Minimum Cost Flow Problems I

AU4606: Network Optimization

AI4702: Network Intelligence and Optimization

Xiaoming Duan
Department of Automation
Shanghai Jiao Tong University

October 23, 2023

Last few lectures

- Maximum flow problems: important concepts
  - Residual graphs
  - s-t cut
  - Augmenting paths
- Maximum flow problems: algorithms
  - Generic augmenting path algorithms $O(m^2 U)$
  - Most improving augmenting path algorithms $O(m \log(mU)(m \log n))$
  - Capacity scaling algorithms $O(m^2 \log U)$
  - Shortest path augmenting path algorithms $O(m^2 n)$
  - Push-relabel algorithms $O(mn^2)$

Today

1. Minimum cost flow problems: formulation
   - What is a minimum cost flow problem?
   - Assumptions and concepts

2. Optimality condition and cycle-canceling algorithms

3. Minimum mean cycle canceling algorithms

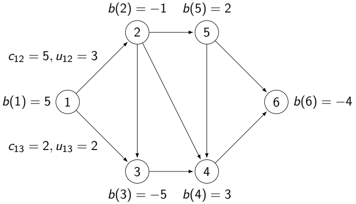
Today

1. Minimum cost flow problems: formulation
   - What is a minimum cost flow problem?
   - Assumptions and concepts

2. Optimality condition and cycle-canceling algorithms

3. Minimum mean cycle canceling algorithms

What is a minimum cost flow problem?



- A directed graph $G = (N, A)$
- Each arc $(i, j) \in A$ has a cost c_{ij} and a capacity constraint u_{ij}
- Each node $i \in N$ has a supply/demand $b(i)$, $\sum_i b(i) = 0$

Define a vector of flow variables f_{ij} over arcs (i, j) such that

1. They are nonnegative
2. They are upper bounded by the flow capacity
3. They satisfy the flow balance equation at all nodes

Minimize the total cost of flow $\sum_{(i,j)\in A} c_{ij} f_{ij}$

What is a minimum cost flow problem?

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in A} c_{ij} f_{ij} \\
\text{subject to} \quad & \sum_{j:(i,j)\in A} f_{ij} - \sum_{j:(j,i)\in A} f_{ji} = b(i) \\
& 0 \le f_{ij} \le u_{ij} \quad \text{for each } (i,j) \in A.
\end{aligned}$$

- A vector $f = \{f_{ij}\}$ satisfying constraints is called a (feasible) *flow*

Minimum cost flow problems: assumptions and concepts

- The graph is directed
- All data (cost, supply/demand, and capacity) are integral
- The supply/demand satisfy $\sum_i b(i) = 0$ and a feasible solution exists
- All arc costs are nonnegative

Minimum cost flow problems: assumptions and concepts

- The graph is directed
- All data (cost, supply/demand, and capacity) are integral
- The supply/demand satisfy $\sum_i b(i) = 0$ and a feasible solution exists
- All arc costs are nonnegative

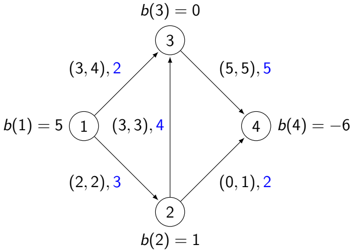
The residual graph $G(f) = (N', A')$ with respect to a feasible flow f in $G = (N, A)$ is defined as

- $N' = N$
- For each arc $(i, j) \in A$
  1. $(i, j) \in A'$ and it has residual capacity $r_{ij} = u_{ij} - f_{ij}$ and cost c_{ij} if $u_{ij} - f_{ij} > 0$
  2. $(j, i) \in A'$ and it has residual capacity $r_{ji} = f_{ij}$ and cost $-c_{ij}$ if $f_{ij} > 0$

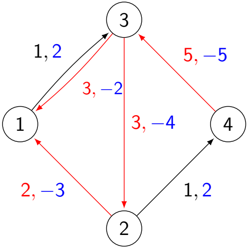
Residual graphs: an example

The residual graph $G(f) = (N', A')$ with respect to a feasible flow f in $G = (N, A)$ is defined as

- $N' = N$
- For each arc $(i, j) \in A$
    1. $(i, j) \in A'$ and it has residual capacity $r_{ij} = u_{ij} - f_{ij}$ and cost c_{ij} if $u_{ij} - f_{ij} > 0$
    2. $(j, i) \in A'$ and it has residual capacity $r_{ji} = f_{ij}$ and cost $-cij$ if $f_{ij} > 0$



(a) Graph G with flow f



(b) Residual graph $G(f)$

Today

1. Minimum cost flow problems: formulation
   - What is a minimum cost flow problem?
   - Assumptions and concepts

2. Optimality condition and cycle-canceling algorithms

3. Minimum mean cycle canceling algorithms

Minimum cost flow problems: optimality condition

Negative cycle optimality condition

A feasible solution f is an optimal solution of the minimum cost flow problem if and only if the residual network $G(f)$ contains no negative cost directed cycle.

