CS7800: Advanced Algorithms

Lecture 9: Generalizing Network Flow

- Minimum cost bipartite matching

Jonathan Ullman

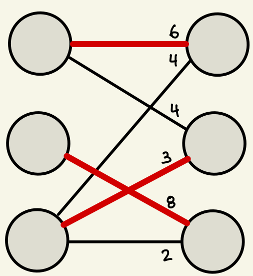
10-07-22

Minimum Cost Perfect Matching

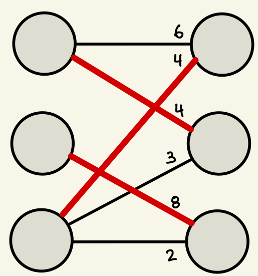
Input: A bipartite graph $G = (L \cup R, E)$ and edge costs $\{c(e)\}$

① $c(e) \geq 0$ ② $|L| = |R|$ ③ G has a perfect matching

Output: A perfect matching M of minimum cost $\sum_{e \in M} c(e)$



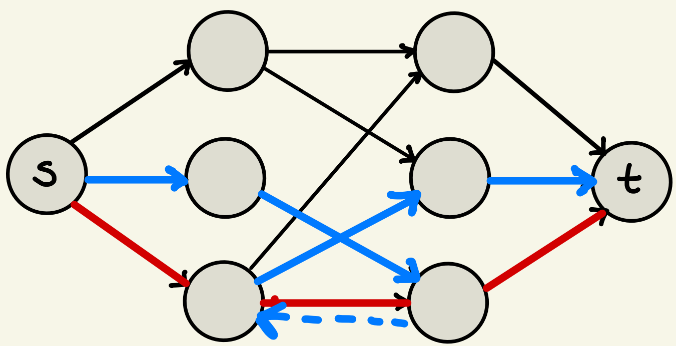
cost 17

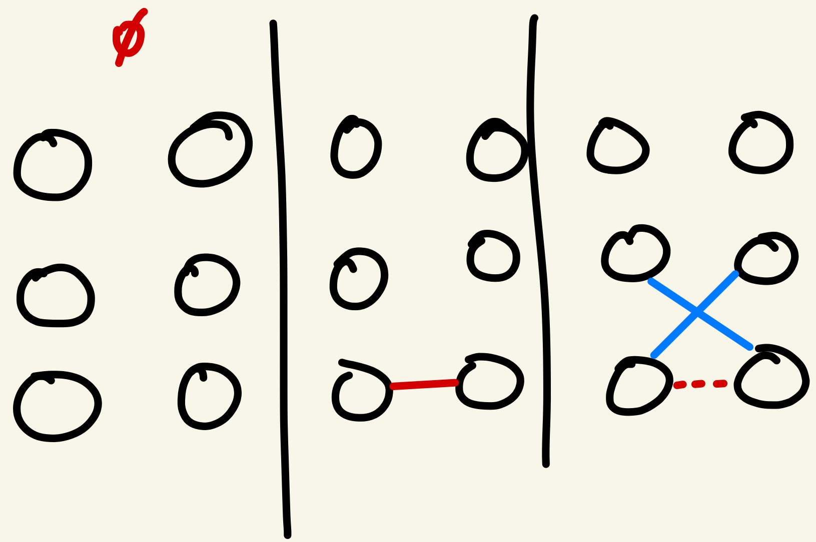


cost 16

Flashback: Maximum Bipartite Matching

What actually happens when we reduce matchings to flows?

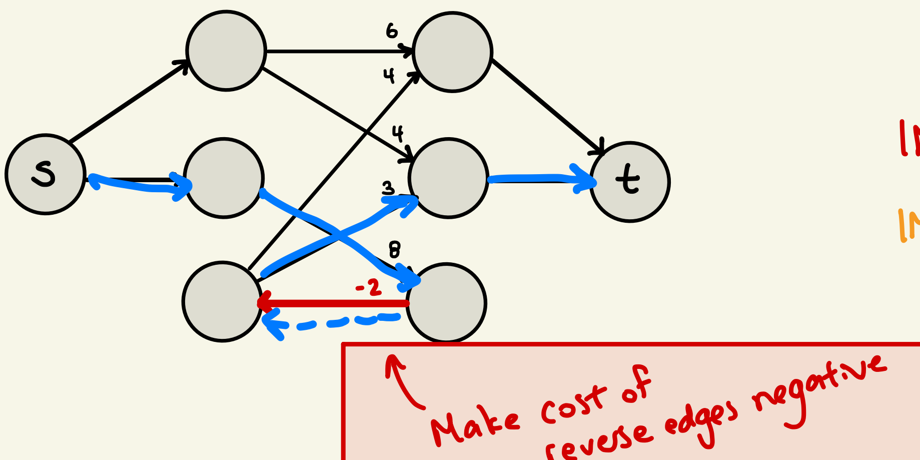




Adding Costs to the Matching

- Let M be a matching, G_M be the residual graph
- Let P be an augmenting path in G_M
- Let M' be the new matching

