MATH 566 : Lecture 25 (11/12/2024)

Today: * SSP algo, example
* MST, cut/path optimality conditions

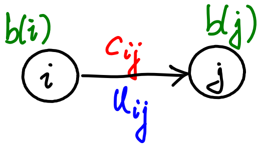
Successive Shortest Path (SSP) Algorithm for MCF

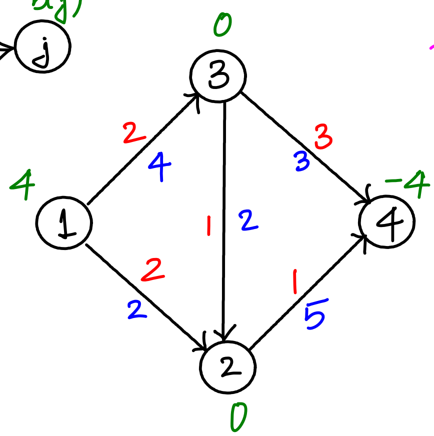
algorithm *successive shortest path*;
begin $\rightarrow$ no need to find feasible flow @start
$x := 0$ and $\pi := 0$;
$e(i) := b(i)$ for all $i \in N$;
initialize the sets $E := \underbrace{\{i : e(i) > 0\}}_{\text{supply nodes}}$ and $D := \underbrace{\{i : e(i) < 0\}}_{\text{demand nodes}}$;
while $E \neq \emptyset$ **do**
begin
select a node $k \in E$ and a node $l \in D$;
determine shortest path distances $d(j)$ from node ~~s~~ k to all other nodes in $G(x)$ with respect to the reduced costs c_{ij}^{π}; $\rightarrow \geq 0$, can use Dijkstra (from step 2 onward)
let P denote a shortest path from node k to node l;
update $\pi := \pi - d$; $\rightarrow$ maintain optimality.
$\delta := \min[e(k), -e(l), \min\{r_{ij} : (i, j) \in P\}]$;
augment δ units of flow along the path P;
update x, $G(x)$, E, D, and the reduced costs;
end;
end;

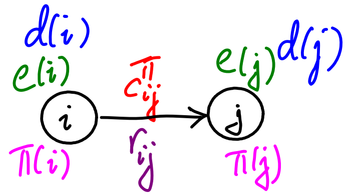
Figure 9.9 Successive shortest path algorithm.

Since we maintain optimality ($c_{ij}^{\bar{\pi}} \geq 0 \ \forall (i,j) \in G(\bar{x})$), we can use efficient algorithms—Dijkstra— to compute the SP distance labels in each iteration after the first one.
When we start, $\bar{\pi} = \bar{0}$ and hence $c_{ij}^{\bar{\pi}} = c_{ij}$ itself, which could be < 0. But after the first SP computation, $c_{ij}^{\bar{\pi}} \geq 0$ is maintained (Lemma 9.11).

