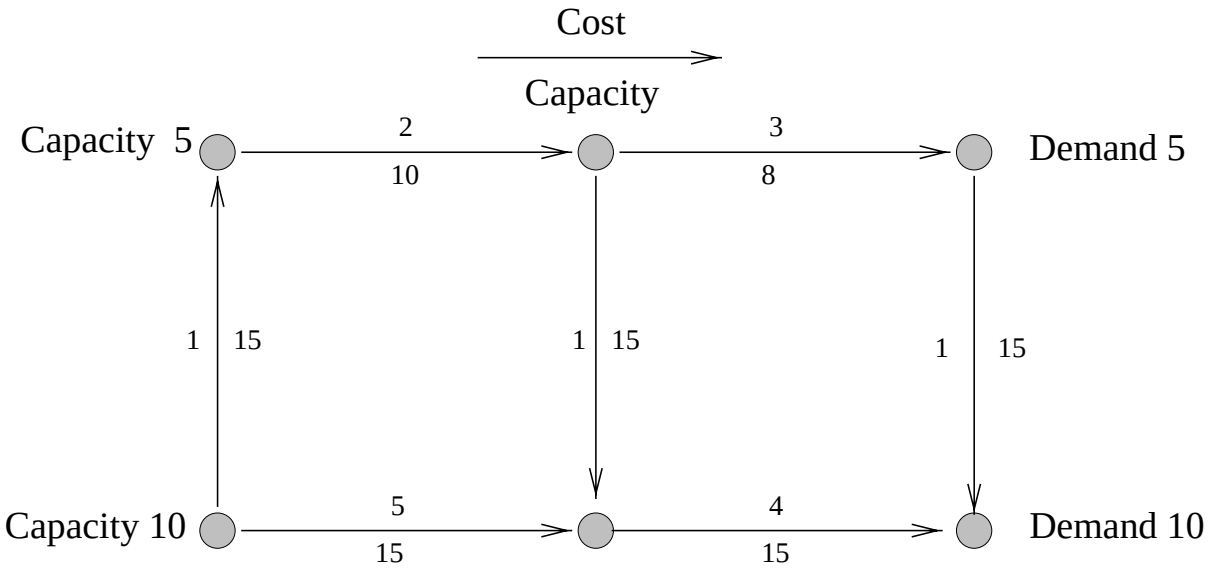
Min Cost Flow - Terminology

We consider a digraph $G = (V(G), E(G))$, in which each edge e has a capacity $u_e \in \mathcal{R}_+$ and a unit transportation cost $c_e \in \mathcal{R}$. Each vertex v furthermore has a demand $b_v \in \mathcal{R}$. If $b_v \geq 0$ then v is a **sink**, and if $b_v < 0$ then v is a **source**. We assume that $b(V) = \sum_{v \in V} b_v = 0$.



The Min Cost Flow problem consists in supplying the sinks from the sources by a flow in the cheapest possible way:

$$\min \sum_{e \in E} c_e x_e$$

$$\forall v \in V \;\; f_x(v) = b_v$$

$$\forall (v, w) \in E \;\; : \;\; 0 \leq x_{vw} \leq u_{vw}$$

Min Cost Flow - Primal LP.

The Min Cost Flow problem is an LP-problem:

	x_{e_1}	x_{e_2}	...	x_{ij}	...	x_{e_m}		
	c_{e_1}	c_{e_2}	...	c_{ij}	...	c_{e_m}		
1	-1	.	...		...		$=$	b_1
2	.	.	...	.	...	.	$=$	b_2
.	.	.	...	.	...	.	$=$	.
i	1	.	...	-1	...	.	$=$	b_i
.	.	.	...	.	...	.	$. =$	
j	.	.	...	1	...	.	$=$	b_j
.	.	.	...	.	...	.	$=$	.
n	.	.	...	.	...	.	$=$	b_n
e_1	-1						$\geq$	$-u_1$
e_2		-1					$\geq$	$-u_2$
.	.	.	.	.	.	.	$\geq$	.
(i,j)				-1			$\geq$	$-u_{ij}$
.	.	.	.	.	.	.	$\geq$	.
e_m						-1	$\geq$	$-u_1$

Min Cost Flow - Dual LP.

The dual variables corresponding to the flow balance equations are denoted y_v, $v \in V$, and those corresponding to the capacity constraints are denoted z_{vw}, $(v,w) \in E$.

