CS 365, Lecture 6
Foundations of Data Science
Boston University

Charalampos E. Tsourakakis

February 8th, 2022

Bayesian inference in the real-world

Reminder: Bayes' theorem

Bayes' theorem (aka Bayes' Law and Bayes' rule) is a direct application of conditional probabilities.

$$\mathbf{Pr}[H|D] = \frac{\mathbf{Pr}[D|H]\mathbf{Pr}[H]}{\mathbf{Pr}[D]}, \text{and } \mathbf{Pr}[D] > 0, \text{or } \ldots$$

posterior $\propto$ likelihood $\times$ prior.

Exact MAP Estimation for Binary Images

Problem: Given (b) can we infer (a)? In other words, can we **restore** the image from its corrupted-by-noise version?

GREIG, PORTEOUS AND SEHEULT

(a) (b)

How to formulate the problem? Any ideas?

Exact MAP Estimation for Binary Images

Let's be Bayesian!

$x = (x_1, \ldots, x_n)$ the original image (shown in (a))

$y = (y_1, \ldots, y_n)$ the observed corrupted image (e.g., the one shown in (b))

Assumption: The records $y_1, \ldots, y_n$ are conditionally independent given x, and each has known conditional density $f(y_i|x_i)$ that depends only on x_i.

By Bayes' theorem:

$$p(x|y) \propto \underbrace{p(y|x)}_{\text{likelihood:how do we compute it?}} \times \underbrace{p(x)}_{\text{prior:what is a good prior?}}$$

Goal: output

$$\boxed{x^* = \arg\max p(x|y)}$$

Likelihood and Prior

Given our assumption, the likelihood function is

$$p(y|x) = \prod_{i=1}^{n} f(y_i|1)^{x_i} f(y_i|0)^{1-x_i}.$$

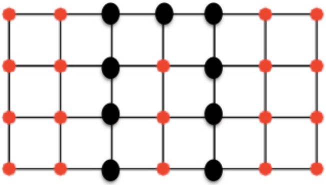
- What kind of patterns would we like the prior to enforce?
- Let's imagine how these characters would look on a binary image:

  a,b,c,f,y,x,z,1,&,$,@

  Homogeneous patches that occasionally change discontinuously

Prior $p(x)$

$$p(x) \propto \exp\left\{\frac{1}{2}\sum_{i\neq j}\beta_{ij}\big(x_i x_j + (1-x_i)(1-x_j)\big)\right\}$$



- For an edge (u, v) where $val(u) = val(v)$

$$x_u x_v + (1-x_v)(1-x_u) = x_u^2 + (1-x_u)^2 = 1.$$

- On the contrary for an edge where $val(u) \neq val(v)$

$$x_u(1-x_u) + (1-x_u)x_u = 0.$$

Exact MAP Estimation for Binary Images

Our MAP inference becomes equivalent to minimizing (details on whiteboard)

$$\sum_{i=1}^{n} x_i \max(0, -\lambda_i) + \sum_{i=1}^{n} \max(0, \lambda_i)(1 - x_i) + \frac{1}{2} \sum_{i \sim j} \beta_{ij}(x_i - x_j)^2,$$

where $\lambda_i = \frac{f(y_i|1)}{f(y_i|0)}$.

Let's rephrase this problem. Suppose $b_{ij} = b$ for all neighboring nodes for simplicity.

Exact MAP Estimation for Binary Images

- We have a $n \times m$ binary matrix
- We impose a grid structure
- We call two neighboring nodes bad if they have different values. We pay K units for each such pair.
- We are allowed to flip the value of any node, but we have to pay R units.
- The total cost is the sum of these two terms. How do we find the best assignment of values to nodes?

Any ideas? Is it NP-hard, poly-time solvable?

Exact MAP Estimation for Binary Images

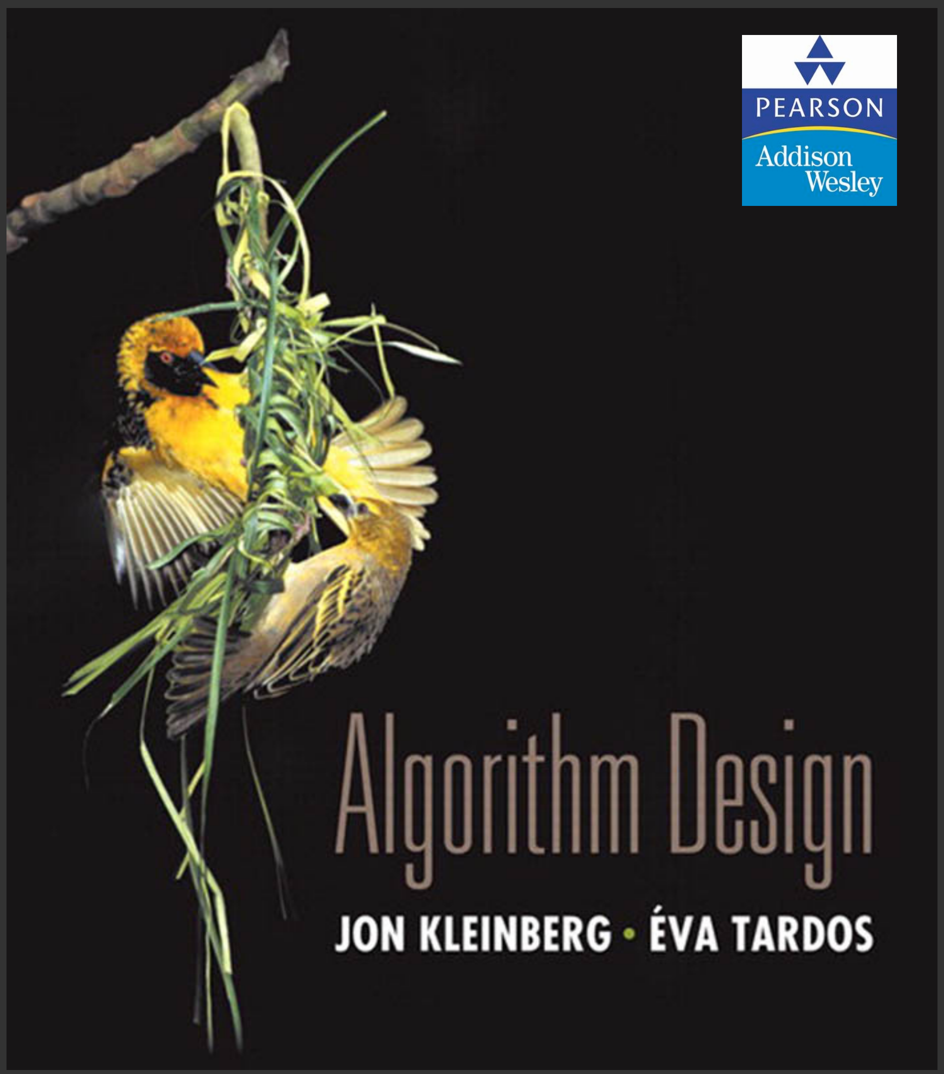
Max flow problem!