This suggests a very natural algorithm!

Minimum cost flow problems: optimality condition

Negative cycle optimality condition

A feasible solution f is an optimal solution of the minimum cost flow problem if and only if the residual network $G(f)$ contains no negative cost directed cycle.

This suggests a very natural algorithm!

Algorithm Cycle-canceling algorithm

1: Establish a feasible flow f in the graph
2: **while** $G(f)$ contains a negative cost cycle **do**
3: Identify a negative cost cycle W
4: $\delta(W) = \min\{r_{ij}, (i,j) \in W\}$
5: Augment $\delta(W)$ units of flow along W and update $G(f)$
6: **end while**

Minimum cost flow problems: optimality condition

Negative cycle optimality condition

A feasible solution f is an optimal solution of the minimum cost flow problem if and only if the residual network $G(f)$ contains no negative cost directed cycle.

This suggests a very natural algorithm!

Algorithm Cycle-canceling algorithm

```
Establish a feasible flow f in the graph
while G(f) contains a negative cost cycle do
    Identify a negative cost cycle W
    δ(W) = min{r_ij, (i, j) ∈ W}
    Augment δ(W) units of flow along W and update G(f)
end while
```

Two questions remain

1. How to identify a negative cost cycle?

Minimum cost flow problems: optimality condition

Negative cycle optimality condition

A feasible solution f is an optimal solution of the minimum cost flow problem if and only if the residual network $G(f)$ contains no negative cost directed cycle.

This suggests a very natural algorithm!

Algorithm Cycle-canceling algorithm

```
1: Establish a feasible flow f in the graph
2: while G(f) contains a negative cost cycle do
3:     Identify a negative cost cycle W
4:     δ(W) = min{r_ij, (i,j) ∈ W}
5:     Augment δ(W) units of flow along W and update G(f)
6: end while
```

Two questions remain

1. How to identify a negative cost cycle? (Bellman-Ford $O(mn)$)

Minimum cost flow problems: optimality condition

Negative cycle optimality condition

A feasible solution f is an optimal solution of the minimum cost flow problem if and only if the residual network $G(f)$ contains no negative cost directed cycle.

This suggests a very natural algorithm!

Algorithm Cycle-canceling algorithm

```
1: Establish a feasible flow f in the graph
2: while G(f) contains a negative cost cycle do
3:     Identify a negative cost cycle W
4:     δ(W) = min{r_ij, (i,j) ∈ W}
5:     Augment δ(W) units of flow along W and update G(f)
6: end while
```

Two questions remain

1. How to identify a negative cost cycle? (Bellman-Ford $O(mn)$)
2. How to establish a feasible flow?

Minimum cost flow problems: establish a feasible flow

How to determine whether the problem is feasible?

$$\sum_{j:(i,j)\in A} f_{ij} - \sum_{j:(j,i)\in A} f_{ji} = b(i)$$

$$0 \leq f_{ij} \leq u_{ij} \quad \text{for each } (i,j) \in A.$$

Build a flow network $G' = (N', A')$ based on $G = (N, A)$ as follows

- Introduce a source node s and a sink node t, and $N' = N \cup \{s, t\}$
- For each node i with $b(i) > 0$, add an arc (s, i) with capacity $b(i)$
- For each node i with $b(i) < 0$, add an arc (i, t) with capacity $-b(i)$

Feasible flows

The minimum cost flow problem is feasible if and only if the maximum flow problem saturates all source arcs.

Cycle-canceling algorithm: analysis

Integrality property

If all arc capacities and supplies/demands of nodes are integer, the minimum cost flow problem always has an integer minimum cost flow.

Complexity

The algorithm terminates within $O(mCU)$ iterations and runs in $O(m^2nCU)$ time.

This is again an exponential time algorithm!

Cycle-canceling algorithm: analysis

Integrality property

If all arc capacities and supplies/demands of nodes are integer, the minimum cost flow problem always has an integer minimum cost flow.

Complexity

The algorithm terminates within $O(mCU)$ iterations and runs in $O(m^2nCU)$ time.

This is again an exponential time algorithm!