$$|M'| = |M| + 1 \qquad \text{cost}(M') = \text{cost}(M) + \text{cost}(P)$$

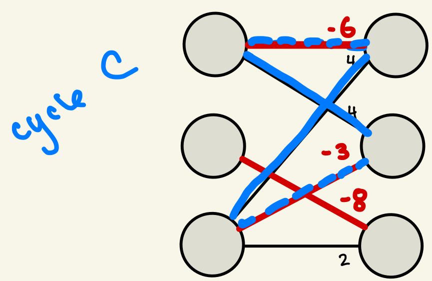


$|M| = 1 \qquad \text{cost}(M) = 2$

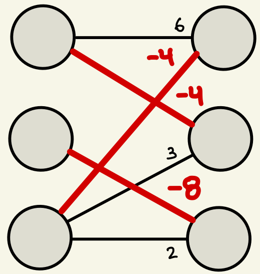
$|M'| = 2 \qquad \text{cost}(M') = 2 - 2 + 3 + 8 = 11$

Understanding Negative Cycles

$wt(C) = 4 - 3 + 4 - 6 = -1$



cost 17



cost 16

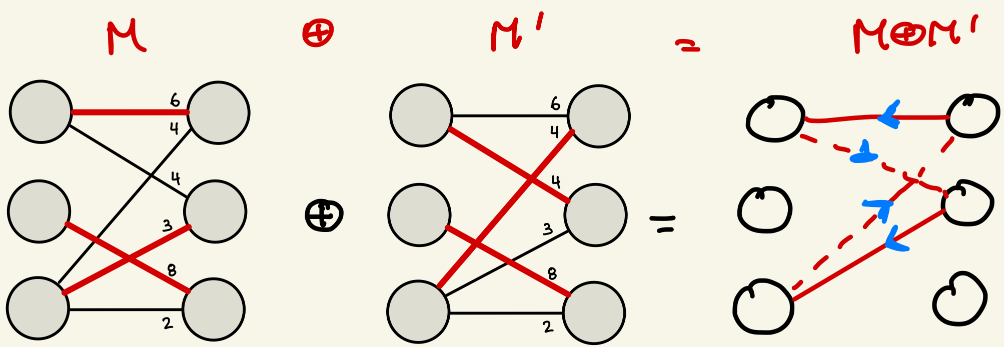
- Any cycle gives a way to go from M to M'
- $\text{cost}(M') = \text{cost}(M) + \text{cost}(C)$

$\Rightarrow$ A negative cycle implies that your matching is not a min cost matching.

Understanding Negative Cycles

<u>Claim</u>: A perfect matching M has minimum cost if and only if G_M contains no negative cost cycles

<u>Proof</u>: ("If direction")



If M, M' are perfect matchings then M⊕M' is a union of cycles.

Let M be "your matching", let M' be a strictly lower cost matching. Then M⊕M' contains a neg-cost cycle

Our Algorithm: Ensuring no negative weight cycles

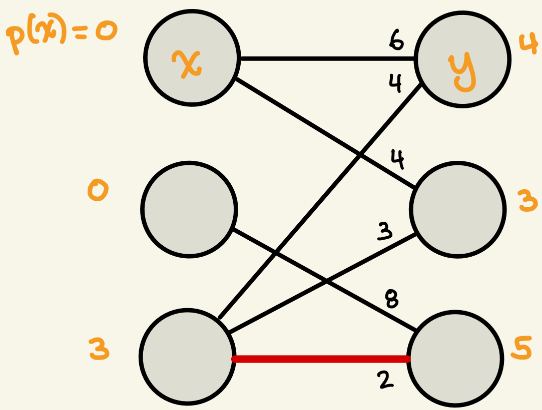
Let $M = \emptyset$

While (M is not perfect):

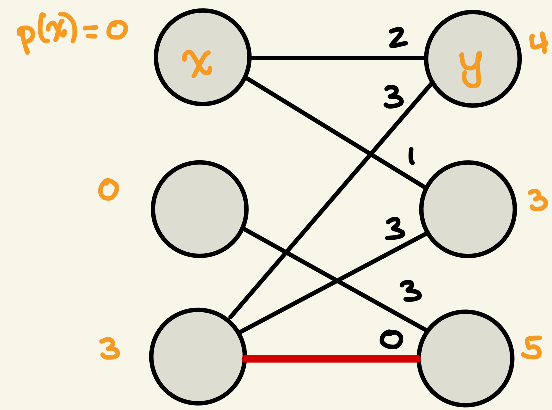
- Find a min cost path P in G_M
- Augment along P to get M'

