**0.1 A faster network flow algorithm.** Augmentation increases the value of the maximum flow by the bottleneck capacity of the selected path; so if we choose paths with large bottleneck capacity, we will be making a lot of progress. A natural idea is to select the path that has the largest bottleneck capacity. Having to find such paths can slow down each individual iteration by quite a bit. We will maintain a so-called *scaling parameter* $\Delta$, and we will look for paths that have bottleneck capacity of at least $\Delta$.

See the algorithm on page 353. Initialize $\Delta$ to be the largest power of 2 that is no larger than the maximum capacity out of $s$.

LEMMA 0.1. *If the capacities are integer-valued, then throughout the Scaling Max-Flow Algorithm the flow and the residual capacities remain integer-valued. This implies that when $\Delta = 1$, $G_f(\Delta)$ is the same as $G_f$, and hence when the algorithm terminates the flow, $f$ is of maximum value.*

LEMMA 0.2. *The number of iterations of the outer* `While` *loop is at most $1 + \lceil \log_2 C \rceil$.*

LEMMA 0.3. *During the $\Delta$-scaling phase, each augmentation increases the flow value by at least $\Delta$.*

LEMMA 0.4. *Let $f$ be the flow at the end of the $\Delta$-scaling phase. There is an $s$-$t$ cut $(A,\ B)$ in $G$ for which $c(A,\ B) \le v(f) + m\Delta$, where $m$ is the number of edges in the graph $G$. Consequently, the maximum flow in the network has value at most $v(f) + m\Delta$.*

*Proof.* Consider an edge $e = (u,\ v)$ in $G$ for which $u \in A$ and $v \in B$. $c_e < f(e) + \Delta$; otherwise, $v \in A$ as well. Also, for any edge $e' = (u',\ v')$ in $G$ for which $u' \in B$ and $v' \in A$, we have $f(e') < \Delta$; otherwise, we have a $s$-$u'$ path in $G_f(\Delta)$. So, all edges $e$ out of $A$ are almost saturated: $c_e < f(e) + \Delta$; and all edges into $A$ are almost empty: $f(e) < \Delta$.

$$
\begin{aligned}
v(f) &= \sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) \\
&\ge \sum_{e \text{ out of } A} (c_e - \Delta) - \sum_{e \text{ in to } A} \Delta \\
&= \sum_{e \text{ out of } A} c_e - \sum_{e \text{ out of } A} \Delta - \sum_{e \text{ in to } A} \Delta \\
&\ge c(A,\ B) - m\Delta.
\end{aligned}
$$

LEMMA 0.5. *The number of augmentations in a scaling phase is at most $2m$.*

THEOREM 0.1. *The Scaling Max-Flow Algorithm in a graph with $m$ edges and integer capacities finds a maximum flow in at most $2m(1 + \lceil \log_2 C \rceil)$ augmentations. It can be implemented to run in at most $O(m^2 \log_2 C)$ time.*

**0.2 The preflow-push maximum-flow algorithm.** The preflow-push algorithm increases the flow on an edge-by-edge basis. For each node $v$ other than the source $s$, we have $\sum_{e \text{ in to } v} f(e) \ge \sum_{e \text{ out of } v} f(e)$. The difference is called *excess* of the preflow at node $v$.

The algorithm is based on the physical intuition that flow naturally finds its way 'downhill'. The 'heights' for this intuition are labels $h(v)$ for each node $v$. We will push flow from nodes with higher labels to those with lower labels. $h(t) = 0$ and $h(s) = n$. For all edges $(v,\ w) \in E_f$ in the residual graph, we have $h(v) \le h(w) + 1$.

Lemma 0.6. *If $s$-$t$ preflow $f$ is compatible with a labeling $h$, then there is no $s$-$t$ path in the residual graph $G_f$.*

Lemma 0.7. *If $s$-$t$ flow $f$ is compatible with a labeling $h$, then $f$ is a flow of maximum value.*

See the algorithm on page 360. 'while there is a node $v \neq t$ with excess $e_f(v) > 0$, then if there is $w$ such that push$(f,\ h,\ v,\ w)$ can be applied (that is, $h(w) < h(v)$) then apply it. Else, relabel $v$ (that is, if for all $w$ with $(v,\ w) \in E_f$, we have $h(w) \geq h(v)$, then increase $h(v)$ by 1).

