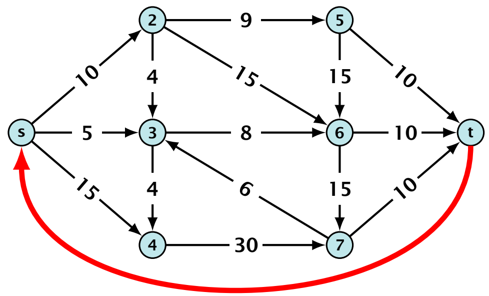
Mincost Flow

Consider the following problem:

$$\begin{aligned} \min \quad & \textstyle\sum_e c(e) f(e) \\ \text{s.t.} \quad & \forall e \in E: \quad 0 \le f(e) \le u(e) \\ & \forall v \in V: \quad f(v) = b(v) \end{aligned}$$

- $G = (V, E)$ is an oriented graph.
- $u : E \to \mathbb{R}_0^+ \cup \{\infty\}$ is the capacity function.
- $c : E \to \mathbb{R}$ is the cost function (note that $c(e)$ may be negative).
- $b : V \to \mathbb{R}$, $\sum_{v \in V} b(v) = 0$ is a demand function.

Solve Maxflow Using Mincost Flow



- Given a flow network for a standard maxflow problem.
- Set $b(v) = 0$ for every node. Keep the capacity function u for all edges. Set the cost $c(e)$ for every edge to 0.
- Add an edge from t to s with infinite capacity and cost -1.
- Then, $\text{val}(f^*) = -\text{cost}(f_{\min})$, where f^* is a maxflow, and $f_{\min}$ is a mincost-flow.

Solve Maxflow Using Mincost Flow

Solve decision version of maxflow:

- Given a flow network for a standard maxflow problem, and a value k.
- Set $b(v) = 0$ for every node apart from s or t. Set $b(s) = -k$ and $b(t) = k$.
- Set edge-costs to zero, and keep the capacities.
- There exists a maxflow of value k if and only if the mincost-flow problem is feasible.

Generalization

Our model:

$$\begin{aligned} \min \quad & \textstyle\sum_e c(e) f(e) \\ \text{s.t.} \quad & \forall e \in E: \quad 0 \le f(e) \le u(e) \\ & \forall v \in V: \quad f(v) = b(v) \end{aligned}$$

where $b : V \to \mathbb{R}$, $\sum_v b(v) = 0$; $u : E \to \mathbb{R}_0^+ \cup \{\infty\}$; $c : E \to \mathbb{R}$;

A more general model?

$$\begin{aligned} \min \quad & \textstyle\sum_e c(e) f(e) \\ \text{s.t.} \quad & \forall e \in E: \quad \ell(e) \le f(e) \le u(e) \\ & \forall v \in V: \quad a(v) \le f(v) \le b(v) \end{aligned}$$

where $a : V \to \mathbb{R}$, $b : V \to \mathbb{R}$; $\ell : E \to \mathbb{R} \cup \{-\infty\}$, $u : E \to \mathbb{R} \cup \{\infty\}$ $c : E \to \mathbb{R}$;

Reduction I

$$\begin{array}{lll} \min & \sum_e c(e)f(e) & \\ \text{s.t.} & \forall e \in E: & \ell(e) \le f(e) \le u(e) \\ & \forall v \in V: & a(v) \le f(v) \le b(v) \end{array}$$

We can assume that $a(v) = b(v)$:

Add new node r.

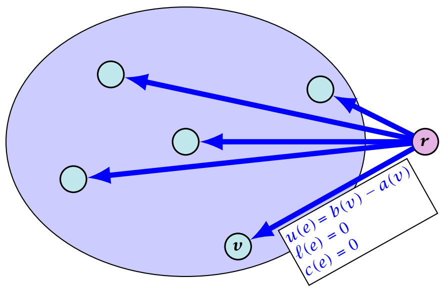
Add edge (r, v) for all $v \in V$.

Set $\ell(e) = c(e) = 0$ for these edges.

Set $u(e) = b(v) - a(v)$ for edge (r, v).

Set $a(v) = b(v)$ for all $v \in V$.

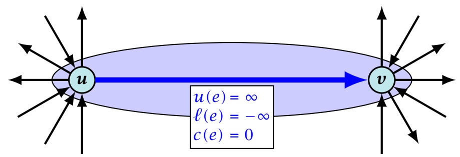
Set $b(r) = \sum_{v \in V} b(v)$.



