MATH 566: Lecture 23 (11/05/2024)

Today:
* MCF — assumptions
* residual network for MCF
* MCF optimality conditions

The **Min-Cost Flow** (MCF) Problem

Recall SP optimality conditions: $d(j) \le d(i) + c_{ij}$ $\forall (i,j) \in A$.

Max flow optimality conditions: No augmenting paths in $G(\bar{x})$.

In MCF, we work with costs, capacities, and supplies/demand. We use $C = \max_{(i,j)\in A} \{|c_{ij}|\}$ and $U = \max_{(i,j)\in A} u_{ij}$ in our discussion.

Optimization model (Linear Program):

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \qquad \text{(total cost)}$$

$$\text{s.t.} \quad \underbrace{\sum' x_{ij}}_{\text{outflow}} - \underbrace{\sum' x_{ji}}_{\text{inflow}} = \underbrace{b(i)}_{\text{supply/demand}} \quad \forall i \in N \qquad \text{(flow balance)}$$

$$0 \le x_{ij} \le u_{ij} \quad \forall (i,j) \in A \qquad \text{(bounds)}$$

Again, we will **not** directly solve MCF problems as linear programs. The focus will be on efficient algorithms. AMO describes several applications modeled as MCF — we will not discuss them here.

Assumptions

1. $l_{ij} = 0$ → else remove nonzero lower bounds (network transformations)

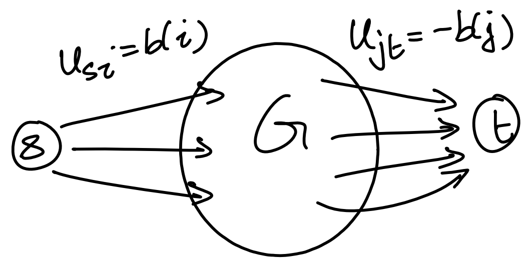
2. All data is integral $(c_{ij}, u_{ij}, b(i))$ for complexity analysis purposes

3. The network is directed.

4. $\sum_{i \in N} b(i) = 0$ (total supply = total demand)
→ else, MCF instance is not feasible

5. The MCF problem has a feasible solution.

Recall: Given an MCF problem instance, we can check if it has a feasible flow by solving a max-flow problem.



Add s, t, (s,i) with $u_{si} = b(i)$ for $\underbrace{i \text{ s.t. } b(i) > 0}_{\text{supply nodes}}$, and (j,t) with $u_{jt} = -b(j)$ for $\underbrace{j \text{ s.t. } b(j) < 0}_{\text{demand nodes}}$.

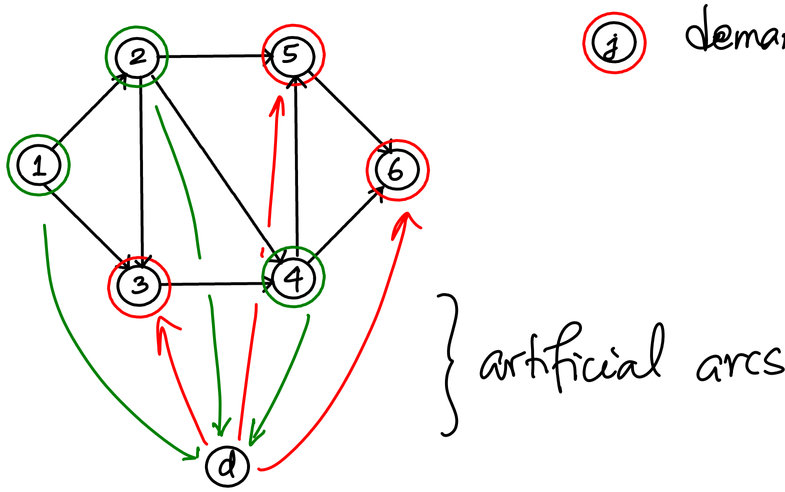
If an s-t max flow saturates all these extra arcs, the original MCF instance has a feasible solution.

Another approach to handle feasibility:

(i) supply node, $b(i) > 0$

(j) demand node, $b(j) < 0$

G



d: dummy node (transshipment node) $b(d) = 0$

Add arcs (i,d) $\forall i \in N$ with $b(i) > 0$, and (d,j) $\forall j \in N$ with $b(j) < 0$, with $c_{id} = c_{dj} = +\infty$ and $u_{id} = u_{dj} = +\infty$. Solve MCF on this modified problem. If the optimal solution has any $x_{id} > 0$ or $x_{dj} > 0$, then the original MCF has no feasible solution. If not, x_{ij} $\forall (i,j) \in A$ (in original network G) is an optimal solution to the MCF problem.

