Minimum-Cost Network Flow

In addition to a capacity, each edge has a real-valued cost per unit of flow.

A minimum-cost (maximum) flow is a maximum flow whose total cost (sum of edge flows times edge costs) is minimum.

Problem: find a minimum-cost flow in a given network.

\[ n = \text{# vertices} \]
\[ m = \text{# edges} \]
\[ U = \text{max capacity (if integers)} \]
\[ C = \text{max cost (if integers)} \]
Two Naive Approaches

(1) Repeat: augment along a cheapest path in the residual network.
Each augmentation takes a shortest path computation.

Shortest paths can be found using Dijkstra's algorithm if costs are kept non-negative using price transformation (primal-dual method of linear programming).

Time: $O(nU(m+n\log n))$ (not polynomial)

(2) repeat
   In network of zero-cost residual edges, find a maximum flow.
   Augment the flow and update the prices.
   (find all paths of a given cost at once)

Time: $O(nC(nm\log(n^2/m)))$ (not polynomial)
\( G = (V, E) \) symmetric directed graph

\((v, w) \in E \iff (w, v) \in E\)

\(|V| = n, \ |E| = m, \ m \geq n \geq 2\)

\( E(v) = \{w | (v, w) \in E\} \)

arc capacities \( u(v, w) : (v, w) \in E \)

arc costs \( c(v, w) : (v, w) \in E \)

cost function is antisymmetric: \( c(v, w) = -c(w, v) \)

Circulation \( f: E \rightarrow \mathbb{R} \)

\( f(v, w) = u(v, w) \quad \forall (v, w) \in E \) \hspace{1cm} \text{capacity constraint}

\( f(v, w) = -f(w, v) \quad \forall (v, w) \in E \) \hspace{1cm} \text{flow antisymmetry}

\( \sum_{w \in E(v)} f(v, w) = 0 \quad \forall v \in V \) \hspace{1cm} \text{flow conservation}

Cost of \( f \): \( \frac{1}{2} \sum_{(v, w) \in E} f(v, w) c(v, w) \)
Reformulated problem: Find a circulation
of minimum cost: add a return arc from root of infinite capacity and
large negative cost \((-\infty)\).

residual capacity \( u'_v (v, w) = u(v, w) - f(v, w) \) or \( f(w, v) \)

residual arc \((v, w): u'_v (v, w) > 0\)

residual cycle: a (simple) cycle of residual arcs

length of cycle = number of arcs, \( l (\Gamma) \)

cost of cycle = sum of arc costs = \( c(\Gamma) \)

mean cost of cycle = \( c(\Gamma) / l(\Gamma) \)

negative cycle: \( c(\Gamma) < 0 \)
Naive Approach

1. Repeat: augment along a cheapest path in the residual graph.
   Each augmentation takes a shortest path computation.
   (1) repeat: find a shortest path using Dijkstra's algorithm
       if costs are kept non-negative using "prices" to
       transform costs (primal-dual method).
       Time: $O(nm \log (n+\frac{m}{n}))$ (not polynomial)

2. repeat: In network of zero-cost edges find a maximum flow.
   A new set of prices and update the prices.
   Time: $O(nm \log (n+\frac{m}{n}))$ (not polynomial)
**Theorem** (Busacker and Saaty, 1965): A circulation f is minimum-cost iff there is no negative residual cycle.

**Algorithm** (Klein, 1967)

1. Find any circulation f (by a max flow computation)

2. While 3 negative cycle $\Gamma$, cancel $\Gamma$ by increasing $f$ on all arcs of $\Gamma$ by $\min \{ u_f (v, w) : (v, w) \in \Gamma \}$.

The choice of cycle can be exponential (or infinite)

**How to choose cycles for canceling to minimize iterations, running time?**

- minimum cost?
- minimum length?
- maximum length?
- maximum capacity?
- maximum cost decrease?
Primal Network Simplex Algorithm: Definitions

If \( (v, w) \) is residual, then \( (w, v) \) is also residual.

A set of residual edges forms a forest.

The algorithm maintains a basic circulation \( f \) and a basis tree \( T \) such that \( T \) contains every residual edge.

Any non-tree arc \((v, w)\) defines a basic cycle \( T_f(v, w) \) consisting of

\((v, w)\) and the path of tree arcs from \( v \) to \( w \).

(We reject each tree edge as consisting of a pair of tree arcs.)

An arc \((v, w)\) is pseudo-residual if it is residual or a tree arc.

A cycle is pseudo-residual if it consists only of pseudo-residual arcs.
Our Results

Minimum-mean cycle canceling: Always cancel a cycle of minimum mean cost.

Theorem: \# cancellations = \( O(nm^2 \log n) \). If costs are integers of maximum magnitude \( C \), \# cancellations = \( O(nm \log (nC)) \).

