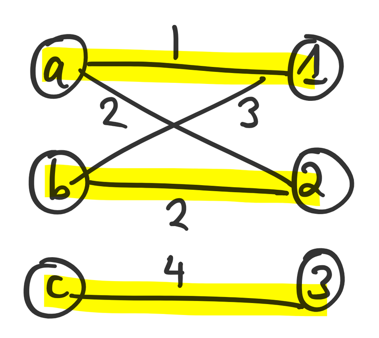
Thursday, October 23, 2025 2:03 AM

Min-Cost Bipartite Matching:



Edges have costs. Find min-cost perfect matching.

Model as a min-cost flow problem. Each edge e has a capacity $c(e)$ and cost $\$(e)$. An s-t flow has cost $\sum_e f(e)\$(e)$.

Find a maximum flow of minimum cost.

$$\text{minimize } \sum_e f(e)\$(e)$$

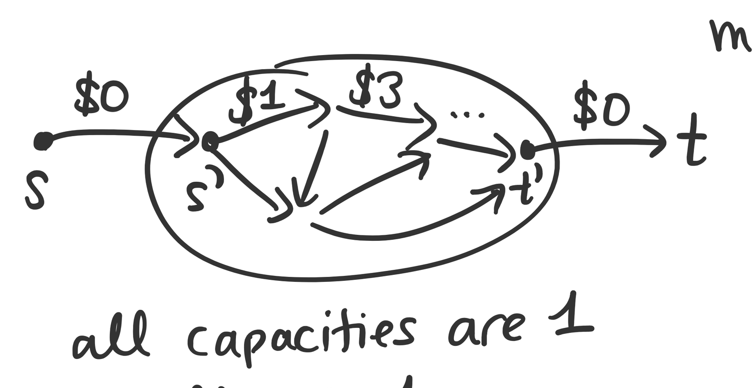
such that $0 \le f(e) \le c(e) \quad \forall e$ [capacity constraint]

$\sum_u f(u,v) = \sum_u f(v,u) \quad \forall v \ne s,t$ [flow conservation]

$\sum_v f(s,v) - \sum_v f(v,s) = F$ [max flow]

max flow value

Example:



all capacities are 1

max flow is 1

min-cost flow is $s' \to t'$ shortest path!

So min-cost flow generalizes s-t shortest path.

Assume uni-directional capacities: $c(u,v) > 0 \Rightarrow c(v,u) = 0$.

Define the residual graph G_f:

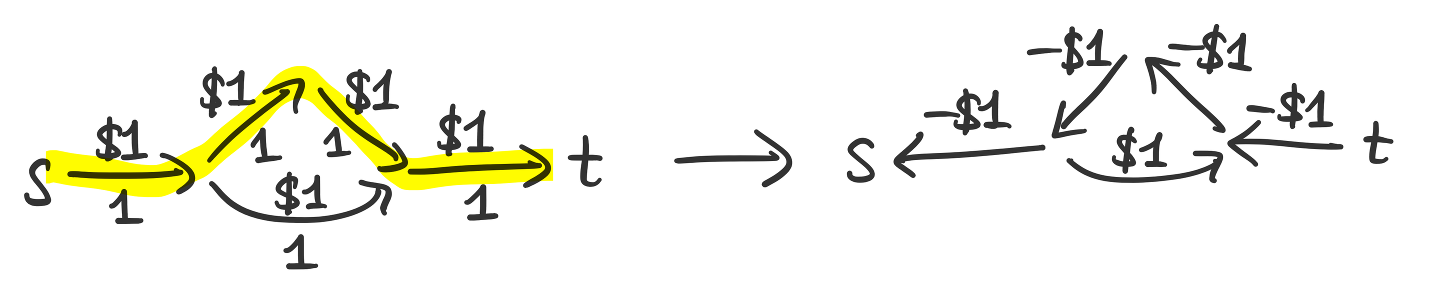
for each edge $(u,v) \in E$ with $f(u,v) > 0$, define

$c_f(u,v) = c(u,v) - f(u,v)$ // can still push $c(u,v) - f(u,v)$ forward

$c_f(v,u) = f(u,v)$ // can undo $f(u,v)$ flow backward

$\$_f(u,v) = \$(u,v)$

$\$_f(v,u) = -\$(u,v)$ // pushing backward undoes cost

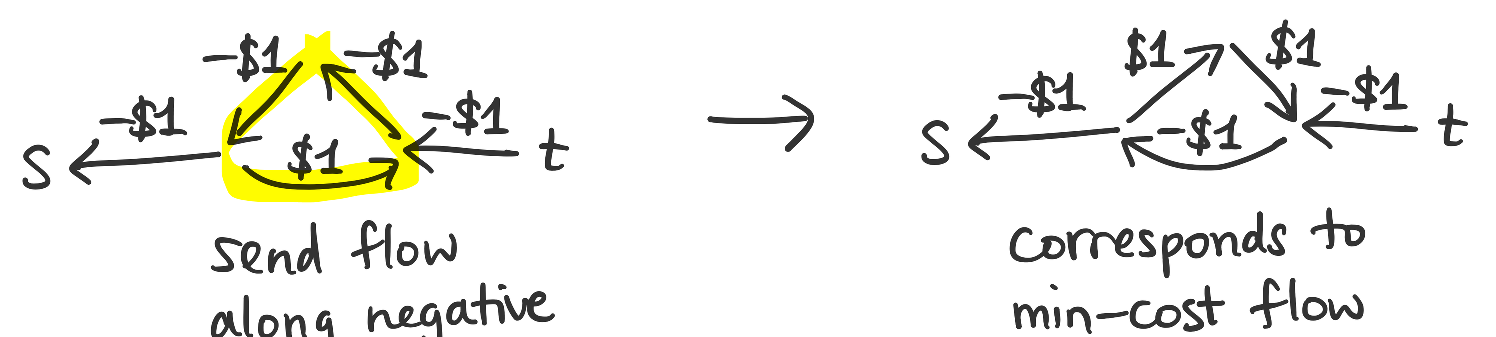


Optimality condition

Define a flow as cost-optimal if lowest cost among all flows of same value.

