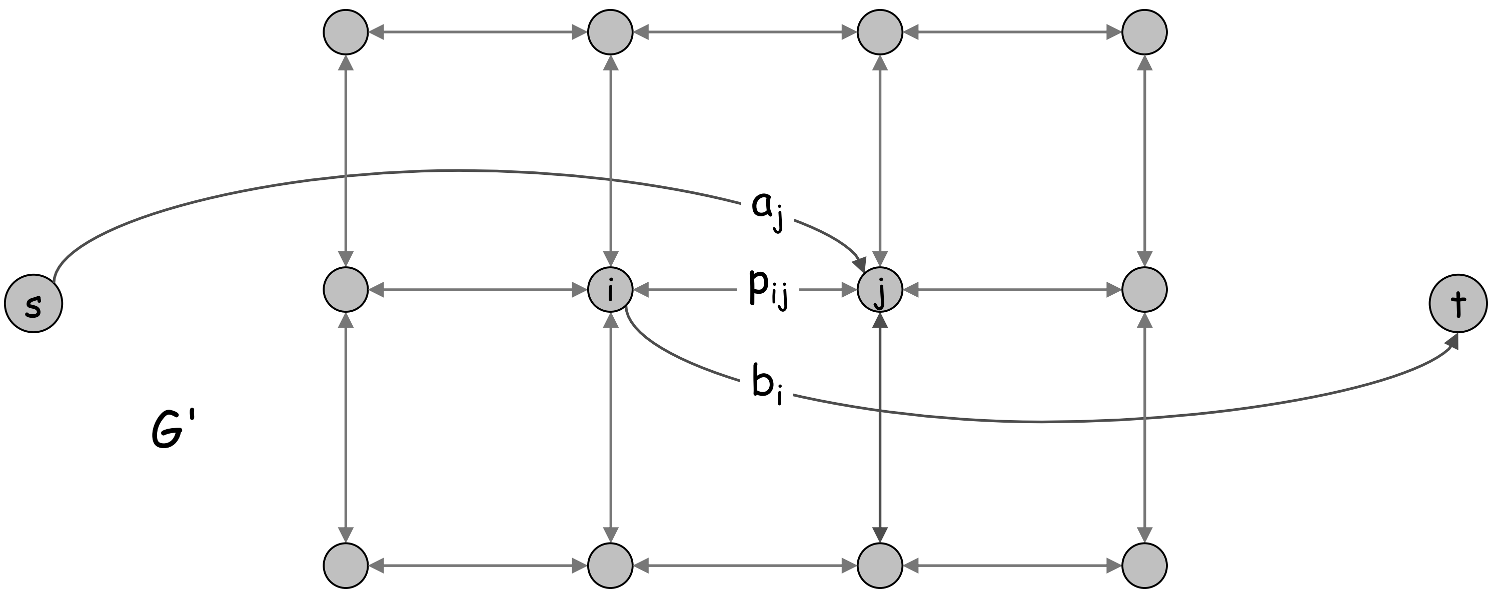


Image Segmentation

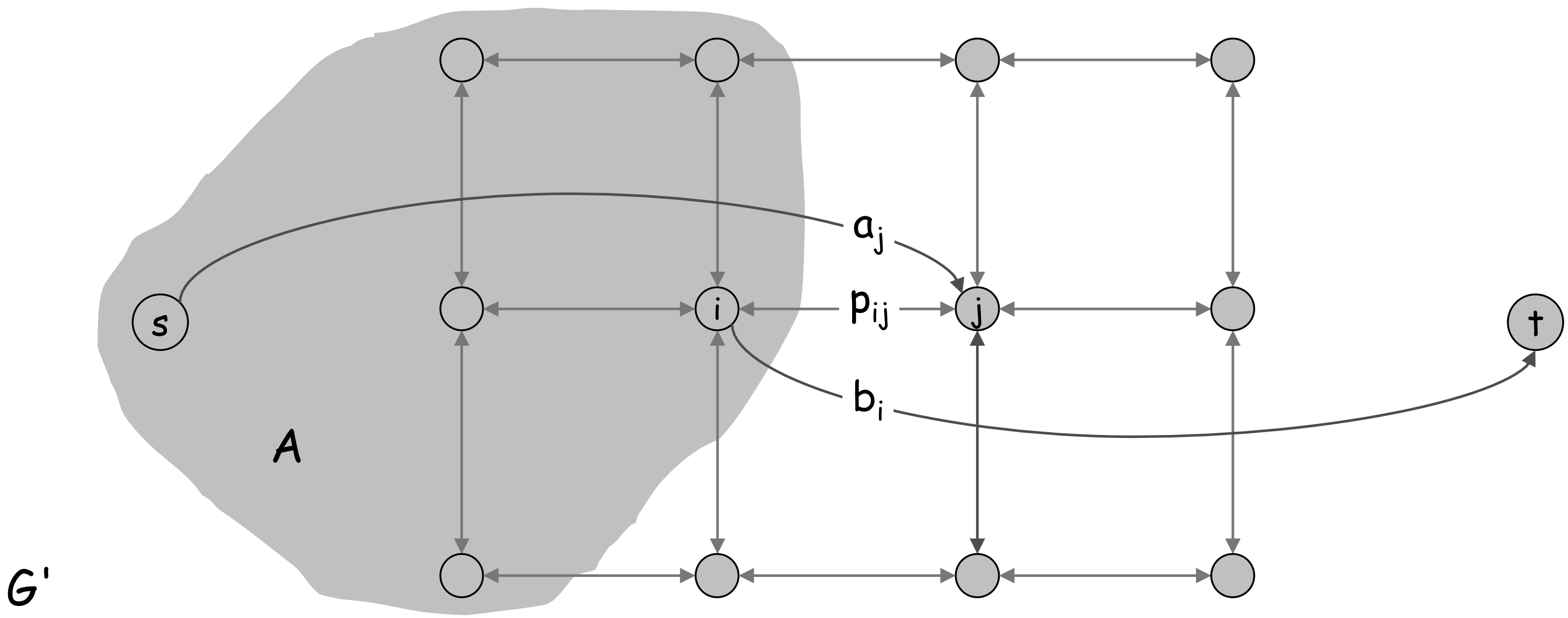
Consider a cut (A, B) in G'.

- A = foreground.

$$cap(A, B) \;=\; \sum_{j \in B} a_j + \sum_{i \in A} b_i \;+ \sum_{\substack{(i,j) \in E \\ i \in A,\; j \in B}} p_{ij}$$

if i and j on different sides, p_{ij} counted exactly once

- Precisely the quantity we want to minimize.



Exercises

1) Write out the reduction as an algorithm (in pseudocode) that uses a min-cut subroutine.

2) Prove that the problem of finding a maximum-weight independent set in **bipartite** graphs can be reduced to minimum cut.
   - Doesn't follow directly from this application, but uses a similar idea

