CSCE 658: Randomized Algorithms

Lecture 16

Samson Zhou

Relevant Supplementary Material

- Chapter 29 in "Introduction to Algorithms", by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein

- Chapters 5.1-5.5 in "The Design of Approximation Algorithms", by David P. Williamson and David B. Shmoys

Linear Programming (Standard Form)

- Maximize a linear objective function:

$$c^\top x = \langle c, x \rangle, \ \ c, x \in \mathbb{R}^n$$

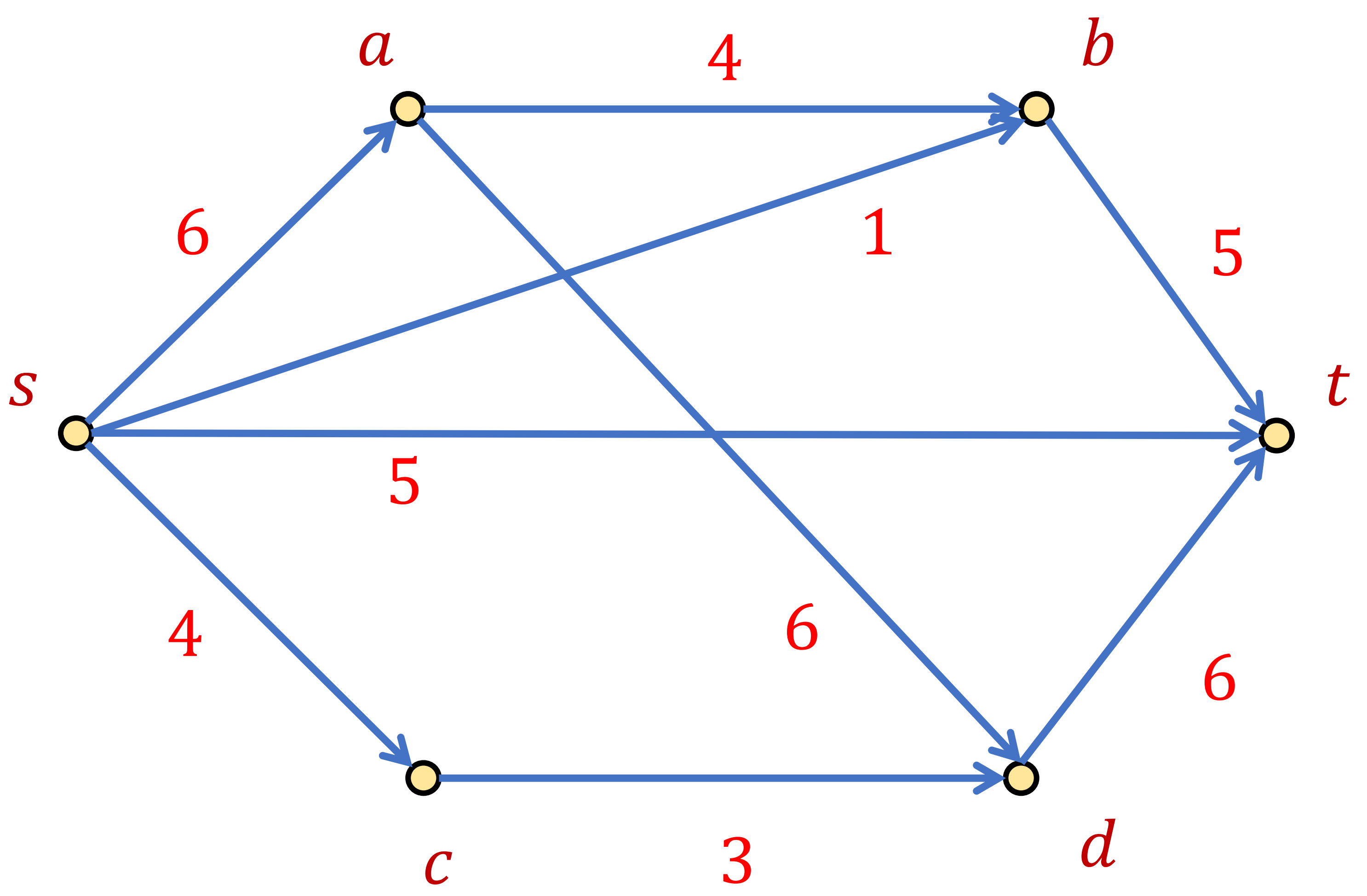
- Subject to constraints:

$$Ax \leq b \text{ for } A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$$
$$x \geq 0 \text{ (entry-wise non-negativity)}$$

