Graph Theory

Dr. Jens Dörpinghaus

doerpinghaus@uni-koblenz.de

Mathematisches Institut
Universität Koblenz-Landau
Campus Koblenz

Federal Institute for Vocational Education and Training (BIBB)
Robert-Schuman-Platz 3
53175 Bonn

Summer 2022

5 Connectivity

- Forests and trees
- Minimum Spanning Trees
- Menger's theorem
- 2-Connected graphs and subgraphs
- The structure of 3-connected graphs

Repetition

Definition 5.1

A graph G is called *connected* if it is non-empty and any two of its vertices are linked by a path in G.

- If $U \subseteq V(G)$ and $G[U]$ is connected, we also call U itself connected (in G).
- Instead of 'not connected' we usually say 'disconnected'.

We consider a variant of traversal that tests if a graph is connected:

Algorithm 6 Test connectivity

1: choose $e \in E$ and set $F = e$, $E' = E \setminus e$
2: **while** $E' \neq \emptyset$ **do**
3: choose $e \in E'$, which is incident with at least one node in $V(F)$.
4: set $F = F \cup e$, $E' = E' \setminus e$

Theorem 5.2

Algorithm 6 terminates with $E = F$ and $V = V(F)$ exactly when G is connected.

Proof.

Exercise. □

Corollary 5.3

Every connected graph with n nodes has at least $n - 1$ edges.

Proof.

- Consider an initial graph F with one edge and two nodes.
- In every step we add one edge and at most one new vertex.
- In every step $|F| \geq |V(F)| - 1$ holds.

$\square$

Definition 5.4

A graph $G = (V, E)$ is called k-connected if we need at least k vertices to disconnect it.

Definition 5.5

A graph $G = (V, E)$ is called k-connected if any two of its vertices can be joined by k independent paths.

Example 5.6

A cycle is 2-connected.

Definition 5.7

The greatest integer k such that G is k-connected is the *connectivity* $\kappa(G)$ of G.

Definition 5.8

If $|G| > 1$ and $G - F$ is connected for every set $F \subseteq E$ of fewer than l edges, then G is called l-edge-connected.

Definition 5.9

The greatest integer l such that G is l-edge-connected is the edge-connectivity $\lambda(G)$ of G.

In particular, we have $\lambda(G) = 0$ if G is disconnected.

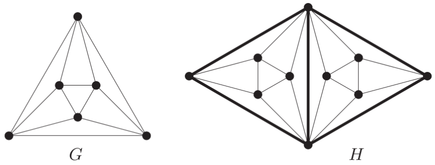


Figure 7: The octahedron G (left) with $\kappa(G) = \lambda(G) = 4$, and a graph H with $\kappa(H) = 2$ but $\lambda(H) = 4$.

The trivial graph is the graph on one vertex.

Lemma 5.10

If G is non-trivial then $\kappa(G) \leq \lambda(G) \leq \delta(G)$.

Definition 5.11

An edge e is called an *intersection edge* if $G \setminus e$ has more connected components than G (more precisely: one more).

- We now consider graphs in which every edge is intersection edge.
- Equivalently, the graph contains no circles.
- A graph without circles is called a *forest*.
- A connected forest is a *tree*.
- A *leaf* in a forest is a node of degree one.
- Trees always have at least two leaves.

Lemma 5.12

The terminal nodes of longest paths in a tree have degree one.

Proof.

- Let W be a path of maximum length and u a terminal node of W.
- Suppose u has another neighbor w.
- If w is not on W, then $[W, w]$ or $[w, W]$ is a longer path than W.
- If w lies on W, then the path from w to u and the edge (u, w) induce a circle.

□

Repetition

Theorem 5.13

The following assertions are equivalent for a graph T:

1. *T is a tree;*
2. *Any two vertices of T are linked by a unique path in T;*
3. *T is minimally connected, i.e. T is connected but $T - e$ is disconnected for every edge $e \in T$;*
4. *T is maximally acyclic, i.e. T contains no cycle but $T + xy$ does, for any two non-adjacent vertices $x, y \in T$.*
5. *G is connected and each edge of G is intersection edge.*

Corollary 5.14 (Kirchoff, 1847)

A graph is connected if and only if it contains a spanning tree.