The dual problem is now:

$$max : \sum_{v \in V} b_v y_v - \sum_{(w,v) \in E} u_{wv} z_{wv}$$

$$\forall (v,w) \in E \ : -y_v + y_w - z_{vw} \leq c_{vw} \quad \Leftrightarrow$$

$$\forall (v,w) \in E \ : -c_{vw} - y_v + y_w \leq z_{vw}$$

$$\forall (v,w) \in E \ : z_{vw} \geq 0$$

$\overline{c}_{vw} = c_{vw} + y_v - y_w$ is called *the reduced cost* for the edge (v,w), and hence $-c_{vw} - y_v + y_w \leq z_{vw}$ is equivalent to

$$-\overline{c}_{vw} \leq z_{vw}$$

When is a set of feasible solutions x, y, og z optimal ?

Min Cost Flow - Optimality conditions I.

If $u_e = +\infty$ (i.e. no capacity constraints for e) then z_e must be 0 and hence $\overline{c}_e \geq 0$ just has to hold (primal optimality condition for the LP).

If $u_e \neq +\infty$ then $z_e \geq 0$ and $z_e \geq -\overline{c}_e$ must hold. z has *negative* coefficient in the objective function - hence the best choice for z is as small as possible: $z_e = max\{0, -\overline{c}_e\}$. Therefore, the optimal value of z_e is uniquely determined from the other variables, and z_e is "unnecessary" in the dual problem.

The complementary slackness conditions (each primal variable times the corresponding dual slack must equal 0, and each dual variable times the corresponding primal slack must equal 0 in optimum) now give:

$$x_{vw} > 0 \ \Rightarrow -\overline{c}_{vw} = z_e = max(0, -\overline{c}_{vw})$$

$$i.e.\ (x_e > 0 \ \Rightarrow -\overline{c}_e \geq 0) \ \equiv \ (\overline{c}_e > 0 \Rightarrow x = 0)$$

and

$$z_e > 0 \ \Rightarrow x_e = u_e$$

$$i.e.\ (-\overline{c}_e > 0 \Rightarrow x_e = u_e) \ \equiv \ (\overline{c}_e < 0 \Rightarrow x_e = u_e)$$

Min Cost Flow - Optimality conditions II.

Summing up: A primal feasible flow satisfying demands in sinks from sources respecting the capacity constraints *is optimal* **if and only if** we can find a dual solution $y_e,\ e \in E$ such that for all $e \in E$ it holds that:

$$\overline{c}_e < 0 \Rightarrow x_e = u_e\ (\neq \infty)$$

$$\overline{c}_e > 0 \Rightarrow x_e = 0$$

All pairs (x, y) of optimal solutions satisfy these conditions.

- and so what ?

Min Cost Flow - Optimality conditions III.

For a legal flow x in G, the residual graph is (like for Max Flow) a graph, in which the paths indicate *how flow excess can be moved* in G given that the flow x already is present. The only difference is that each edge has a cost assigned.

The residual graph $G(x)$ for G wrt. x is defined by

$$V(G(x)) = V(G)$$

$$E(G(x)) = E_x =$$

$$\{(v,w)|(v,w) \in E \wedge x_{vw} < u_{vw}\} \cup \{(v,w)|(w,v) \in E \wedge x_{wv} > 0\}$$

The unit cost c'_{vw} for an edge with $x_{vw} < u_{vw}$ is c_{vw}, while c'_{vw} for an edge with $x_{wv} > 0$ is $-c_{vw}$.