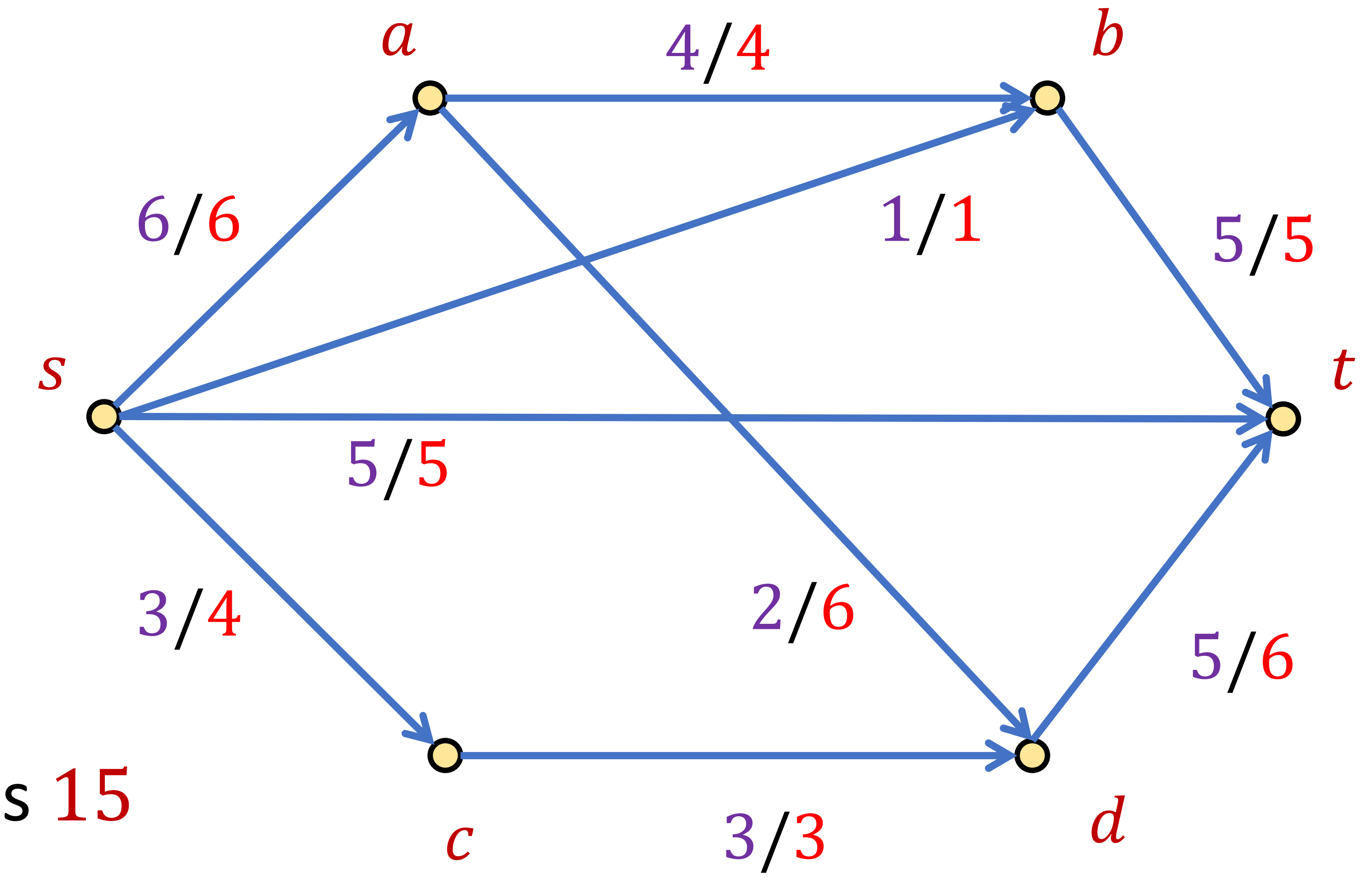
Max $s - t$ Flow in a Directed Graph

- Input: A directed graph $G = (V, E)$, capacities $c_{(u,v)}$ for each edge $(u, v) \in E$, source vertex s, and sink vertex t
- A *flow* is assignment of weights to edges so that:
  - Capacity constraint: the flow of an edge does not exceed its capacity
  - Conservation of flow: sum of flows entering a node equals sum of flows exiting a node, except for s and t
- Goal: Route as much flow as possible from s to t

Max $s-t$ Flow in a Directed Graph



Max $s - t$ Flow in a Directed Graph



Max flow is 15

Linear Program for Max $s - t$ Flow

- What variables do we want?

- Flow $f_{(u,v)}$ for each edge (u, v)

- What constraints do we want?
- Capacity constraint, conservation of flow

Linear Program for Max $s-t$ Flow