Want to somehow maintain the invariant that M' has no negative cycles

Using prices to guide the search



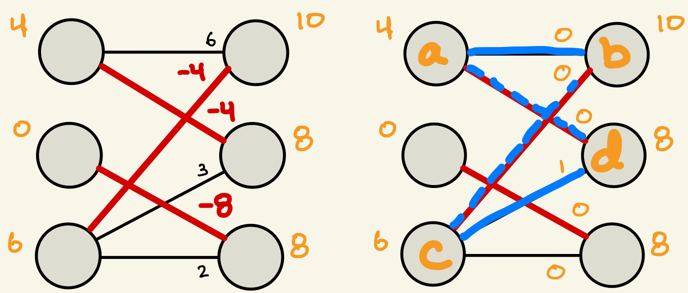
graph with prices
on nodes



"adjusted costs"

$c_p(e) = c(e) + p(x) - p(y)$

Using prices to guide the search



graph G and matching M
with compatible prizes
and adjusted costs

For any alternating cycle the total cost doesn't change.

Compatible prizes:

① if $x \in L$ is unmatched, $p(x) = 0$

② for all edges $e = (x, y)$

$$c_p(e) = c(e) + p(x) - p(y) \geq 0$$

③ for all edges $e = (x, y) \in M$

$$c_p(e) = c(e) + p(x) - p(y) = 0$$

Claim: If there exist compatible prizes for M, then G_M has no negative weight cycles

$$\sum_{e \in C} c_p(e) = \sum_{e \in C} c(e)$$

Our Algorithm: Ensuring no negative weight cycles

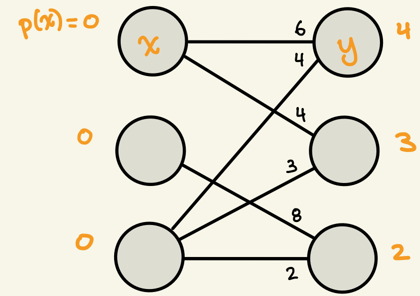
Let $M = \emptyset$

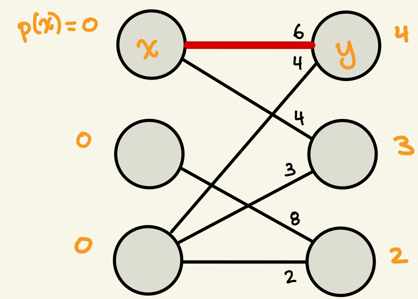
Let $p(x) = 0$ for $x \in L$

Let $p(y) = \min_{e \text{ into } y} c(e)$ for $x \in L$

While (M is not perfect):

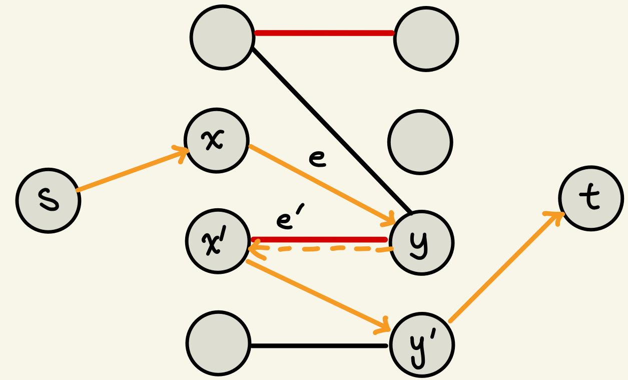
- Find a min cost path P in G_M
- Augment along P to get M'
- Find new compatible prices





How can we get new prices?

Updating the compatible prices



Cool Fact:

- Let p be compatible for M
- Let $d_{p,M}(v)$ be the dist from s to v in the residual G_M
- Let M' be obtained by augmenting the min cost path in G_M

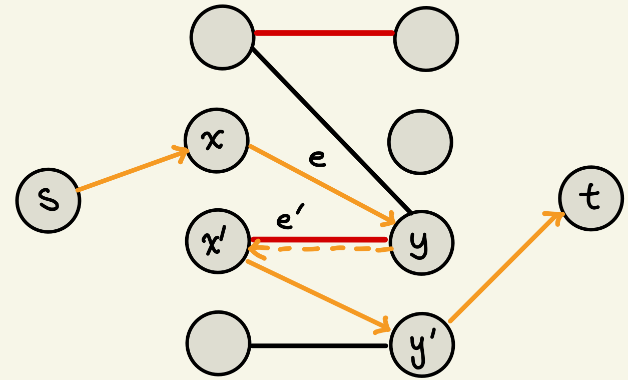
Then $p'(v) = p(v) + d_{p,M}(v)$ is compatible for M'

