Advanced Algorithms

February 3, 2026

Logistics

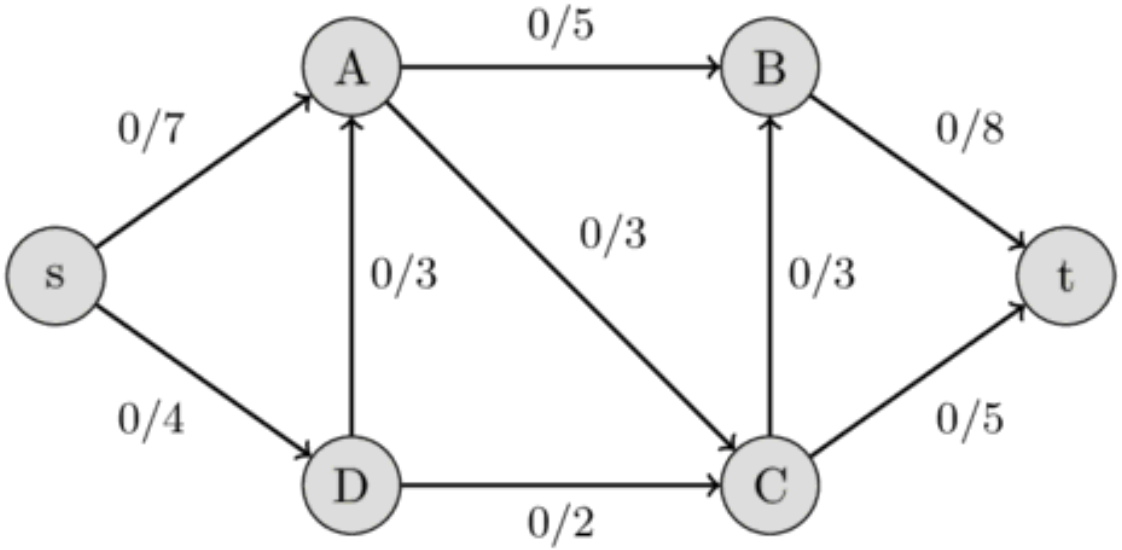
- Questions about Exercise 1?

- Exercise 2 out!
  - Due next week
  - Is shorter than usual because . . .

- Assignment 1 out!
  - Due in 2 weeks on gradescope
  - Work together, come to mentor sessions, office hours, etc.

Review

- Max-flow problem
- Ford-Fulkerson Algorithm
- Optimality Conditions



Graph cuts

- Given a flow network, an s, t – cut is any partition of the vertices into A, B where $s \in A$, $t \in B$.

- The capacity of a cut is:
  - Sum of capacities of all edges outgoing from A to B

- For *any* flow f and cut (A,B) we have value(f) $\leq$ capacity (A,B)

- In fact, the max-flow is always **equal** to the minimum capacity cut

Network flow: quiz 1

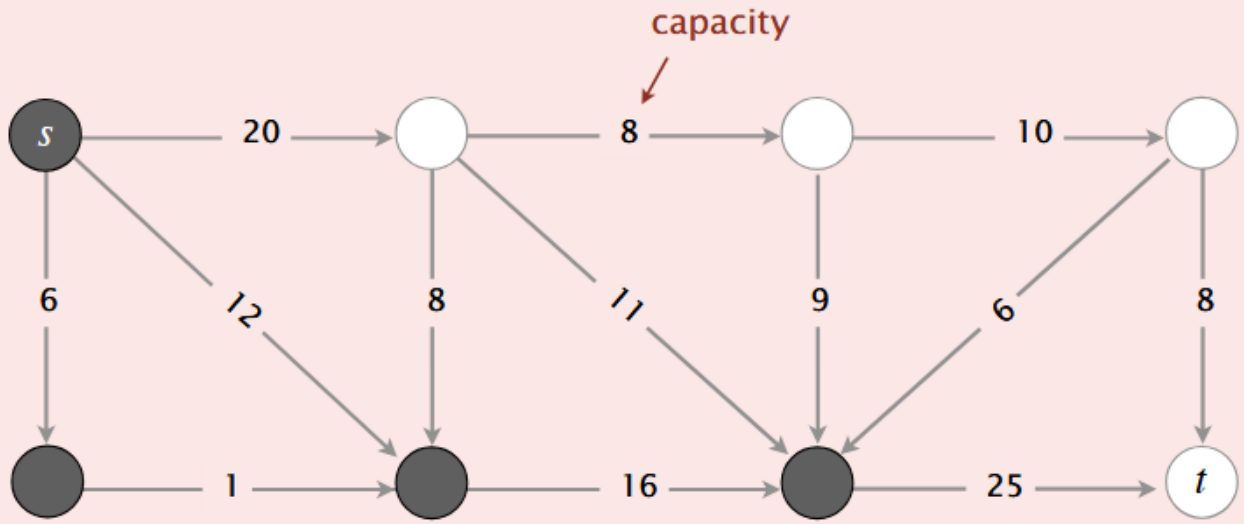
Which is the capacity of the given *st*-cut?