- Maximize: $\sum_{v:(s,v)\in E} f_{(s,v)} - \sum_{v:(v,s)\in E} f_{(v,s)}$

- Subject to:

$$f_{(u,v)} \geq 0 \text{ for all } (u,v) \in E$$
$$f_{(u,v)} \leq c_{(u,v)} \text{ for all } (u,v) \in E$$
$$\sum_{u:(u,v)\in E} f_{(u,v)} = \sum_{w:(v,w)\in E} f_{(v,u)} \text{ for all } v \neq s,t$$

Dual Program for Max $s-t$ Flow

- Minimize: $\sum_{v:(u,v)\in E} c_{(u,v)} d_{(u,v)}$, where $d_{(u,v)}$ indicates whether (u,v) crosses the cut, $c_{(u,v)}$ is the capacity of (u,v)

- Subject to:

$$d_{(u,v)} \geq 0 \text{ for all } (u,v) \in E$$

$$d_{(u,v)} - z_u + z_v \geq 0 \text{ for all } (u,v) \in E, u \neq s, v \neq t$$

$$d_{(s,v)} + z_v \geq 1 \text{ for all } (s,v) \in E$$

$$d_{(u,t)} - z_u \geq 0 \text{ for all } (u,t) \in E$$

Cuts

- A cut $C = S_1, S_2$ of a graph G is a partition of the vertices V into a set S_1 and the remaining vertices $S_2 = V - S_1$