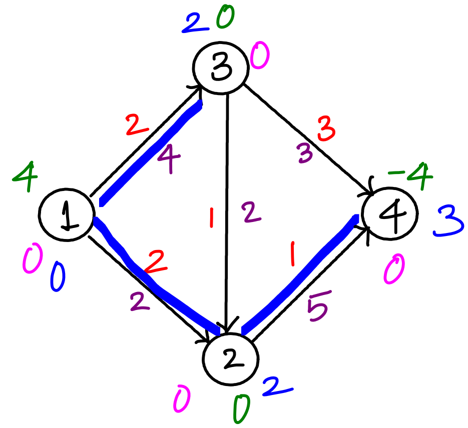
SSP Algorithm: Example







$E = \{1\},\ D = \{4\}$

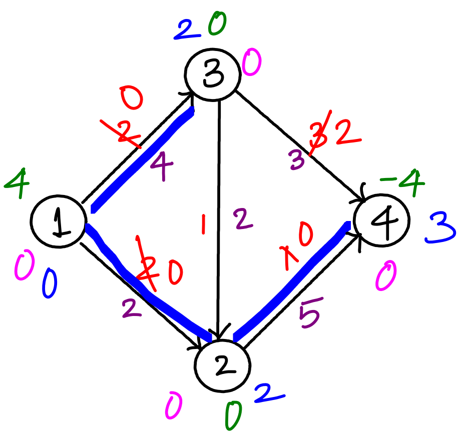


Iteration 1: SP from k=1

d(i)'s

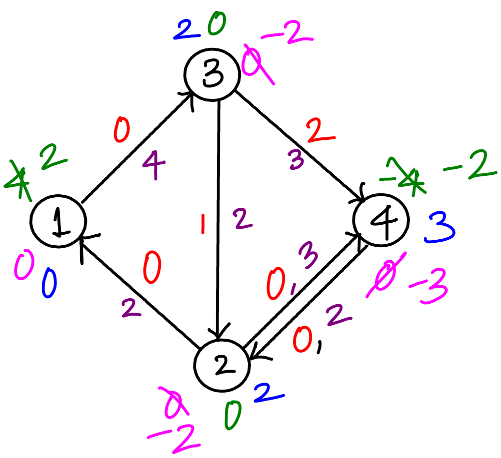
(SP tree shown in **blue**)