Suppose a flow is not cost-optimal. How to lower cost?

- Sending flow along a cycle in G_f does not change flow value.
- If cycle is negative in G_f, then lower cost.



send flow along negative cycle

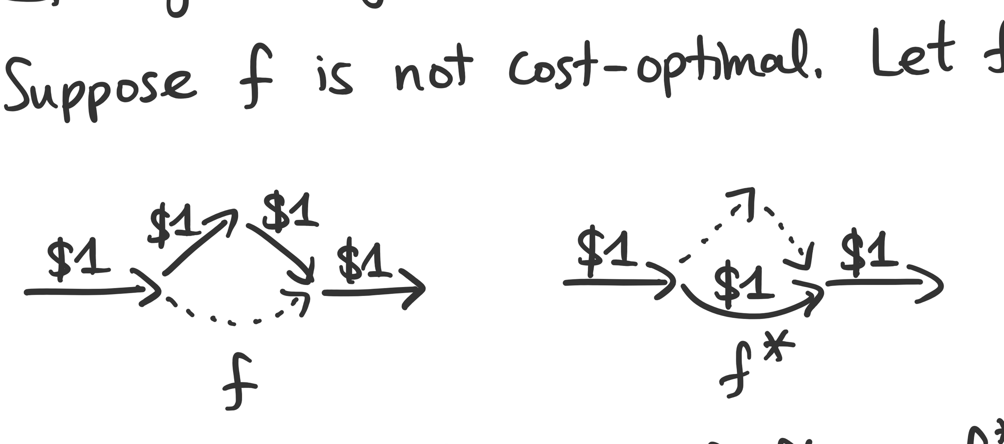
Corresponds to min-cost flow of value 1

Lemma: a flow f is cost-optimal iff there is no negative cost cycle in G_f.

Proof:

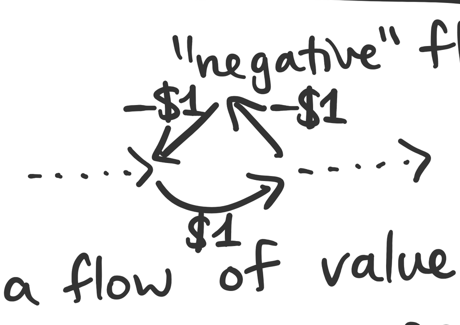
($\Rightarrow$) If negative cycle in G_f, then can lower cost, so not cost-optimal.

($\Leftarrow$) Suppose f is not cost-optimal. Let f^* be cost-optimal flow.



Consider the difference of flows $f^* - f$

"negative" flow corresponds to reversed edge



It is a flow of value $\varnothing$ (called a circulation) in G_f of cost $cost(f^* - f) = cost(f^*) - cost(f) < 0$.

In a circulation, flow in = flow out everywhere.

Fact: can decompose a circulation into cycles of flow

Sum of costs of cycles $= cost(f^* - f) < 0$.

Therefore, some cycle has negative cost.

Cycle-canceling algorithm for min-cost max-flow:

- Find a max-flow using any algorithm
- While exists a negative cost cycle in G_f, send flow along that cycle

Running time?

- $O(Fm)$ time to find initial max flow of value $F \le \sum_e c(e)$
- Maximum possible cost is $\sum_e c(e)\$(e) \le m \cdot \max_e c(e) \cdot \max_e \$(e)$
- Each iteration lowers cost by ≥ 1, runs in $O(mn)$ by Bellman-Ford
- Total time $O(m^2 n \max_e c(e) \max_e \$(e))$.

Augmenting paths algorithm for min-cost max-flow

Idea: run Ford-Fulkerson, but always pick cheapest augmenting path

It turns out this will never create negative cycles in G_f. So every intermediate flow is cost-optimal, and the final flow is min-cost max-flow.

Lemma: Consider augmenting a flow f by the shortest s-t path in G_f. Let f' be the new flow. Then, (s-t distance in $G_{f'}$) $\ge$ (s-t distance in G_f).

In particular, there is no negative cycle.

Proof: Suppose not. Define $d(v) =$ s-v distance in G_f.

Suppose there is an s-t path P in $G_{f'}$ of cost $< d(v)$.

Let v be earliest vertex on this path s.t.

s-v segment in P has cost $c_P(v) < c(v)$

Let u be the vertex immediately before v on P, so

s-u segment in P has cost $c_P(u) \ge c(u)$

By construction, $c_P(u) + \$_{f'}(u,v) = c_P(v)$.

Can the edge (u,v) exist in G_f? If so, then

$$c(u) + \$_f(u,v) \ge c(v) \quad \text{[triangle inequality]}$$

but that contradicts $c_P(v) < c(v)$, $c_P(u) \ge c(u)$.

So, edge (u,v) does not exist in G_f. So the augmenting path in G_f must have used edge (v,u), "freeing" the reverse (u,v).

So (v,u) is part of the shortest s-t path, and

$$c(v) + \$_f(v,u) = c(v)$$

Since $\$_f(v,u) = -\$_{f'}(u,v)$, we get

$$c(u) + \$_f(u,v) = c(v)$$

again, which is a contradiction.