- An edge (u, v) crosses the cut C if $u \in S_1$ and $v \in S_2$

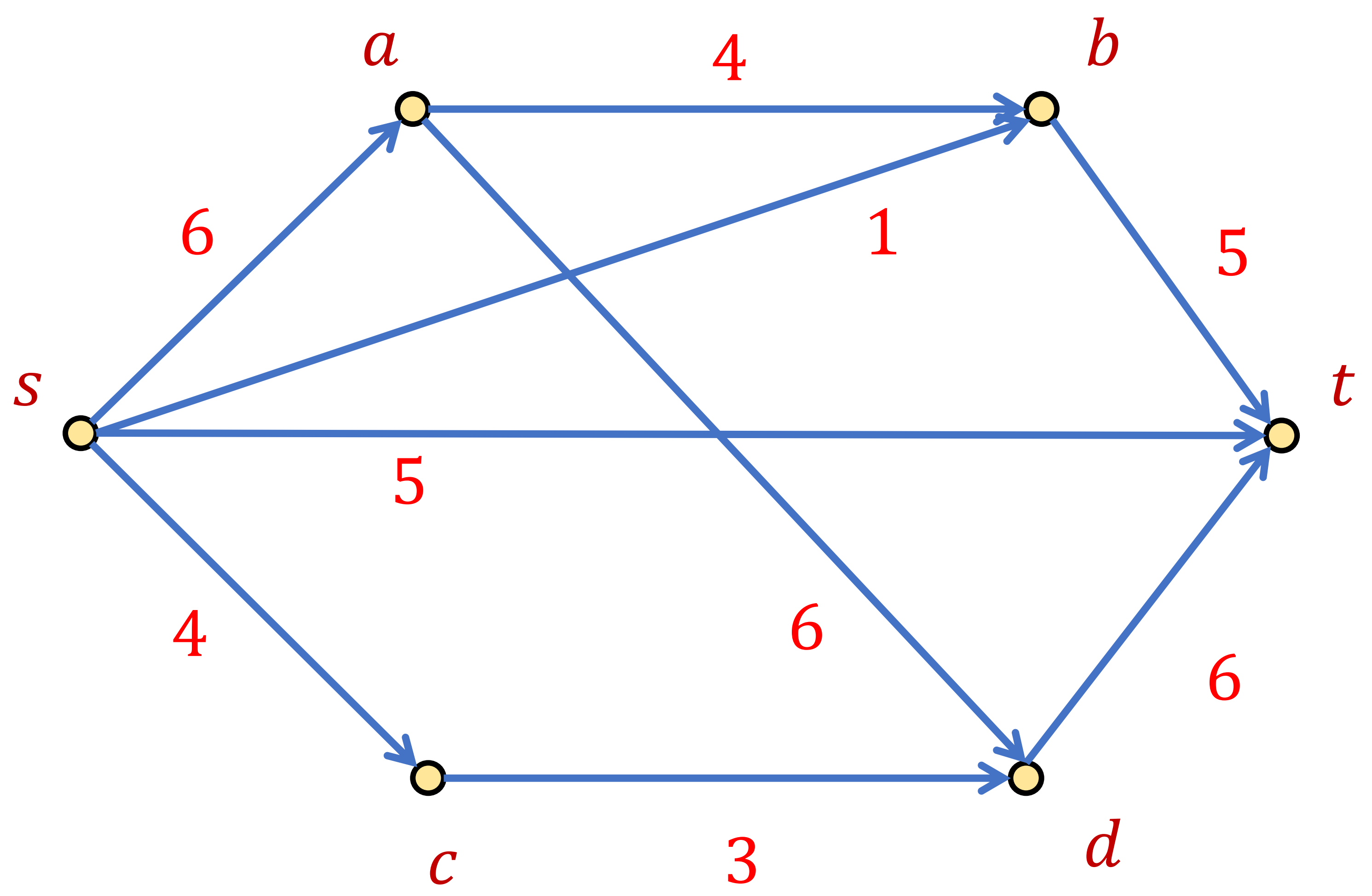
- The size of the cut C is the number of edges that cross C