Two similar improvements as in augmenting path algorithms

- Augmenting along a cycle with maximum $-\delta(W)\sum_{(i,j)\in W} c_{ij}$
  - Identify such cycles is a hard problem
- Augmenting flow along a negative cycle with minimum mean cost
  - Mean cost of a cycle W: $\frac{\sum_{(i,j)\in W} c_{ij}}{|W|}$

Today

1. Minimum cost flow problems: formulation
   - What is a minimum cost flow problem?
   - Assumptions and concepts
2. Optimality condition and cycle-canceling algorithms
3. Minimum mean cycle canceling algorithms

Algorithm Minimum mean cycle canceling algorithm

1: Establish a feasible flow f in the graph
2: **while** $G(f)$ contains a negative cost cycle **do**
3: Identify a negative cost cycle W that has the minimum $\frac{\sum_{(i,j)\in W} c_{ij}}{|W|}$
4: $\delta(W) = \min\{r_{ij}, (i,j) \in W\}$
5: Augment $\delta(W)$ units of flow along W and update $G(f)$
6: **end while**

Minimum mean cycle canceling algorithm: procedure

Algorithm Minimum mean cycle canceling algorithm

Establish a feasible flow f in the graph
while $G(f)$ contains a negative cost cycle **do**
    Identify a negative cost cycle W that has the minimum $\frac{\sum_{(i,j)\in W} c_{ij}}{|W|}$
    $\delta(W) = \min\{r_{ij}, (i,j) \in W\}$
    Augment $\delta(W)$ units of flow along W and update $G(f)$
end while

- How to find a cycle that has minimum mean cost?
- What is the complexity of this algorithm?

Minimum mean cost cycle: computing minimum mean cost

- Let s be a node that has directed paths to all other nodes
- Let $d^k(j)$ be length of shortest directed walk from s to j containing exactly k arcs
- These distances $d^k(j)$ can be computed in $O(mn)$ by

$$d^k(j) = \min_{(i,j)\in A} \{d^{k-1}(i) + c_{ij}\}$$

Minimum mean cost

Let μ^* be the cost of a minimum mean cost cycle, i.e.,

$$\mu^* = \min_W \frac{\sum_{(i,j)\in W} c_{ij}}{|W|}$$

then

$$\mu^* = \min_{j\in N} \max_{0\leq k\leq n-1} \frac{d^n(j) - d^k(j)}{n-k}$$

With μ^*, we do the following

- Subtract μ^* from the cost of each arc, i.e., $c'_{ij} = c_{ij} - \mu^*$
- Compute distances $d'(j)$ in modified graph with c'_{ij}
- Let $c^d_{ij} = c'_{ij} + d'(i) - d'(j) \geq 0$
- Claim: cycles consisting of zero cost arcs have minimum mean costs

In summary, minimum mean cost cycle can be identified in $O(mn)$

Minimum mean cycle canceling algorithm: analysis

Algorithm Minimum mean cycle canceling algorithm

Establish a feasible flow f in the graph
while $G(f)$ contains a negative cost cycle **do**
    Identify a negative cost cycle W that has the minimum $\frac{\sum_{(i,j)\in W} c_{ij}}{|W|}$
    $\delta(W) = \min\{r_{ij}, (i,j) \in W\}$
    Augment $\delta(W)$ units of flow along W and update $G(f)$
end while

- First establish that the algorithm is weakly polynomial
- Then show it is actually strongly polynomial

Concepts: node potentials, reduced cost, and ϵ-optimality

Node potentials

A node potential $p : N \to \mathbb{R}$ is an assignment of reals to nodes.

Reduced cost with respect to node potential

The reduced cost of an arc (i, j) w.r.t. potential p is $c_{ij}^p = c_{ij} + p(i) - p(j)$.

Node potentials

A node potential $p : N \to \mathbb{R}$ is an assignment of reals to nodes.

Reduced cost with respect to node potential

The reduced cost of an arc (i, j) w.r.t. potential p is $c_{ij}^p = c_{ij} + p(i) - p(j)$.

Negative cycle optimality condition

A feasible solution f is an optimal solution of the minimum cost flow problem if and only if the residual network $G(f)$ contains no negative cost directed cycle.

Concepts: node potentials, reduced cost, and ϵ-optimality

Node potentials

A node potential $p : N \to \mathbb{R}$ is an assignment of reals to nodes.

Reduced cost with respect to node potential

The reduced cost of an arc (i, j) w.r.t. potential p is $c_{ij}^p = c_{ij} + p(i) - p(j)$.

Negative cycle optimality condition

A feasible solution f is an optimal solution of the minimum cost flow problem if and only if the residual network $G(f)$ contains no negative cost directed cycle.

Optimality conditions

Let f be a feasible flow, then the following statements are equivalent

1. f is a minimum cost flow
2. Residual network $G(f)$ contains no negative cost cycle
3. There exists a potential p such that $c_{ij}^p \geq 0$ for $(i, j) \in G(f)$

Concepts: node potentials, reduced cost, and ϵ-optimality

Optimality conditions

Let f be a feasible flow, then the following statements are equivalent

1. f is a minimum cost flow
2. Residual network $G(f)$ contains no negative cost cycle
3. There exists a potential p such that $c_{ij}^p \geq 0$ for $(i, j) \in G(f)$

ϵ-optimality

A flow f is ϵ-optimal for $\epsilon > 0$ and some node potentials p if the reduced arc cost of each arc is at least $-\epsilon$.