7.11 Project Selection

Project Selection

can be positive or negative

Projects with prerequisites.

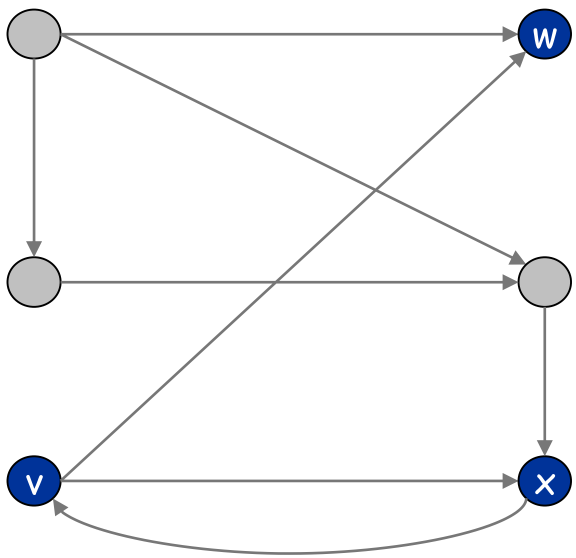
- Set P of possible projects. Project v has associated revenue p_v.
  - some projects generate money: create interactive e-commerce interface, redesign web page
  - others cost money: upgrade computers, get site license
- Set of prerequisites E. If (v, w) $\in$ E, can't do project v and unless also do project w.
  - (P,E) is a DAG
- A subset of projects $A \subseteq P$ is feasible if the prerequisite of every project in A also belongs to A.

Project selection. Choose a feasible subset of projects to maximize revenue.