Lemma 0.8. *Throughout the algorithm; (1) the labels are nonnegative integers; (2) $f$ is a preflow, and if the capacities are integral, then the preflow $f$ is integral; (3) the preflow $f$ and labeling $h$ are compatible. If the algorithm returns a preflow $f$, the $f$ is a flow of maximum value.*

Lemma 0.9. *Let $f$ be a preflow. If the node $v$ has excess, then there is a path in $G_f$ from $v$ to the source $s$.*

*Proof.* Let $A$ denote all the nodes $w$ such that there is a path from $w$ to $s$ in the residual graph $G_f$. We show all nodes with excess are in $A$. No edges $e = (x,\ y)$ leaving $A$ can have positive flow, as $f(e) > 0$ gives rise to a reverse edge $(y,\ x)$ and then $y$ is in $A$ as well. The sum of excess of $B$ is 0, since each individual excess is non-negative, then all are 0. □

Lemma 0.10. *Throughout the algorithm, all nodes have $h(v) \leq 2n - 1$.*

*Proof.* $h(v) - h(s) \leq |P| \leq n - 1$. □

Lemma 0.11. *Throughput the algorithm, each node is relabeled at most $2n - 1$ times, and the total number of relabeling operations is less than $2n^2$.*

Lemma 0.12. *Throughput the algorithm, then number of saturating push operations is at most $2nm$.*

Lemma 0.13. *Throughout the algorithm, the number of non-saturating push operations is most $2n^2m$.*

## 0.3 Hall's theorem.

Theorem 0.2. *Consider a bipartite graph $G = (X,\ Y,\ E)$. Then $G$ either has a perfect matching or there is a subset $A \subset X$ such that $|N(A)| < |A|$.*

*Proof.* We show that if the value of the maximum flow is $< n$, then there exists a subset $A$ such that $|N(A)| < |A|$. That is, there exists a cut $(A',\ B')$ with capacity $< n$. At first one can modify $(A',\ B')$ so as to ensure that $N(A) \subseteq A'$. By moving $y$ from $B'$ to $A'$, we do not increase the capacity of the cut. Since all neighbors of $A$ belong to $A'$, we see that the only edges out of $A'$ are either edges that leave the source $s$ or that enter the sink $t$. Thus $c(A',\ B') = |X \bigcap B'| + |Y \bigcap A'|$. Note $|X \bigcap B'| = n - |A|$ and $|Y \bigcap A'| \geq |N(A)|$. $c(A',\ B') < n$ implies that $n - |A| + |N(A)| \leq |X \bigcap B'| + |Y \bigcap A'| = c(A',\ B') < n$. So, $|A| > |N(A)|$. □

## 0.4 Circulation with demands.

- Directed graph $G = (V,\ E)$.
- Edge capacities $c(e)$, $e \in E$.
- Node supply and demands $d(v)$, $v \in V$. demand if $d(v) > 0$; supply if $d(v) < 0$; transshipment if $d(v) = 0$.

A *circulation* is a function that satisfies: (1) for each $e \in E$: $0 \le f(e) \le c(e)$ (capacity); (2) for each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation). Circulation problem: given $(V,\ E,\ c,\ d)$, does there exist a circulation?

LEMMA 0.14. Necessary condition*: Sum of supplies = sum of demands.* $\sum_{v:d(v)>0} d(v) = \sum_{v:d(v)<0} -d(v) := D$.

*Proof.* Sum conservation constraints for every demand node $v$. □

Max flow formulation.

- Add new source $s$ and sink $t$.
- For each $v$ with $d(v) < 0$, add edge $(s,\ v)$ with capacity $-d(v)$.
- For each $v$ with $d(v) > 0$, add edge $(v,\ t)$ with capacity $d(v)$.
- Claim: $G$ has circulation if and only if $G'$ has max flow of value $D$, which saturates all edges leaving $s$ and entering $t$.

THEOREM 0.3. **Integrality theorem**. *If all capacities and demands are integers, and there exists a circulation, then there exists one that is integer-valued.*

LEMMA 0.15. *Given $(V,\ E,\ c,\ d)$, there does not exists a circulation if and only if there exists a node partition $(A,\ B)$ such that $\sum_{v\in B} d_v > cap(A,\ B)$. That is, demand by nodes in $B$ exceeds supply of nodes in $B$ plus max capacity of edges going from $A$ to $B$.*

Feasible circulation.

- Directed graph $G = (V,\ E)$.
- Edge capacities $c(e)$ and lower bounds $l(e)$, $e \in E$.
- Node supply and demands $d(v)$, $v \in V$.

A *circulation* is a function that satisfies: (1) for each $e \in E$: $l(e) \le f(e) \le c(e)$ (capacity); (2) for each $v \in V$: $\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = d(v)$ (conservation). Circulation problem with lower bounds. Given $(V,\ E,\ l,\ c,\ d)$, does there exists a a circulation?

Idea. Model lower bounds with demands. (1) send $l(e)$ units of flow along edge $e$; (2) update demands of both endpoints.

THEOREM 0.4. *There exists a circulation in $G$ if and only if there exists a circulation in $G'$. If all demands, capacities, and lower bounds in $G$ are integers, then there is a circulation in $G$ that is integer-valued.*

*Proof.* $f(e)$ is a circulation in $G$ if and only if $f'(e) = f(e) - l(e)$ is a circulation in $G'$. □