

DM545/DM871
Linear and Integer Programming

Lecture 13
Network Flows, Cntd

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

1. Duality in Network Flow Problems

2. Network Simplex

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

1. Duality in Network Flow Problems

2. Network Simplex

Shortest Path - Dual LP

$$z = \min \sum_{ij \in A} c_{ij} x_{ij}$$

$$\sum_{j:ji \in A} x_{ji} - \sum_{j:ij \in A} x_{ij} = 1 \quad \text{for } i = s \quad (\pi_s)$$

$$\sum_{j:ji \in A} x_{ji} - \sum_{j:ij \in A} x_{ij} = 0 \quad \forall i \in V \setminus \{s, t\} \quad (\pi_i)$$

$$\sum_{j:ji \in A} x_{ji} - \sum_{j:ij \in A} x_{ij} = -1 \quad \text{for } i = t \quad (\pi_t)$$

$$x_{ij} \geq 0 \quad \forall ij \in A$$

Dual problem:

$$g^{LP} = \max \pi_s - \pi_t$$
$$\pi_j - \pi_i \leq c_{ij} \quad \forall ij \in A$$

Hence, the shortest path can be found by potential values π_i on nodes such that $\pi_s = z, \pi_t = 0$ and $\pi_j - \pi_i \leq c_{ij}$ for $ij \in A$

Maximum (s, t) -Flow

Adding a backward arc from t to s :

$$\begin{aligned} z &= \max x_{ts} \\ \sum_{j:ji \in A} x_{ij} - \sum_{j:ij \in A} x_{ji} &= 0 & \forall i \in V & \quad (\pi_i) \\ x_{ij} &\leq u_{ij} & \forall ij \in A & \quad (w_{ij}) \\ x_{ij} &\geq 0 & \forall ij \in A & \end{aligned}$$

Dual problem:

$$\begin{aligned} g^{LP} &= \min \sum_{ij \in A} u_{ij} w_{ij} \\ \pi_i - \pi_j + w_{ij} &\geq 0 & \forall ij \in A \\ \pi_t - \pi_s &\geq 1 \\ w_{ij} &\geq 0 & \forall ij \in A \end{aligned}$$

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

$$g^{LP} = \min \sum_{ij \in A} u_{ij} w_{ij} \quad (1)$$

$$\pi_i - \pi_j + w_{ij} \geq 0 \quad \forall ij \in A \quad (2)$$

$$\pi_t - \pi_s \geq 1 \quad (3)$$

$$w_{ij} \geq 0 \quad \forall ij \in A \quad (4)$$

- Without (3) all potentials would go to 0.
- Keep w low because of objective function
- Keep all potentials low \rightsquigarrow (3) $\pi_s = 0, \pi_t = 1$
- Cut C : on left =1 on right =0. Where is the transition?
- Vars w identify the cut $\rightsquigarrow \pi_j - \pi_i + w_{ij} \geq 0 \rightsquigarrow w_{ij} = 1$

$$w_{ij} = \begin{cases} 1 & \text{if } ij \in C \\ 0 & \text{otherwise} \end{cases}$$

for those arcs that minimize the cut capacity $\sum_{ij \in A} u_{ij} w_{ij}$

- Complementary slackness: $w_{ij} = 1 \implies x_{ij} = u_{ij}$

Theorem

A strong dual to the max (st) -flow is the minimum (st) -cut problem:

$$\min_X \left\{ \sum_{ij \in A: i \in X, j \notin X} u_{ij} : s \in X \subset V \setminus \{t\} \right\}$$

Optimality Condition

- Ford Fulkerson augmenting path algorithm $O(m|x^*|)$
- Edmonds-Karp algorithm (augment by shortest path) in $O(nm^2)$
- Dinic algorithm in layered networks $O(n^2m)$
- Karzanov's push relabel $O(n^2m)$

Min Cost Flow - Dual LP

$$\begin{aligned} \min \quad & \sum_{ij \in A} c_{ij} x_{ij} \\ \sum_{j:ji \in A} x_{ij} - \sum_{j:ij \in A} x_{ji} &= b_i & \forall i \in V & \quad (\pi_i) \\ x_{ij} &\leq u_{ij} & \forall ij \in A & \quad (w_{ij}) \\ x_{ij} &\geq 0 & \forall ij \in A & \end{aligned}$$