$\bar{\pi} \leftarrow \bar{\pi} - \bar{d}$



$P_1 = 1\text{-}2\text{-}4.\quad \delta = \min\{e(1), -e(4), r_{12}, r_{24}\}$

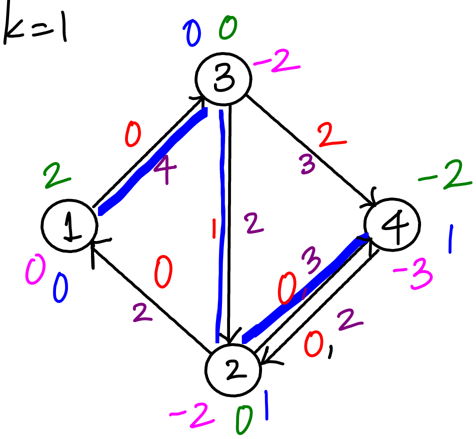
$= \min\{4, 4, 2, 5\} = 2$

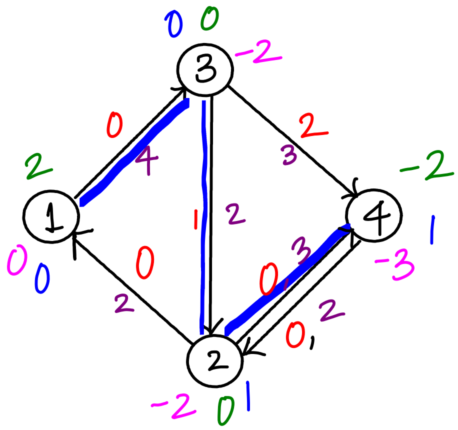


Iteration 2

$E = \{1\},\ D = \{4\}.$

SP from k=1



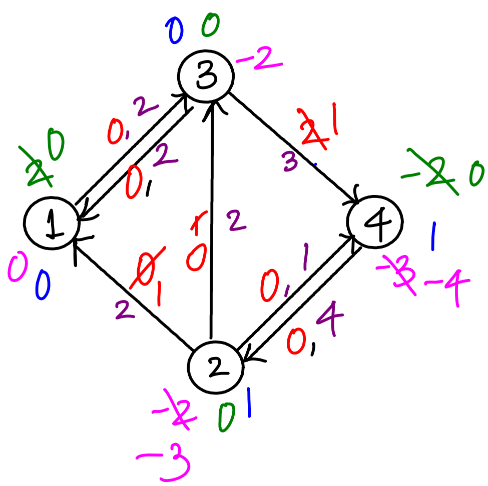


$P_2 = 1\text{-}3\text{-}2\text{-}4$

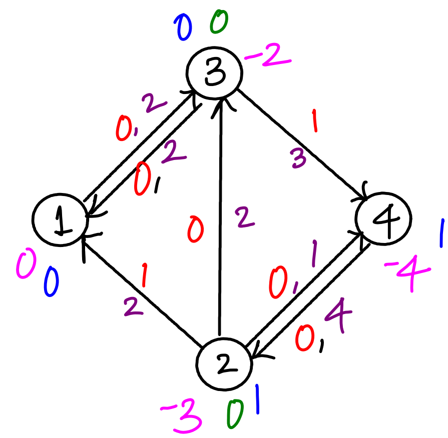
$$\delta = \min\{e(1), -e(4), r_{13}, r_{32}, r_{24}\}$$

$$= \min\{2, 2, 4, 2, 3\} = 2$$

$$\bar{\pi} \leftarrow \bar{\pi} - \bar{d}$$



Flow is optimum, as $E = \phi = D$ now.



flow in the original network:

$x_{12} = 2$, $x_{13} = 2$, $x_{32} = 2$, $x_{24} = 4$

Minimum Spanning Trees (MST) (AMO Chapter 13)

We switch to undirected networks, i.e., arcs are undirected.

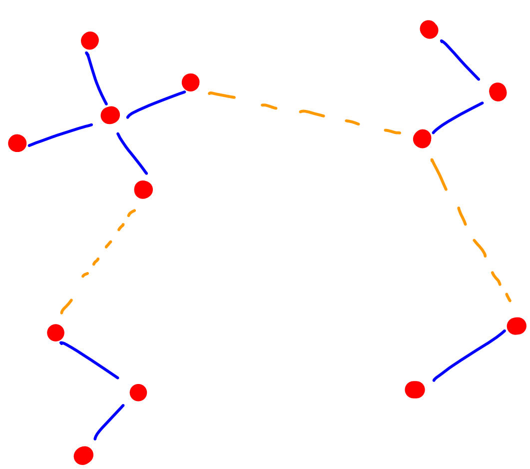
$G=(N,A)$, but now A has undirected arcs (or edges).

A **spanning tree** T of G is a connected, acyclic subgraph that spans all nodes in N. T has $n-1$ arcs.

A **minimum spanning tree** (MST) is a spanning tree that has the smallest total cost $\sum_{(i,j)\in T} c_{ij}$.

An application: Cluster analysis.

see AMO for more applications



Can start building MST by assembling smaller trees. Stop at a specified cut-off value (for c_{ij}).

The connected components (subtrees) form the clusters.

Optimality Conditions

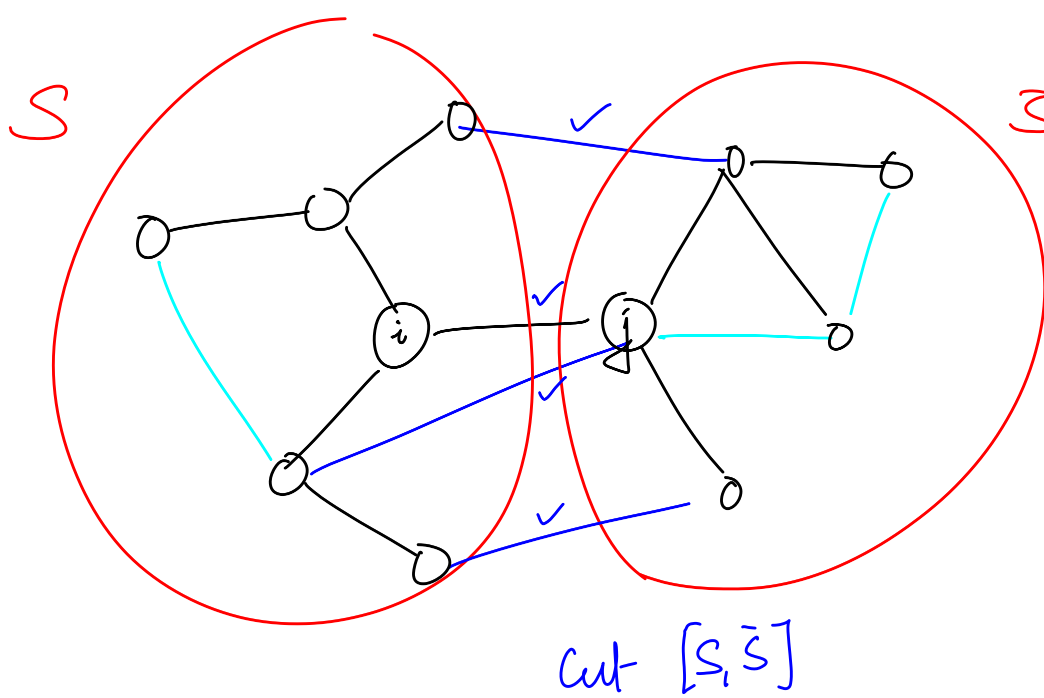
- cut optimality conditions
- path optimality conditions