Minimum $s - t$ Cut

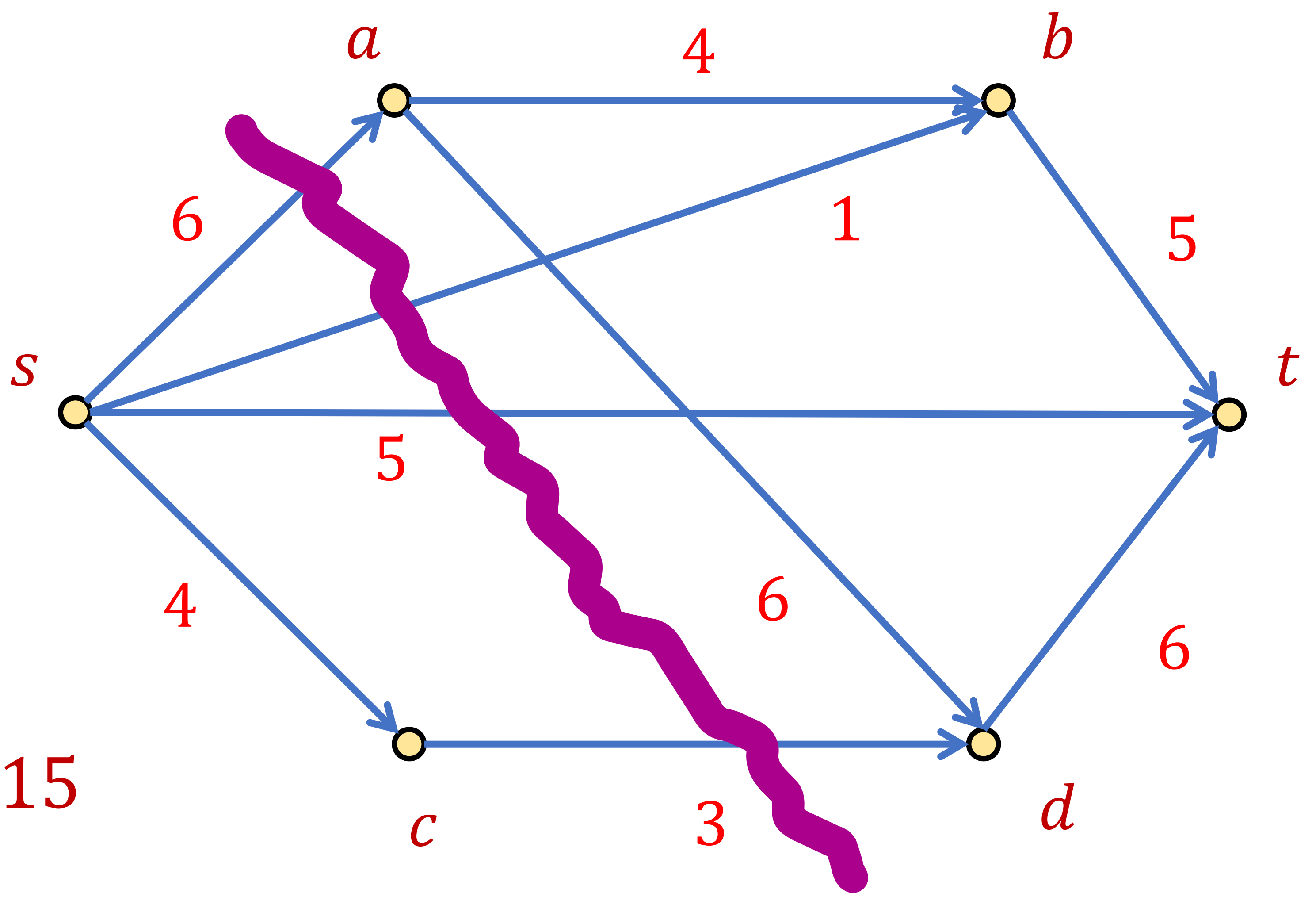
- The minimum cut of a graph is the size of the smallest cut across all pairs of sets of vertices S_1 and $S_2 = V - S_1$

- Find the minimum cut of a graph G that separates s and t

What is the minimum $s-t$ cut of the graph?



What is the minimum $s - t$ cut of the graph?



Min cut is 15

Linear Program for Min $s - t$ Cut

- What variables do we want?

- Variables $d_{(u,v)}$ for each edge (u, v) indicating whether it crosses the cut

- Set $d_{(u,v)} \geq z_u - z_v, z_v - z_u$, where $z_u \in \{0,1\}$ indicates whether u is on the side of s
- Need $d_{(s,v)} \geq 1 - z_v, d_{(u,t)} \geq z_u$

Linear Program for Min $s-t$ Cut

- Minimize: $\sum_{v:(u,v)\in E} c_{(u,v)} d_{(u,v)}$, where $d_{(u,v)}$ indicates whether (u,v) crosses the cut, $c_{(u,v)}$ is the capacity of (u,v)

- Subject to:

$$d_{(u,v)} \geq 0 \text{ for all } (u,v) \in E$$
$$z_u \in \{0,1\} \text{ for all } u \in V$$
$$d_{(u,v)} - z_u + z_v \geq 0 \text{ for all } (u,v) \in E, u \neq s, v \neq t$$
$$d_{(s,v)} + z_v \geq 1 \text{ for all } (s,v) \in E$$
$$d_{(u,t)} - z_u \geq 0 \text{ for all } (u,t) \in E$$