Why is ϵ-optimality useful?

Concepts: node potentials, reduced cost, and ϵ-optimality

Optimality conditions

Let f be a feasible flow, then the following statements are equivalent

1. f is a minimum cost flow
2. Residual network $G(f)$ contains no negative cost cycle
3. There exists a potential p such that $c_{ij}^p \geq 0$ for $(i, j) \in G(f)$

ϵ-optimality

A flow f is ϵ-optimal for $\epsilon > 0$ and some node potentials p if the reduced arc cost of each arc is at least $-\epsilon$.

Why is ϵ-optimality useful?

ϵ-optimality and optimality

Suppose the costs c_{ij} are integers. If a flow f is ϵ-optimal for some $\epsilon < \frac{1}{n}$, then f is a minimum-cost flow.

Analysis: weak polynomiality

Let $\epsilon(f)$ be the minimum ϵ for which f is ϵ-optimal.

Nonoptimal flows

If f is not a minimum-cost flow, then

$$\epsilon(f) = -\min_{W \in G(f)} \frac{\sum_{(i,j) \in W} c_{ij}}{|W|} = -\mu^*.$$

Thus, $\epsilon(f)$ can be computed in $O(mn)$.

Analysis: weak polynomiality

Let $\epsilon(f)$ be the minimum ϵ for which f is ϵ-optimal.

Nonoptimal flows

If f is not a minimum-cost flow, then

$$\epsilon(f) = -\min_{W \in G(f)} \frac{\sum_{(i,j) \in W} c_{ij}}{|W|} = -\mu^*.$$

Thus, $\epsilon(f)$ can be computed in $O(mn)$.

Continuous improvement

Let f be a flow and $f^{(k)}$ be the flow after k cycle cancelations, then

- (Monotonicity) $\epsilon(f^{(1)}) \leq \epsilon(f)$
- (Convergence) $\epsilon(f^{(m)}) \leq (1 - \frac{1}{n})\epsilon(f)$

Weak polynomiality

The minimum-mean cycle canceling algorithm takes $O(mn \log(nC))$ iterations and $O(m^2 n^2 \log(nC))$ overall time.

Analysis: strong polynomiality

ϵ-fixed arcs

An arc $(i, j) \in A$ is ϵ-fixed if the flow f_{ij} on (i, j) is the same for all ϵ'-optimal flows whenever $\epsilon' \leq \epsilon$.

Due to monotonicity of $\epsilon(f)$, the flow on ϵ-fixed arc does not change!

Analysis: strong polynomiality

ϵ-fixed arcs

An arc $(i, j) \in A$ is ϵ-fixed if the flow f_{ij} on (i, j) is the same for all ϵ'-optimal flows whenever $\epsilon' \leq \epsilon$.

Due to monotonicity of $\epsilon(f)$, the flow on ϵ-fixed arc does not change!

Conditions for ϵ-fixed arcs

Let f be an ϵ-optimal flow with respect to node potential p. If $|c_{ij}^p| \geq 2n\epsilon$, then arc (i, j) is ϵ-fixed.

If arcs become ϵ-fixed every few iterations, then we can bound iteration $\#$.

Analysis: strong polynomiality

ϵ-fixed arcs

An arc $(i,j) \in A$ is ϵ-fixed if the flow f_{ij} on (i,j) is the same for all ϵ'-optimal flows whenever $\epsilon' \leq \epsilon$.

Due to monotonicity of $\epsilon(f)$, the flow on ϵ-fixed arc does not change!

Conditions for ϵ-fixed arcs

Let f be an ϵ-optimal flow with respect to node potential p. If $|c_{ij}^p| \geq 2n\epsilon$, then arc (i,j) is ϵ-fixed.

If arcs become ϵ-fixed every few iterations, then we can bound iteration $\#$.

Strong polynomiality

The minimum-mean cycle canceling algorithm takes $O(nm^2 \log n)$ iterations and $O(n^2 m^3 \log n)$ overall time.

Upcoming

Week 1-8 (AU4606 & AI4702):

- Introduction (1 lecture)
- Preparations (3 lectures)
  - basics of graph theory
  - algorithm complexity and data structure
  - graph search algorithm
- Shortest path problems (3 lectures)
- Maximum flow problems (5 lectures)
- Minimum cost flow problems (this and next few lectures)
- Introduction to multi-agent systems (1 lecture)
- Introduction to cloud networks (1 lecture)

Week 9-16 (AU4606):

- Simplex and network simplex methods (2 lectures)
- Global minimum cut problems (3 lectures)
- Minimum spanning tree problems (3 lectures)