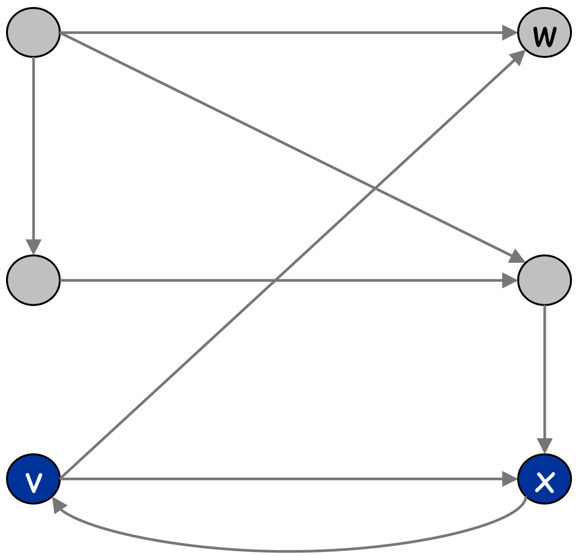
Project Selection: Prerequisite Graph

Prerequisite graph.

- Include an edge from v to w if w is a prerequisite to v.
  - $(v, w) \in E \;\Rightarrow\; v$ requires w
- {v, w, x} is feasible subset of projects.
- {v, x} is infeasible subset of projects.



feasible



infeasible

Project Selection: Min Cut Formulation

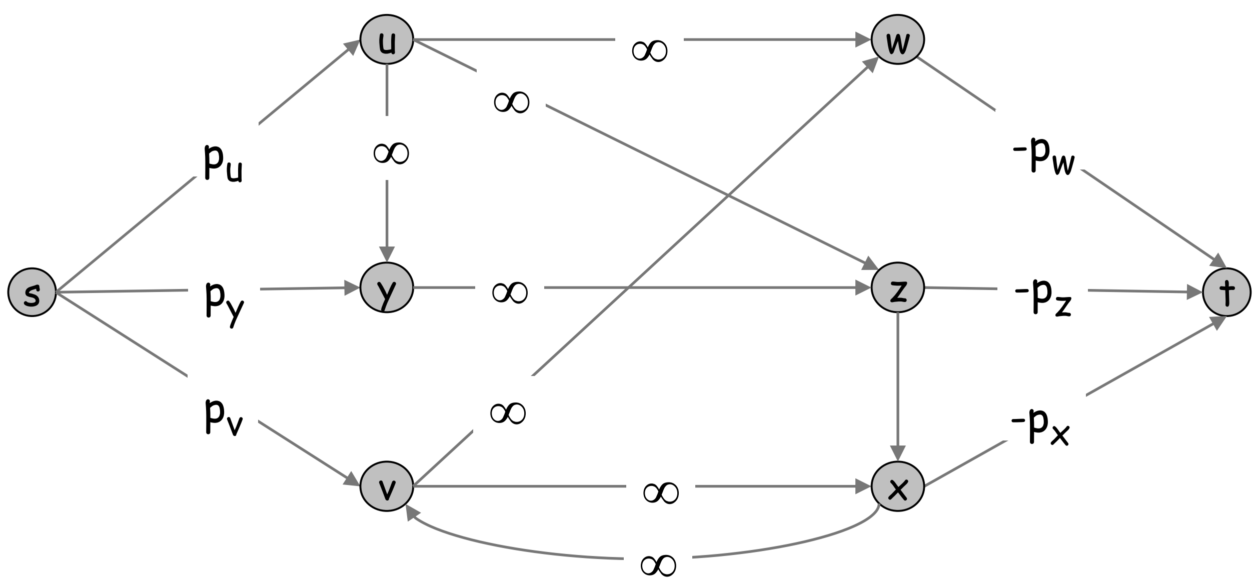
Min cut formulation.

- Add source s, sink t
- Assign capacity __ to all prerequisite edges.
- Add edge (s, v) with capacity __ if ___.
- Add edge (v, t) with capacity __ if ___.

Project Selection: Min Cut Formulation

Min cut formulation.

- Assign capacity ∞ to all prerequisite edges.
- Add source s, sink t
- Add edge (s, v) with capacity p_v if $p_v > 0$.
- Add edge (v, t) with capacity $-p_v$ if $p_v < 0$.
- For notational convenience, define $p_s = p_t = 0$.

