DM545

Linear and Integer Programming

Lecture 11
More on Network Flows

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

1. Duality in Network Flow Problems

2. More on Network Flows

Outline

1. Duality in Network Flow Problems

2. More on Network Flows

Shortest Path - Dual LP

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

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

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

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

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

Dual problem:

$$g^{LP} = \max \pi_t - \pi_s$$

$$\pi_j - \pi_i \leq c_{ij} \qquad \forall ij \in A$$

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

Maximum (s,t)-Flow

Adding a backward arc from t to s:

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

Dual problem:

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

$$
\begin{aligned}
g^{LP} &= \min \sum_{ij \in A} u_{ij} w_{ij} && && (1)\\
\pi_i - \pi_j + w_{ij} &\geq 0 && \forall ij \in A && (2)\\
\pi_t - \pi_s &\geq 1 && && (3)\\
w_{ij} &\geq 0 && \forall ij \in A && (4)
\end{aligned}
$$

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

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

  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:

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

Min Cost Flow - Dual LP

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

$$\sum_{j:ij \in A} x_{ij} - \sum_{j:ji \in A} x_{ji} = b_i \qquad \forall i \in V \qquad (\pi_i)$$

$$x_{ij} \leq u_{ij} \qquad \forall ij \in A \qquad (w_{ij})$$

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

Dual problem:

$$\begin{array}{lll} (1) & \max \sum_{i \in V} b_i \pi_i - \sum_{ij \in E} u_{ij} w_{ij} & \\ (2) & -c_{ij} - \pi_i + \pi_j \leq w_{ij} & \forall ij \in E \\ (3) & w_{ij} \geq 0 & \forall ij \in A \end{array}$$

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

  Hence:
  $\bar{c}_e < 0$ then $x_e = u_e \neq \infty$
  $\bar{c}_e > 0$ then $x_e = 0$

Outline

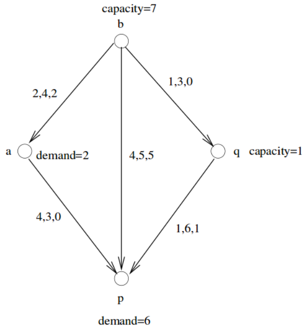
1. Duality in Network Flow Problems

2. More on Network Flows

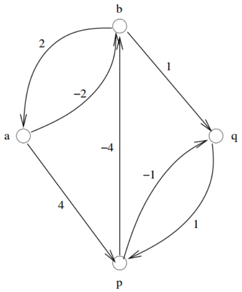
Residual Network

Residual Network $N(x)$**:**
how flow excess can be moved in G given that a flow x already exists
replace arc $ij \in N$ with arcs:

	residual capacity	cost
ij :	$r_{ij} = u_{ij} - x_{ij}$	c_{ij}
ji :	$r_{ji} = x_{ij}$	$-c_{ij}$



(N, c, u, x)



$(N(x), c')$

Min Cost Flow Algorithms

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.

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

Minimum spanning tree

Definition

Given a graph $G = (V, E)$

- a forest is a subgraph $G' = (V, E')$ containing no cycles
- a tree is a subgraph $G' = (V, E')$ that is a forest and is connected ($\exists$ a (uv)-path $\forall u, v \in V$)

Proposition

A graph $G = (V, E)$ is a tree iff

- *it is a forest containing exactly $n - 1$ edges*
- *it is an edge minimal connected graph spanning V*
- *it contains a unique path between every pair of nodes of V*
- *the addition of an edge not in E creates a unique cycle.*

Solvable via greedy algorithm (Kruskall)

$$\min \sum_{e \in E} c_e x_e \tag{5}$$

$$\sum_{e \in E} x_e = n - 1 \tag{6}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \qquad \text{for } 2 \leq |S| \leq n-1 \tag{7}$$

$$x_e \geq 0 \qquad \text{for } e \in E \tag{8}$$

$$x \in \mathbb{Z}^{|E|} \tag{9}$$

Theorem

The convex hull of the incidence vectors of the forests in a graph is given by the constraints (2)-(3)

Network simplex

- Improved version of the simplex method for network flows (still not polynomial but performs well in practice)
- it goes through same basic steps at each iteration:
  finding basic variable + determining leaving variable + solving for the new basis
- executes these steps exploiting network structure without needing a simplex tableau
- Key idea: network representation of basic feasible solutions

- in min cost flow formulation one of the node constraints is redundant (summing all these constraints yields zero on both sides - $\sum_i b_i = 0$)
- with $n-1$ non redundant node constraints, we have just $n-1$ basic variables for a basic solution
  each basic variable x_{ij} represents the flow though arc ij: basic arcs
- basic arcs never form undirected cycles...
- hence they form a spanning tree

Max Flow Algorithms

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 $O(n^2 m)$
- Karzanov's push relabel $O(n^2 m)$

Matching Algorithms

Matching: $M \subseteq E$ of pairwise non adjacent edges

- bipartite graphs
- arbitrary graphs
- cardinality (max or perfect)
- weighted

Assignment problem $\equiv$ min weighted perfect bipartite matching $\equiv$ special case of min cost flow

bipartite cardinality

Theorem

The cardinality of a max matching in a bipartite graph equals the value of a maximum (s,t)-flow in N_{st}.

$\leadsto$ Dinic $O(\sqrt{n}m)$

Theorem (Optimality condition (Berge))

A matching M in a graph G is a maximum matching iff G contains no M-augmenting path.

$\leadsto$ augmenting path $O(\min(|U|,|V|),m)$

bipartite weighted
build up algorithm $O(n^3)$
bipartite weighted: Hungarian method $O(n^3)$

minimum weight perfect matching
Edmonds $O(n^3)$

Theorem (Hall's (marriage) theorem)

A bipartite graph $B = (X, Y, E)$ *has a matching covering* X *iff:*

$$|N(U)| \geq |U| \quad \forall U \subseteq X$$

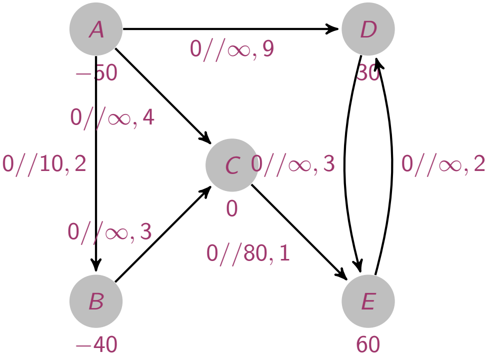
Theorem (König, Egeavary theorem)

Let $B = (X, Y, E)$ *be a bipartite graph. Let* M^* *be the maximum matching and* V^* *the minimum vertex cover:*

$$|M^*| = |V^*|$$

ILP in Excel

A company produces the same product at two different factories (A and B) and then the product must be shipped to two warehouses, where either factory can supply either warehouse. The distribution network is shown below where C is a distribution center. There are costs and bounds on the amount of product to ship through the connections



What problem is it? Transhippment problem (ie, min cost flow)

See file `mincost.xlsx`
If Solver is not there, click **Tools**, select **Add-Ins**, **Solver Add-in** and OK.
Then **Tools**, **Solve**

What if $\sum b(v) \neq 0$?

Summary

1. Duality in Network Flow Problems

2. More on Network Flows