Dual problem:

$$\max \sum_{i \in V} b_i \pi_i - \sum_{ij \in E} u_{ij} w_{ij} \quad (1)$$

$$-c_{ij} - \pi_i + \pi_j \leq w_{ij} \quad \forall ij \in E \quad (2)$$

$$w_{ij} \geq 0 \quad \forall ij \in A \quad (3)$$

- define reduced costs $\bar{c}_{ij} = c_{ij} + \pi_i - \pi_j$, hence (2) becomes $-\bar{c}_{ij} \leq w_{ij}$
- $u_e = \infty$ then $w_e = 0$ (from obj. func) and $\bar{c}_{ij} \geq 0$ (from 2)
- $u_e < \infty$ then $w_e \geq 0$ and $w_e \geq -\bar{c}_{ij}$ then $w_e = \max\{0, -\bar{c}_{ij}\}$, hence w_e is determined by others and irrelevant
- Complementary slackness th. for optimal solutions:
each primal variable \times the corresponding dual slack must be equal 0, ie, $x_e(\bar{c}_e + w_e) = 0$;
 - $x_e > 0$ then $-\bar{c}_e = w_e = \max\{0, -\bar{c}_e\}$,
 $x_e > 0 \implies -\bar{c}_e \geq 0$ or equivalently (by negation) $\bar{c}_e > 0 \implies x_e = 0$
 each dual variable \times the corresponding primal slack must be equal 0, ie, $w_e(x_e - u_e) = 0$;
 - $w_e > 0$ then $x_e = u_e$
 $-\bar{c}_e > 0 \implies x_e = u_e$ or equivalently $\bar{c}_e < 0 \implies x_e = u_e$

Hence:

$$\bar{c}_e > 0 \text{ then } x_e = 0$$

$$\bar{c}_e < 0 \text{ then } x_e = u_e \neq \infty$$

Min Cost Flow Algorithms

The conditions derived can be used to define a solution approach for the minimum cost flow problem.

Directed cycle \equiv circuit

Note that if a set of potentials $\pi_i, i \in V$ are given, and the cost of a circuit wrt. the reduced costs for the edges ($\bar{c}_{ij} = c_{ij} + \pi_j - \pi_i$) are calculated, the cost remains the same as the original costs as the potentials are “telescoped” to 0.

Theorem (Optimality conditions)

Let x be feasible flow in $N(V, A, l, u, b)$ then x is min cost flow in N iff $N(x)$ contains no directed cycle of negative cost.

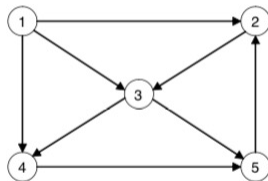
Note that a (directed) circuit with negative cost in $N(x)$ corresponds to a negative cost cycle in N , if costs are added for forward edges and subtracted for backward edges.

- Cycle canceling algorithm with Bellman Ford Moore for negative cycles $O(nm^2 UC)$,
 $U = \max |u_e|$, $C = \max |c_e|$
- Build up algorithms $O(n^2 mM)$, $M = \max |b(v)|$

1. Duality in Network Flow Problems

2. Network Simplex

Min Cost Flow



$$A = \begin{bmatrix} 1 & 1 & 1 & & & & & \\ -1 & & & 1 & & & & -1 \\ & -1 & & -1 & 1 & 1 & & \\ & & -1 & & -1 & & 1 & \\ & & & & & -1 & -1 & 1 \end{bmatrix}$$

$$\begin{array}{rcll} \min & 10x_{12} + 8x_{13} + x_{14} + 2x_{23} + x_{34} + 4x_{35} + 12x_{45} - 7x_{52} & & \\ & x_{12} & +x_{13} & +x_{14} & & & & = 10 \\ & -x_{12} & & & +x_{23} & & & -x_{52} = 4 \\ & & -x_{13} & & -x_{23} & +x_{34} & +x_{35} & = 0 \\ & & & -x_{14} & & -x_{34} & & +x_{45} = -6 \\ & & & & & & -x_{35} & -x_{45} & +x_{52} = -8 \end{array}$$

$$x \geq 0$$

- A is not full-rank: adding all rows \rightsquigarrow null vector, i.e., the rows of A are not linearly indep.
- Since we assume that total supply equal total demand, i.e., $\sum_{i \in V} b_i = 0$ then $\text{rank}[A] = \text{rank}[A \ b]$.
- Hence, one of the equations can be canceled.

- assume network N is connected
- **cycle**: here, a set of arcs forming a closed path (i.e., a path in which the first and the last node of the path coincide) when ignoring their orientation
- **spanning tree**: here, a tree that reaches every node (it coincides with the classical notion of spanning tree if one disregards arc orientation).

Theorem (Spanning Trees)

For an undirected graph $D' = (N, A')$, the following are equivalent:

- $G' = (N, E)$ is a tree (acyclic and connected);
- $G' = (N, E)$ is acyclic and has $n - 1$ arcs; and
- $G' = (N, E)$ is connected and has $n - 1$ arcs.