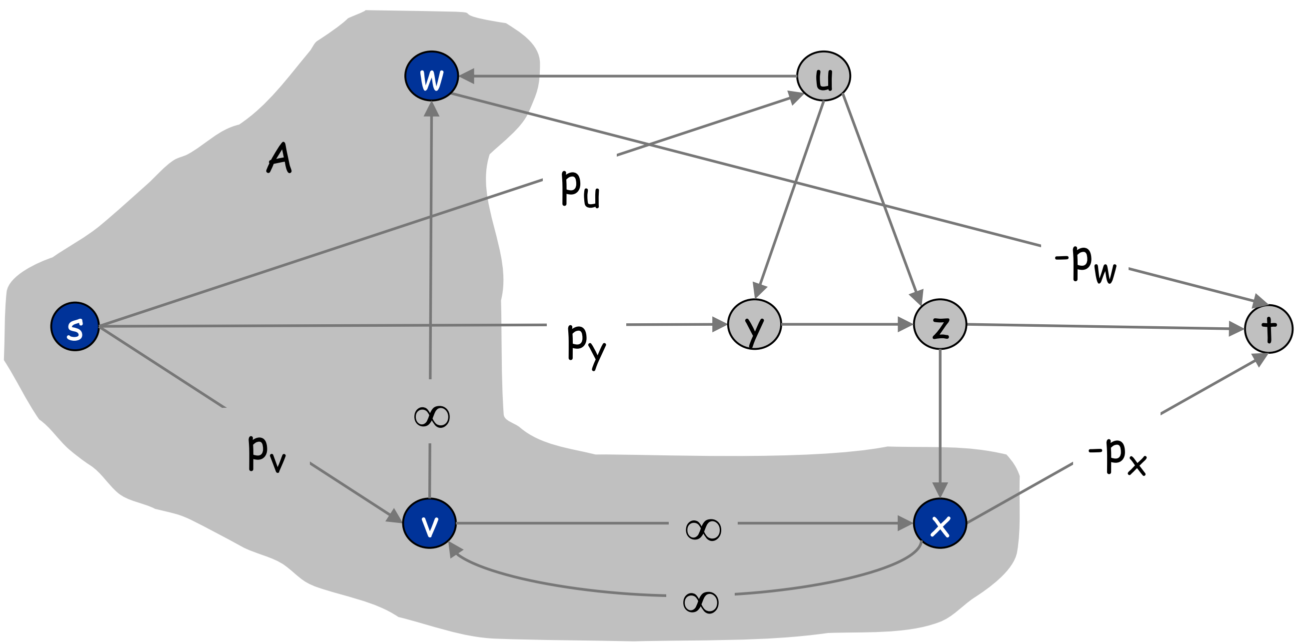


Project Selection: Min Cut Formulation

Claim. (A, B) is min cut iff A – { s } is optimal set of projects.

- Infinite capacity edges ensure A – { s } is feasible.
- Max revenue because:

$$
\begin{aligned}
cap(A,\ B) &= \sum_{v \in B:\ p_v > 0} p_v + \sum_{v \in A:\ p_v < 0} (-p_v) \\
&= \underbrace{\sum_{v:\ p_v > 0} p_v}_{\text{constant}} - \sum_{v \in A} p_v
\end{aligned}
$$



