MAT344 Week 7

2019/Oct/21

1 This week

This week, we are talking about

1. Homeomorphic graphs
2. Criteria for planarity
3. Minimum weight spanning trees

2 More on the Euler characteristic

Corollary 2.1. *Let G be a connected planar graph. Then every planar drawing of G has the same number of faces.*

Proof. $|V(G)|$ and $|E(G)|$ are determined independently of the drawing, and $|F(G)| = 2 - |V(G)| + |E(G)|$.

Q.E.D.

Corollary 2.2 (Theorem 5.12. in [KT17])**.** *If G is a connected planar graph and $|V| \geq 3$, then*

$$|E| \leq 3|V| - 6.$$

If, in addition, G has no cycles of length less than 4, then

$$|E| \leq 2|V| - 4$$

Proof. We will give a combinatorial proof, counting the number of edges we encounter as we move around the boundary of each face.

First, note that every edge is adjacent to either one or two faces. If it is adjacent to two faces, it separates them, and as we move around both of those faces, we will count the edge once for each face. If an edge is adjacent to just one face, it will still be counted twice as we move around the boundary of the face (once we move toward the "outside" of the face and once when we move toward the "inside"). So every edge is counted exactly twice, so our count is $2|E|$.

Secondly, we look at all the faces and count how many edges surround that face. There must be at least 3 edges around each face, unless there is just one face (in this case G is a tree, so $|E| = |V| - 1$ and since $|V| \geq 3$, we are done). Therefore, we have counted at least $3|F|$ edges (there is some overcount here, but that's okay, we are looking for an inequality).

From the above, we learn that $2|E| \geq 3|F|$, or, equivalently

$$|F| \leq \frac{2|E|}{3}.$$

Using Euler's formula, $|F| = 2 - |V| + |E|$, we get

$$2 \leq |V| - |E| + \frac{2|E|}{3},$$

and some algebra yields

$$|E| \leq 3|V| - 6.$$

For the second part, we assume that G has no cycles of length less than 4. In this case, every face must be surrounded by at least 4 edges, and the rest of the argument is unchanged.

So we get the estimate $2|E| \geq 4|F|$, or

$$|F| \leq \frac{|E|}{2}.$$

And applying Euler's formula yields

$$2 - |V| + |E| \leq \frac{|E|}{2}$$

from which we get

$$|E| \leq 2|V| - 4.$$

Q.E.D.

Corollary 2.3. *$K_{3,3}$ is not planar.*

Proof. Consider the case of $K_{3,3}$. It has 6 vertices and 9 edges. It also has no cycles of length 3 (or any odd-length cycles, since it's bipartite). Any planar graph that has 6 vertices and contains no 3-cycles can have at most

$$2 \cdot 6 - 4 = 8$$

edges. So $K_{3,3}$ is not planar.

Q.E.D.

3 Homeomorphic graphs and Kuratowski's Theorem

It turns out that K_5 and $K_{3,3}$ are the two graphs that determine planarity. Let us say what this means. We already know that any graph containing either a copy of K_5 or $K_{3,3}$ as a subgraph can not be planar. Consider the graph on Figure 1.

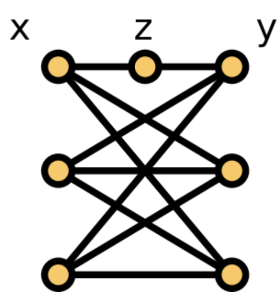

Figure 1: A graph homeomorphic to $K_{3,3}$

The first thing that we notice is that the graph looks very similar to $K_{3,3}$, except that we subdivided an edge between the top two vertices by adding an extra vertex z. In particular, the vertices x and y are no longer adjacent.

Definition 3.1. *If G is a graph and xy is an edge, then we can form a new graph G' called an* ***elementary subdivision*** *of G by adding a new vertex z and replacing the edge xy by edges xz and zy.*

As a result, the graph no longer contains a copy of $K_{3,3}$. The only possible subgraph that could be a copy of $K_{3,3}$ would be all vertices except z (for degree reasons), but as x is not adjacent to y, we can conclude that this graph does not contain $K_{3,3}$.

Should this affect planarity? As we are drawing the edges in the plane, all we care about are edge crossings. Having the vertex z in our graph adjacent to both x and y does not change the drawing problem in any meaningful way (note that we can not cheat by making an edge "pass through" z, since z is not adjacent to any vertex other than x and y).