- Source s, sink t
- Arc of capacity R from s to each node u with value 0.
- Arc of capacity R from each u node with value 1 to sink t.
- Directed arcs from each node u to its neighbors with capacity K.

Details on whiteboard.

Assigned reading: Exact maximum a posteriori estimation for binary images

`https://github.com/tsourolampis/cs365-spring22/blob/main/greig-porteous-seheult.pdf`



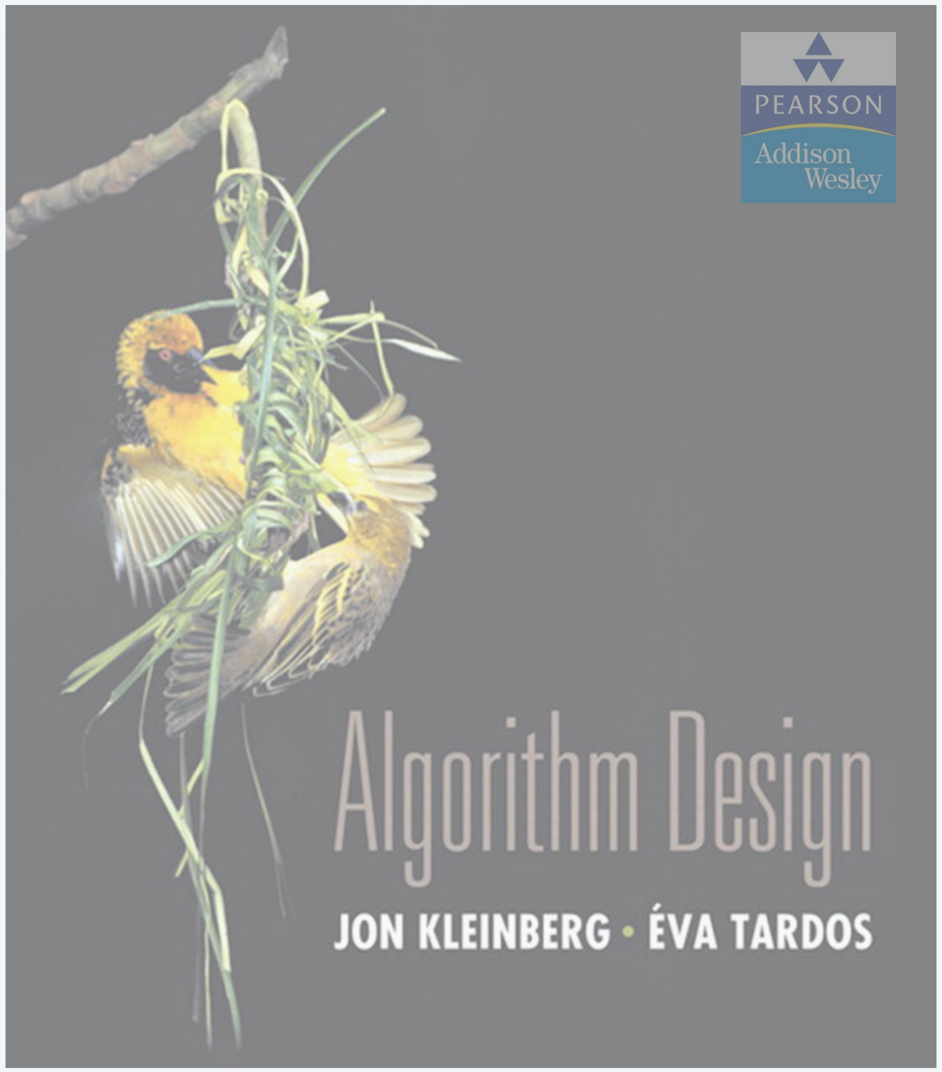
Lecture slides by Kevin Wayne

http://www.cs.princeton.edu/~wayne/kleinberg-tardos

7. Network Flow I

- *max-flow and min-cut problems*
- *Ford–Fulkerson algorithm*
- *max-flow min-cut theorem*
- *capacity-scaling algorithm*
- *shortest augmenting paths*
- *Dinitz' algorithm*
- *simple unit-capacity networks*

Last updated on 11/1/21 4:31 PM



SECTION 7.1

7. NETWORK FLOW I

- *max-flow and min-cut problems*
- *Ford–Fulkerson algorithm*
- *max-flow min-cut theorem*
- *capacity-scaling algorithm*
- *shortest augmenting paths*
- *Dinitz' algorithm*
- *simple unit-capacity networks*