This option is motivated by the use of artificial variables in linear programming (the big-M simplex method).

Assumptions (for MCF, continued)

6. There is an uncapacitated directed path between every pair of nodes. Can add extra arc (i,j) with $u_{ij} = \infty$, $c_{ij} = +\infty$ if needed.

7. $c_{ij} \geq 0 \ \forall (i,j) \in A$ (else, we can use arc reversal)

We need $u_{ij} < \infty$ for this transformation.

If $u_{ij} = \infty$, use $u_{ij} = B > \sum_{u_{ij} \text{ is finite}} u_{ij} + \sum_{b(i)>0} b(i)$.

finite, but large enough to act as $+\infty$.

8. If some $u_{ij} = +\infty$, we assume there is no negative cost cycle of infinite capacity.

If there is such a cycle, the problem is unbounded.
Else, we could replace $u_{ij} = \infty$ with B as shown above.

Notice that we get Assumption 8 when Assumption 7 is satisfied. At the same time, it is helpful to list them both separately.

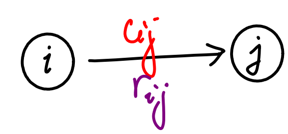
Residual Network for Min-Cost Flow

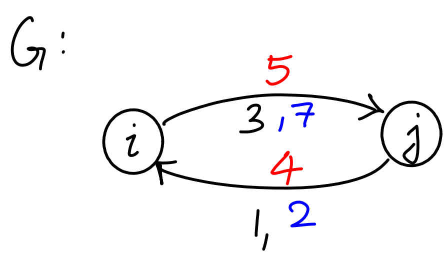
Recall: residual network for max flow: $r_{ij} = u_{ij} - x_{ij} + x_{ji}$. But in MCF c_{ij} may not be same as c_{ji}, so we cannot combine residual capacities usually.

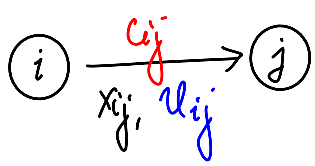
$G(\bar{x})$ for MCF: For arc (i,j) we get

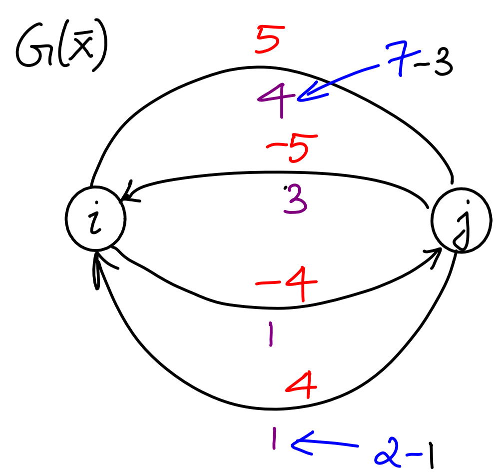
$$r_{ij} = u_{ij} - x_{ij} \text{ with cost } c_{ij}, \text{ and}$$
$$r_{ji} = x_{ij} \text{ with cost } -c_{ij} \text{ (or } c_{ji} = -c_{ij} \text{ here).}$$

Of course, only those arcs (i,j) with $r_{ij} > 0$ appear in $G(\bar{x})$.









Recall that in max flow, we could combine the residual capacities arising from different arcs. In particular, we had $r_{ij} = (u_{ij} - x_{ij}) + x_{ji}$. But in MCF, the costs are different, and hence we cannot combine.

Reduced Costs

Recall, SP optimality conditions: $d(j) \le d(i) + c_{ij} \;\; \forall (i,j) \in G$.

Equivalently, $\bar{c}^{d}_{ij} = c_{ij} + d(i) - d(j) \geqslant 0 \;\; \forall (i,j) \in G$.

Def For a set of node potentials $\pi(i)$, $i \in N$, $c^{\bar{\pi}}_{ij} = c_{ij} - \pi(i) + \pi(j)$ is the **reduced cost** of (i,j) **with respect to $\bar{\pi}$**.

In the SP case, we take $\pi(i) = -d(i)$.

Optimality Conditions for MCF

We extend the SP optimality conditions to $G(\bar{x})$ for MCF to define MCF optimality conditions. We then devise algorithms for MCF that check for these optimality conditions repeatedly, and modify $\bar{x}$ and $G(\bar{x})$ to correct any violations.

We present three different optimality conditions. Naturally, they are equivalent. The first two are specified on $G(\bar{x})$, while the third is specified on G (original network).

1. Negative Cycle Optimality Conditions

AMO Theorem 9.1 A feasible flow $\bar{x}$ is an optimal solution to the MCF problem iff there is no negative-cost cycle in $G(\bar{x})$.

