Discrete Optimization 2010
Lecture 4
Minimum-Cost Flows

Marc Uetz
University of Twente

m.uetz@utwente.nl
Outline

1. Remarks on Max-Flow and Min-Cut
2. Flow Decomposition
3. Min-Cost Flows
Weak Duality Theorem

The value of any flow is at most equal to the capacity of any cut. So for any flow $x$ and any cut $[S, T]$,

$$v(x) \leq u(S, T).$$

Strong Duality Theorem

There is a flow $x$ and a cut $[S, T]$ such that

$$v(x) = u(S, T).$$
Figure 2
From Harris and Ross [1955]: Schematic diagram of the railway network of the Western Soviet Union and Eastern European countries, with a maximum flow of value 163,000 tons from Russia to Eastern Europe, and a cut of capacity 163,000 tons indicated as 'The bottleneck'.

The max-flow min-cut theorem
In the RAND Report of 19 November 1954, Ford and Fulkerson [1954] gave (next to defining the maximum flow problem and suggesting the simplex method for it) the max-flow min-cut theorem for undirected graphs, saying that the maximum flow value is equal to the minimum capacity of a cut separating source and terminal. Their proof is not constructive, but for planar graphs, with source and sink on the outer boundary, they give a polynomial-time, constructive method. In a report of 26 May 1955, Robacker [1955a] showed that the max-flow min-cut theorem can be derived also from the vertex-disjoint version of Menger’s theorem.

As for the directed case, Ford and Fulkerson [1955] observed that the max-flow min-cut theorem holds also for directed graphs. Dantzig and Fulkerson [1955] showed, by extending the results of Dantzig [1951a] on integer solutions for the transportation problem to the (declassified by Pentagon in 1999, on request of Lex Schrijver)
### Reminder: LP Duality (I)

**Primal LP (P)**

\[
\begin{align*}
\text{max} & \quad c^t \cdot x & \text{dual variables} \\
\text{s.t.} & \quad Ax \leq b & \alpha \\
& \quad x \geq 0
\end{align*}
\]

**Dual LP (D)**

\[
\begin{align*}
\text{min} & \quad b^t \cdot \alpha \\
\text{s.t.} & \quad A^t \alpha \geq c \\
& \quad \alpha \geq 0
\end{align*}
\]

**Theorem (LP Duality)**

For all feasible \( x \) and \( \alpha \), we have \( c^t \cdot x \leq b^t \cdot \alpha \), and “=” for optimal primal and dual solutions (if both are finite)
Reminder: LP Duality (II)

\[
\begin{align*}
\text{max} & \quad c^t \cdot x & & \text{dual variables} \\
\text{s.t.} & \quad Ax = b & & \pi \\
& \quad Bx \leq u & & \alpha \\
& \quad x \geq 0
\end{align*}
\]

\[
\begin{align*}
\text{min} & \quad (b^t, u^t) \cdot \begin{pmatrix} \pi \\ \alpha \end{pmatrix} \\
\text{s.t.} & \quad [A^t, B^t] \begin{pmatrix} \pi \\ \alpha \end{pmatrix} \geq c \\
& \quad \alpha \geq 0
\end{align*}
\]

Theorem (LP Duality)

For all feasible \( x \) and \( \begin{pmatrix} \pi \\ \alpha \end{pmatrix} \), we have \( c^t \cdot x \leq (b^t, u^t) \cdot \begin{pmatrix} \pi \\ \alpha \end{pmatrix} \), and “=” for optimal primal and dual solutions (if both are finite)
Complementary Slackness

Consider primal and dual solutions \(x\) and \(\begin{pmatrix} \pi \\ \alpha \end{pmatrix}\), if

- \(x_i > 0 \Rightarrow [A^t, B^t]_i \begin{pmatrix} \pi \\ \alpha \end{pmatrix} = c_i\)
  - primal \(\geq 0\) variable nonzero \(\Rightarrow\) dual constraint with =