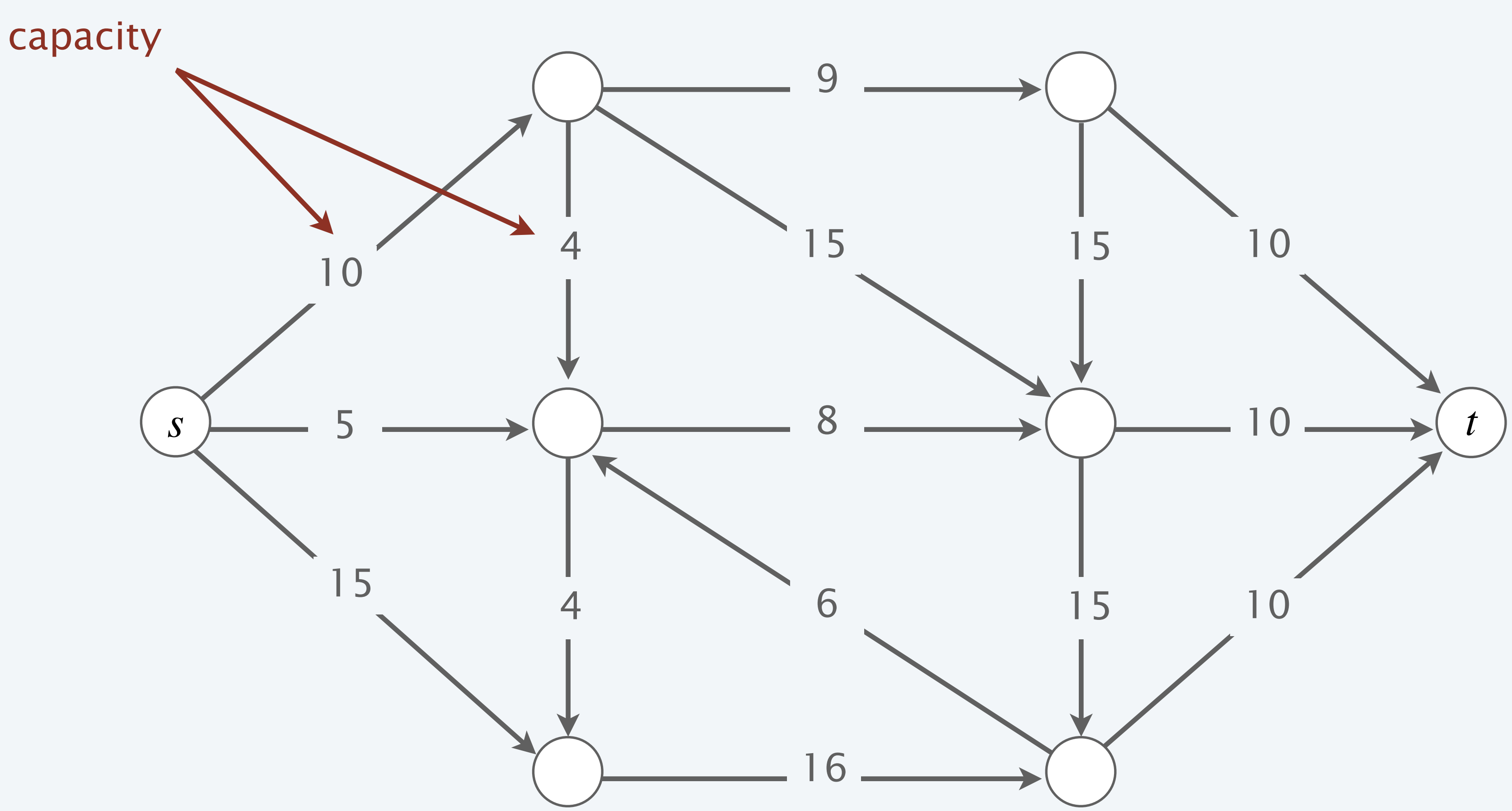
Flow network

A flow network is a tuple $G = (V, E, s, t, c)$.

- Digraph (V, E) with source $s \in V$ and sink $t \in V$.
- Capacity $c(e) \geq 0$ for each $e \in E$.

assume all nodes are reachable from s

Intuition. Material flowing through a transportation network; material originates at source and is sent to sink.

