Section 4.3

The Dinic Algorithm and Layered Networks

LAYERED NETWORKS

A **layered network** is a network in which the vertices have been layered. The layers and their structure are determined by the present flow. Forward arcs $e = u \to v$ such that $f(e) < c(e)$ and backward arcs $e = v \to u$ such that $f(e) > 0$ are called **useful arcs**. We denote the useful arcs from layer L_i to layer L_{i+1} by U_{i+1}. We build layers on a breadth-first search using only useful arcs. As arcs become saturated, we have fewer and fewer useful arcs in relayering the network.

THE LAYERING ALGORITHM

Algorithm 4.3.1 The Layering Algorithm

Input: A network N and flow f.

Output: A sequence of layers of vertices $L_0, L_1, \ldots, L_d$ or the message that the present flow is maximum.

Method: A modified BFS using only useful arcs.

LAYERING ALGORITHM (CONCLUDED)

1. $L_0 \leftarrow \{s\}$ and $i \leftarrow 0$.
2. Set $T \leftarrow \{v \mid v \notin \bigcup_{j=0}^{i} L_j$ and there is $e = u \rightarrow v$ or $v \rightarrow u$ such that e is useful$\}$.
3. If $T = \emptyset$, say the present flow is maximum and halt.
4. If $t \in T$,
   then $k \leftarrow i + 1$ and $L_k \leftarrow \{t\}$ and halt with the layers $L_0, L_1, \ldots, L_k$;
   else $L_{i+1} \leftarrow T$, set $i \leftarrow i + 1$ and go to step 2.

COMMENTS ON LAYERING

- Consecutive layers are joined only by useful arcs.
- We seek a **maximal flow** f^* in the layered network.
- This means that a flow f^* such that for every path $s = v_0, e_1, v_1, \ldots, e_d, v_d = t$, where $v_i \in L_i$ and $e_i \in U_{i+1}$, there is at least one saturated arc e.
- That is, every feasible augmenting path with vertices in consecutive layers has an arc whose flow is at capacity.

PHASES

The process of finding a layered network, then finding the maximal flow on the layered network and improving the original flow is called a **phase**. We can bound the number of phases needed in order to find a maximal flow.

LENGTH OF A LAYERED NETWORK

- The **length** of a layered network is the index of the final layer.
- This is also a measure of the length of an augmenting path.
- We denote the length of a layered network obtained in the j^{th} phase by $len(j)$.
- We denote the a^{th} layer in the b^{th} phase by $L_a(b)$.

A BOUND ON THE NUMBER OF PHASES

Theorem 4.3.1: If phase $m + 1$ is not the final phase, then $len(m + 1) > len(m)$, and hence, the number of phases is at most $|V| - 1$.

STACKS

- A **stack** is a last in-first out information storage and retrieval device. (Think of a stack of trays in a cafeteria.)
- The act of placing X on the top of the stack ST will be denoted by ST <== X.
- The act of removing X from the top of the stack ST will be denoted by X <== ST.
- These are the only two operations allowed on the stack.

DINIC'S MAXIMAL FLOW ALGORITHM

Algorithm 4.3.2 Dinic's Maximal Flow Algorithm.

Input: A layered network N with $f(e) = 0$ and e marked unblocked.

Output: A maximal flow on N.

DINIC'S ALGORITHM (CONCLUDED)

1. Let $v \leftarrow s$ and empty the stack ST.
2. If all arcs to the next layer are blocked, then
   if $s = v$, then halt and note that the present flow is maximal.
   else e <== ST (say $e = uv$), mark e as "blocked," $v \leftarrow u$, and repeat step 2.
3. Chose an unblocked arc $e = vu$ with u in the next layer, ST <== e and let $v \leftarrow u$. If v does not equal t, then go to step 2.
4. The edges on ST form an augmenting path P. Find the minimum slack Δ on P. For every arc e on P, set $f(e) \leftarrow f(e) + \Delta$ and if $f(e) = c(e)$, mark e as "blocked." Go to step 1.