Since we know that the matrix A is not full-rank, a basis of A consists of only $n - 1$ linearly independent columns of A . These columns correspond to a collection of arcs of the flow network.

Theorem

*Given a connected flow network, letting A be its incidence matrix, a submatrix B of size $(n - 1) \times (n - 1)$ is a **basis** of A **if and only if** the arcs associated with the columns of B form a **spanning tree**.*

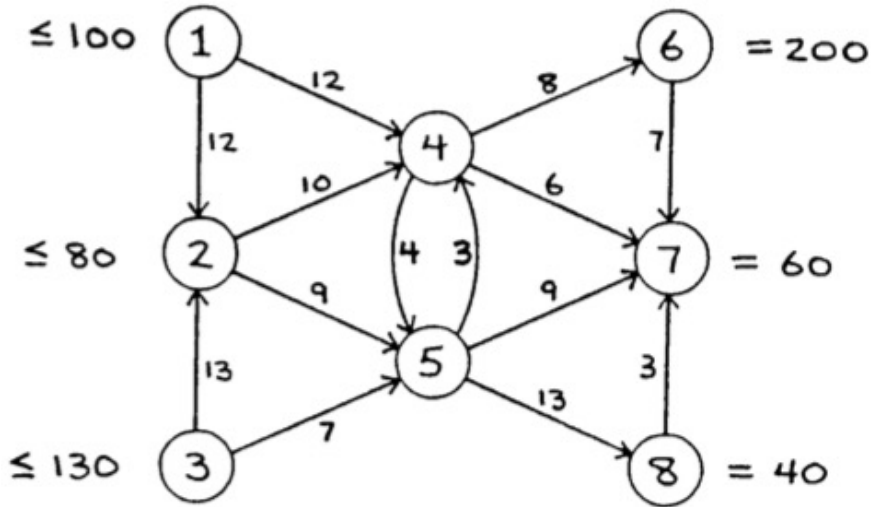
Proof:

if columns from A correspond to a spanning tree \implies they are lin. indep., B is upper triangular
if a subset of columns of A are a basis \implies they are $n - 1$ and acyclic

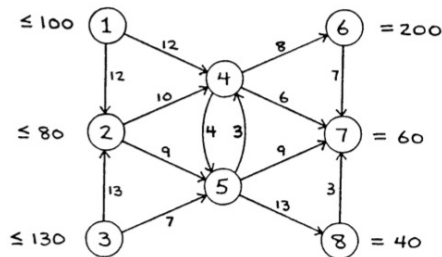
Hence, all basic feasible solutions explored by the simplex algorithm are spanning trees of the flow network.

As for any LP, also in min-cost flow problems there are feasible, infeasible and degenerate bases.
(feasible if $x_B = A_B^{-1}b \geq 0$).

Example

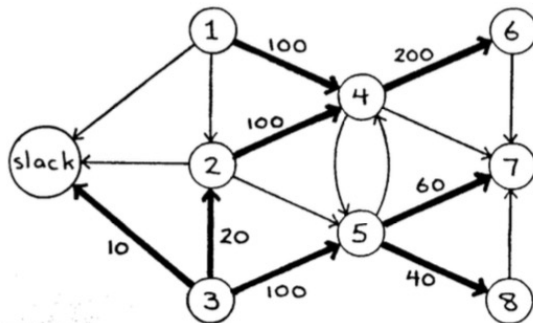


Example



$$\begin{aligned}
 &12x_{12} + 12x_{14} + 10x_{24} + 9x_{25} + 13x_{32} + 7x_{35} + 4x_{45} + 8x_{46} + 6x_{47} + 3x_{54} + 9x_{57} + 13x_{58} + 7x_{67} + 3x_{87} \\
 &+ x_{12} + x_{14} + x_{24} + x_{25} - x_{32} + x_{32} + x_{35} + s_1 = 100 \\
 &- x_{12} + x_{14} + x_{24} + x_{25} - x_{32} + x_{32} + x_{35} + s_2 = 80 \\
 &- x_{14} - x_{24} - x_{25} - x_{35} + x_{32} + x_{35} + s_3 = 130 \\
 &- x_{14} - x_{24} - x_{25} - x_{35} + x_{45} + x_{46} + x_{47} - x_{54} = 0 \\
 &- x_{35} - x_{45} + x_{54} + x_{57} + x_{58} = 0 \\
 &- x_{46} + x_{67} = -200 \\
 &- x_{47} - x_{57} - x_{67} - x_{87} = -60 \\
 &- x_{58} + x_{87} = -40 \\
 &x_{12}, x_{14}, x_{24}, x_{25}, x_{32}, x_{35}, x_{45}, x_{46}, x_{47}, x_{54}, x_{57}, x_{58}, x_{67}, x_{87}, s_1, s_2, s_3 \geq 0
 \end{aligned}$$