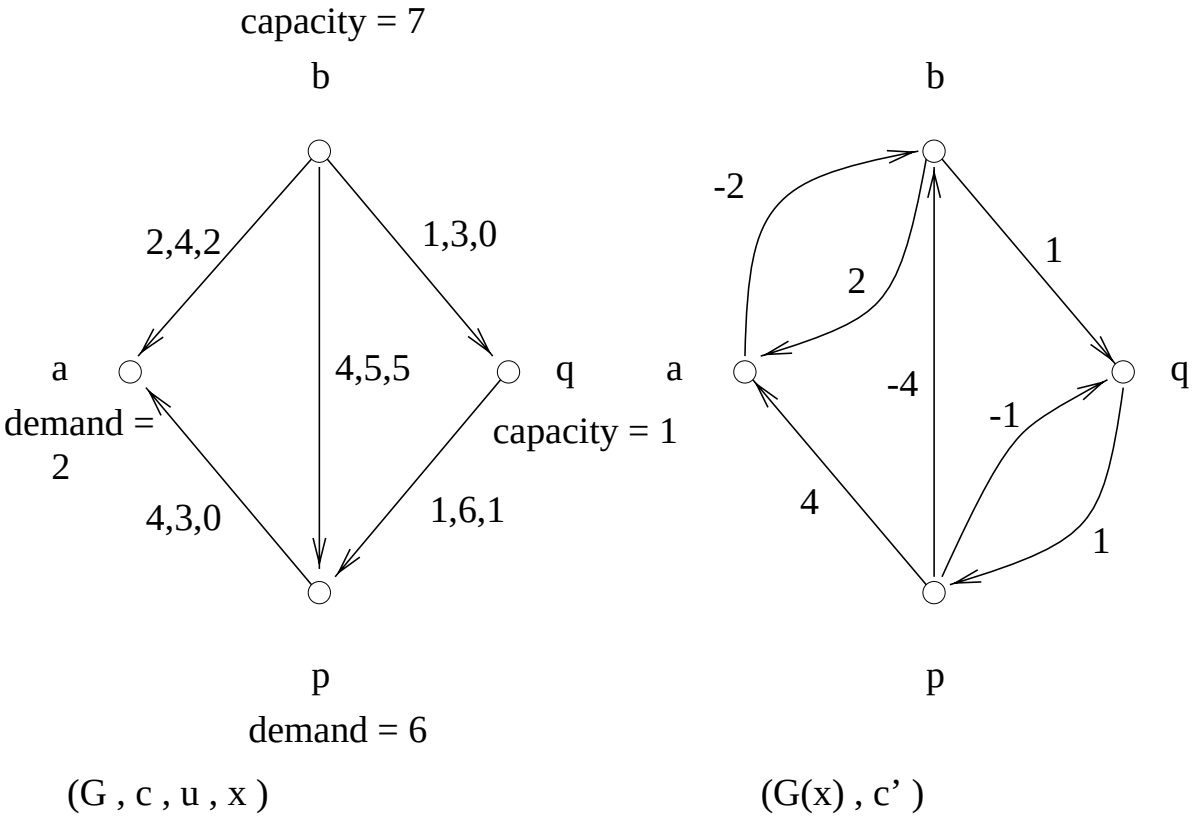
Note that a dicircuit with negative cost in $G(x)$ corresponds to a negative cost circuit in G, if cost are added for forward edges and subtracted for backward edges.

Note also that if a set of vertex potentials $y_v,\ v \in V$ are given, and the cost of a circuit wrt. the reduced costs for the edges ($\overline{c}_{vw} = c_{vw} + y_v - y_w$) are calculated, the cost remains the same as the original costs - the vertex potentials are "telescoped" to 0.

Min Cost Flow - Negative cost circuits.

A primal feasible flow satisfying sink demands from sources and respecting the capacity constraints *is optimal* **if and only if** an x-augmenting circuit with negative c-cost (or negative $\overline{c}$-cost - there is no difference) does not exist.

This is the idea behind the identification of optimal solutions in the *Network Simplex Algorithm*.



Transshipment - Network Simplex I.

We consider first the Transshipment-problemet (Min Cost Flow without capacity constraints).

Reminder: A *tree* in a digraph is a set $T \subseteq E$, such that this is a tree in the underlying undirected graph.

A *tree solution* for the transshipment-problem give by $G = (V, E)$, the demands $b_v,\ v \in V$ and the costs $c_{vw},\ (v, w) \in E$, is a flow $x \in \mathcal{R}^{\mathcal{E}}$ satisfying:

$$\forall v \in V :\ f_x(v) = b_v$$

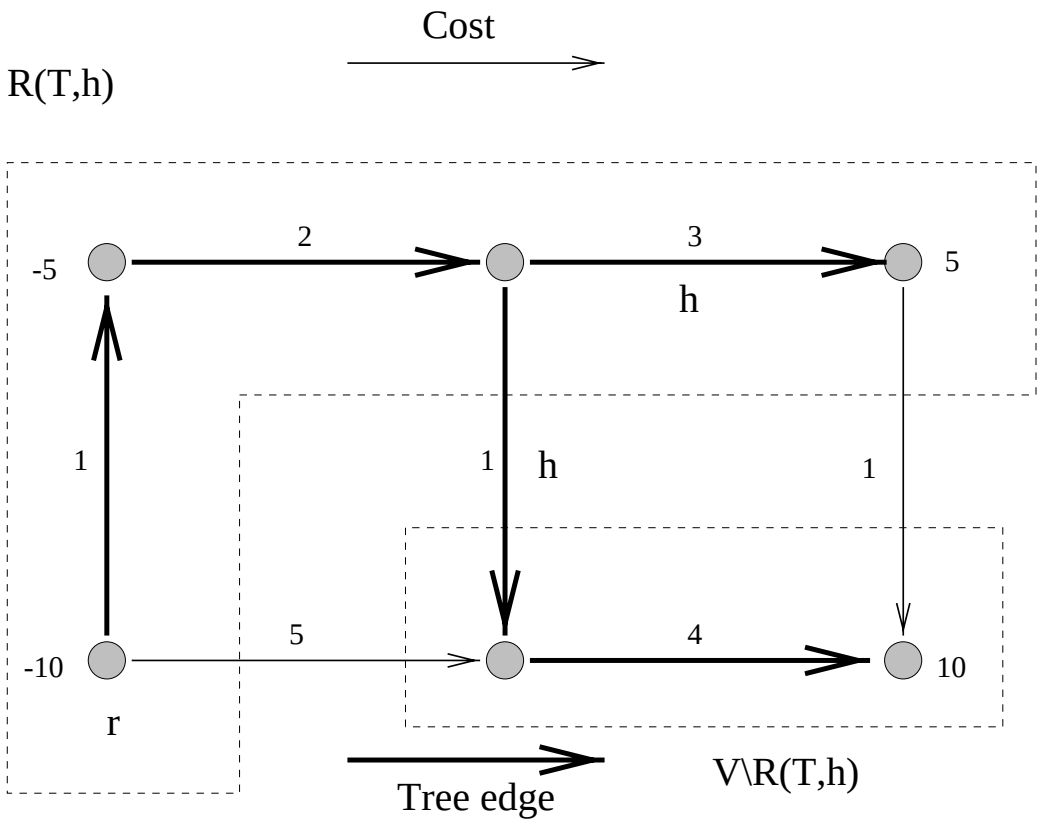
$$\forall e \notin T :\ x_e = 0$$

In words: No flow in edges not in T, and all demands satisfied. Note that tree solutions are normally *not* feasible, since edges with negative flow may exist (cf. basic solution and *feasible* basic solution in LP).

Transshipment - Network Simplex II.

Does a tree solution exist for any tree T ? YES !!!

Select a vertex r called the *root* of T. Consider an edge h in T. h "splits" V into two: a part containing r denoted $R(T,h)$ and a remainder - $V \setminus R(T,h)$.