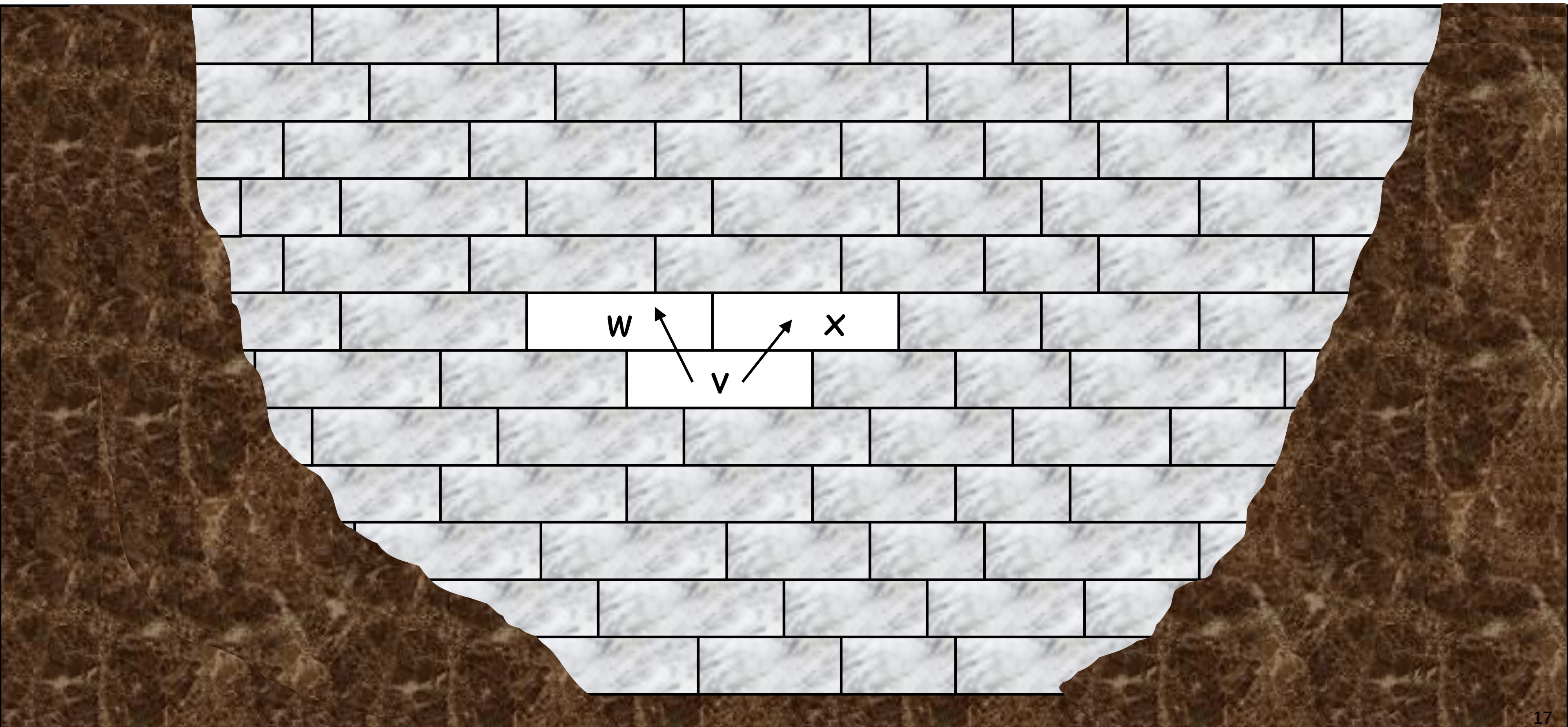
Exercise

- Write out the reduction as an algorithm in pseudocode that uses a min-cut subroutine.

Application: Open Pit Mining

Open-pit mining. (studied since early 1960s)

- Blocks of earth are extracted from surface to retrieve ore.
- Each block v has net value p_v = value of ore - processing cost.
- Can't remove block v before w or x.



7.7 Circulations

Circulation with Demands

Circulation with demands.

- Directed graph G = (V, E).
- Edge capacities c(e), e ∈ E.
- Node supply and demands d(v), v ∈ V.

demand if d(v) > 0; supply if d(v) < 0; transshipment if d(v) = 0

Def. A circulation is a function that satisfies:

- For each e ∈ E: $0 \leq f(e) \leq c(e)$ (capacity)
- For each v ∈ V: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