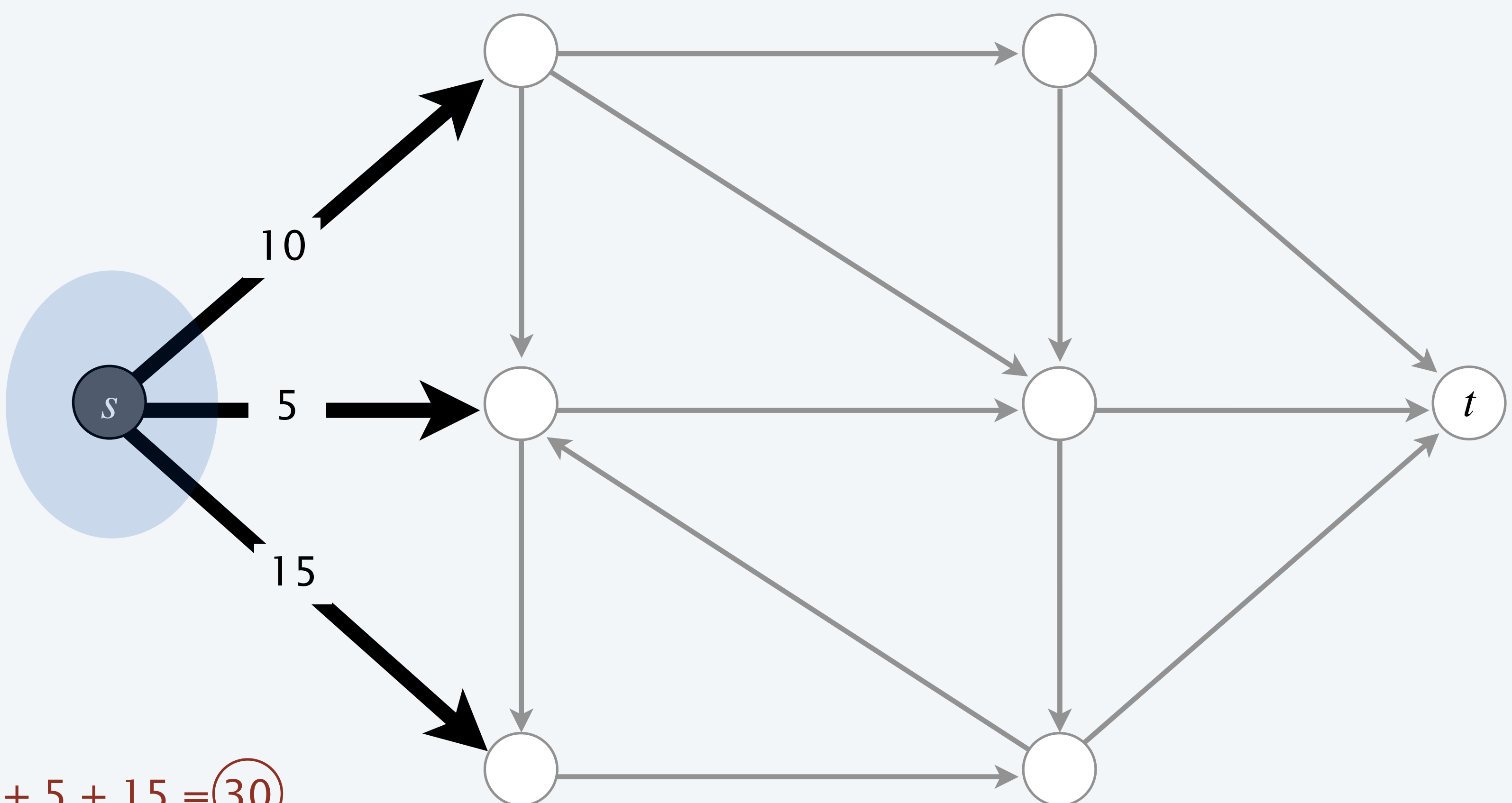


Minimum-cut problem

Def. An *st*-cut (cut) is a partition (A, B) of the nodes with $s \in A$ and $t \in B$.

Def. Its capacity is the sum of the capacities of the edges from A to B.

$$cap(A,B) = \sum_{e \text{ out of } A} c(e)$$



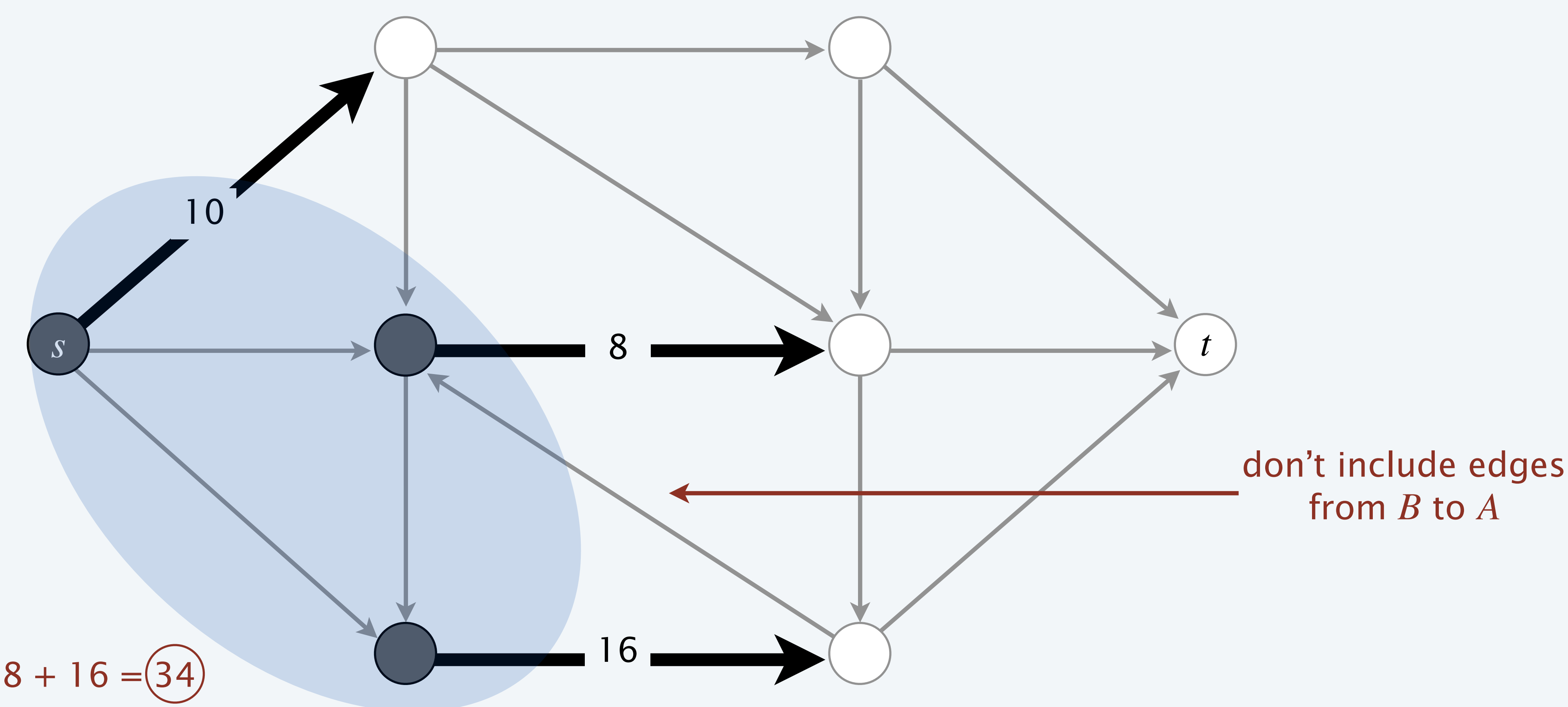
capacity = 10 + 5 + 15 = 30

Minimum-cut problem

Def. An st-cut (cut) is a partition (A, B) of the nodes with $s \in A$ and $t \in B$.

