Graphs and Network Flows
IE411

Lecture 12

Dr. Ted Ralphs

References for Today's Lecture

- Required reading
  - Sections 21.1–21.2
- References
  - AMO Chapter 6
  - CLRS Sections 26.1–26.2

Flows in Network

- For the remainder of the course, a *capacitated network*, or just simply a *network*, will refer to some combination of
  - a directed graph $G = (N, A)$,
  - a *cost* and a *capacity* associated with each arc, and
  - a *supply* or *demand* associated with each node.
- A *flow* in a given network is a set of nonnegative integer values that correspond to the flow of some commodity from tail to head.
- Generally, we will require that flows be *balanced*.
- We will define what balanced means in different contexts.

Circulations

- The simplest kind of flow is a *circulation*.
- This is a flow in which the flow into each node equals the flow out.
- The *flow decomposition theorem* tells us that any circulation is the sum of flows along at most $|A|$ arcs.
- This is easy to show, but has important consequences.

s-t Flows

- An *s-t flow* is a flow in which flow in equals flow out at all nodes except for special nodes s and t.
- The requirement for nodes s and t is simply that the flow out of s must be equal to the flow into t.
- Node s is referred to as the *source* and node t is referred to as the *sink*.
- The *value* of the flow is the flow out of s.
- An *s-t* flow can be easily converted into a circulation by adding an arc from t to s with infinite capacity.
- In this augmented network, the flow value is the flow on arc (t, s).

The Maximum Flow Problem

- Maximum Flow Problem: Given a capacitated network $G = (N, A)$ and two designated nodes s and t, find the s-t flow of maximum value.
- Two types of algorithms
  - Augmenting path algorithms
  - Preflow-push algorithms
- Correctness of algorithms relies on *Max-Flow Min-Cut Theorem*

Linear Programming Formulation

Given a network $G = (N, A)$ with a non-negative capacity u_{ij} associated with each arc $(i, j) \in A$ and two nodes s and t, find the maximum flow from s to t that satisfies the arc capacities.

$$
\begin{aligned}
\text{Maximize} \quad & v && && (1)\\
\text{subject to} \quad & \sum_{j:(s,j)\in A} x_{sj} - \sum_{j:(j,s)\in A} x_{js} &=\;& v && (2)\\
& \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} &=\;& 0 \quad \forall i \in N \setminus \{s, t\} && (3)\\
& \sum_{j:(t,j)\in A} x_{tj} - \sum_{j:(j,t)\in A} x_{jt} &=\;& -v && (4)\\
& x_{ij} &\leq\;& u_{ij} \quad \forall (i, j) \in A && (5)\\
& x_{ij} &\geq\;& 0 \quad \forall (i, j) \in A && (6)
\end{aligned}
$$

Assumptions

1. The network is directed.
2. All capacities are non-negative integers.
3. The network does not contain a directed path from node s to node t composed only of infinite capacity arcs.
4. Whenever arc (i, j) belongs to A, arc (j, i) also belongs to A.
5. The network does not contain parallel arcs.

Residual Network

- Suppose that an arc (i, j) with capacity u_{ij} carries x_{ij} units of flow.
- Then, we can send up to $u_{ij} - x_{ij}$ additional units of flow.
- We can also send up to x_{ij} units of flow backwards, canceling the existing flow and decreasing the flow cost.
- The *residual network* $G(x^0)$ is defined with respect to a given flow x^0 and consists of arcs with positive residual capacity.
- Note that if for some pair of nodes i and j, G already contains both (i, j) and (j, i), the residual network may contain parallel arcs with different residual capacities.

Residual Network Example

Definitions

- A *cut* is a partition of the node set N into two parts S and $\bar{S} = N \setminus S$.
- An $s-t$ *cut* is defined with respect to two distinguished nodes s and t and is a cut $[S, \bar{S}]$ such that $s \in S$ and $t \in \bar{S}$.
- A *forward arc* with respect to a cut is an arc (i, j) with $i \in S$ and $j \in \bar{S}$.
- A *backward arc* with respect to a cut is an arc (i, j) with $i \in \bar{S}$ and $j \in S$.

Definitions (cont)

- The *capacity of an $s-t$ cut* is the sum of the capacities of the forward arcs in the cut.
- A *minimum cut* is the $s-t$ cut whose capacity is minimum among all $s-t$ cuts.
- The *residual capacity* of an $s-t$ cut is the sum of the residual capacities of the forward arcs in a cut.

Weak Duality

Property 1. [6.1] *The value of any feasible flow is less than or equal to the capacity of any cut in the network.*

Proof: Let x be an arbitrary flow with value v. Let $[S, \bar{S}]$ be an arbitrary cut. Then we need to show that $v \leq u[S, \bar{S}]$.

Implications of Property 1

Suppose x^* is a feasible flow with value v^* and $[S, \bar{S}]$ is a cut with capacity v^*.

- Since $u[S, \bar{S}] = v^*$ is an upper bound on the maximum flow, then x^* must be a maximum flow.
- Since x^* is a feasible flow with value v^*, any cut must have a capacity of at least v^*. $[S, \bar{S}]$ has a capacity of v^*, so therefore $[S, \bar{S}]$ is a minimum cut.

Property 2. [6.2] *For any flow x of value v, the additional flow that can be sent from s to t is less than or equal to the residual capacity of any $s-t$ cut.*

Generic Augmenting Path Algorithm

- An *augmenting path* is a directed path from the source to the sink in the *residual* network.
- The *residual capacity* of an augmenting path is the minimum residual capacity of any arc in the path, which we denote by δ.
  - By definition, $\delta > 0$.
  - When the network contains an augmenting path, we can send additional flow from the source to the sink.

Generic Augmenting Path Algorithm

Input: A network $G = (N, A)$ and a vector of capacities $u \in \mathbb{Z}^A$
Output: x represents the maximum flow from node s to node t

$x \leftarrow 0$
while $G(x)$ contains a directed path from s to t **do**
  identify an augmenting path P from s to t
  $\delta \leftarrow \min\{r_{ij} : (i, j) \in P\}$
  augment the flow along P by δ units and update $G(x)$ accordingly.
end while