If h starts in $R(T,h)$ and ends in $V \setminus R(T,h)$ then set

$$x_h = b(V \setminus R(T,h)) = -b(R(T,h))$$

and if h starts in $V \setminus R(T,h)$ and ends in $R(T,h)$ then set

$$x_h = b(R(T,h))$$

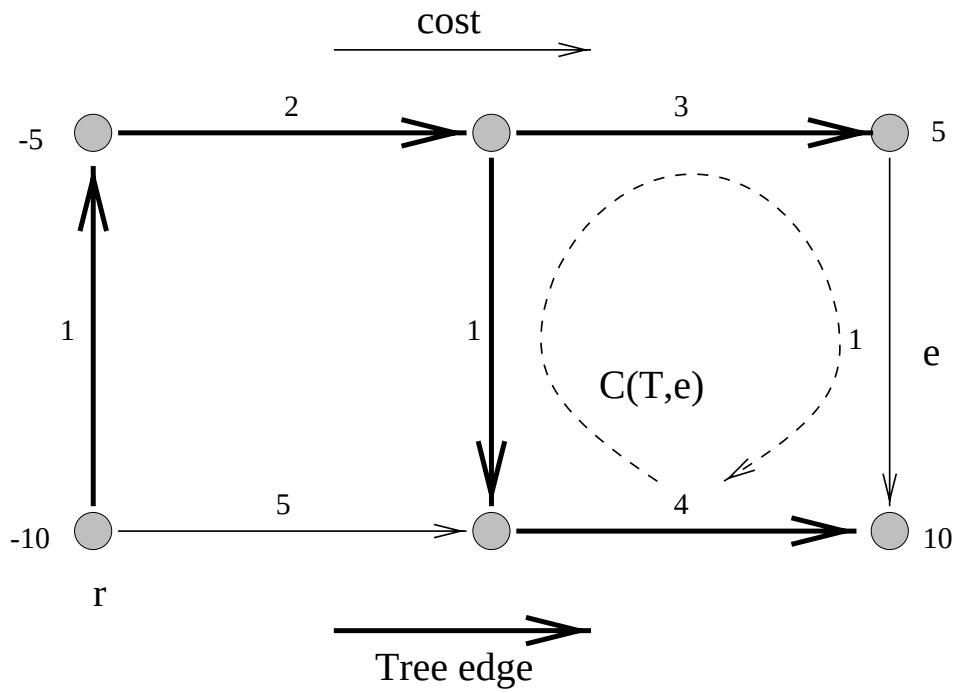
Transshipment - Network Simplex III.

- The tree solution for T is unique.
- If (G, c, b) has a feasible solution, it also has a feasible tree solution.
- If (G, c, b) hars an optimal solution, it also has an optimal tree solution.

The Network Simplex Method moves from tree solution to tree solution using negative cost circuits $C(T, e)$ consisting of tree edges *and exactly one* non-tree edge e (think of the tree edges as basic variables and the non-tree edge as the non-basic variable of a Simplex iteration):

- $C(T, e) \subseteq T \cup \{e\}$
- e is forward in $C(T, e)$

Transshipment - Network Simplex IIII.



Set the vertex potential $y_v,\ v \in V$ to the cost of the path in T from r to v (counted with sign: plus for a forward edge, minus for a backward edge). It now holds:

For all $v, w \in V$, the cost of the path from v to w in T equals $y_w - y_v$ (why ?). But then the reduced costs $\overline{c}_{vw}$ (defined by $c_{vw} + y_v - y_w$) satisfy:

$$\forall e \in T:\ \overline{c}_e = 0$$

$$\forall e \notin T:\ \overline{c}_e = c(C(T,e))$$

Transshipment - Network Simplex V.

If T determines a *feasible* tree solution x and $C(T,e)$ has non-negative cost $\forall e \notin T$ (i.e. $\overline{c}_e \geq 0$), then x is optimal.