Proof

$(\Rightarrow)$ If there is a negative cycle $G(\bar{x})$, then augment flow along it to reduce total cost. Hence $\bar{x}$ is not an optimal flow. Hence by the contrapositive, if $\bar{x}$ is optimal, $G(\bar{x})$ has no negative cycle.

$(\Leftarrow)$ Let $\bar{x}$ be a feasible flow, and $G(\bar{x})$ has no negative cycle. We want to show $\bar{x}$ is optimal. $G(\bar{x})$ has $\leq 2m$ arcs

Assume $\bar{x}^0$ is an optimal flow, and $\bar{x}^0 \neq \bar{x}$. Then we can decompose the difference $\bar{x}^0 - \bar{x}$ into at most $\underline{2m}$ cycles. see below for why
The sum of the costs of all these cycles is $\bar{c}^T\bar{x}^0 - \bar{c}^T\bar{x}$.
Since $G(\bar{x})$ has no negative cycles, it must hold that
$\bar{c}^T\bar{x}^0 - \bar{c}^T\bar{x} \geq 0 \Rightarrow \bar{c}^T\bar{x}^0 \geq \bar{c}^T\bar{x}$. But $\bar{x}^0$ is an optimal solution, i.e., $\bar{c}^T\bar{x}^0 \leq \bar{c}^T\bar{x}$. So, $\bar{c}^T\bar{x} = \bar{c}^T\bar{x}^0$.

Hence $\bar{x}$ is also an optimal solution. $\square$

Note: Here $\bar{c}^T\bar{x} = \sum_{(i,j)\in A} c_{ij}x_{ij}$, with $\bar{x} = \begin{array}{c} 1 \\ 2 \\ \vdots \\ \\ m \end{array}\begin{bmatrix} \\ \\ x_{ij} \\ \\ \\ \end{bmatrix}$, the vector of flows x_{ij}.

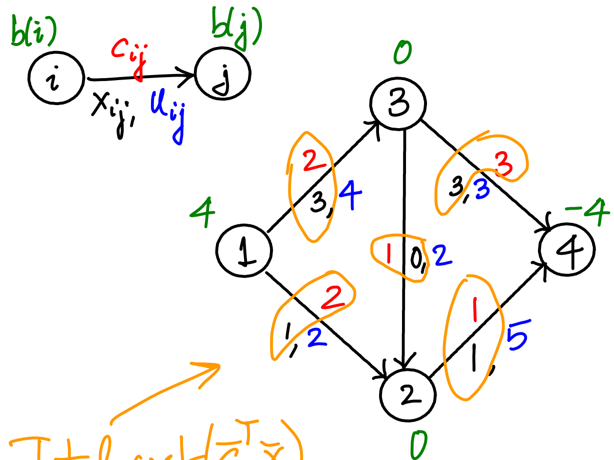
Also, since both $\bar{x}^0$ and $\bar{x}$ are feasible flows, their difference $\bar{x}^0 - \bar{x}$ can be decomposed into only cycle flows. Due to the flow balance constraints, and since $b(i)$ is same (for both $\bar{x}^0$ and $\bar{x}$), we cannot get any path flows.

Negative Cycle Canceling Algorithm

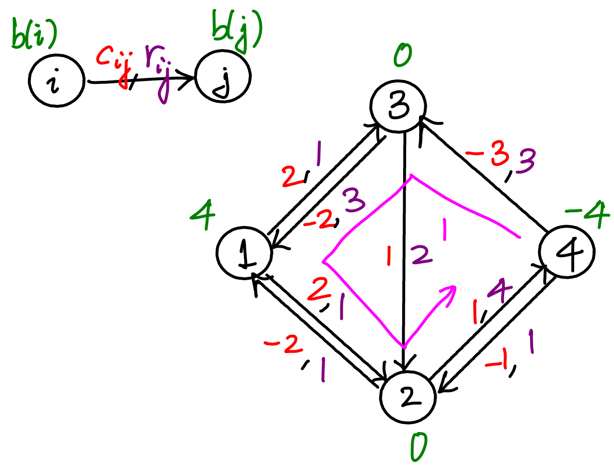
* Start with a feasible flow $\bar{x}$.
* Use FIFO label correcting algorithm to identify a negative cycle in $G(\bar{x})$.
* augment, update $\bar{x}$, $G(\bar{x})$; repeat.