- \(\alpha_j > 0 \Rightarrow [B]_j x = u_j\)
  - dual \(\geq 0\)-variable nonzero \(\Rightarrow\) primal constraint with =

then \(x\) and \(\begin{pmatrix} \pi \\ \alpha \end{pmatrix}\) are optimal solutions for the primal and dual \((P)\) and \((D)\), respectively, and \(c^t \cdot x = (b^t, u^t) \cdot \begin{pmatrix} \pi \\ \alpha \end{pmatrix}\).
Maximum Flow: Simplification

Question: is MinCut indeed the LP-dual of MaxFlow?

- introduce arc \((t, s)\) with capacity \(\infty\)
- then flow balance \(= 0\) at all nodes
- objective is maximize flow on arc \((t, s)\)
LP Formulation

\[ x_{ij} = \text{amount of flow through arc } (i, j) \]

\[
\begin{align*}
\text{max} & \quad x_{ts} \\
\text{s.t.} & \quad \sum_{j: (i, j) \in A} x_{ij} - \sum_{j: (j, i) \in A} x_{ji} = 0 \quad \forall \ i \in V \quad (1) \\
& \quad x_{ij} \leq u_{ij} \quad \forall \ (i, j) \in A \quad (2) \\
& \quad x \geq 0 \quad (3)
\end{align*}
\]

(1) is flow balance
(2) arc capacities
(3) non-negativity
# Node-Arc Incidence Matrix

$A = \text{node-arc incidence matrix}$

$x = \text{flow vector}$

then Flow Balance is just $Ax = 0$

<table>
<thead>
<tr>
<th></th>
<th>$(s,i)$</th>
<th>$(s,j)$</th>
<th>$(i,j)$</th>
<th>$(i,t)$</th>
<th>$(j,t)$</th>
<th>$(t,s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>$i$</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$j$</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$t$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

$\begin{pmatrix}
X_{si} \\
X_{sj} \\
X_{ij} \\
X_{it} \\
X_{jt} \\
X_{ts}
\end{pmatrix} = 0$
LP Formulation Max Flow

- $A = \text{node-arc incidence matrix}$
- $E = \text{identity matrix}$
- $u = \text{vector of arc capacities}$

$x = \text{flow vector}$

$$\begin{align*}
\max & \quad x_{ts} \\
\text{s.t.} & \quad Ax = 0 \\
& \quad Ex \leq u \\
& \quad x \geq 0
\end{align*}$$

$$\pi = (\pi_i)_{i \in V} \quad \alpha = (\alpha_{ij})_{(i,j) \in A}$$
An Example for the Primal-Dual Pair
Max-Flow and Min-Cut

Max-Flow Problem:
\[ V \xrightarrow{3} 1 \xleftarrow{2} 0 \xrightarrow{t} \] problem: minimize flow along arc (t,s)

The primal LP:
\[ \text{max } c^T x \]
\[ \text{st. } Ax = b \quad \text{flow balance} \]
\[ Ex \leq u \quad \text{capacities} \]
\[ x \geq 0 \]

where \( c = (0,0,1) \) = minimize flow along (t,s).

So this gives the following on our example:

\[
\begin{bmatrix}
    x_{sv} \\
    x_{st} \\
    x_{tv} \\
    x_{ts} \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    1 & 1 & 0 & -1 \\
    -1 & 0 & 1 & 0 \\
    0 & -1 & -1 & 1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    1 \\
    0 \\
    0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    3 \\
    2 \\
    1 \\
\end{bmatrix}
\]