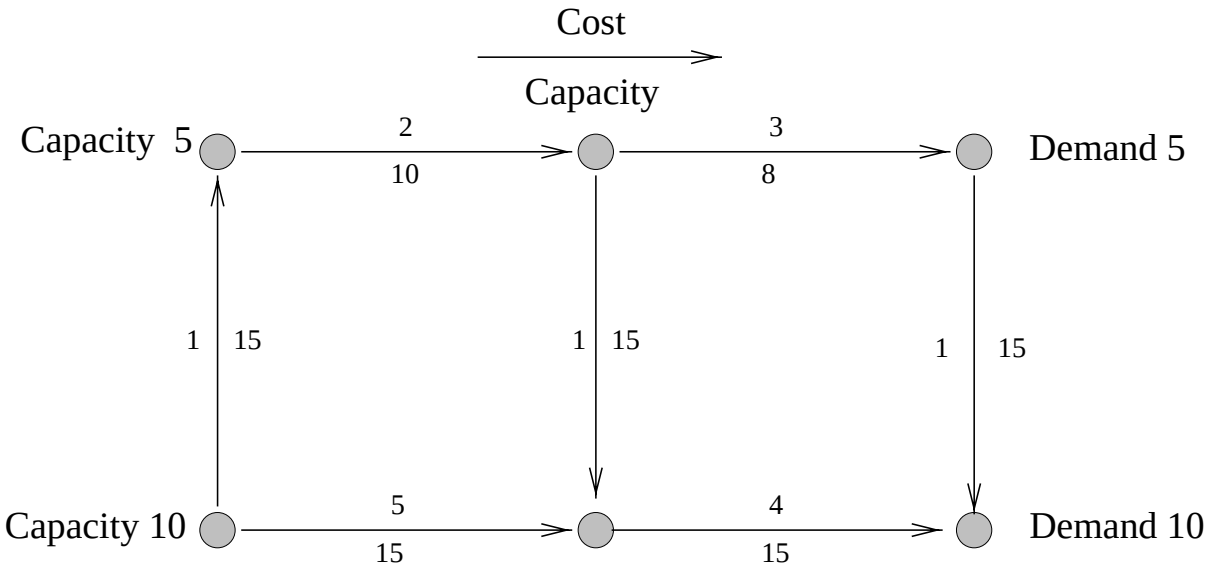
Network Simplex algorithm

1. Find a tree T with a corresponding feasible tree solution x. Select an $r \in V$ as root for T.
2. Compute y_v as the length of the $r-v$–path in T, costs counted with sign (plus for forward edges, minus for backward edges).
3. **while** $\exists e=(v,w)$ $s.t$ $\overline{c}_{vw} = c_{vw} + y_v - y_w < 0$ **do**
   - Select one of these;
   - If alle edges in $C(T,e)$ are forward - STOP - the problem is "unbounded";
   - Find $\theta = min\{x_j |$ j backward in $C(T,e)\}$ and an edge h with $x_h = \theta$;
   - Increase x with θ along $C(T,e)$ - increase flow in forward edges and decrease flow in backward edges;
   - $T := (T \cup \{e\}) \setminus \{h\}$;
   - update y as in (2) above;

   endwhile

What if edge capacities are present ? I

We consider the Min Cost flow - problem given by the network $G = (V, E)$ with demands $b_v,\ v \in V$, capacities $u_{vw},\ (v, w) \in E$ and costs $c_{vw},\ (v, w) \in E$.



Remember the optimality conditions for a feasible flow $x \in \mathcal{R}^{\mathcal{E}}$:

$$\overline{c}_e < 0 \Rightarrow x_e = u_e\ (\neq \infty)$$

$$\overline{c}_e > 0 \Rightarrow x_e = 0$$

What if edge capacities are present ? II

The concept of a tree solution is extended to capacity constrained problems - a non-tree edge e may have flow either 0 or u_e. $E \setminus T$ is hence split into two edge sets, L and U. Edges in L must have flow 0 - edges in U must be filled to their capacity. The tree solution x must satisfy:

$$\forall v \in V : \; f_x(v) = b_v$$

$$\forall e \in L : \; x_e = 0$$

$$\forall e \in U : \; x_e = u_e$$

As before, the tree solution for T is unique:

Select a vertex r called the *root* of T. Consider an edge h in T. h "splits" V into two: a part containing r denoted $R(T,h)$ and a remainder - $V \setminus R(T,h)$.

Consider now the *modified demand* $B(V \setminus R(T,h))$ in $V \setminus R(T,h)$ *given that a tree solution is sought*:

