Network Flow III

Minimum-cost flows

Network Flow Recap Recap

Directed graph

Capacities : $0 \leq f(u,v) \leq c(u,v)$

Conservation: "flow in = flow out"

Residual graph

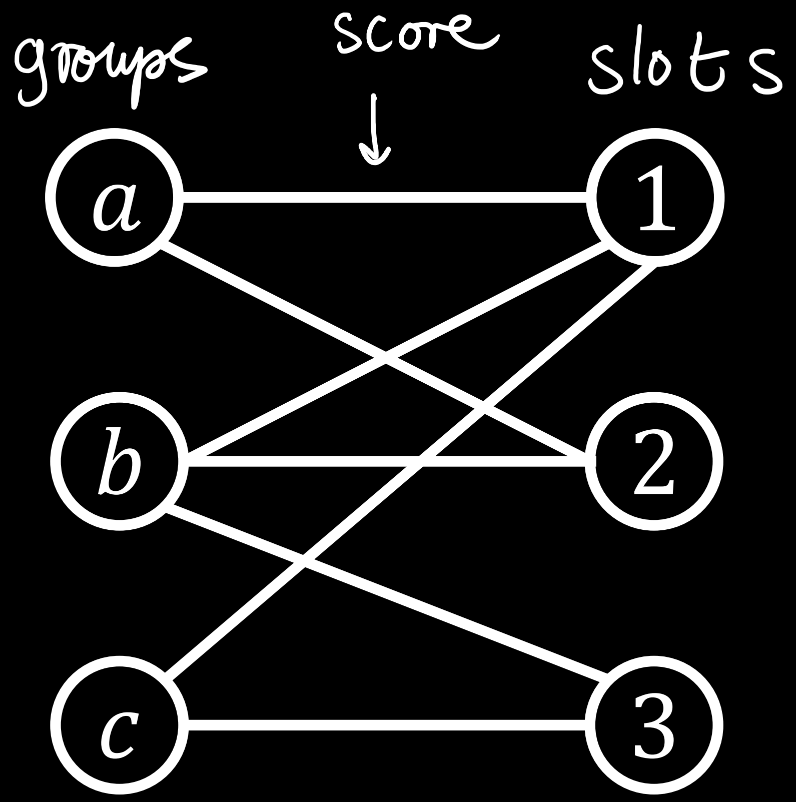
$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \\ f(v,u) & \text{if } (v,u) \in E \end{cases}$$

Motivation

What if we have prefs

Find min-weight perfect matching

Matchings



Minimum-cost Flows

Directed graph, has capacities, conservation

Add $\boxed{\text{costs}}$ to the edges $\$(e)$

Goal: minimize cost

$$\text{cost}(f) = \sum_{e \in E} \$(e) \cdot f(e)$$

<u>Note</u> Cost is <u>per unit</u> of flow

Assumptions

- Allow negative costs!
- Negative-cost cycles allowed!
  - Breaks some algorithms

The residual graph

- Can we generalize it?

$$C_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \$ = \$(u,v) \\ f(v,u) & \$ = -\$(v,u) \end{cases}$$

Residual network with costs

An augmenting path algorithm

minimum-cost path (Dijkstra's) ← only if no negatives

Use Bellman-Ford! ← work with negatives

(Algorithm assumes no neg cycles)