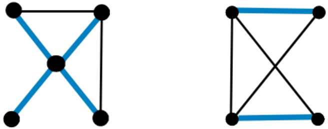
Proof.

- Let T be a spanning tree of a graph G
- since T is connected, the top graph G is also connected
- conversely, let G be connected
- then G contains a minimal connected subgraph T, where every edge is an intersection edge.
- by Theorem 1.21 T is a tree.

□

Definition 5.15

A forest $F \subseteq G$ is called maximal if for each component G' of G the subgraph $F \cap G'$ is a spanning tree.



The blue edges form a spanning, non-maximal Forest.

Corollary 5.16

Let G be a graph with k components and $F \subseteq G$. F is a maximal forest if and only if F has k components and $n - k$ edges.

Proof.

Follows from Theorem 1.21 and Corollary 5.14. □

Each edge of a spanning tree can be replaced by a suitable edge of another spanning tree:

Corollary 5.17

Let T, T' be the edge sets of two spanning trees of a connected graph. Then for every $e \in T \setminus T'$ there exists an $e' \in T' \setminus t$ such that $T \setminus e \cup e'$ is again a spanning tree.

Proof.

Exercise. □

- The structural properties of spanning trees allow to determine trees with the lowest possible weight.
- Let $G = (V, E)$ be a connected graph
- with weights $c(e)$.
- We search for a spanning tree $T = (V, F)$ of G such that

$$c(T) = \sum_{e \in F} c(e)$$

is minimal among all spanning trees

Algorithm 7 Jarnik (1930), Prim (1957), Dijkstra (1959)

choose $v \in V$ arbitrarily and set $U = \{v\}$, $T = \emptyset$
while $\exists$ a minimum-weighted edge $e = (u, v) \in E$ with $u \in U$, $v \in V \setminus U$ **do**
    set $U = U \cup v$ and $T = T \cup e$

Algorithm 8 Borůvka (1926), Kruskal (1956)

```
Set $F = \emptyset$
**for** $i = 1, ..., |V| - 1$ **do**
    choose a minimal-weighted edge $e \in E \setminus F$, which does not close a circle in $F$.
    set $F = F \cup e$, $E = E \setminus e$
```

Algorithm 9 Dual Greedy Algorithm (Kruskal (1956)

1: Set $F = E$
2: **while** F contains edges that are not intersection edges **do**
3: choose a maximal weighted edge $e \in F$ which is not an intersection edge
4: set $F = F \setminus e$

Theorem 5.18

Algorithm 9 computes a minimum spanning tree.

Proof.

- F is spanning and contiguous at any point of time.
- At the end every edge is an intersection edge, i.e. F is a tree according to Theorem 1.21.
- Suppose F has not minimal weight.
- Let T be a minimal spanning tree with $|T \cap F|$ maximal.
- Consider the first edge $e \in T \setminus F$, which the algorithm removes.
- With Corollary 5.17, there exists an edge $e' \in F \setminus T$ such that $S = T \cup e' \setminus e$ is a spanning tree.
- In the step in which e was removed, e' could also have been removed
- Thus:

$$c(e) \geq c(e') \text{ and hence } c(S) \leq c(T)$$

- Then S is a minimal spanning tree with $|S \cap F| > |T \cap F|$.

$\square$

Kruskal's algorithm can be implemented with suitable data structures in such a way that it has a running time of $\mathcal{O}(m \log m)$ or $\mathcal{O}(m + n \log n)$

The following theorem is one of the cornerstones of graph theory.