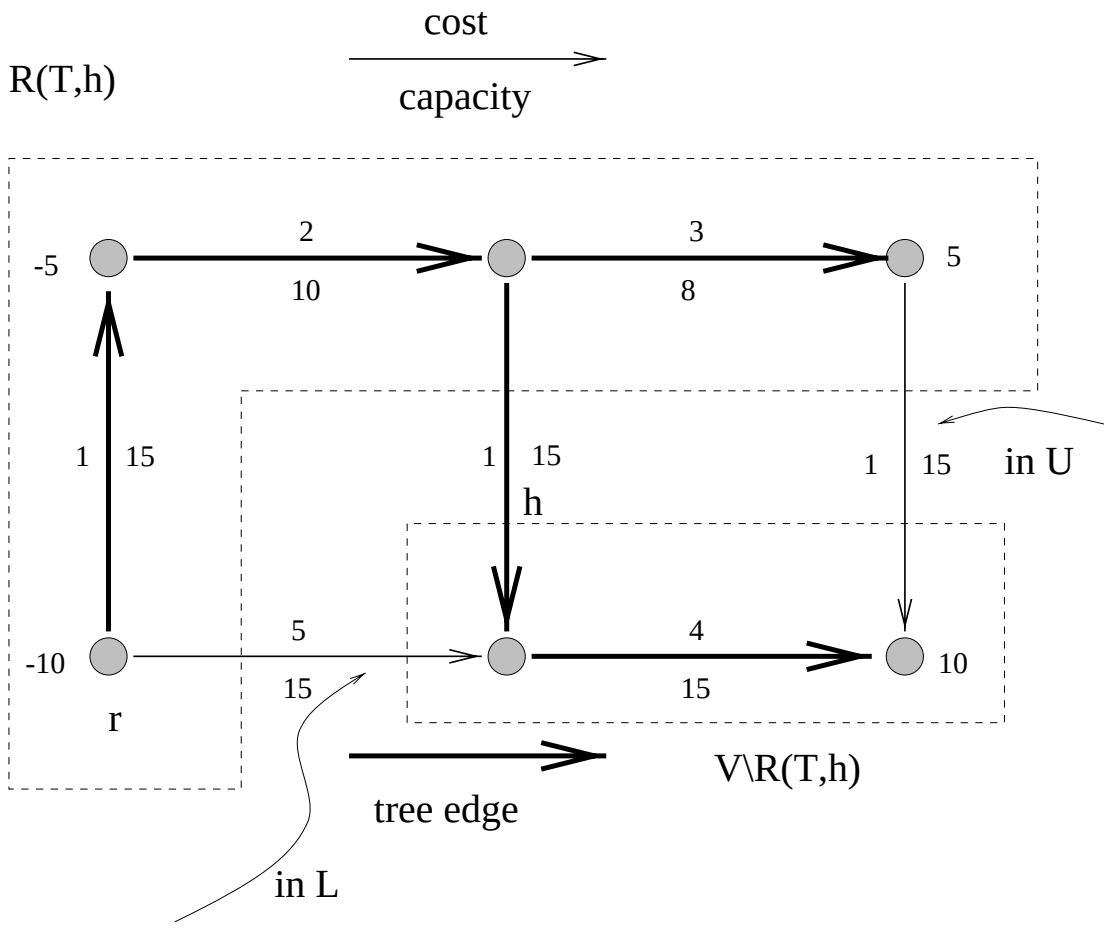
$$\begin{aligned} & B(V \setminus R(T,h)) = b(V \setminus R(T,h)) \\ & - \sum\nolimits_{\{(v,w) \in U | v \in R(T,h),\; w \in V \setminus R(T,h)\}} u_{vw} \\ & + \sum\nolimits_{\{(v,w) \in U | v \in V \setminus R(T,h),\; w \in R(T,h)\}} u_{vw} \end{aligned}$$