A. 11 (20 + 25 – 8 – 11 – 9 – 6)

B. 34 (8 + 11 + 9 + 6)

C. 45 (20 + 25)

D. 79 (20 + 25 + 8 + 11 + 9 + 6)



In the news!

NUMBER THEORY

Networks Hold the Key to a Decades-Old Problem About Waves

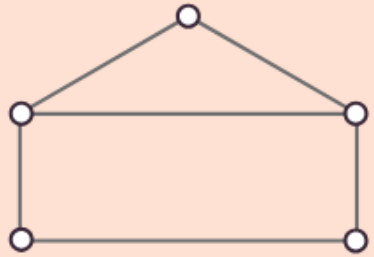
6 |

Mathematicians are still trying to understand fundamental properties of the Fourier transform, one of their most ubiquitous and powerful tools. A new result marks an exciting advance toward that goal.

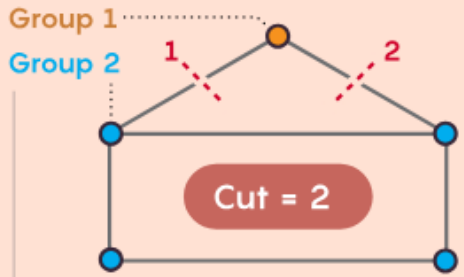
January 28, 2026

To Cut a Graph

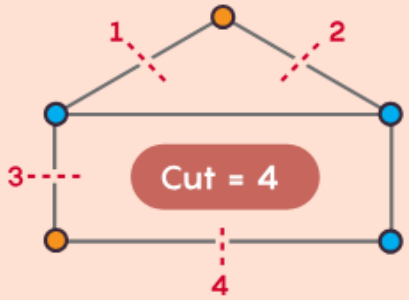
The "MaxCut" problem asks you to split a graph's nodes into two groups so that as many edges connect the groups as possible. The problem has applications in physics, engineering, and other fields.



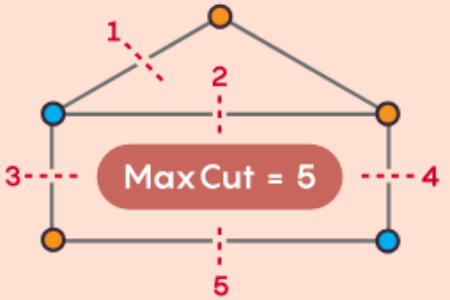
There are many different ways to split this graph's nodes into two groups.



Here's one way. Count how many edges connect nodes from different groups. These edges form the graph's "cut."



Here's another way. More edges connect the two groups.