Def. Its capacity is the sum of the capacities of the edges from A to B.

$$cap(A, B) = \sum_{e \text{ out of } A} c(e)$$



capacity = 10 + 8 + 16 = 34

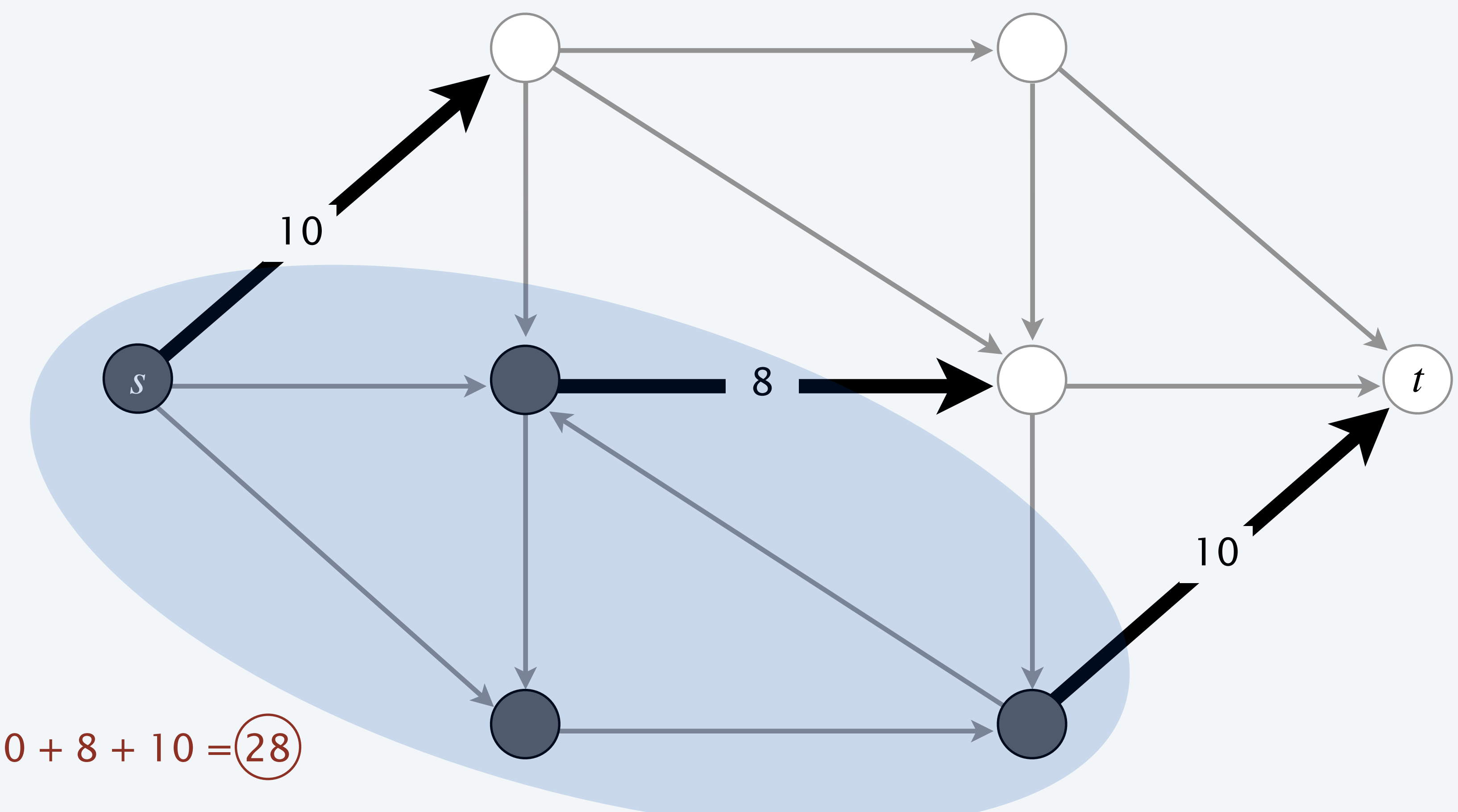
Minimum-cut problem

Def. An *st*-cut (cut) is a partition (A, B) of the nodes with $s \in A$ and $t \in B$.

Def. Its capacity is the sum of the capacities of the edges from A to B.

$$cap(A, B) = \sum_{e \text{ out of } A} c(e)$$

Min-cut problem. Find a cut of minimum capacity.