Now remember that the dual is the following LP

\[
\text{min } (0,u) \cdot (\Pi, d) \\
\text{st. } [A^T, E^T] \begin{bmatrix} \Pi \\ \alpha \end{bmatrix} \geq c \\
\alpha > 0
\]

which becomes the following on our example:

\[
\begin{bmatrix}
    d_{sv} \\
    d_{st} \\
    d_{tv} \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    3 \\
    2 \\
    1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    \Pi_s \\
    \Pi_t \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    5 \times v \\
    1 -1 0 1 0 0 0 \\
    1 0 -1 0 1 0 0 \\
    0 1 -1 0 0 1 0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    d_{sv} \\
    d_{st} \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    \Pi_s \\
    \Pi_t \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    1 \\
    0 \\
    1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    0 \\
    1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    0 \\
    0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    1 \\
    1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    d_{sv} + \Pi_s - \Pi(v) \geq 0 \\
    d_{st} + \Pi(s) - \Pi(t) \geq 0 \\
    d_{tv} + \Pi(v) - \Pi(t) \geq 0 \\
    d_{ts} + \Pi(s) - \Pi(t) \geq 0 \\
\end{bmatrix}
\]

and setting \( \Pi(s) = 0 \) and \( d_{ts} = 0 \), then in

\[
\begin{bmatrix}
    d_{sv} > \Pi(t) \\
    d_{st} > \Pi(v) \\
    d_{tv} > \Pi(v) \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    \Pi(t) > 0 \\
\end{bmatrix}
\]
Dual LP Formulation

\( \pi_i \) = node variables, \( \alpha_{ij} \) = arc variables

\[
\begin{align*}
\text{min} \quad & \sum_{(i,j) \in A} \alpha_{ij} u_{ij} \\
\text{s.t.} \quad & \alpha_{ij} \geq \pi(j) - \pi(i) \\
& \alpha_{ts} \geq \pi(s) - \pi(t) + 1 \\
& \alpha \geq 0
\end{align*}
\]

Observations

- may assume w.l.o.g. \( \pi(s) = 0 \)
  - why? because if \((\pi, \alpha)\) is solution, so is \((\pi \pm \text{const}, \alpha)\)
- since \(u_{ts} = \infty\), need \(\alpha_{ts} = 0\), so \(\pi(t) \geq \pi(s) + 1 = 1\)
- hence: in dual LP, like to set all \(\alpha_{ij} = 0\), but due to the constraints, need to set \(\alpha_{ij} \geq 1\) on some \((s, t)\)-cut
Intuition: Dual of MaxFlow is the MinCut Problem

The cut \([S, T]\) is given by partition of nodes with \(\pi = 0\) and \(\pi = 1\)
Theorem

The primal and dual linear programming problems for maximum flow and minimum cut, respectively, have integer optimal solutions (if all $u_{ij}$ integer).

Proof: (Primal) Augmenting path algorithm solves the problem, and only augments by integer amounts.

(Dual) And, for any maximum flow, we can construct a cut, i.e., an integer solution of the dual of the same value.
Theorem

There are instances (with irrational capacities) where the Ford-Fulkerson augmenting path algorithm may fail to converge to the optimum.

An example (not treated in all detail in the lecture) is available via the link [http://dx.doi.org/10.1016/0304-3975(95)00022-O](http://dx.doi.org/10.1016/0304-3975(95)00022-O)

Note: The Edmonds-Karp algorithm works nevertheless, and computes the optimum after $O(nm)$ flow augmentations.
Outline

1. Remarks on Max-Flow and Min-Cut
2. Flow Decomposition
3. Min-Cost Flows
Flow Decomposition Theorem

Theorem

Given an network $G = (V, a)$ with $n$ nodes and $m$ arcs. Every flow $x$ in $G$ can be decomposed into flow on at most $n + m$ paths and at most $m$ cycles. (not unique)

decomposition possible in 2 path flows and 3 cycle flows
Proof: Flow Decomposition Theorem

**Excess node** if outflow $> \text{inflow}$, **deficit node** if outflow $< \text{inflow}$

Start with any $(s, t)$-flow (then $s =$ excess node, $t =$ deficit node)

- start at any excess node and go along flow carrying arc $(s, v)$
- by flow balance, $v$ has outgoing flow (or is deficit node)
- continue until
  - ends up at a deficit node (path flow $P$)
  - ends at previously visited node (cycle flow $C$)
- decrease flow along $P$ or $C$ by **maximum possible amount**