Dual Program for Max $s - t$ Flow

- Minimize: $\sum_{v:(u,v)\in E} c_{(u,v)} d_{(u,v)}$, where $d_{(u,v)}$ indicates whether (u, v) crosses the cut, $c_{(u,v)}$ is the capacity of (u, v)
- Subject to:

$$d_{(u,v)} \geq 0 \text{ for all } (u, v) \in E$$

$$d_{(u,v)} - z_u + z_v \geq 0 \text{ for all } (u, v) \in E, u \neq s, v \neq t$$

$$d_{(s,v)} + z_v \geq 1 \text{ for all } (s, v) \in E$$

$$d_{(u,t)} - z_u \geq 0 \text{ for all } (u, t) \in E$$

Min Cut-Max Flow Theorem?

- Recall: the max-flow min-cut theorem states the maximum flow through any graph between any fixed source and sink is exactly equal to the minimum cut

- However, the dual LP to the max-flow problem is a fractional problem, while the LP for the min-cut problem requires integral solutions

Linear Programming (Standard Form)

- Maximize a linear objective function:

$$c^\top x = \langle c, x \rangle,\ \ c, x \in \mathbb{R}^n$$

- Subject to constraints:

$$Ax \le b \text{ for } A \in \mathbb{R}^{m \times n},\ b \in \mathbb{R}^m$$
$$x \ge 0 \text{ (entry-wise non-negativity)}$$

Integer Linear Programming (Standard Form)

- Maximize a linear objective function:

$$c^\top x = \langle c, x \rangle, \ c \in \ \mathbb{R}^n, x \in \mathbb{Z}^n$$

- Subject to constraints:

$$Ax + s = b \text{ for } A \in \mathbb{R}^{m \times n}, s, b \in \mathbb{R}^m$$
$$s, x \geq 0 \text{ (entry-wise non-negativity)}$$

Integer Linear Programming (Standard Form)

- Integer linear programming is NP-hard (solves vertex cover)

- When constraint is $Ax = b$, the matrix A and the vector b all have integer entries, and A is totally unimodular (every square submatrix has determinant $-1,0,1$), then the vertices of the LP polytope are integers
- Can use standard LP algorithms

MAX-SAT Revisited

- In the MAX-SAT problem, the input is a CNF formula $f(x_1, \ldots, x_n)$ with m clauses $C_1, \ldots., C_m$

- The goal is to assign values to $x_1, \ldots, x_n$ to maximize the number of satisfied clauses

MAX-SAT Revisited

- Suppose we assign each variable x_i a separate random TRUE/FALSE value
- For each $i \in [m]$, we have $\Pr[C_i \text{ is FALSE}] \leq 1/2$
- By a linearity of expectation, the expected number of satisfied clauses is at least $m/2$

- Random assignment gives (at least) a $\frac{1}{2}$-approximation in expectation

Derandomization of MAX-SAT

- How to get an algorithm that achieves $\frac{1}{2}$?

- Method of conditional expectation
  - Set x_1 to be the value with the higher conditional expectation
  - Random assignment is a $\frac{1}{2}$-approximation in expectation, so there is a value of x_1 that is a $\frac{1}{2}$-approximation in expectation
  - Iterate

Better Algorithm for MAX-SAT

- First suppose there is no unit clause $\overline{x_i}$ (will remove assumption later)

- Set each x_i to be TRUE with probability $p > 1/2$ independently

Better Algorithm for MAX-SAT

- Claim: The probability that any given clause is satisfied is $\min(p, 1-p^2)$
- $\min(p, 1-p^2)$ is maximized ≈ 0.618 for $p = \frac{1}{2}(\sqrt{5}-1)$
- If there is no unit clause $\overline{x_i}$, there is a ≈ 0.618 approximation algorithm for MAX-SAT

MAX-SAT Revisited (Integer Program)