capacity = 10 + 8 + 10 = 28

Network flow: quiz 1

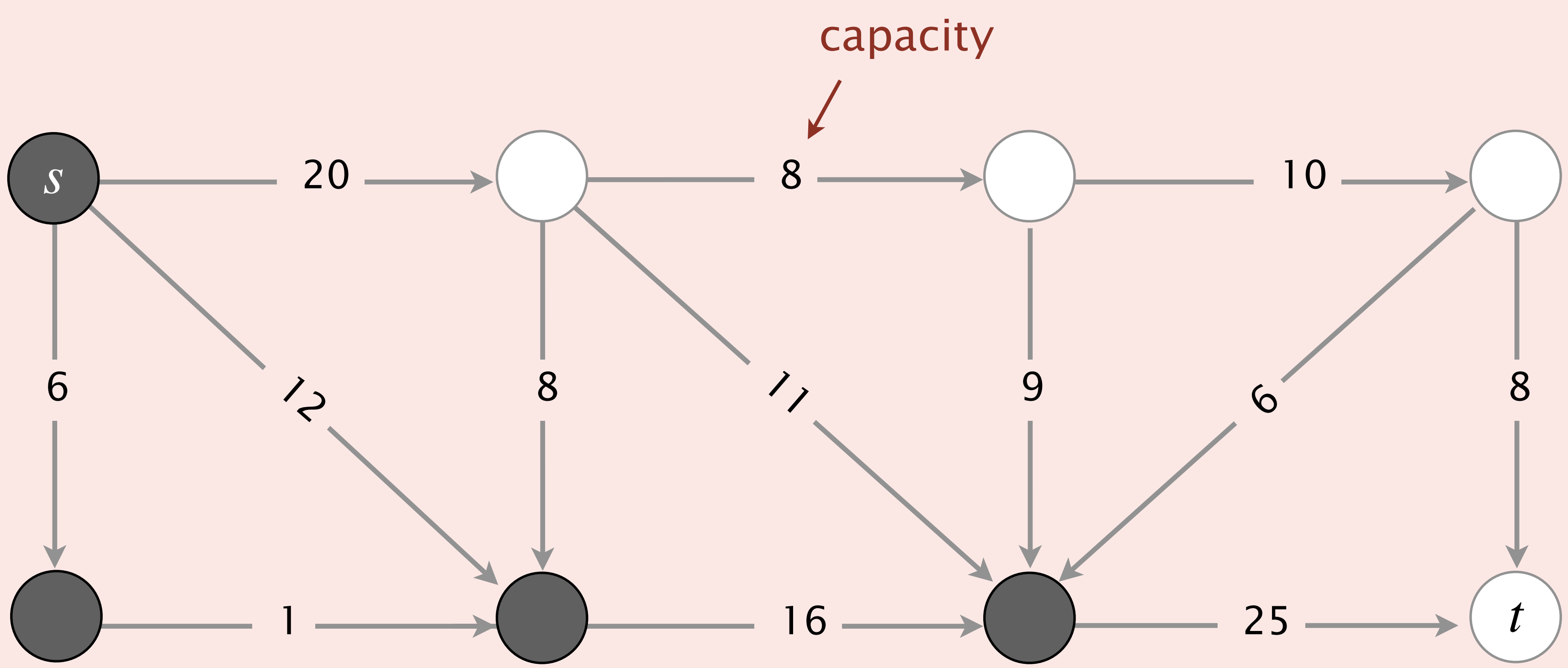
Which is the capacity of the given *st*-cut?

A. 11 (20 + 25 − 8 − 11 − 9 − 6)

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

C. 45 (20 + 25)

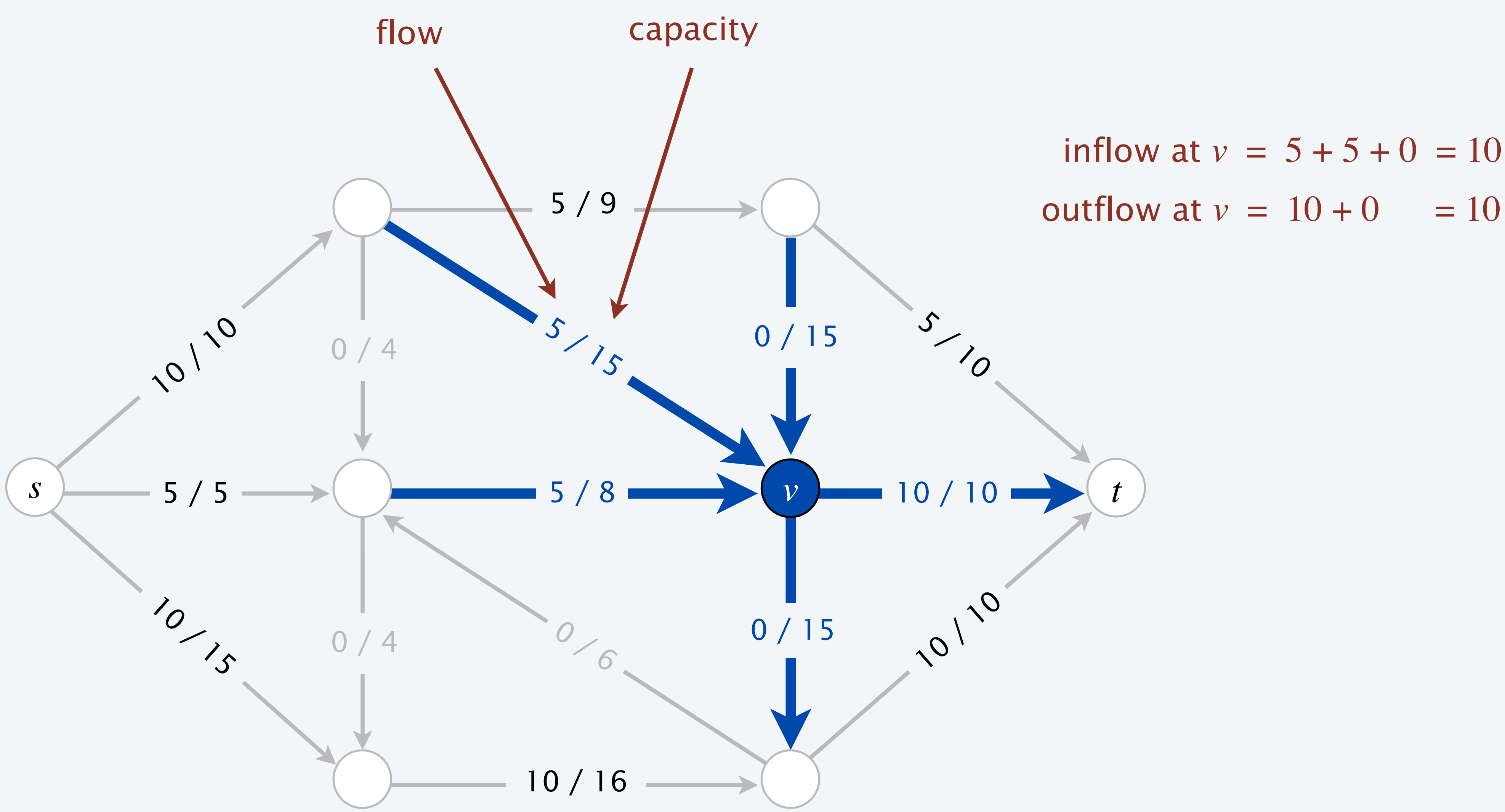
D. 79 (20 + 25 + 8 + 11 + 9 + 6)