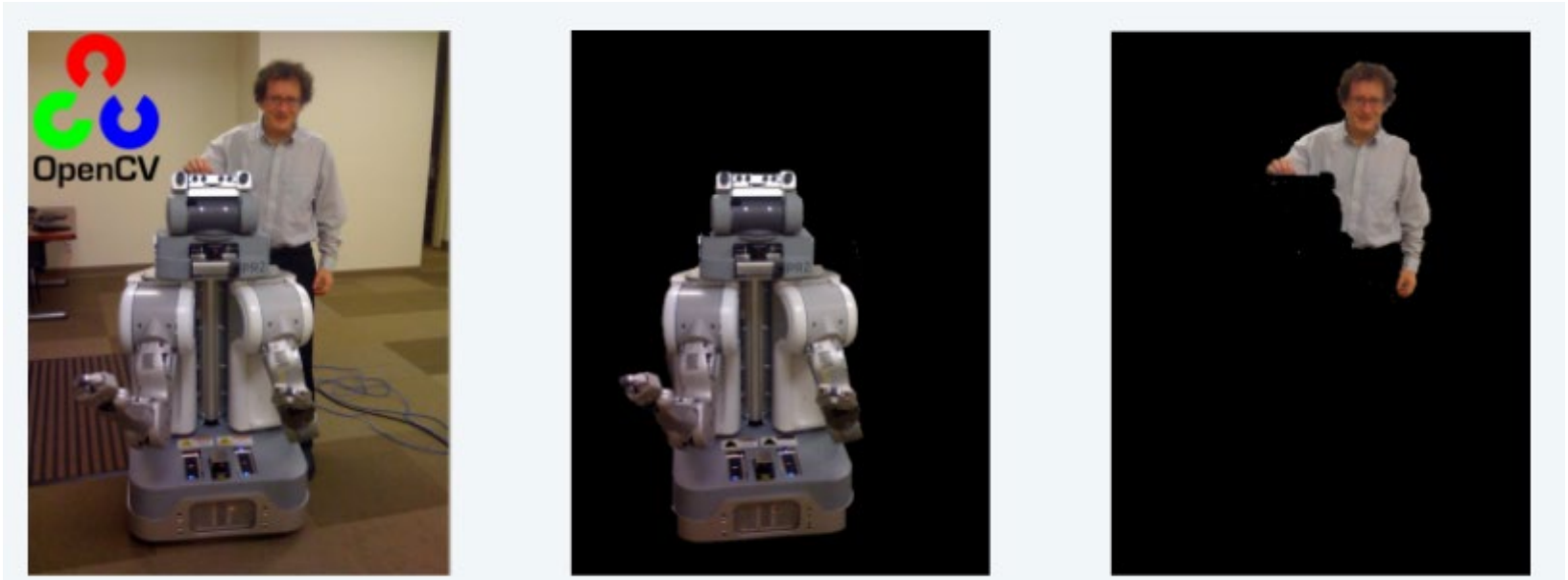


The way of partitioning nodes that involves the greatest number of edges is called the MaxCut.

Image Segmentation

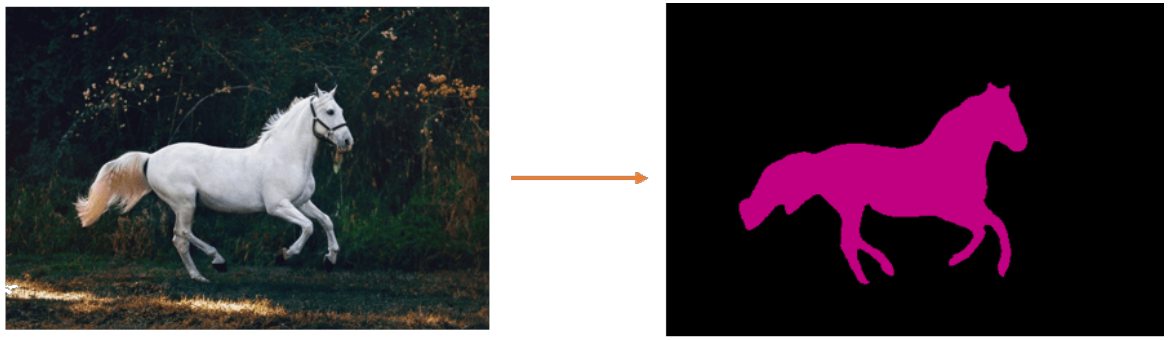
Image segmentation: separate the pixels of an image into distinct shapes corresponding to contiguous figures