Cheapest Augmenting Path algorithm

Dealing with negative-cost cycles

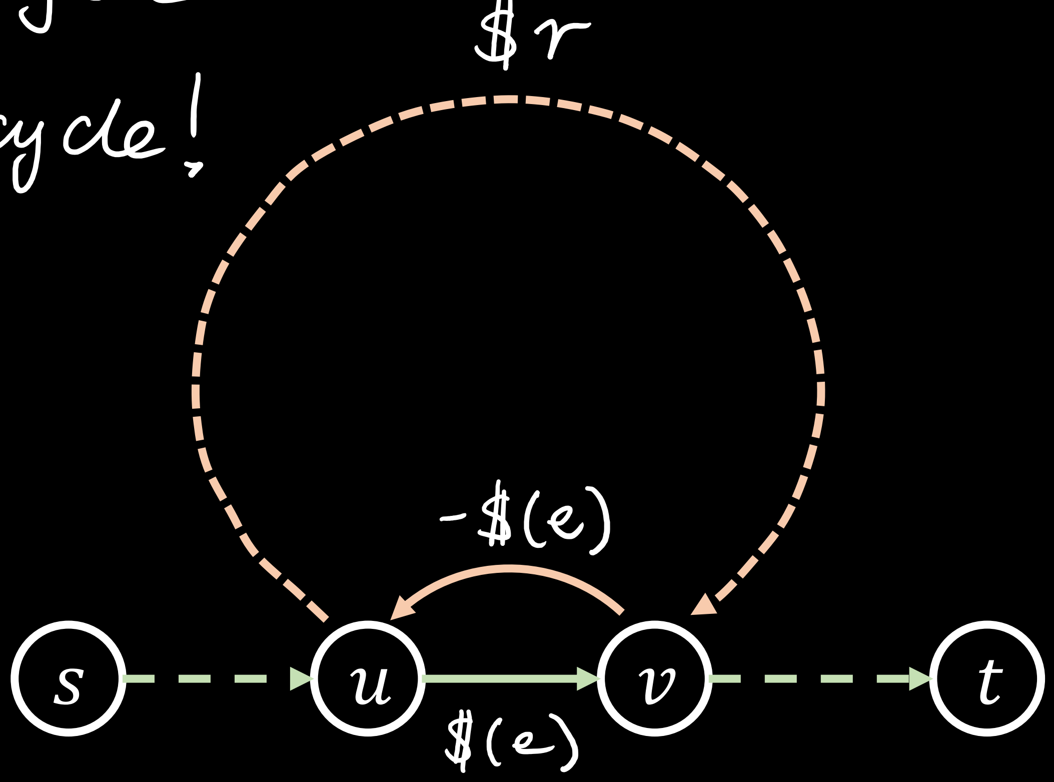
WTS don't create a negative cycle

Suppose we make a negative cycle!

$$\text{cost} = -\$(e) + r < 0$$

$$r < \$(e)$$

Contradiction!