Notation Arcs in a spanning tree are called tree arcs, while arcs not in the spanning tree are non-tree arcs.

Observation

(1) For every non-tree arc (k, l), there is a unique path in T connecting k and l.

(2) Deleting a tree arc (i, j) from a spanning tree divides N into two disjoint subsets $S, \bar{S}$. The arcs of G (k, l) with $k \in S$, $l \in \bar{S}$ form a cut $[S, \bar{S}]$.

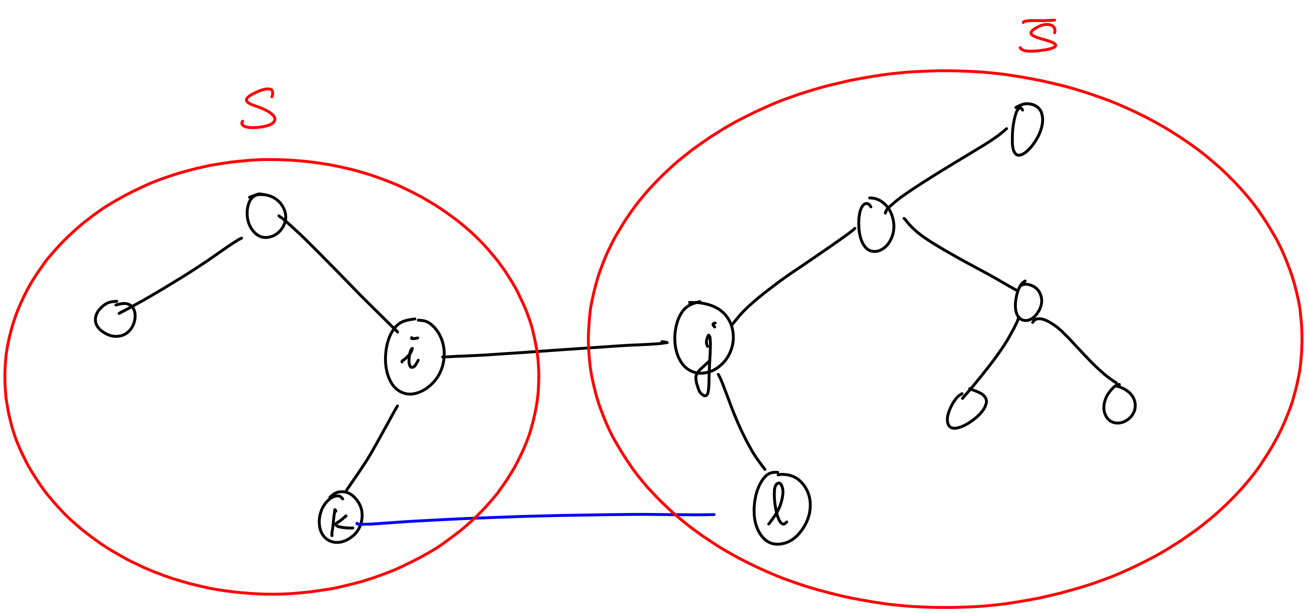


Cut Optimality Conditions

A spanning tree T is an MST iff $\forall (i,j) \in T$, $c_{ij} \le c_{kl}$ $\forall (k,l) \in [S, \bar{S}]$, where $[S, \bar{S}]$ is the cut formed by deleting (i,j) from T.

