Minimum-cost circulation

Flow network = directed graph $G=(V,E)$

- capacity $c(e)$ for each edge
- cost $\$(e)$ for each edge — cost per unit of flow

Feasible circulation $f: E \to \mathbb{R}$

$0 \le f(e) \le c(e)$ for every $e \in E$

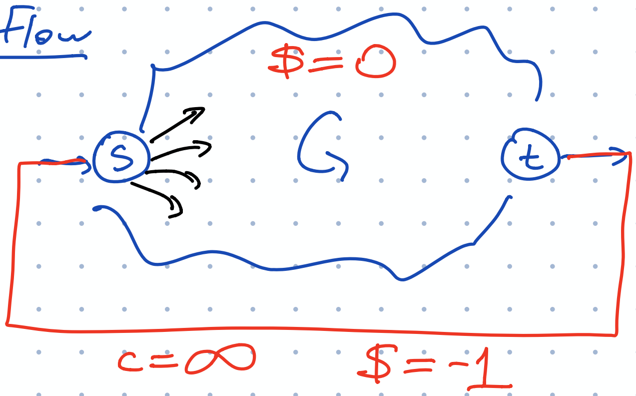
$\sum_u f(u \to v) = \sum_w f(v \to w)$ for every $v \in V$

Minimize Cost $\$(f) := \sum_{u \to v} f(u \to v) \cdot \$(u \to v)$

If $\$(e) \ge 0$, 0 is min. cost circulation

Only interesting if $\$(e) < 0$ for some edges.

MaxFlow



Any flow in G

$\uparrow\downarrow$

Circulation in G'

$f(t \to s) = |f|$

$\$(f) = -|f|$

Cycle Canceling

$\$(u \to v) = -\$(v \to u)$

$f \leftarrow 0$

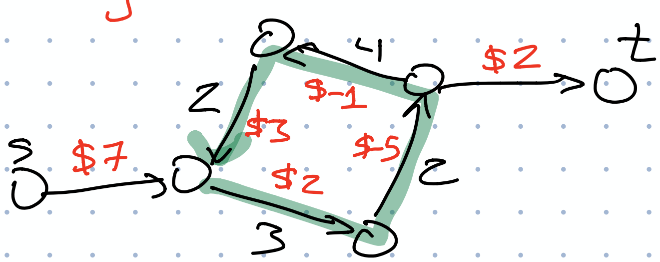
while G_f has any neg. cycles.

  $C \leftarrow$ cycle in G_f with negative cost

  $F \leftarrow \min_{e \in C} c_f(e)$

  $f \leftarrow f + F \cdot C$

return f



If alg. terminates, returns a min-cost circulation

Integer capacities $\longrightarrow$ $O(VE \cdot |\$(f^*)|)$
Intege costs

$O(VE)$ — Find neg cycles [Bellman-Ford]

Each iteration decrease $\$(f)$ by ≥ 1

Heuristics
- Cycle with min cost — NP-hard
- Cycle decreases $\$(f)$ most
- Cycle min # edges — NP-hard
- Minimum-mean cycle — $\frac{\$(C)}{\#e(C)}$

$O(VE)$ time (Karp)

$O(VE^2 \log U)$ min-cost circulation [Goldberg Tarjan]

Min-cost flows (in general form):
- Directed graph $G=(V,E)$
  - capacities $c(e) \geq 0$
  - lower bounds $0 \leq \ell(e) \leq c(e)$
  - balance $b(v)$ $\quad \sum_v b(v) = 0$
  - costs $\$(e)$

Feasible flow

$$\sum_u f(u \to v) - \sum_w f(v \to w) = b(v) \quad \text{for all } v \in V$$

$$\ell(e) \leq f(e) \leq c(e)$$

Min-cost

$$\$(f) := \sum_e \$(e) \cdot f(e)$$

Start with $f(e)=0$ for every edge

$\downarrow$ $f = \ell$ $O(E)$

Feasible pseudoflow

$\downarrow$ Augment supply $\to$ demand paths (FF) in G_f $O(EV)$

Feasible balanced flow

$\downarrow$ Cancel neg cycles in G_f

~~Min-cost~~ feasible balanced flow $\longrightarrow$ no neg-cost cycles in G_f

Locally optimal

Successive Shortest Paths [FF 50's, others]

1. Feasible
2. Locally optimal
3. Balanced

$$f(u\to v) = \begin{cases} c(u\to v) & \text{if } \$(u\to v) < 0 \\ \ell(u\to v) & \text{if } \ell(u\to v) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Every edge ~~cycle~~ in G_f has $\$ \geq 0$ $\to$ f is loc. opt.