Central problem in image processing



Foreground / background segmentation

As a start: distinguish the foreground from the background



Problem formulation

- Form groups of 3-4

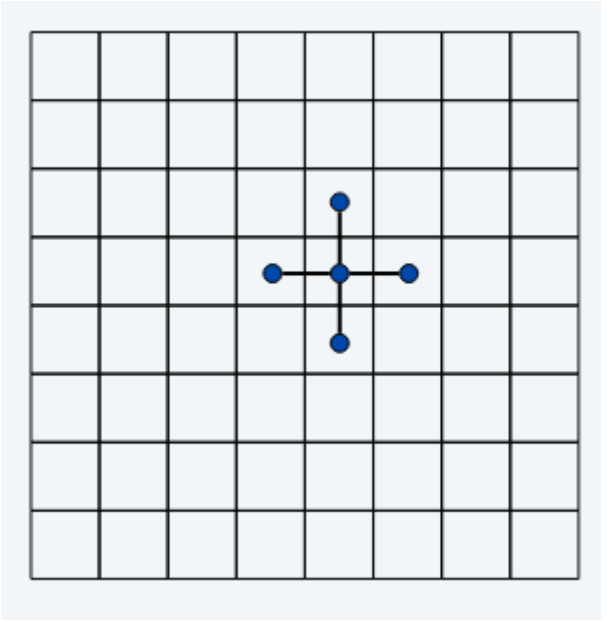
- Fuzzy problem: Given an image, decide which pixels are in the foreground and which are in the background

- Your task: model this as an optimization problem
  - What is the input to your problem? I.e. what do you need from the image
  - What is a feasible solution?
  - How to compare feasible solutions?

Themes?

An approach

- We will consider each pixel in the image as a vertex.
- Each pixel (node) is adjacent to its “neighbors”
- These could be the pixels immediately to the left, right, up and down of it.
- Confidence of foreground / background given, as well as penalty for separation.



An approach

Goals:

- Accuracy: if $a_i > b_i$, prefer to label i in foreground.
- Smoothness: if many neighbors of i are labeled foreground, we should be inclined to label i as foreground.

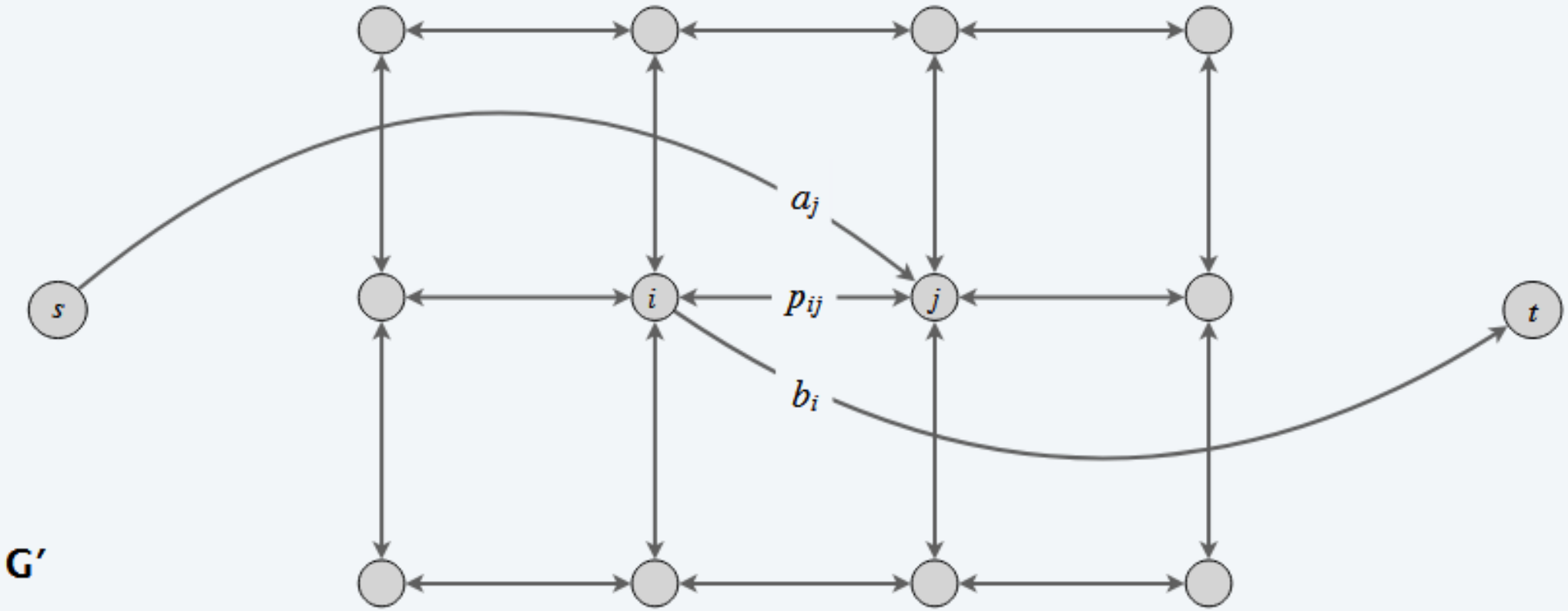
What is a good objective function?