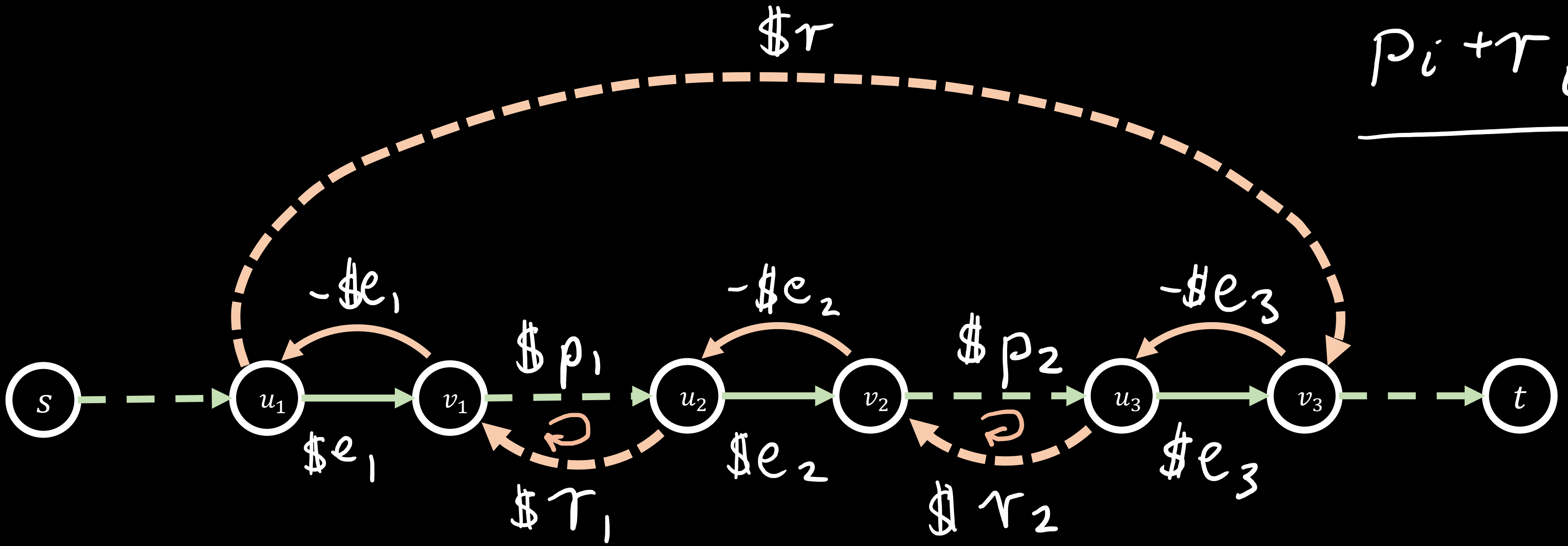


Dealing with negative-cost cycles

$$r - \sum_{i=1}^{k-1} p_i - \sum_{i=1}^{k} \$(e_i) \leq r + \sum_{i=1}^{k-1} r_i - \sum_{i=1}^{k} \$(e_i) < 0$$

$$r < \sum p_i + \sum \$(e_i)$$ ↯ contradiction!

$p_i + r_i \geq 0$



Analysis of cheapest augmenting paths

C.A.P. doesn't create negative cycles $\Rightarrow$ terminates

Runtime: $O(nmF)$ assuming integer capacities

Is this optimal?

Optimality Criteria

<u>Def</u> Cost optimal — Flow is cost optimal if is cheapest of all flows of <u>same value</u>

<u>Recall</u> Flow is not max if $\exists$ augmenting path

How can change a flow – change the cost
– dont change value

<u>Augmenting cycles</u>

Theorem: Cost optimality

A flow f is cost optimal iff there are no negative cycles in G_f

Suppose $\exists$ negative cycle. Can augment along it and reduce cost of f. Therefore f is not cost optimal.

Theorem: Cost optimality

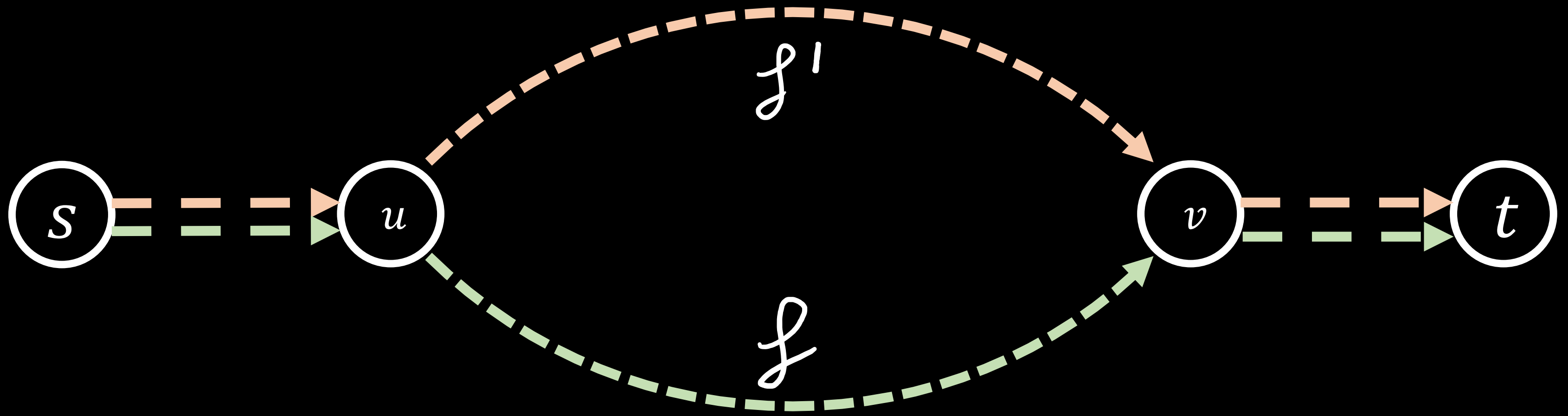
Suppose f is not cost optimal $\Rightarrow$ $\exists$ negative cycle