Example

G



Total cost $(\bar{c}^T\bar{x})$ $= 18$ here

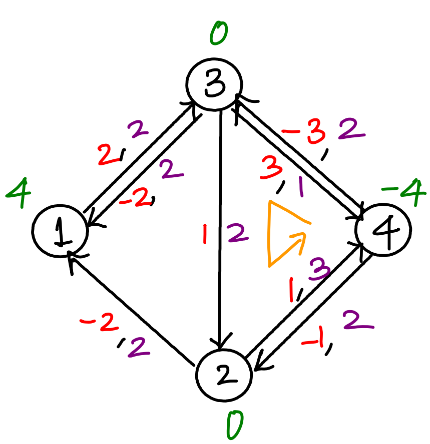


$W_1 = 1\text{-}2\text{-}4\text{-}3\text{-}1$

$c(W_1) = -2$

$\delta(W_1) = 1 \quad (r_{12} = 1)$

Augment $\delta = 1$ along W_1.



$W_2 = 2\text{-}4\text{-}3\text{-}2$

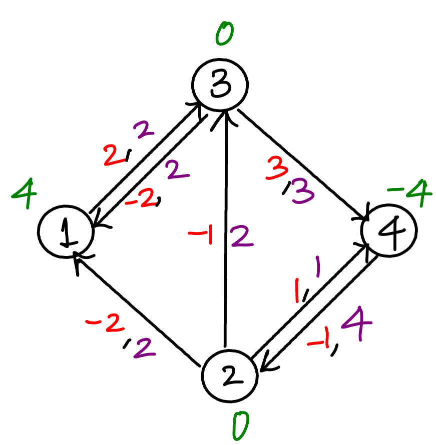
$c(W_2) = -1$, $\delta(W_2) = 2$

r_{32} and r_{43}.

augment

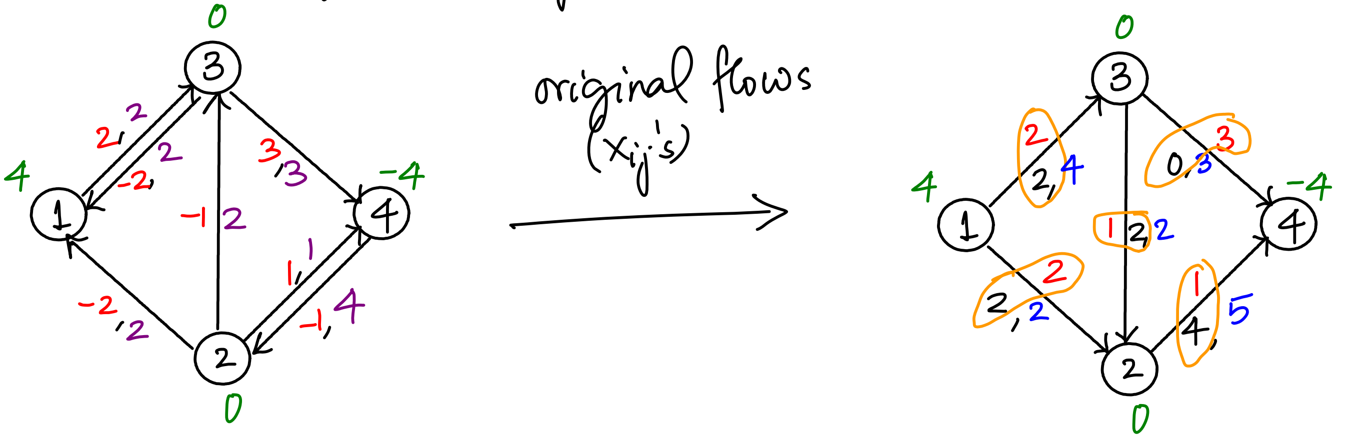
$\delta(W_2) = 2$

$\longrightarrow$



No more negative cycles, so flow is optimum.

We recover the original flows x_{ij}, similar to how we did it for max flow:



$\bar{c}^T \bar{x} = 14$ here

originally, $\bar{c}^T \bar{x} = 18$.

Finiteness and Complexity

AMO Theorem 9.10 If u_{ij} and $b(i)$ are all integers, the negative cycle canceling algorithm maintains an integral flow (solution) in each iteration.

Proof Initial feasible flow can be found using a max flow, which gives an integral flow. In each iteration, the bottleneck capacity is integral.

Theorem The negative cycle canceling algorithm is <u>finite</u> if all data is finite and integral.

→ terminates after a finite # iterations

Proof We use $-mCU$ as a lower bound and mCU as an upper bound for the total cost. Hence, the maximum change in total cost is $2mCU$.

In each iteration, total cost is decreased by at least 1. Hence, the algorithm terminates in at most $2mCU$ augmentations.

FIFO label correcting algorithm in each step takes $O(mn)$ time. Hence the overall time complexity of the negative cycle canceling algorithm is $O(m^2 n C U)$. $\square$