**Path flow $P$:** either some arc gets 0-flow, or one excess or deficit node has balance 0 $\Rightarrow \leq n+m$ path flows

**Cycle flow $C$:** an arc gets 0-flow $\Rightarrow \leq m$ cycle flows

**After $\leq n + m$ iterations, have overall 0-flow**
Outline

1. Remarks on Max-Flow and Min-Cut
2. Flow Decomposition
3. Min-Cost Flows
Tolstoi: Minimum Cost Transportation Problem, 1930

After 10 steps, when the transports from all 10 factories have been set, Tolstoi ‘verifies’ the solution by considering a number of cycles in the network, and he concludes that his solution is optimum:

Thus, by use of successive applications of the method of differences, followed by a verification of the results by the circle dependency, we managed to compose the transportation plan which results in the minimum total kilometrage.

The objective value of Tolstoi’s solution is 395,052 kiloton-kilometers. Solving the problem with modern linear programming tools (CPLEX) shows that Tolstoi’s solution indeed is optimum. But it is unclear how sure Tolstoi could have been about his claim that his solution is optimum. Geographical insight probably has helped him in growing convinced of the optimality of his solution. On the other hand, it can be checked that there exist feasible solutions that have none of the negative-cost cycles considered by Tolstoi in their residual graph, but that are yet not optimum.

Later, Tolstoi [1939] described similar results in an article entitled Methods of removing irrational transportations in planning in the September 1939 issue of Sotsialisticheski Transport. The methods were also explained in the book Planning Goods Transportation by Pariiskaya, Tolstoi, and Mots [1947].

According to Kantorovich [1987], there were some attempts to introduce Tolstoi’s work by the appropriate department of the People’s Commissariat of Transport.

Kantorovich 1939

Apparently unaware (by that time) of the work of Tolstoi, L.V. Kantorovich studied a general class of problems, that includes the transportation problem. The transportation problem formed the big motivation for studying linear programming. In his memoirs, Kantorovich [1987] wrote how questions from practice motivated him to formulate these problems:
Minimum Cost Flow Problem

arc capacities $u_{ij}$ and per-unit arc costs $c_{ij}$

$$\begin{align*}
\text{min} & \quad \sum_{(i,j) \in A} c_{ij} x_{ij} \\
\text{s.t.} & \quad \sum_{j : (s,j) \in A} x_{sj} - \sum_{j : (j,s) \in A} x_{js} = \nu \quad \text{send $\nu$ units} \\
& \quad \sum_{j : (i,j) \in A} x_{ij} - \sum_{j : (j,i) \in A} x_{ji} = 0 \quad \forall \ i \neq s, t \\
& \quad 0 \leq x_{ij} \leq u_{ij}
\end{align*}$$

numbers at arcs: $(c_a, u_a)$
Minimum Cost Flow Problem

Or, more generally...

- $A =$ node arc incidence matrix
- $c =$ vector of arc costs
- $u =$ vector of arc capacities
- $b =$ vector of node balances, with $\sum_{i \in V} b_i = 0$

\[
\begin{align*}
\text{min} & \quad c \cdot x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \leq u \\
& \quad x \geq 0
\end{align*}
\]

Previous special case: $b_s = \nu$, $b_t = -\nu$, $b_i = 0$ otherwise
Min-Cost Flow: Assumptions & Remarks

- Arc costs $c$ and capacities $u$ are integers, and $u \geq 0$
- Can restrict to $(s, t)$-flow problem of value $v$
  (any general problem can be transformed into this version)
- There exists a feasible solution
  (can be checked beforehand by Max-Flow algorithm)
- There exists a path between any two nodes
  (otherwise, add dummy arcs with high costs)
The Residual Graph

For given flow $x$, have

- forward arcs: possible flow increase, capacity $u_{ij} - x_{ij}$, cost $c_{ij}$
- backward arcs: possible flow decrease, capacity $x_{ij}$, cost $-c_{ij}$

(costs, capacities)
Theorem (Negative Cost Cycle Optimality Condition)

A flow $x$ (of value $v$) is a minimum cost flow if and only if the residual graph $G(x)$ has no negative cost directed cycle.
Negative Cost Cycle Condition

Necessity

Obvious!