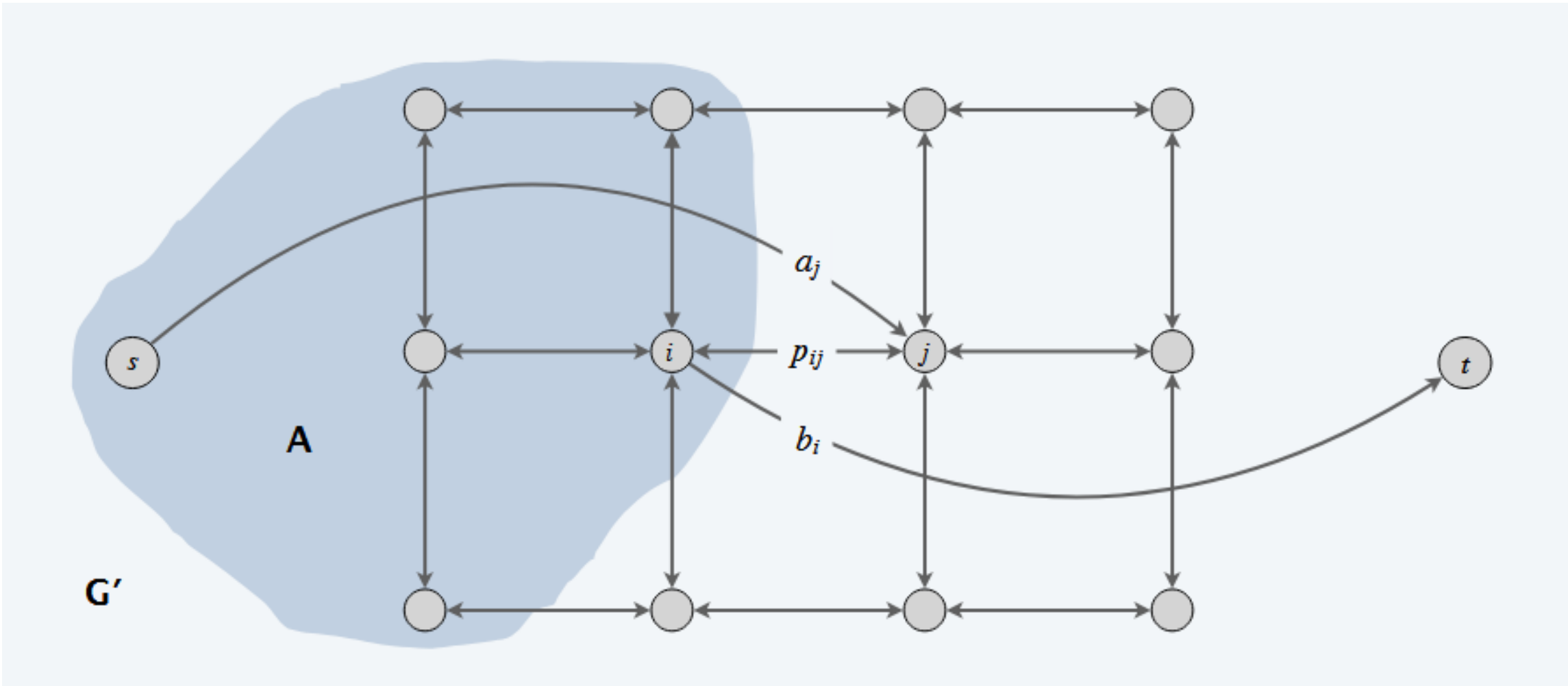
Find a partition (A, B) maximizing: $\sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{i \in A, b \in B, ij \in E} p_{ij}$

The Minimum Cut Problem

- Similarities?

- Differences?


a_j
p_{ij}
b_i
s
i
j
t
G′


a_j
p_{ij}
b_i
s
i
j
t
A
G′