Proof

($\Longrightarrow$) Assume T is an MST but $c_{ij} > c_{kl}$ for some $(i,j) \in T$ and $(k,l) \in [S, \bar{S}]$. Then we can replace (i,j) by (k,l) in T to obtain another spanning tree with smaller total cost, contradicting minimality of T.



($\Leftarrow$) Let T be a spanning tree and $c_{ij} \le c_{kl}$ holds.
We want to show T is an MST.

Let T° be an MST and $T^\circ \neq T$.

$\Rightarrow \exists (i,j) \in T$ such $(i,j) \notin T^\circ$.

$\Rightarrow \exists (k,l) \in T^\circ$ such that $k \in S$, $l \in \bar{S}$, where $[S, \bar{S}]$ is the cut obtained by deleting (i,j) from T.
The pieces in S and $\bar{S}$ need to be connected somehow in T°.

T satisfies cut optimality conditions $\Rightarrow c_{ij} \le c_{kl}$.

T° is an MST $\Rightarrow c_{kl} \le c_{ij}$.

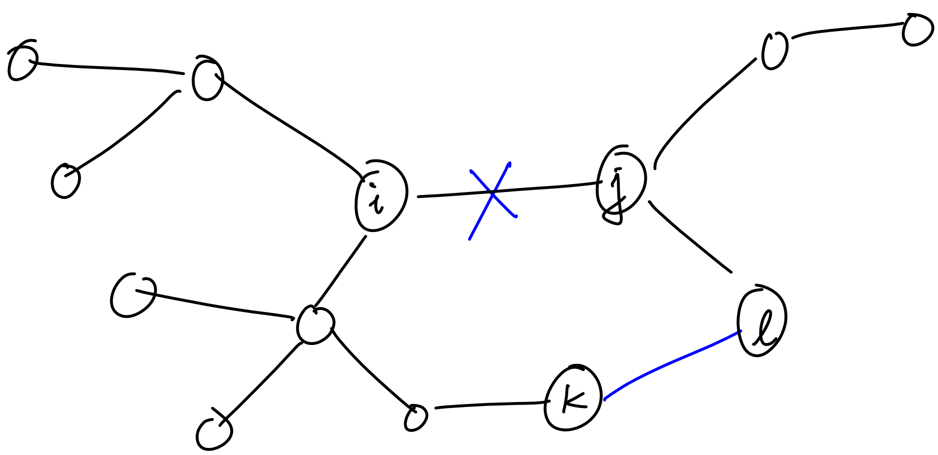
$\Rightarrow c_{ij} = c_{kl}$.

Hence we can replace (i,j) in T with (k,l). Repeat this process until $T = T^\circ$. We have not changed the total cost all along, so T must be an MST as well. $\square$

Path Optimality Conditions

A spanning tree T is an MST iff for every non-tree arc $(k,l) \in G$, $c_{ij} \le c_{kl}$ $\forall (i,j)$ in the path connecting k and l in T.

Proof If T is an MST and $c_{ij} > c_{kl}$, then replacing (i,j) with (k,l) gives a contradiction.



$(\Leftarrow)$ Let T be a spanning tree satisfying the path optimality conditions. We show that T will satisfy the cut optimality conditions.

$(k,l) \in [S, \bar{S}] \Rightarrow$ There is a unique path in T connecting k and l. (i,j) is the only arc connecting S and $\bar{S}$. Hence (i,j) is in this path.

Path optimality $\Rightarrow$ $c_{ij} \le c_{kl}$. This holds $\forall (k,l) \in [S, \bar{S}]$.

$\Rightarrow$ cut optimality conditions hold. $\square$