Theorem 5.19 (Menger, 1927)

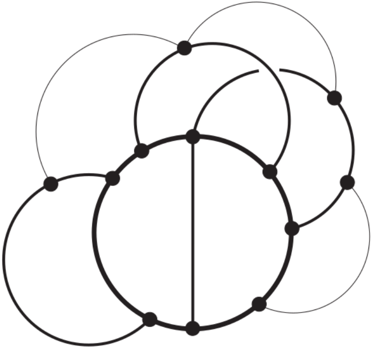
Let $G = (V, E)$ be a graph and $A, B \subseteq V$. Then the minimum number of vertices separating A from B in G is equal to the maximum number of disjoint $A - B$ paths in G.

Definition 5.20

Given a graph H, we call P an H-path if P is non-trivial and meets H exactly in its ends. In particular, the edge of any H-path of length 1 is never an edge of H.

Proposition 1

A graph is 2-connected if and only if it can be constructed from a cycle by successively adding H-paths to graphs H already constructed.



Proposition 2

A graph is 2-connected if and only if it can be constructed from a cycle by successively adding H-paths to graphs H already constructed.

Proof.

"$\Leftarrow$" Clearly, every graph constructed as described is 2-connected.
"$\Rightarrow$" Let a 2-connected graph $G = (V, E)$ be given.

- G contains a cycle and hence has a maximal subgraph H constructible as above.
- Since any edge $(x, y) \in E \setminus E(H)$ with $x, y \in H$ would define an H-path, H is an induced subgraph of G.
- Thus if $H \neq G$, then by the connectedness of G there is an edge (v, w) with $v \in G - H$ and $w \in H$.
- As G is 2-connected, $G - w$ contains a $v - H$ path P.
- Then $[w, v, P]$ is an H-path in G, and $H \cup [w, v, P]$ is a constructible subgraph of G larger than H.
- This contradicts the maximality of H.

$\square$

- In this section we describe how every 3-connected graph can be obtained from a K_4 by a succession of elementary operations preserving 3-connectedness.
- Given an edge e in a graph G, let us write $G \circ e$ for the *multigraph* obtained from $G - e$ by suppressing any end of e that has degree 2 in $G - e$.

Lemma 5.22

Let e be an edge in a graph G. If $G \circ e$ is 3-connected then so is G.

Proof.

- Thinking of G as obtained from $G \circ e$ by adding e, let us call the vertices of $G \circ e$ the old vertices of G, and any other vertex of G (which will be an end of e) a new vertex.
- Remembering that $G \circ e$, being 3-connected, has no parallel edges, it is easy to see that, in G, no two vertices x_1 , x_2 can separate a new vertex from all the old vertices.
- So it suffices to show that x_1, x_2 cannot separate two old vertices.
- If they did, then those old vertices would be separated in $G \circ e$ by x_1' and x_2', where either $x_i' = x_i$ or, if x_i is new, x_i' is the edge of $G \circ e$ subdivided by x_i. By Lemma 5.10, this contradicts our assumption that $G \circ e$ is 3-connected.

□

Lemma 5.23

Every 3-connected graph $G \neq K_4$ has an edge e such that $G \circ e$ is another 3-connected graph.

Theorem 5.24 (Tutte, 1966)

A graph G is 3-connected if and only if there exists a sequence $G_0, ..., G_n$ of graphs such that

1. $G_0 = K_4$ *and* $G_n = G$;
2. G_{i+1} *has an edge* e *such that* $G_i = G_{i+1} \circ e$, *for every* $i < n$.

Moreover, the graphs in any such sequence are all 3-connected.

Proof.

- If G is 3-connected, use Lemma 5.23 to find $G_n, ..., G_0$ in turn.
- Conversely, if $G_0, ..., G_n$ is any sequence of graphs satisfying (1) and (2), then all these graphs, and in particular $G = G_n$, are 3-connected by Lemma 5.22.

□

- Theorem 5.24 enables us to construct, recursively, the entire class of 3-connected graphs.
- Starting from K_4, we simply add to every graph already constructed a new edge in every way compatible with (2):
  - between two already existing vertices, between newly inserted subdividing vertices (not on the same edge),
  - or between one old vertex and one new subdividing vertex.
- A second method of reducing 3-connected graphs to K_4 is contracting edges.