Updating the compatible prices

Cool Fact:

- Let p be compatible for M
- Let $d_{p,M}(v)$ be the dist from s to y in the residual G_M
- Let M' be obtained by augmenting the min cost path in G_M

Then $p'(v) = p(v) + d_{p,M}(v)$ is compatible for M'



Case 1: $e' = (x', y) \in M$

$$d_{p,M}(x') = d_{p,M}(y) - c_p(e')$$

$$d_{p,M}(x') = d_{p,M}(y)$$

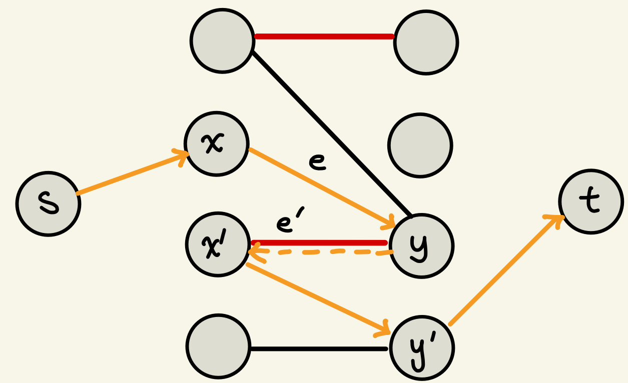
$$\Rightarrow \quad c_{p'}(e') = c_p(e') = 0$$

Updating the compatible prices

Cool Fact:

- Let p be compatible for M
- Let $d_{p,M}(v)$ be the dist from s to y in the residual G_M
- Let M' be obtained by augmenting the min cost path in G_M

Then $p'(v) = p(v) + d_{p,M}(v)$ is compatible for M'



Case 2: $e=(x,y) \in M' \setminus M$

$$d_{p,M}(y) = d_{p,M}(x) + c_p(e)$$

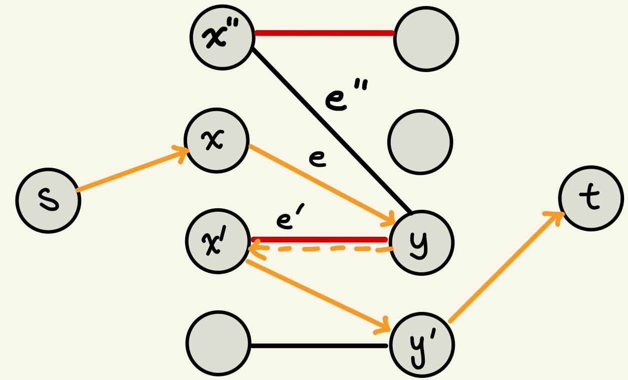
$$c_p(e) - c_{p'}(e) = d_{p,M}(x) - d_{p,M}(y)$$
$$= c_p(e)$$

Updating the compatible prices

Cool Fact:

- Let p be compatible for M
- Let $d_{p,M}(v)$ be the dist from s to y in the residual G_M
- Let M' be obtained by augmenting the min cost path in G_M

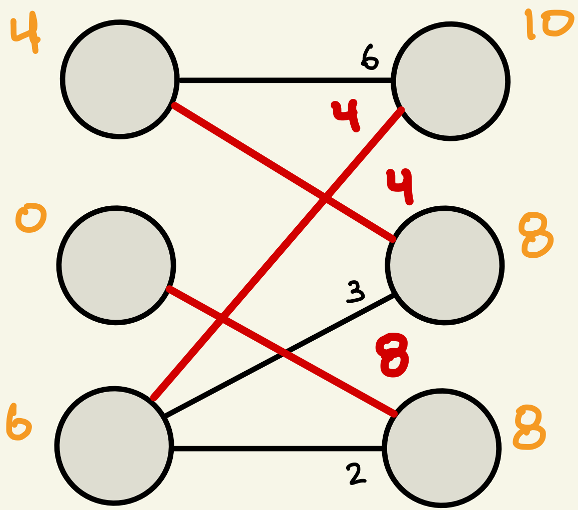
Then $p'(v) = p(v) + d_{p,M}(v)$ is compatible for M'



Case 3: $e'' = (x'', y) \in M$

$$d_{p,M}(y) \leq d_{p,M}(x'') + c_p(e'')$$

Interpreting the Prices



Goal: $\underset{M}{\text{minimize}} \sum_{e} c(e)$

A matching is an equilibrium if every node on the left is maximizing the reward $p(y) - c(x,y)$

$\forall (x,y) \in M \quad p(y) - c(x,y) \geq p(y') - c(x,y)$

for every y'