Reduction II

$$\begin{array}{lll} \min & \sum_e c(e)f(e) & \\ \text{s.t.} & \forall e \in E: & \ell(e) \le f(e) \le u(e) \\ & \forall v \in V: & f(v) = b(v) \end{array}$$

We can assume that either $\ell(e) \neq -\infty$ or $u(e) \neq \infty$:

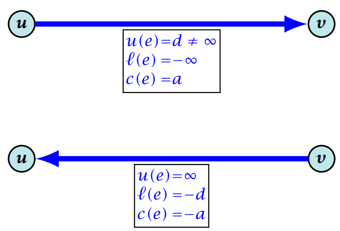


If $c(e) = 0$ we can simply contract the edge/identify nodes u and v

Reduction III

$$\begin{array}{lll} \min & \sum_e c(e)f(e) & \\ \text{s.t.} & \forall e \in E: & \ell(e) \le f(e) \le u(e) \\ & \forall v \in V: & f(v) = b(v) \end{array}$$

We can assume that $\ell(e) \neq -\infty$:

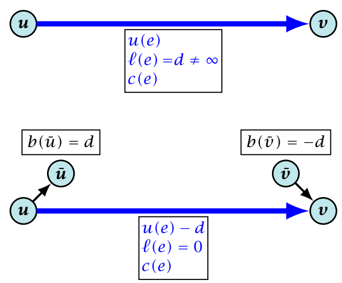


Replace the edge by an edge in opposite direction.

Reduction IV

$$\begin{array}{lll} \min & \sum_e c(e)f(e) & \\ \text{s.t.} & \forall e \in E: & \ell(e) \le f(e) \le u(e) \\ & \forall v \in V: & f(v) = b(v) \end{array}$$

We can assume that $\ell(e) = 0$:



The added edges have infinite capacity and cost $c(e)/2$.

Applications

Caterer Problem

- She needs to supply r_i napkins on N successive days.
- She can buy new napkins at p cents each.
- She can launder them at a fast laundry that takes m days and cost f cents a napkin.
- She can use a slow laundry that takes $k > m$ days and costs s cents each.
- At the end of each day she should determine how many to send to each laundry and how many to buy in order to fulfill demand.
- Minimize cost.

Residual Graph

The residual graph for a mincost flow is exactly defined as the residual graph for standard flows, with the only exception that one needs to define a cost for the residual edge.

For a flow of z from u to v the residual edge (v,u) has capacity z and a cost of $-c((u,v))$.

15 Mincost Flow

A circulation in a graph $G = (V,E)$ is a function $f : E \to \mathbb{R}^+$ that has an excess flow $f(v) = 0$ for every node $v \in V$ (G may be a directed graph instead of just an oriented graph).

A circulation is feasible if it fulfills capacity constraints, i.e., $f(e) \le u(e)$ for every edge of G.

15 Mincost Flow

$g = f^* - f$ is obtained by computing $\Delta(e) = f^*(e) - f(e)$ for every edge $e = (u,v)$. If the result is positive set $g((u,v)) = \Delta(e)$ and $g((v,u)) = 0$; otw. set $g((u,v)) = 0$ and $g((v,u)) = -\Delta(e)$.

Lemma 85

A given flow is a mincost-flow if and only if the corresponding residual graph G_f does not have a feasible circulation of negative cost.

⇒ Suppose that g is a feasible circulation of negative cost in the residual graph.

Then $f + g$ is a feasible flow with cost $\text{cost}(f) + \text{cost}(g) < \text{cost}(f)$. Hence, f is not minimum cost.

⇐ Let f be a non-mincost flow, and let f^* be a min-cost flow. We need to show that the residual graph has a feasible circulation with negative cost.

Clearly $f^* - f$ is a circulation of negative cost. One can also easily see that it is feasible for the residual graph.

15 Mincost Flow

Lemma 86

A graph (without zero-capacity edges) has a feasible circulation of negative cost if and only if it has a negative cycle w.r.t. edge-weights $c : E \to \mathbb{R}$.

Proof.

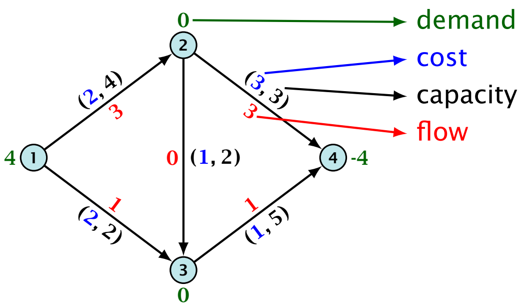
- Suppose that we have a negative cost circulation.
- Find directed path only using edges that have non-zero flow.
- If this path has negative cost you are done.
- Otherwise send flow in opposite direction along the cycle until the bottleneck edge(s) does not carry any flow.
- You still have a circulation with negative cost.
- Repeat.

15 Mincost Flow

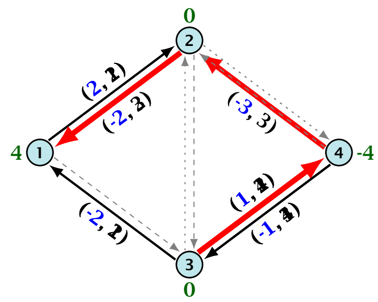
Algorithm 51 CycleCanceling$(G = (V, E), c, u, b)$

establish a feasible flow f in G
while G_f contains negative cycle **do**
    use Bellman-Ford to find a negative circuit Z
    $\delta \leftarrow \min\{u_f(e) \mid e \in Z\}$
    augment δ units along Z and update G_f

15 Mincost Flow



15 Mincost Flow



15 Mincost Flow

Lemma 87

The improving cycle algorithm runs in time $\mathcal{O}(nm^2CU)$, *for integer capacities and costs, when for all edges* e, $|c(e)| \leq C$ *and* $|u(e)| \leq U$.