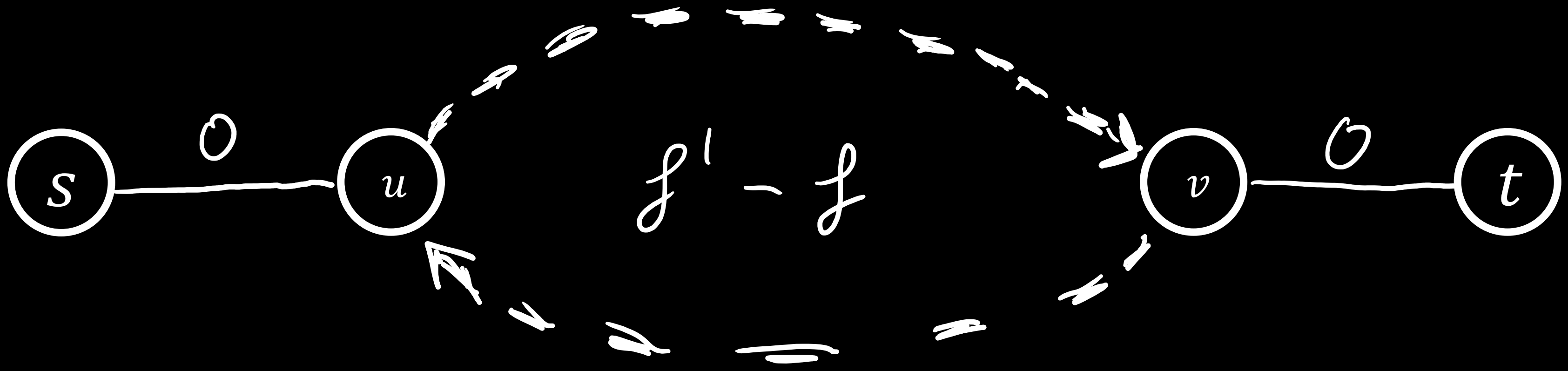
$\exists$ f' of the <u>same value</u> that is cheaper

Define difference between flows $f' - f$

- If $f'(u,v) \geq f(u,v)$ then (u,v) has $f'(u,v) - f(u,v)$
- Otherwise, (v,u) has flow of $f(u,v) - f'(u,v)$

Theorem: Cost optimality





Theorem: Cost optimality

Conservation: Flow-in = Flow-out → collection of cycles

Feasible: in G_f

- positive $f'(e) - f(e) \leq c(e) - f(e) = c_f(e)$
- negative: $f(u,v) - f'(u,v) \leq f(u,v) = c_f(v,u)$

$$\text{cost}(f' - f) = \text{cost}(f') - \text{cost}(f) < 0$$

At least one cycle is negative

$\square$

Completing the Analysis of CAP

C.A.P never creates a negative cycle

$\Rightarrow$ Cost optimal

$\Rightarrow$ Min-cost max flow ✓

Another Algorithm: Cycle Canceling

Start with a max flow

while $\exists$ negative cost cycle

find it and augment it ← Bellman-Ford

Assumption: Integers capacities, costs

Works when there are negative cycles in input!

Analysis of Cycle Canceling

max cost $\leq mUC$

$U =$ max cap

$C = \max |\text{cost}|$

Iteration takes $O(nm)$

$\Rightarrow$ CC runs in $O(nm^2 UC)$

✓