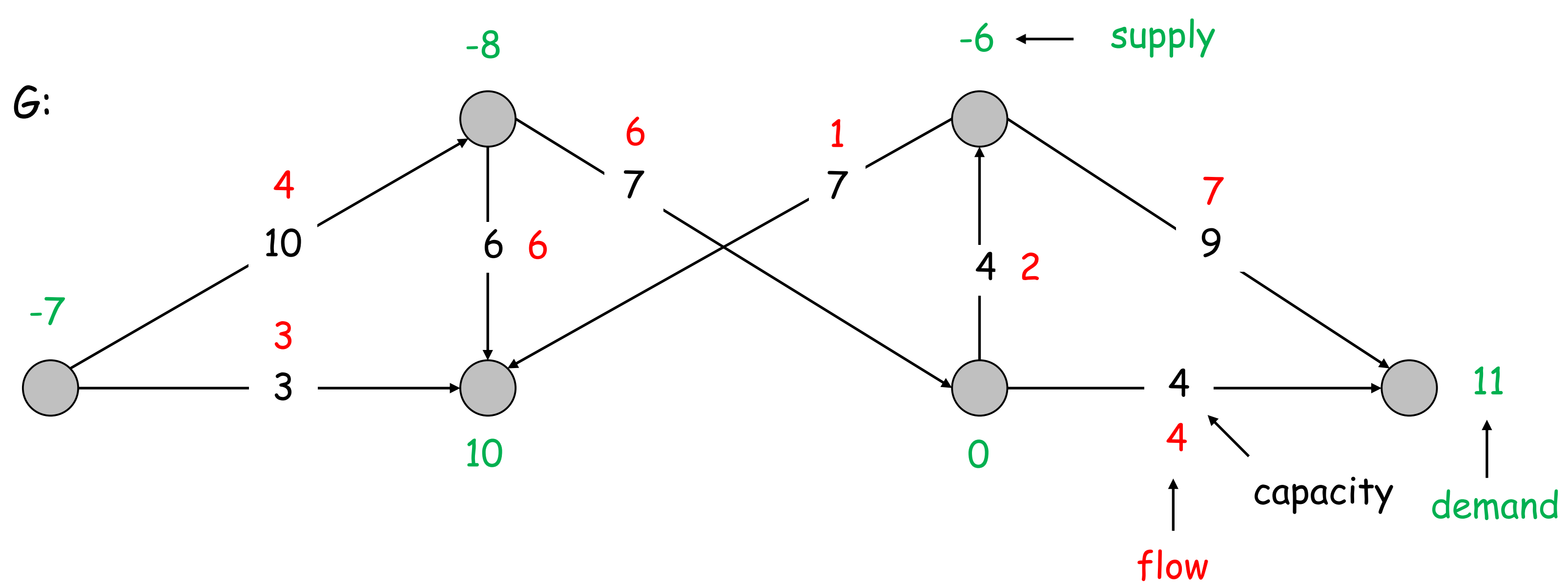
Circulation problem: given (V, E, c, d), does there exist a circulation?

Circulation with Demands

Necessary condition: sum of supplies = sum of demands.

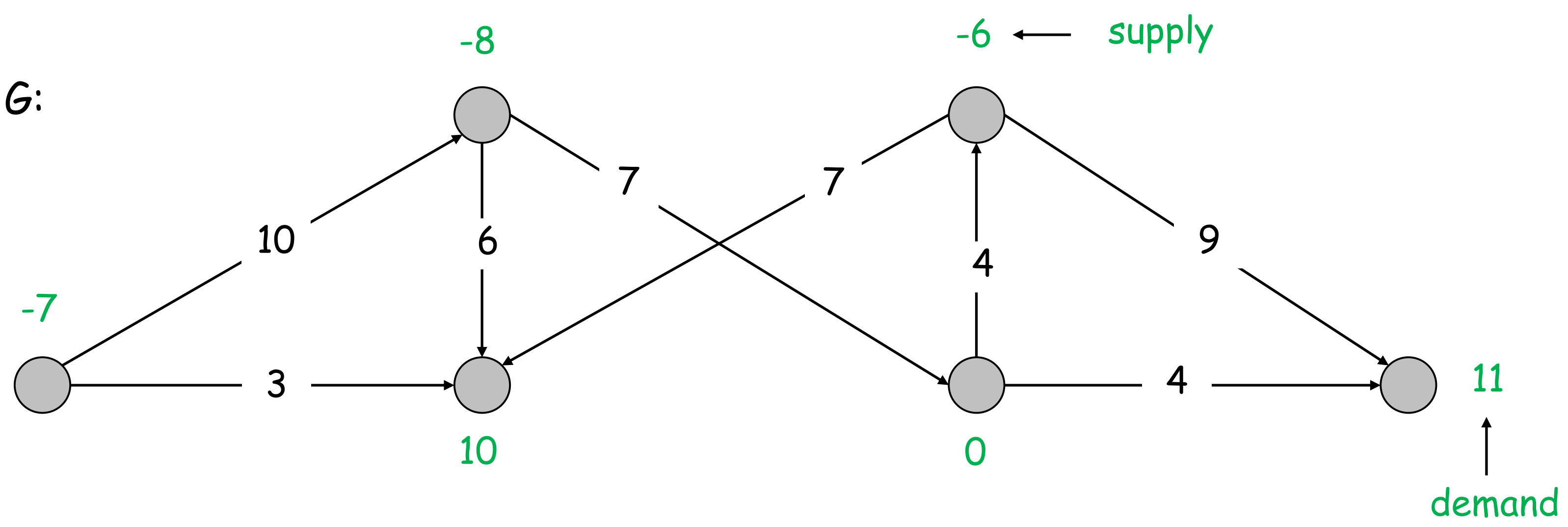
$$\sum_{v : d(v) > 0} d(v) = \sum_{v : d(v) < 0} -d(v) =: D$$

Proof. Sum conservation constraints for every demand node v.