Example



$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ -1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

- solve $Bx_B = b$ in value of variables to check feasibility; easy because of structure or because done by updates.
- solve $\pi^T B = c_B^T$ in π (dual potential variables to derive reduced costs); easy because of structure of B .
- calculate $\bar{c}_{ij} = c_{ij} + \pi_j - \pi_i$

$$\pi_1 - \pi_4 = 12$$

$$\pi_2 - \pi_4 = 10$$

$$\pi_3 - \pi_2 = 13$$

$$\pi_3 - \pi_5 = 7$$

$$\pi_4 - \pi_6 = 8$$

$$\pi_5 - \pi_7 = 9$$

$$\pi_5 - \pi_8 = 13$$

$$\pi_3 = 0$$

$$\pi_3 = 0 \text{ and } \pi_3 - \pi_5 = 7 \Rightarrow \pi_5 = -7$$

$$\pi_5 = -7 \text{ and } \pi_5 - \pi_8 = 13 \Rightarrow \pi_8 = -20$$

$$\pi_5 = -7 \text{ and } \pi_5 - \pi_7 = 9 \Rightarrow \pi_7 = -16$$

$$\pi_3 = 0 \text{ and } \pi_3 - \pi_2 = 13 \Rightarrow \pi_2 = -13$$

$$\pi_2 = -13 \text{ and } \pi_2 - \pi_4 = 10 \Rightarrow \pi_4 = -23$$

$$\pi_4 = -23 \text{ and } \pi_4 - \pi_6 = 8 \Rightarrow \pi_6 = -31$$

$$\pi_4 = -23 \text{ and } \pi_1 - \pi_4 = 12 \Rightarrow \pi_1 = -11$$

$$d_{12} = c_{12} - \pi_1 + \pi_2 = 12 - (-11) + (-13) = 10$$

$$d_{25} = c_{25} - \pi_2 + \pi_5 = 9 - (-13) + (-7) = 15$$

$$d_{45} = c_{45} - \pi_4 + \pi_5 = 4 - (-23) + (-7) = 20$$

$$d_{54} = c_{54} - \pi_5 + \pi_4 = 3 - (-7) + (-23) = -13$$

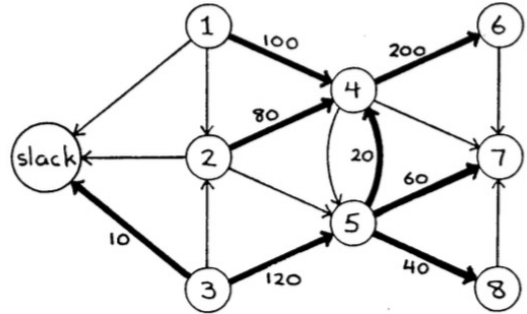
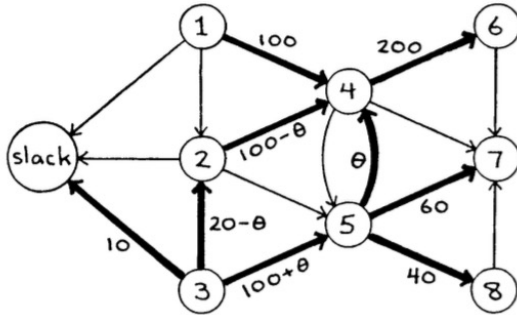
$$d_{47} = c_{47} - \pi_4 + \pi_7 = 6 - (-23) + (-16) = 13$$

$$d_{67} = c_{67} - \pi_6 + \pi_7 = 7 - (-31) + (-16) = 22$$

$$d_{87} = c_{87} - \pi_8 + \pi_7 = 3 - (-20) + (-16) = 7$$

$$d_1 = 0 - \pi_1 = -(-11) = 11$$

$$d_2 = 0 - \pi_2 = -(-13) = 13$$



How much can we increase the flow θ through
(54)?
Until (32) reaches zero

- It can be proved that, because the basis corresponds to a tree, the equations can always be solved by simple substitution.
- The order of substitution can always be found by “walking around the tree”.
- Efficient implementations further reduce the cost of determining π by updating it as they walk around the tree, rather than computing it anew at each iteration.
- When the network simplex steps are to be carried out by a computer, it is not so obvious how
- A few concise and clever data structures are used to represent the basis tree in a way that allows the walk around the tree and finding the circuit induced by the entering arc efficiently.
- The data structures can themselves be efficiently updated as the tree changes from iteration to iteration.