$\ell(u\to v) \leq f(u\to v)$ $\to$ f is feasible

Input to SSP:

Dir graph $G=(V,E)$

- $c(e) \geq 0$ for each edge
- $\$(e) \geq 0$ " " "
- $b(v)$ for each vertex $\sum_v b(v) = 0$

Successive SP:

$f \leftarrow 0$ $\qquad$ f is feasible, f is loc. opt. because $\$(e) \geq 0$

while f is not balanced

$\quad$ Build G_f $\qquad$ $\leftarrow$ but here $\$_f(e) < 0$ is possible

$\quad$ $s \leftarrow$ any node with $b_f(s) < 0$ $\qquad$ [supply]

$\quad$ $t \leftarrow$ any node reachable from s with $b_f(t) > 0$ $\qquad$ (demand)

$\quad$ $\sigma \leftarrow$ shortest path from s to t in G_f $\Leftarrow$ $\boxed{\text{Bellman-Ford}}$ $O(VE)$

$\quad$ $\quad$ $\min \sum_e \$(e)$

$\quad$ Augment f along σ $\qquad$ $\leadsto$ f is always feasible

Claim: f is locally optimal!

return f

$\Rightarrow$ final f is balanced and feasible and locally optimal

---

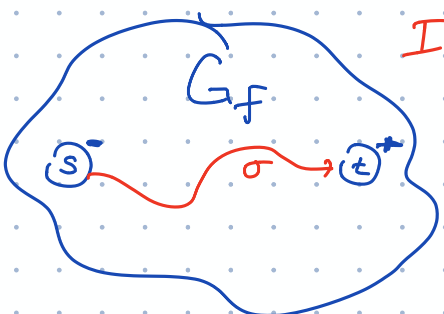
Recall shortest paths only well-defined if no neg. cycles in G_f

$\Longleftrightarrow$ f is locally optimal

Need f to be always loc. opt for algo to be well-defined

Claim: At every iteration of SSP, G_f has no neg cycles.

Proof: by induction on # iteration $\qquad$ base case: $f=0$



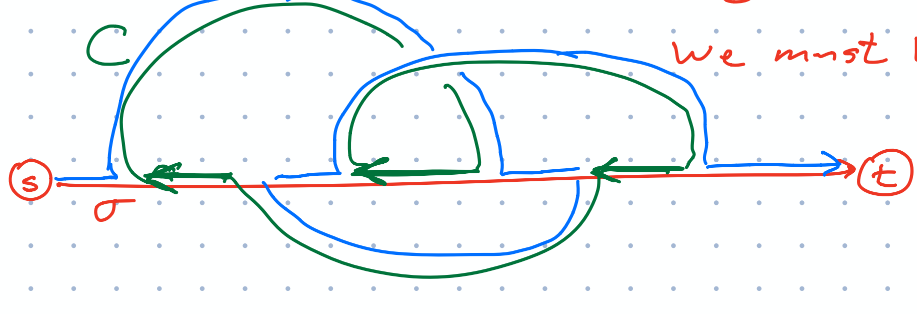
IH: At start G_f has no neg cycles

Let $f' = f + F \cdot \sigma$

Suppose $G_{f'}$ has neg cycle C

At least one edge e of C is not in G_f

we must have $rev(e) \in \sigma$



There is a path σ' that follows σ edges forward
C edges forward

$\$(\sigma') \le \$(\sigma + C) = \$(\sigma) + \$(C) < \$(\sigma)$

Contradiction! σ is a shortest path! $\square$

B = total imbalance $= \sum_v |b(v)|$

Integer balances $\Longrightarrow$ $\boxed{O(VEB) \text{ time}}$

$\downarrow$

$\rightarrow$ $\boxed{O(BE \log V) \text{ time}}$ $\leftarrow$

$\downarrow$ Orlin

Arb. balances $\rightarrow$ $\boxed{O(E^2 \log^2 V) \text{ time}}$ $\leftarrow$

How to speed up SSP

G_f has negative edges $\longrightarrow$ we can't use Dijkstra.

Assign prices $\pi(v)$ to each vertex
potentials

$u \longrightarrow v \longrightarrow w$

$+\pi(u)$ $+\$(u \to v)$
$-\pi(v)$
$+\pi(v)$ $+\$(v \to w)$ $-\pi(w)$

reduced cost $\$^{\pi}(u \to v)$
$= \pi(u) + \$(u \to v) - \pi(v)$

reduced cost of any path $s \rightsquigarrow t$ $= \pi(s) + \$(s \rightsquigarrow t) - \pi(t)$

reduced cost of any cycle $\$^{\pi}(C) = \$(C)$

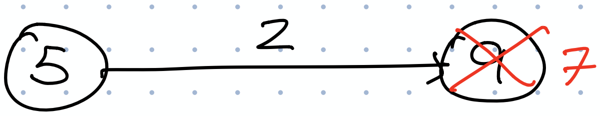
shortest paths wrt $\$^{\pi}$ $\equiv$ shortest paths wrt $\$$

Can we choose $\pi(v)$ so that $\$^{\pi}(e) \ge 0$?

Bellman Ford $\$$ $\longrightarrow$ Dijkstra $\$^{\pi}$

$\boxed{\text{YES!}}$ Pick any vertex s

Let $\pi(v)$ = shortest path distance (wrt \$) from s to v
$= \text{dist}(v)$



$u \to v$ is <u>tense</u> if $\text{dist}(u) + \$(u \to v) > \text{dist}(v)$

If dist correct, no edge is tense

$\$^{\text{dist}}(u \to v) = \text{dist}(u) + \$(u \to v) - \text{dist}(v) \geq 0$

<u>Circular?</u> We need s.p distances to compute s.p distance quickly??

$\boxed{\text{We can use dists at each iteration as prices in the } \underline{\text{next}} \text{ iteration}}$

Proof: See notes

$\hookrightarrow$ Each iteration of SSP in $O(E \log V)$ time