- Maximize: $\sum_{j \in [m]} Z_j$
- Subject to:

$$\sum_{i: x_i \in C_j} y_i + \sum_{i: \overline{x_i} \in C_j} (1 - y_i) \geq Z_j \text{ for all } j \in [m]$$

$$Z_j \in \{0,1\} \text{ for all } j \in [m]$$

$$y_i \in \{0,1\} \text{ for all } i \in [n]$$

MAX-SAT Revisited (LP Relaxation)

- Maximize: $\sum_{j \in [m]} Z_j$

- Subject to:

$$\sum_{i: x_i \in C_j} y_i + \sum_{i: \overline{x_i} \in C_j} (1 - y_i) \geq Z_j \text{ for all } j \in [m]$$
$$0 \leq Z_j \leq 1 \text{ for all } j \in [m]$$
$$0 \leq y_i \leq 1 \text{ for all } i \in [n]$$

Randomized Rounding for MAX-SAT

- Let y_i^* and z_j^* be the optimal solution to the LP relaxation
- Set $x_i = 1$ with probability y_i^*

- $\Pr[C_j \text{ is not satisfied}] = \prod_{i \in P_j}(1 - y_i^*) \prod_{i \in N_j} y_i^*$, where we split clause C_j into positive literals P_j and negative literals N_j

Randomized Rounding for MAX-SAT

$$\Pr\left[C_j \text{ is not satisfied}\right] = \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^*$$

(AM-GM)

$$\leq \left[\frac{1}{|C_j|}\left(\sum_{i \in P_j} (1 - y_i^*) + \sum_{i \in N_j} y_i^*\right)\right]^{|C_j|}$$

$$= \left[1 - \frac{1}{|C_j|}\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*)\right)\right]^{|C_j|}$$

$$= \left[1 - \frac{z_j^*}{|C_j|}\right]^{|C_j|}$$

Randomized Rounding for MAX-SAT

$$\Pr[C_j \text{ is satisfied}] \geq 1 - \left[1 - \frac{z_j^*}{|C_j|}\right]^{|C_j|}$$

(concavity) $$\geq 1 - \left[1 - \frac{1}{|C_j|}\right]^{|C_j|} z_j^*$$

$$\geq \left(1 - \frac{1}{e}\right) z_j^*$$

Randomized Rounding for MAX-SAT

- Let y_i^* and z_j^* be the optimal solution to the LP relaxation
- Set $x_i = 1$ with probability y_i^*

- $\left(1 - \frac{1}{e}\right) \approx 0.6321$-approximation algorithm

MAX-SAT Summary

- Random assignment gives ≈ 0.618-approximation

$$\Pr[C_j \text{ is satisfied}] \geq \left(1 - 2^{-|C_j|}\right) \geq z_j^* \left(1 - 2^{-|C_j|}\right)$$

- Randomized rounding gives ≈ 0.6321-approximation

$$\Pr[C_j \text{ is satisfied}] \geq z_j^* \left(1 - \left[1 - \frac{1}{|C_j|}\right]^{|C_j|}\right)$$

MAX-SAT Summary

- Random assignment gives ≈ 0.618-approximation

$$\Pr[C_j \text{ is satisfied}] \geq \left(1 - 2^{-|C_j|}\right) \geq z_j^* \left(1 - 2^{-|C_j|}\right)$$

- Randomized rounding gives ≈ 0.6321-approximation

$$\Pr[C_j \text{ is satisfied}] \geq z_j^* \left(1 - \left[1 - \frac{1}{|C_j|}\right]^{|C_j|}\right)$$

MAX-SAT Summary

- Random assignment gives ≈ 0.618-approximation

$$\Pr\left[C_j \text{ is satisfied}\right] \geq \left(1 - 2^{-|C_j|}\right) \geq z_j^* \left(1 - 2^{-|C_j|}\right)$$

- Randomized rounding gives ≈ 0.6321-approximation

$$\Pr\left[C_j \text{ is satisfied}\right] \geq z_j^* \left(1 - \left[1 - \frac{1}{|C_j|}\right]^{|C_j|}\right)$$

- How do these behave across values of $|C_j|$?