If h starts in $R(T,h)$ and ends in $V \setminus R(T,h)$ the set

$$x_h = B(V \setminus R(T,h))$$

and if h starts in $V \setminus R(T,h)$ and ends in $R(T,h)$ then set

$$x_h = -B(V \setminus R(T,h))$$

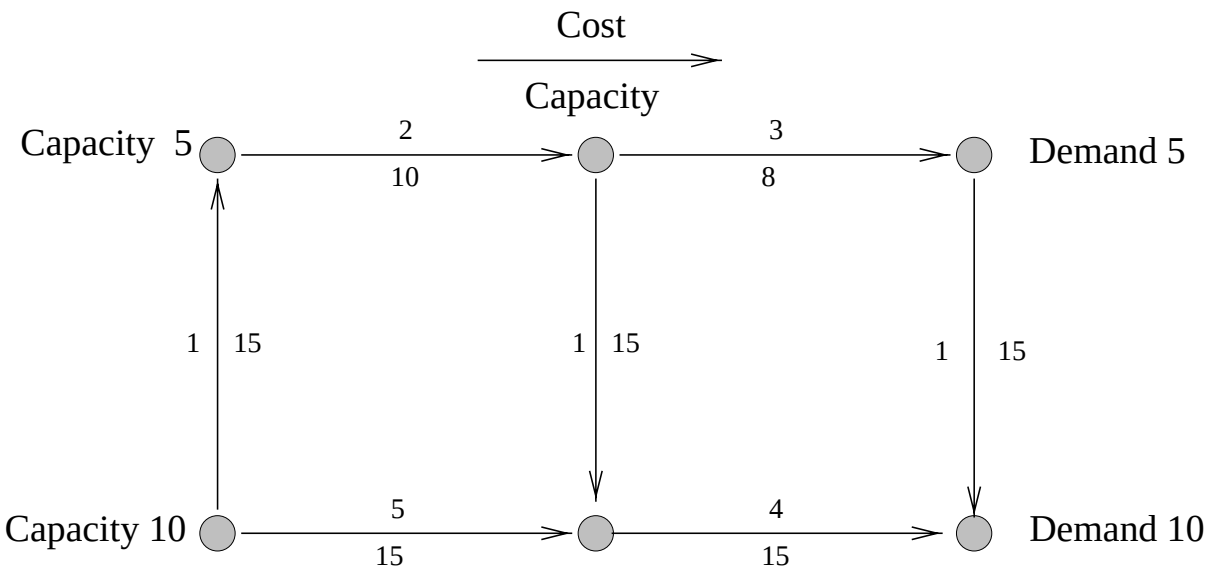


Network Simplex for Min Cost Flow I

Idea: Start with a feasible tree solution - find it e.g using a Max Flow algorithm.

We now search for *either* a non-tree edge e with $x_e = 0$ and negative reduced cost *or* a non-tree edges e with $x_e = u_e$ and positive reduced cost. Given e and (T, L, U), the corresponding circuit is denoted $C(T, L, U, e)$.

In this, the flow must be *increased* in forward edges respecting capacity constraints and *decreased* in backward edges respect the non-negativity constraints.



Network Simplex for Min Cost Flow II.

1. Find a tree T with a corresponding feasible tree solution x. Select an $r \in V$ as root for T.
2. Compute y_v as the length of the $r - v$–path in T, costs counted with sign (plus for forward edges, minus for backward edges).
3. **while** $(\exists e = (v,w) \in L \; s.t \; \overline{c}_{vw} = c_{vw} + y_v - y_w < 0) \; \vee \; (\exists e = (v,w) \in U \; s.t \; \overline{c}_{vw} = c_{vw} + y_v - y_w > 0)$ **do**
   - Select one of these;
   - If no edge in $C(T, L, Ue)$ is backward and no forward edge has limited capacity STOP - the problem is unbounded;
   - Find $\theta_1 = min\{x_j |$ j backward in $C(T, L, U, e)\}$, $\theta_2 = min\{u_j - x_j |$ j forward in $C(T, L, U, e)\}$, and an edge h giving rise to $\theta = min\{\theta_1, \theta_2\}$;
   - Increase x by θ along $C(T, L, U, e)$;
   - $T := (T \cup \{e\}) \setminus \{h\}$;
   - Update L, U by removing e and inserting h in the relevant one of L and U;
   - Update y as in (2) above;

   endwhile

Transshipment - the LP-version I

We consider a digraph $G = (V(G), E(G))$, in which each edge e has a capacity $u_e \in \mathcal{R}_+$ and a transportation cost per unit $c_e \in \mathcal{R}$. Each vertex v furthermore has a demand $b_v \in \mathcal{R}$. If $b_v \geq 0$ then v is a **sink**, and if $b_v < 0$ then v is a **source**. We assume that $b(V) = \sum_{v \in V} b_v = 0$.

A **basis** is a set of edges constituting a spanning tree in the undirected graph underlying the network, i.e. connected, circuit-free and with $|V| - 1$ edges. A basis is also called a *basis tree*, and the edges *basis-edges*.

A **feasible basic solution** is a feasible way of sending flow from sources to sinks satisfying the demands and with no loops, such that *at most* $|V| - 1$ edges carry flow. If less than $|V| - 1$ have positive flow, a sufficient number of other edges is added to form a basis (these will have 0 flow).

Transshipment - the LP-version II

Iterative step: Suppose a feasible basis is given. For each edge, a reduced cost d_{ij} is defined by $d_{ij} = c_{ij} + y_i - y_j$.

1: Select a vertex r and assign it vertex potential 0: $y_r := 0$.

2: calculate y_v for each other vertex using the basis edges and the fact that for these, $d_{ij} = 0$.

3: Now all $y_v, v \in V$ are known. Compute d_{ij} based on on the definition for each non-basis edge.

4: If all d_{ij} are non-negative STOP - optimum has been identified.

5: Select an edge (v, w) with $d_{ij} < 0$ (and this must be a non-basis edge).

6: Update the flow in that circuit, which is the result of the addition of (v, w) to the basis tree, such that at least one basis edge (i, j) gets a resulting flow of 0.

7: The new basis results from the exchange of $(v, w$ and (i, j) in the basis tree. Go to step 1.