5

Connectivity

P. Danziger

A *network* can be modeled as a set of nodes joined by links. Usually a network carries a *flow* of some material, either physical or virtual.
For example, the nodes may be junctions and the link physical pipes carrying oil or water. Alternately, the nodes may be computers (routers) and the links connections, the flow is then information. Another example would be with the links being roads, the nodes and the flow being a traffic flow. In such a situation it is natural to assign a *capacity* to each edge. A capacity of c on an edge indicates that no more than c of the indicated quantity may flow along a given edge.
We usually assume that a network is initially connected.

1 Cuts and Connectivity

Given a network one consideration is the *robustness* of the network. One measure of this is how many nodes or links would have to fail before the network becomes disconnected, so there are parts which cannot communicate with each other.
For this section we consider that all initial graphs G are connected. If the initial graph G is not connected we consider the connected components individually.

1.1 Vertex Connectivity

Definition 1 *Given a graph $G = (V, E)$:*

- *A separating set or vertex cut set is a set of vertices $S \subseteq V$, such that $G - S$ is disconnected.*
- *The connectivity of G, $\kappa(G)$, is the minimum size of $S \subseteq V$, such that $G - S$ is disconnected (a disconnecting set), or has only one vertex.*
- *A graph is $k-$connected if and only if its connectivity is at least k.*

Notes

- If a graph is $k-$connected, it is $\ell-$connected for every $\ell \leq k$
- Saying a graph is $1-$connected is equivalent to saying that it is connected.
- A graph is $0-$connected if and only if it is disconnected.
- $\kappa(G) = \min\{|S| \mid S \subseteq V(G)$ and S is a separating set $\}$
- There is a separating set of size $\kappa(G)$, but no separating set of size $\kappa(G) - 1$.

1.2 Edge Connectivity

Definition 2 *Given a graph $G = (V, E)$:*

- *A disconnecting set is a set of edges $F \subseteq E$, such that $G - F$ is disconnected.*
- *A graph is $k-$edge connected if and only if every disconnecting set has at least k edges.*
- *The edge connectivity of G, $\lambda(G)$, is the minimum size of a disconnecting set.*

Notation 3 *Given a graph G and a subsets of vertices $S, T \subseteq V(G)$, we represent the set of edges between them by $[S, T]$. Formally:*

$$[S, T] = \{e \in E(G) \mid e = u\,v, u \in S, v \in T\}.$$

Note that given any set S $[S, \overline{S}]$ is always a disconnecting set, called an edge cut.

Theorem 4 (Whitney 1932) *For any graph G,*

$$\kappa(G) \leq \kappa'(G) \leq \delta(G)$$

where $\delta(G)$ is the minimum degree in G.

2 Blocks

Definition 5 *A block in a graph G is a connected subgraph of G which contains no cut vertex.*

Theorem 6 *If a Depth First Search starting at a verttex v is used to generate a spanning tree of a graph G, T, then taking v as the root, every chord of co$-T$ connects an ancestor to a descendant.*

Proof: Suppose $e = u\,v$ is an edge of co$-T$ and that u is visited before v. Thus when DFS(u) is called v is still unlabeled and as a neighbour of u, DFS(v) will not terminate before DFS(u) is called. □
So chords do not go across branches of a DFS spanning tree.

2.1 Block Search algorithm

Input A (di)graph G
Idea We traverse the graph with a DFS starting at x, creating a DFS spanning tree of G with root x.

We keep an index P for each vertex, this variable will be the same for every vertex in the block (part). We keep checking for back edges to an already visited vertex, if one of these is found, the value of P is updated.

Initialization:
    for all $v \in V(G)$, $D(v) = 0$.
    $F = \emptyset$
    $i = 1$

Recursion:
Block(v)
    $P(v) = D(v) = i$
    $i = i + 1$
    For all $u \in N(v)$
        Push the edge $u\,v$ onto the stack
        if $D(u) == 0$ then
            parent[u] = v
            Block(u)
            If $P(u) \geq D(v)$ then
                Pop all edges up to and including $u\,v$ from the stack and mark them as a block
            $P(v) = \min(P(v), P(u))$
        Else if (u has already been visited and) $u \neq$ parent[v] then
                $P(v) = \min(P(v), D(u))$
        end if
    end for

Call: while there is $v \in V$ for which $D(v) = 0$
    Block(v)