Maximum-flow problem

Def. An *st*-flow (flow) f is a function that satisfies:

- For each $e \in E$: $\quad 0 \le f(e) \le c(e) \quad$ [capacity]
- For each $v \in V - \{s, t\}$: $\quad \sum_{e \text{ in to } v} f(e) = \sum_{e \text{ out of } v} f(e) \quad$ [flow conservation]

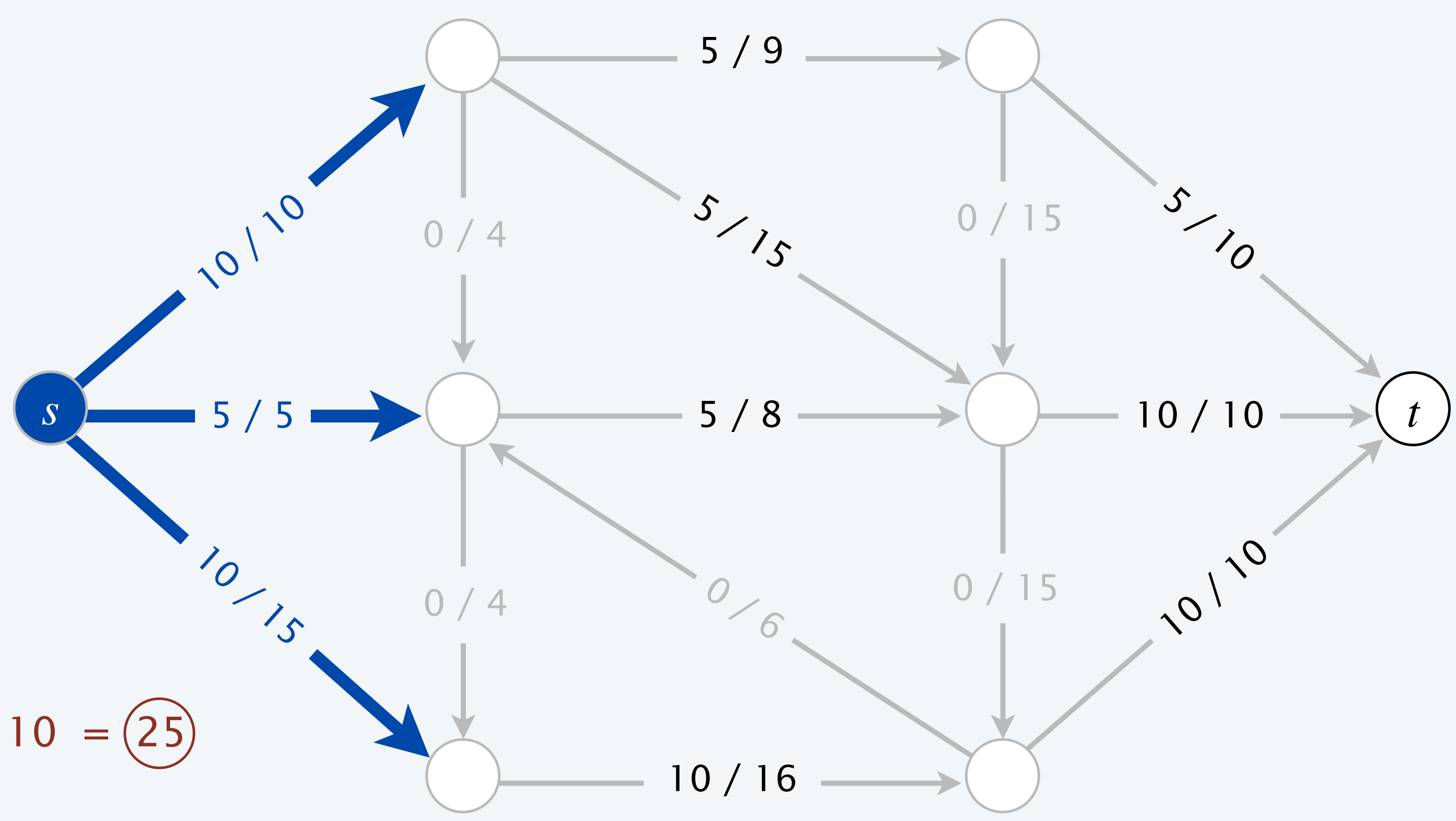


Maximum-flow problem

Def. An *st*-flow (flow) f is a function that satisfies:

- For each $e \in E$: $\quad 0 \le f(e) \le c(e)$ [capacity]
- For each $v \in V - \{s, t\}$: $\quad \sum_{e \text{ in to } v} f(e) = \sum_{e \text{ out of } v} f(e)$ [flow conservation]

