Flow Variants

Lecture 16
March 30, 2021

Most slides are courtesy Prof. Chekuri
Generalizations of Flow

We have seen $s-t$ flow. Flow problems admit several generalizations and variations.

- Demands and Supplies (we have already seen them)
- Circulations
- Lower bounds in addition to upper bounds
- Minimum cost flows and circulations
- Flows with losses
- Flows with time delays
- Multi-commodity flows
- . . .

Many applications, connections, algorithms.
Part I

Circulations
Circulations

**Definition**

**Circulation** in a network $G = (V, E)$, is a function $f : E \rightarrow \mathbb{R}^{\geq 0}$ s.t.

1. **Conservation Constraint:** For each vertex $v$:

   $$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

2. **Capacity Constraint:** For each edge $e$, $f(e) \leq c(e)$

No source or sink. $f(e) = 0$ for all $e$ is a valid circulation.
Circulation with lower bounds

Circulations are useful mainly in conjunction with lower bounds. Given a network $G = (V, E)$ with capacities $c : E \rightarrow \mathbb{R}^{\geq 0}$ and lower bounds $\ell : E \rightarrow \mathbb{R}^{\geq 0}$.

**Definition**

**Circulation** in a network $G = (V, E)$, is function $f : E \rightarrow \mathbb{R}^{\geq 0}$ s.t.

1. **Conservation Constraint**: For each vertex $v$:

   $$\sum_{e \text{ into } v} f(e) = \sum_{e \text{ out of } v} f(e)$$

2. **Capacity Constraint**: For each edge $e$, $f(e) \leq c(e)$

3. **Lower bound Constraint**: For each edge $e$, $f(e) \geq \ell(e)$
Circulation problem

Problem

<table>
<thead>
<tr>
<th>Input</th>
<th>A network $G$ with capacity $c$ and lower bound $\ell$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>Find a feasible circulation</td>
</tr>
</tbody>
</table>

Simply a feasibility problem.

Observation: As hard as the $s$-$t$ maxflow!
Reducing Max-flow to Circulation

Decision version of max-flow.

### Problem

**Input**  
A network $G$ with capacity $c$ and source $s$ and sink $t$ and number $F$.

**Goal**  
Is there an $s$-$t$ flow of value at least $v$ in $G$?
Reducing Max-flow to Circulation

Decision version of max-flow.

**Problem**

**Input** A network $G$ with capacity $c$ and source $s$ and sink $t$ and number $F$.

**Goal** Is there an $s$-$t$ flow of value at least $v$ in $G$?

Given $G,s,t$ create network $G'$ as follows:

1. set $\ell(e) = 0$ for each $e$ in $G$
2. add new edge $(t, s)$ with lower bound $v$ and upper bound $\infty$
Reducing Max-flow to Circulation

Decision version of max-flow.

**Problem**

<table>
<thead>
<tr>
<th>Input</th>
<th>A network $G$ with capacity $c$ and source $s$ and sink $t$ and number $F$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>Is there an $s$-$t$ flow of value at least $v$ in $G$?</td>
</tr>
</tbody>
</table>

Given $G, s, t$ create network $G'$ as follows:

1. set $\ell(e) = 0$ for each $e$ in $G$
2. add new edge $(t, s)$ with lower bound $v$ and upper bound $\infty$

**Claim**

There exists a flow of value $v$ from $s$ to $t$ in $G$ if and only if there exists a feasible circulation in $G'$. 
Reducing Circulation to Max-Flow

Circulation problem can be reduced to $s-t$ flow and hence they are polynomial-time equivalent. See Kleinberg-Tardos Chapter 7 for details of the reduction.
Reducing Circulation to Max-Flow

Circulation problem can be reduced to \( s-t \) flow and hence they are polynomial-time equivalent. See Kleinberg-Tardos Chapter 7 for details of the reduction.

Important properties of circulations:

- Reduction shows that one can find in \( O(mn) \) time a feasible circulation in a network with capacities and lower bounds.
- If edge capacities and lower bounds are integer valued then there is always a feasible integer-valued circulation.
- Hoffman’s circulation theorem is the equivalent of maxflow-mincut theorem.
- Circulation can be decomposed into at most \( m \) cycles in \( O(mn) \) time.
1. Design survey to find information about $n_1$ products from $n_2$ customers.

2. Can ask customer questions only about products purchased in the past.

3. Customer can only be asked about at most $c'_i$ products and at least $c_i$ products.

4. For each product need to ask at least $p_i$ consumers and at most $p'_i$ consumers.
Reduction to Circulation

1. include edge $(i, j)$ is customer $i$ has bought product $j$
2. Add edge $(t, s)$ with lower bound 0 and upper bound $\infty$.
   - Consumer $i$ is asked about product $j$ if the integral flow on edge $(i, j)$ is 1
Part II

Minimum Cost Flows
Minimum Cost Flows

1. **Input:** Given a flow network $G$ and also edge costs, $w(e)$ for edge $e$, and a flow requirement $F$.