Sufficiency

Given $\bar{x}$, \( \exists \) negative cost cycle in $G(\bar{x})$, consider feasible flow $x'$

- show that $x' - \bar{x}$ is feasible in $G(\bar{x})$
  - non-trivial, but straightforward, just check constraints

- $x' - \bar{x}$ has flow balance 0 at all nodes
  - both have flow value $\nu$

- hence, $x' - \bar{x}$ is a sum of cycle flows (flow decomposition)

- each has nonnegative costs, by our assumption

- thus, $cx' = c(x' - x + \bar{x}) = c(x' - x) + cx \geq cx$
Algorithm 1: Cycle Canceling

**input**: network \((G, u)\) capacities \(u \geq 0\), costs \(c\), flow value \(v\), and nodes \(s, t \in V\)

**output**: \(x = \) minimum-cost \((s, t)\)-flow with value \(v\)

compute \((s, t)\)-flow \(x\) with value \(v\) [e.g., augmenting paths];

**while** \((\exists \) negative cost dicycle \(C\) in \(G(x)\)) **do**

\[
x_C = \text{maximum flow along } C;
\]
reduce cost by updating flow, \(x = x + x_C\);
update residual capacities, i.e., re-compute \(G(x)\);

- Termination if \(\not\exists\) negative cost cycle with unbounded capacity: because each time the cost decreases by \(\geq 1\)
- Correctness follows from negative cycle condition
- Computation time not polynomial in general (Exercise)
Theorem (Reduced Cost Optimality Condition)

A flow $x$ (of value $v$) is a minimum cost flow if and only if for some set of node labels $\pi(i), i \in V$, the reduced cost optimality condition holds in the residual graph $G(x)$:

$$c_{ij}^\pi := c_{ij} - \pi(i) + \pi(j) \geq 0 \quad \forall (i, j) \in G(x)$$

Compare to reduced cost optimality condition for LP (Simplex): “No column with negative reduced costs”
Intuition about Reduced Cost Condition

\[ -\pi(i) \leq -\pi(j) + c_{ij} \]

\(-\pi(i)\) = min. cost of getting 1 unit flow into \(i\) in \(G(x)\)

then the \(-\pi\)'s should better fulfill \(\Delta\)-inequality,

\[ c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j) \geq 0 \]

Such \(-\pi\)'s (called a node potential) can only exist in \(G(x)\) if \(x\) is a min-cost flow, for otherwise there is a negative cost directed cycle.

Note: min-cost flow would never use an arc with \(c_{ij}^\pi > 0\) (why?)
Reduced Cost Optimality Condition

Necessity

Consider min-cost flow $x$, know $G(x)$ has no negative cost cycle

- Compute cheapest $(s, i)$ path lengths $=: d(i)$ in $G(x)$ [exist]
- define $\pi(i) = -d(i)$, then $d(j) \leq d(i) + c_{ij}$ ($\Delta$-ineq.)
- $\iff c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j) \geq 0$

Sufficiency (via neg. cycle condition)

Let $x$ be flow and $\pi$ be node labels that fulfill the condition that $c_{ij}^\pi = c_{ij} - \pi(i) + \pi(j) \geq 0$ in $G(x)$. Consider any cycle $C$ in residual graph $G(x)$, costs are

$$\sum_{(i,j) \in C} c_{ij} = \sum_{(i,j) \in C} [c_{ij} - \pi(i) + \pi(j)] = \sum_{(i,j) \in C} c_{ij}^\pi \geq 0$$