Def. The value of a flow f is: $val(f) = \sum_{e \text{ out of } s} f(e) - \sum_{e \text{ in to } s} f(e)$



value = 5 + 10 + 10 = 25

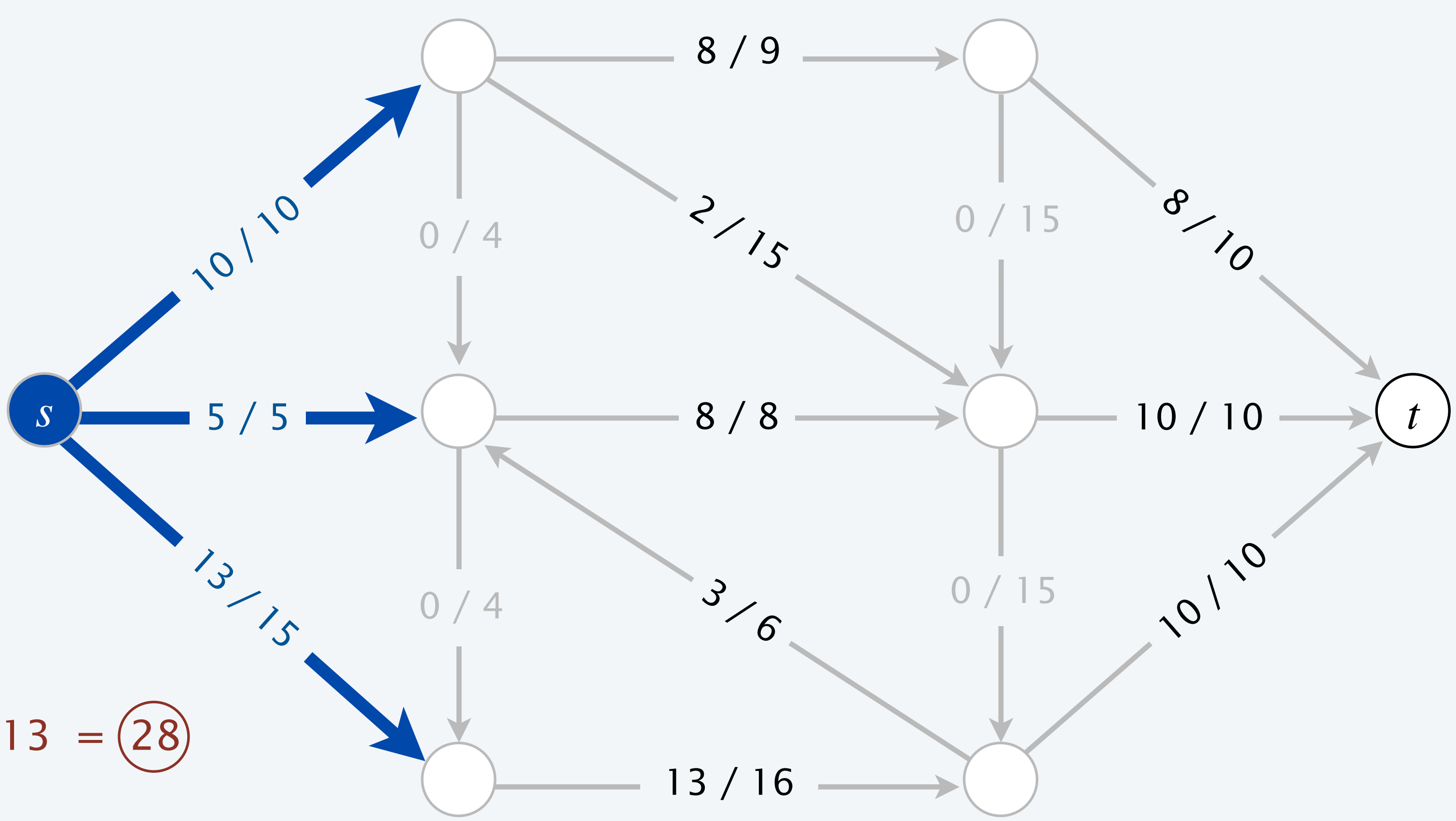
Maximum-flow problem

Def. An *st*-flow (flow) f is a function that satisfies:

- For each $e \in E$: $0 \leq f(e) \leq c(e)$ [capacity]
- For each $v \in V - \{s, t\}$: $\sum_{e \text{ in to } v} f(e) = \sum_{e \text{ out of } v} f(e)$ [flow conservation]

Def. The value of a flow f is: $val(f) = \sum_{e \text{ out of } s} f(e) - \sum_{e \text{ in to } s} f(e)$

Max-flow problem. Find a flow of maximum value.



value = 10 + 5 + 13 = 28

Max-flow min-cut theorem

Max-flow min-cut theorem. Value of a max flow = capacity of a min cut.

strong duality

MAXIMAL FLOW THROUGH A NETWORK

L. R. FORD, JR. AND D. R. FULKERSON

Introduction. The problem discussed in this paper was formulated by T. Harris as follows:

"Consider a rail network connecting two cities by way of a number of intermediate cities, where each link of the network has a number assigned to it representing its capacity. Assuming a steady state condition, find a maximal flow from one given city to the other."

ON THE MAX FLOW MIN CUT THEOREM OF NETWORKS

G. B. Dantzig
D. R. Fulkerson

P-826

April 15, 1955

A Note on the Maximum Flow Through a Network*

P. ELIAS†, A. FEINSTEIN‡, AND C. E. SHANNON§

Summary—This note discusses the problem of maximizing the rate of flow from one terminal to another, through a network which consists of a number of branches, each of which has a limited capacity. The main result is a theorem: The maximum possible flow from left to right through a network is equal to the minimum value among all simple cut-sets. This theorem is applied to solve a more general problem, in which a number of input nodes and a number of output nodes are used.

from one terminal to the other in the original network passes through at least one branch in the cut-set. In the network above, some examples of cut-sets are (d, e, f), and (b, c, e, g, h), (d, g, h, i). By a *simple cut-set* we will mean a cut-set such that if any branch is omitted it is no longer a cut-set. Thus (d, e, f) and (b, c, e, g, h) are simple