Circulation with Demands

Max flow formulation.



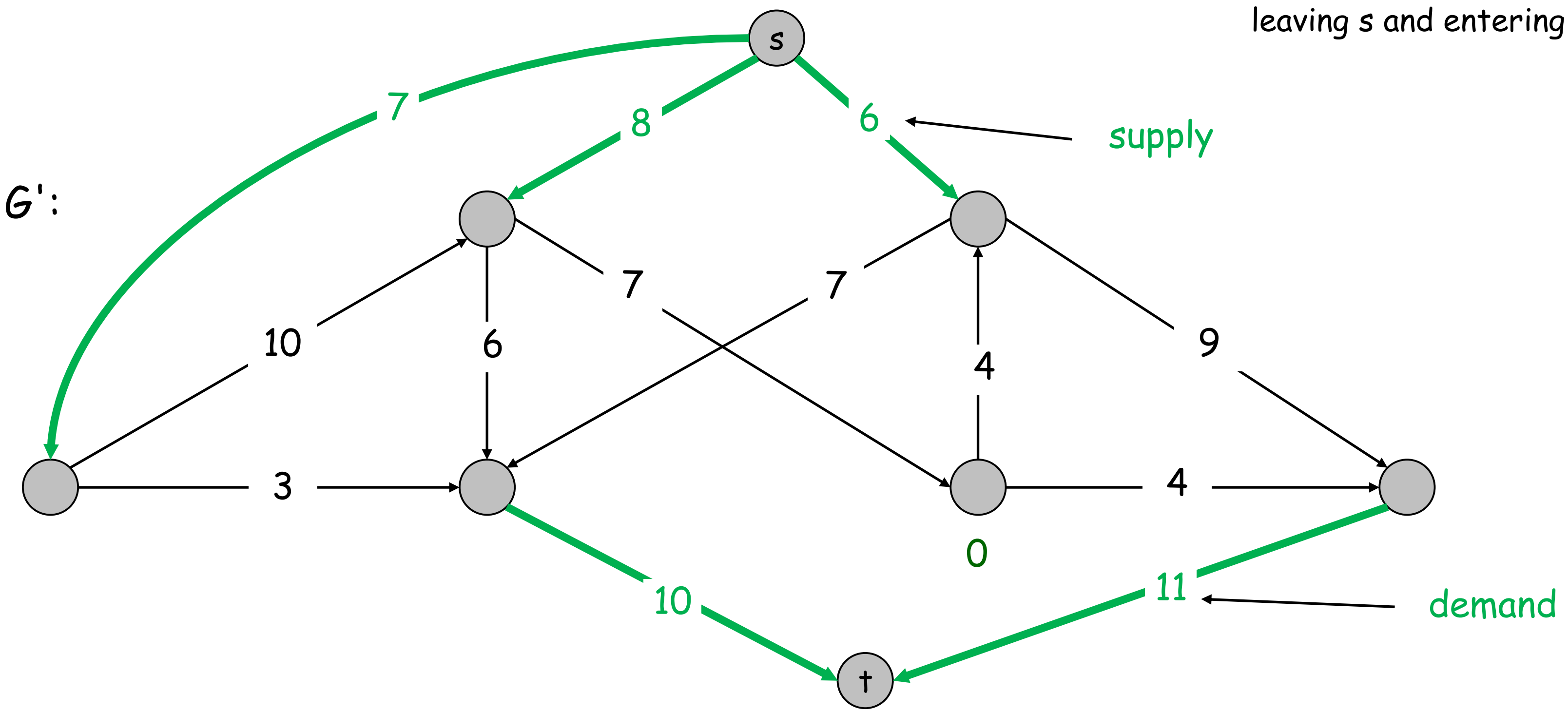
Circulation with Demands

Max flow formulation.

- Add new source s and sink t.
- For each v with $d(v) < 0$, add edge (s, v) with capacity $-d(v)$.
- For each v with $d(v) > 0$, add edge (v, t) with capacity $d(v)$.