Time to find a minimum mean cycle = \( O(nm) \) (Karp, 1978)

A variant of this approach gives a "practical" algorithm with a running time of \( O(nm \log n \min \{ \log (nC), m \log n \}) \).
Price Function (Dual Variables)

\[ p: V \to \mathbb{R} \quad \text{reduced arc cost} \quad c_p(v,w) = c(v,w) + p(v) - p(w) \]

**Theorem (Ford and Fulkerson, 1962):** A circulation \( f \) is minimum-cost iff \( \exists p \) such that, \( \forall (v,w) \in E \),

\[ u_f(v,w) > 0 \text{ implies } c_p(v,w) > 0. \]

\( \varepsilon \)-optimality

For \( \varepsilon > 0 \), a circulation \( f \) is \( \varepsilon \)-optimal with respect to a price function \( p \) iff, \( \forall (v,w) \in E \),

\[ u_f(v,w) > 0 \text{ implies } c_p(v,w) \geq -\varepsilon. \]

\( \varepsilon(f) = \min \{ \varepsilon \geq 0 \text{ for which } f \text{ is } \varepsilon \text{-optimal with respect to some } p \} \)

**Theorem (Bertsekas, 1986):** If costs are integral and \( \varepsilon < \frac{1}{n} \), any \( \varepsilon \)-optimal circulation is minimum-cost.
Idea: minimum mean cycle canceling reduces $\epsilon(f)$ by a measurable amount, after enough cancellations.

Note: The cost of a cycle is the same as its reduced cost.

Key question: What is $\epsilon(f)$?

Let $\mu(f)$ be the mean cost of a minimum mean residual cycle with respect to circulation $f$.

Theorem: $\epsilon(f) = \max \{0, -\mu(f)\}$.

Proof: Use properties of shortest paths, e.g. shortest paths exist iff there are no negative cycles.
Analysis of Minimum Mean Cycle Canceling

**Lemma:** Canceling a minimum mean cycle cannot increase $\varepsilon(f)$.

**Lemma:** After $m$ cancellations, $\varepsilon(f)$ has decreased by a factor of $(n-1)/n$ or better.

**Theorem:** $\#\text{cancellations} = O(nm \log (nc))$.

**Lemma:** If $f$ and $f'$ are both $\varepsilon$-optimal and $|c_p(v,w)| > 2n\varepsilon$, where $f$ is $\varepsilon$-optimal with respect to $p$, then $f(v,w) = f'(v,w)$.

**Theorem:** $\#\text{cancellations} = O(nm^2 \log n)$. 
A "Practical" Variant

Maintain a price function $p$ along with a circulation $f$.

Call an arc $(v,w)$ eligible if $u_f(v,w) > 0$ and $c_p(v,w) < 0$.

Let $\epsilon(f,p) = \min \{ c_p(v,w) \mid u_f(v,w) > 0 \} \cup \{ 0 \}$.

Algorithm

0. Let $f$ be any circulation and let $p = 0$.  

1. Repeat until $\epsilon(f,p) < 1/n$:

   a. Find an eligible cycle and cancel it.

   b. If the subgraph of eligible arcs is acyclic, modify $p$ to reduce $\epsilon(f,p)$ by a factor of at least $(n-1)/n$. 
Analysis

There are at most \( n \) iterations of 1a between iterations of 1b.

All iterations of 1a between two iterations of 1b take a total of \( O(m \log n) \) time using dynamic trees.

One iteration of 1b takes \( O(m) \) time.

\( O(n) \) iterations of 1b reduce \( \epsilon(f,p) \) by a constant factor.

\[
\therefore O(nm \log n \log(nC)) \text{ total time.}
\]

If every \( n \text{th} \) iteration of 1b reduces \( \epsilon(f,p) \) as much as possible, then the amortized time per iteration of 1b is still \( O(m) \) (every \( n \text{th} \) takes \( O(nm) \)).

\[
\therefore O(nm^2/(\log n)^2) \text{ total time.}
\]
<table>
<thead>
<tr>
<th>Date</th>
<th>Discoverer</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1972</td>
<td>Edmonds and Karp</td>
<td>$O(n \log^2 n)$</td>
</tr>
<tr>
<td>1980</td>
<td>Reif</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>1980</td>
<td>Rei</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>1984</td>
<td>Tardos</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>1984</td>
<td>Orin</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>1985</td>
<td>Fujishige</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>1985</td>
<td>Bland and Jansen</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>1986</td>
<td>Galil and Tardos</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>1987</td>
<td>Goldberg and Tarjan</td>
<td>$O(n \log n)$</td>
</tr>
</tbody>
</table>