2. **Goal:** Find a *minimum cost* flow of value $F$ from $s$ to $t$.

3. **Goal:** Find a *minimum cost* maximum $s$-$t$ flow.

Given flow $f : E \rightarrow R^+$, cost of flow $= \sum_{e \in E} w(e)f(e)$.

**Note:** costs can be negative. An optimum solution may need cycles.
Minimum Cost Flows

1. **Input:** Given a flow network $G$ and also edge costs, $w(e)$ for edge $e$, and a flow requirement $F$.

2. **Goal:** Find a *minimum cost* flow of value $F$ from $s$ to $t$.

3. **Goal:** Find a *minimum cost* maximum $s$-$t$ flow.

Given flow $f : E \rightarrow R^+$, cost of flow $= \sum_{e \in E} w(e)f(e)$.

**Note:** costs can be negative. An optimum solution may need cycles.

Much more general than the shortest path problem.
Minimum Cost Flow: Facts

1. The problem can be solved efficiently in polynomial time.
   - $O(nm \log C \log(nW))$ time algorithm where $C$ is the maximum edge capacity and $W$ is the maximum edge cost.

2. $O(m \log n(m + n \log n))$ time strongly polynomial time algorithm.

3. For integer capacities, there is always an optimum solution in which flow is integral.
Residual graph when there are costs:

**Definition**

For a network $G = (V, E)$ and flow $f$, the residual graph $G_{f,w} = (V', E')$ of $G$ with respect to $f$ and $w$ is

1. $V' = V$,

2. **Forward Edges**: For each edge $e \in E$ with $f(e) < c(e)$, we add $e \in E'$ with capacity $c(e) - f(e)$. Cost $w'(e) = w(e)$.

3. **Backward Edges**: For each edge $e = (u, v) \in E$ with $f(e) > 0$, we add $(v, u) \in E'$ with capacity $f(e)$. Cost $w'(e) = -w(e)$. 
**Question:** Suppose $f$ is a max $s$-$t$ flow in $G$. When is $f$ a min-cost a minimum cost max-flow?
Min-Cost Flow: Optimality Condition

**Question:** Suppose $f$ is a max $s$-$t$ flow in $G$. When is $f$ a min-cost a minimum cost max-flow?

If and only if there is no negative-cost cycle in $G_f$.

- If there is a negative cost cycle we can augment along the cycle and reduce the cost of $f$ (note that value of $f$ does not change).
- Suppose $f'$ is another maxflow of less cost. One can show that $f' - f$ is a circulation in $G_f$ (since both are maxflows) which means that $f' - f$ can be decomposed into cycles. Since $f'$ has less cost than $f$ there must be a negative cost cycle.
Min-Cost Flow: Cycle-canceling algorithm

Goal: Given $G$ with integer capacities, non-negative weights, find $s$-$t$ maxflow of with minimum cost.

Cycle-Canceling-Alg

Compute a maxflow $f$ in $G$ (ignoring costs)

$G_{f,w}$ is residual graph of $G$ with respect to $f$

while there is a negative weight cycle $C$ in $G_{f,w}$ do

let $C$ be a negative weight cycle in $G_{f,w}$

Augment one unit of flow along $C$ and update $f$

Construct new residual graph $G_{f,w}$

Output $f$

Like Ford-Fulkerson the run-time is pseudo-polynomial in costs. Can be implemented to run in $O(m^2 nCW)$ time where $C = \max_e c(e)$ and $W = \max_e |w(e)|$. 


Goal: Given \( G \) with integer capacities, \textbf{non-negative} weights, and integer \( k \), find \( s-t \) flow of value \( k \) with minimum cost.

**Successive-Shortest-Path-Alg**

for every edge \( e \), \( f(e) = 0 \)

\( G_{f,w} \) is residual graph of \( G \) with respect to \( f \)

\begin{verbatim}
while \( v(f) < k \) and \( G_{f,w} \) has a simple \( s-t \) path do
    let \( P \) be a \textit{shortest} \( s-t \) path in \( G_{f,w} \)
    Augment one unit of flow along \( P \) and update \( f \)
    Construct new residual graph \( G_{f,w} \).
\end{verbatim}

Algorithm gives optimum solution. Shows existence of integral optimum solution for integer capacities. Run time is \( O(mk \log m) \), and in the worst-case, \( O(mC \log m) \).
Can we find a maxflow of maximum profit?