MAX-SAT Summary

- When $|C_j|$ is small, $\left(1-2^{-|C_j|}\right)$ is small and $\left(1-\left[1-\frac{1}{|C_j|}\right]^{|C_j|}\right)$ is large
- When $|C_j|$ is large, $\left(1-2^{-|C_j|}\right)$ is large and $\left(1-\left[1-\frac{1}{|C_j|}\right]^{|C_j|}\right)$ is small

Choosing the Better of Two Solutions

- Run randomized rounding and random assignment and take the better of the two solutions

$$\Pr[C_j \text{ is satisfied}] \geq z_j^* \max\left(\left(1-2^{-|C_j|}\right), \left(1-\left[1-\frac{1}{|C_j|}\right]^{|C_j|}\right)\right)$$

- We have $\max(\mathrm{a}, \mathrm{b}) \geq \frac{a+b}{2}$

Choosing the Better of Two Solutions

- Run randomized rounding and random assignment and take the better of the two solutions

$$\Pr[C_j \text{ is satisfied}] \geq z_j^* \cdot \frac{1}{2}\left(\left(1 - 2^{-|C_j|}\right) + \left(1 - \left[1 - \frac{1}{|C_j|}\right]^{|C_j|}\right)\right)$$

- For $|C_j| = 1, \frac{1}{2}\left(\left(1 - 2^{-|C_j|}\right) + \left(1 - \left[1 - \frac{1}{|C_j|}\right]^{|C_j|}\right)\right) = \frac{3}{4}$

Choosing the Better of Two Solutions

- Run randomized rounding and random assignment and take the better of the two solutions

$$\Pr\left[C_j \text{ is satisfied}\right] \geq z_j^* \cdot \frac{1}{2}\left(\left(1 - 2^{-|C_j|}\right) + \left(1 - \left[1 - \frac{1}{|C_j|}\right]^{|C_j|}\right)\right)$$

- For $\left|C_j\right| = 2, \frac{1}{2}\left(\left(1 - 2^{-|C_j|}\right) + \left(1 - \left[1 - \frac{1}{|C_j|}\right]^{|C_j|}\right)\right) = \frac{3}{4}$

Choosing the Better of Two Solutions

- For $|C_j| \geq 3$:

$$\frac{1}{2}\left(\left(1-2^{-|C_j|}\right)+\left(1-\left[1-\frac{1}{|C_j|}\right]^{|C_j|}\right)\right) \geq \frac{1}{2}\left(1-\frac{1}{e}\right)+\frac{1}{2}\cdot\frac{7}{8}$$

$$\approx 0.753 \geq \frac{3}{4}$$

Choosing the Better of Two Solutions

- Run randomized rounding and random assignment and take the better of the two solutions

$$\Pr\left[C_j \text{ is satisfied}\right] \geq \frac{3}{4} \cdot z_j^*$$

- By linearity of expectation, $\frac{3}{4}$-approximation algorithm

Nonlinear Randomized Rounding for MAX-SAT

- Let y_i^* and z_j^* be the optimal solution to the LP relaxation
- Previously: Set $x_i = 1$ with probability y_i^*
- What if we set $x_i = 1$ with probability $f(y_i^*)$?

- $\Pr[C_j \text{ is not satisfied}] = \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$, where we split clause C_j into positive literals P_j and negative literals N_j

Nonlinear Randomized Rounding for MAX-SAT

- $\Pr[C_j \text{ is not satisfied}] = \prod_{i \in P_j} (1 - f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$
- Suppose we set $1 - 4^{-x} \leq f(x) \leq 4^{x-1}$
- $\Pr[C_j \text{ is not satisfied}] = \prod_{i \in P_j} 4^{-y_i^*} \prod_{i \in N_j} 4^{y_i^* - 1}$

$$= 4^{-\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*)\right)}$$

$$\leq 4^{-z_j^*}$$

- $\Pr[C_j \text{ is satisfied}] \geq \frac{1}{4}$

Nonlinear Randomized Rounding for MAX-SAT

- Let y_i^* and z_j^* be the optimal solution to the LP relaxation
- Set $x_i = 1$ with probability $f(y_i^*)$

- By linearity of expectation, $\frac{3}{4}$-approximation algorithm