Claim: G has circulation iff G' has max flow of value D.

saturates all edges leaving s and entering t



Circulation with Demands

Integrality theorem. If all capacities and demands are integers, and there exists a circulation, then there exists one that is integer-valued.

Proof. Follows from max flow formulation and integrality theorem for max flow.

Characterization. Given (V, E, c, d), there does not exist a circulation iff there exists a node partition (A, B) such that $\Sigma_{v \in B} d_v > cap(A, B)$

Effective demand by nodes in B exceeds max capacity of edges going from A to B

Proof idea. Look at min cut in G'.

Circulation with Demands and Lower Bounds

Feasible circulation.

- Directed graph $G = (V, E)$.
- Edge capacities $c(e)$ and lower bounds $\ell(e)$, $e \in E$.
- Node supply and demands $d(v)$, $v \in V$.

Def. A circulation is a function that satisfies:

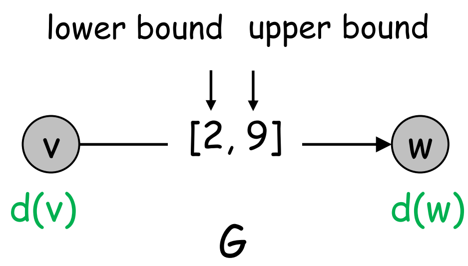
- For each $e \in E$: $\ell(e) \leq f(e) \leq c(e)$ (capacity)
- For each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation)

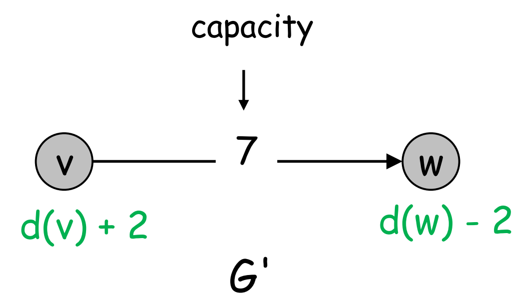
Circulation problem with lower bounds. Given (V, E, ℓ, c, d), does there exist a circulation?

Circulation with Demands and Lower Bounds

Idea. Model lower bounds with demands.

- Send $\ell(e)$ units of flow along edge e.
- Update demands of both endpoints.





Theorem. There exists a circulation in G iff there exists a circulation in G'. If all demands, capacities, and lower bounds in G are integers, then there is a circulation in G that is integer-valued.

Proof sketch. $f(e)$ is a circulation in G if and only if $f'(e) = f(e) - \ell(e)$ is a circulation in G'.

Census Tabulation (Exercise 7.39)

Feasible matrix rounding.

- Given a p-by-q matrix $D = \{d_{ij}\}$ of real numbers.
- Row i sum = a_i, column j sum b_j.
- Round each d_{ij}, a_i, b_j up or down to integer so that sum of rounded elements in each row (column) equals row (column) sum.
- Original application: publishing US Census data.

Goal. Find a feasible rounding, if one exists.

3.14	6.8	7.3	17.24
9.6	2.4	0.7	12.7
3.6	1.2	6.5	11.3
16.34	10.4	14.5	

original matrix

3	7	7	17
10	2	1	13
3	1	7	11
16	10	15	

feasible rounding

Census Tabulation

Feasible matrix rounding.

- Given a p-by-q matrix $D = \{d_{ij}\}$ of real numbers.
- Row i sum = a_i, column j sum b_j.
- Round each d_{ij}, a_i, b_j up or down to integer so that sum of rounded elements in each row (column) equals row (column) sum.
- Original application: publishing US Census data.

Goal. Find a feasible rounding, if one exists.

Remark. "Threshold rounding" can fail.

0.35	0.35	0.35	1.05
0.55	0.55	0.55	1.65
0.9	0.9	0.9	

original matrix

0	0	1	1
1	1	0	2
1	1	1	

feasible rounding

Census Tabulation

Theorem. Feasible matrix rounding always exists.

Proof. Formulate as a circulation problem with lower bounds.

- Original data provides circulation (all demands = 0).
- Integrality theorem $\Rightarrow$ integral solution $\Rightarrow$ feasible rounding. ▪

3.14	6.8	7.3	17.24
9.6	2.4	0.7	12.7
3.6	1.2	6.5	11.3
16.34	10.4	14.5