Lesson 4: Maximum-flow Problems

Notes

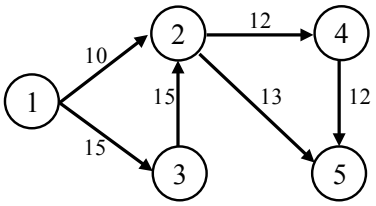
Book acknowledgment:

Goals

- Maximum Flow

1 Try it on your own

For the following network, the arc labels refer to the maximum allowable (capacity) flow on that arc. Given these capacities, determine the maximum amount of flow f that can be sent from 1 to 5.



How did you go about determining f? How could you make your strategy into an algorithm?

2 Problem Definition

2.1 Notation

Let $c_{uv} \geq 0$ be the capacity on each arc $(u, v) \in A$. For each $(u, v) \in A$, then $(v, u) \notin A$. Determine the maximum amount of flow that can be transmitted from s to t without violating arc capacities or flow conservation among all nodes (incoming flow at each node is equal to the outgoing flow at that node).

3 Ford-Fulkerson Algorithm

The Ford-Fulkerson Algorithm is an augmenting-path algorithm that iteratively increases the maximum s-t flow f by pushing from s to t on a sequence of arcs having positive residual capacity. A residual graph of a flow network is a graph that indicates additional possible flow given a current set of flows. If there is a path from s to t in the residual network, then additional flow can be sent from s to t. Every arc of a residual network has a residual capacity equal to the original capacity of the arc minus the current flow on that arc.

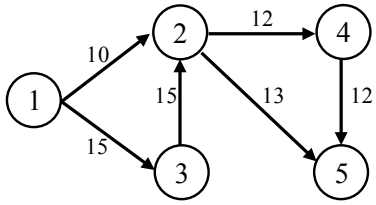
Written Steps:

1. Set $f = 0$.
2. While there is an augmenting s-t path in the residual network:
   - Add this path-flow to f.
3. Return f.

Pseudocode:

1. Set $f = 0$, $\bar{x}_{ij} = 0,\ \forall (i, j) \in A$, $\bar{c}_{ij} = c_{ij},\ \forall (i, j) \in A$, and $\bar{c}_{ji} = 0,\ \forall (i, j) \in A$. If $\bar{c}_{ij} > 0$, then (i, j) is found among those arcs A' in the residual network.
2. While a positive residual capacity path exists among the arcs in A':
   a. Let p be any positive residual capacity path among those arcs in A'. Let A_p contain all arcs on path p, and set the flow $f_p = \min\{\bar{c}_{ij} : (i, j) \in A_p\}$.
   b. Increase the total flow on each arc $(i, j) \in A_p$ by setting $\bar{x}_{ij} = \bar{x}_{ij} + f_p,\ \forall (i, j) \in A_p \cap (A \cap A')$ and $\bar{x}_{ij} = \bar{x}_{ij} - f_p,\ \forall (j, i) \in A_p \cap (A' \setminus A)$.
   c. Update residual capacities $\bar{c}_{ij} = \bar{c}_{ij} - f_p,\ \forall (i, j) \in A_p \cap (A \cap A')$ and $\bar{c}_{ij} = \bar{c}_{ij} + f_p,\ \forall (j, i) \in A_p \cap (A' \setminus A)$.
   d. Update A': For all $(i, j) \in A'$, if $\bar{c}_{ij} = 0$, then remove (i, j) from A'.
   e. Set $f = f + f_p$.
3. Return maximum flow f.

Using the Ford-Fulkerson Algorithm, determine the maximum amount of flow f that can be sent from 1 to 5.



4 Push-relabel Algorithms

The push-relabel algorithm is generally more efficient than the Ford-Fulkerson Algorithm. Similar to Ford-Fulkerson, the Push-relable also works on a residual graph. Alternatively, the push-relabel algorithm is drastically different. It works in a decentralized manner. As opposed to examining the entire residual network to find an augmenting path, the P-R algorithm works on one node at a time. During each step of the Ford-Fulkerson, the flow conservation at each node is maintained to be 0. Alternatively, the P-R algorithm allows for the in-flow at each node to exceed the out-flow at that node before reaching the final flow. At termination, though, flow conservation is conserved.

Notes:

- Each node has an associated height value and excess flow value.
- The height value helps to determine whether a node can send more flow to an adjacent node. The P-R algorithm only allows for a node to send flow to other nodes having lower height values.
- The excess flow value is the difference of total in-flow and the total out-flow at each node.
- Each arc has an associated flow value and capacity.

High-level Steps:

1. Initialize the height and flow of every node to be 0. Initialize the height of the source node to be equal to the number of vertices in the network. Initalize the flow on each arc to be 0. For all nodes adjacent to the source, flow and excess flow is equal to the capacity initially.
2. While a vertex has excess flow:
    a. If there is an adjacent node with a smaller height in the residual graph, push flow from the node to the adjacent node having a smaller height. The amount of flow to push is equal to the minimum between the excess flow at that node and the capacity on that arc.
    b. Otherwise, determine the minimum height adjacent node, and increase that heigh value by one.

Implementation Steps:

1. Set $f = 0$, $\bar{x}_{ij} = 0,\ \forall (i,j) \in A$, $\bar{c}_{ij} = c_{ij},\ \forall (i,j) \in A$, and $\bar{c}_{ji} = 0,\ \forall (i,j) \in A$. If $\bar{c}_{ij} > 0$, then (i,j) exists in A' in the residual network. Set $h_i = 0$ and $e_i = 0,\ \forall i \in V$. Finally, set $h_s = |V|$ and $e_s = \sum_{j:(s,j)\in A} c_{sj}$.

2. While $e_i > 0$ for some node $i \in V \setminus \{t\}$:

   a. While there exists some node u for which $h_u < h_i$ and $(i, u) \in A'$:

      i. If $(i, u) \in A \cap A'$, then increase the flow on (i, u) by setting $\bar{x}_{iu} = \bar{x}_{iu} + \min\{e_i,\ \bar{c}_{iu}\}$. Set $\bar{c}_{iu} = \bar{c}_{iu} - \min\{e_i,\ \bar{c}_{iu}\}$. Update A'

      ii. Otherwise, if $(i, u) \in A' \setminus A$, the decrease the flow on (u, i) by setting $\bar{x}_{ui} = \bar{x}_{ui} - \min\{e_i,\ \bar{c}_{iu}\}$. Set $\bar{c}_{ui} = \bar{c}_{ui} + \min\{e_i,\ \bar{c}_{iu}\}$. Update A'.

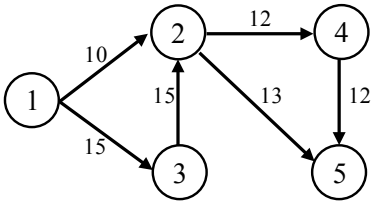
      iii. Set $e_i = e_i - \min\{e_i,\ \bar{c}_{iu}\}$ and $e_u = e_u + \min\{e_i,\ \bar{c}_{iu}\}$.

   b. Determine node u for which $h_u = \min\{h_j : (u, j) \in A'\}$.

   c. Set $h_u = h_u + 1$.

3. Return maximum flow value e_t.

Using the Push-Relabel Algorithm, determine the maximum amount of flow f that can be sent from 1 to 5.



5 Ford-Fulkerson vs. Push-Relabel

- What are the similarities between Ford-Fulkerson and the Push-Relabel?

- What are the differences between Ford-Fulkerson and the Push-Relabel Algorithms?

- The P-R algorithm has a running time of $O(V^2E)$ and the Ford-Fulkerson Algorithm has a running time of $O(VE^2)$.