Definition 3.2. *Two graphs G_1 and G_2 are said to be* ***homeomorphic*** *if they can be obtained from some graph G by a sequence of elementary subdivisions.*

Exercise 3.3. *If you know some topology, prove that graphs that are homeomorphic as graphs are also homeomorphic as topological spaces (with the subspace topology from $\mathbb{R}^2$).*

Our argument above can be generalized to the following proposition

Proposition 3.4. *Any graph containing a subgraph homeomorphic to K_5 or $K_{3,3}$ is nonplanar.*

Exercise 3.5. *Show that the Petersen graph (see Figure 2) is not planar by finding a subgraph homeomorphic to $K_{3,3}$.*

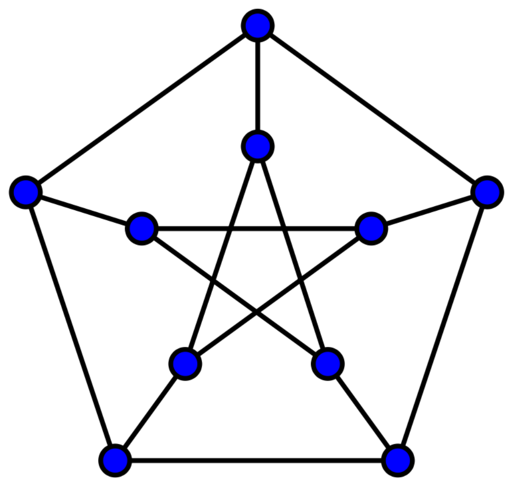

Figure 2: The Petersen graph

What is much more surprising is that the converse of Proposition 3.4 is also true:

Theorem 3.6 (Kuratowski, 1930, Theorem 5.13 in [KT17])**.** *A graph is planar if and only if it does not contain a subgraph homeomorphic to K_5 or $K_{3,3}$.*

In this week's tutorials you'll see two operations on graphs, **edge deletion** and **edge contraction**. For a graph G with an edge xy, we can construct two new graphs:

- $G - \{xy\}$ by **deleting** the edge, leaving the vertices alone;
- $G/\{xy\}$ by **contracting** the edge, combining x and y into one vertex, and removing any multiple edges or loops.

For example, deleting any edge in the cycle graph C_4 gives P_4, and contracting any edge gives C_3.

Definition 3.7 (Definition 15.19 in [Mor17])**.** *Let G be a graph. Then H is a* ***minor*** *of G if we can construct H from G by a sequence of edge deletions, edge contractions and vertex deletions.*

Theorem 3.8 (Wagner, 1937, Theorem 15.20 in [Mor17])**.** *A graph is planar if and only if it has no minor isomorphic to K_5 or $K_{3,3}$.*

Exercise 3.9. *Find a minor in the graph on Figure 1 isomorphic to $K_{3,3}$.*

Exercise 3.10. *Find a minor of the Petersen graph (see Figure 2) isomorphic to K_5.*

4 Minimum weight spanning trees (Ch.12.1 in [KT17])

We have looked at graphs as representing many real-world scenarios before. However, in many cases, a graph does not contain all the necessary information that would model the problem accurately. For example, if we have a graph that represnts nodes in a future network, and we want to establish connections between these nodes, it is likely that the cost of establishing connections between different nodes are different.

For example, consider the graph in 3. The nodes may represent locations of computers, and the labels (weights) on the edges represent the cost of laying the cables necessary for the connection between two given locations. We would like all computers to be able to communicate with each other, and we want to minimize the cost of establishing the connection.

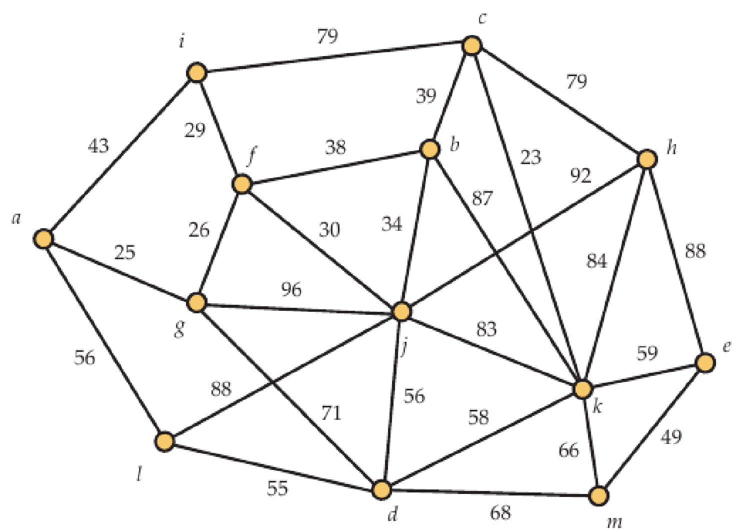

Figure 3: A proposed network

Definition 4.1. *Let $G = (V_G, E_G)$ be a graph, and let $w : E_G \to \mathbb{R}_{\geq 0}$ be a function. For any edge $e \in E_G$, the quantity $w(e)$ is called the **weight** of e. If $H = (V_H, E_H) \subseteq G$ is a subgraph of G, then $w(H) = \sum_{e \in E_H} w(e)$.*

In mathematical terms, we are looking for a spanning tree in this graph that has a minimal sum of weights. We know that a mimimal spanning tree is out there, but enumerating over all cases is not really an option, so we need a more clever algorithm.

This question is one that can be solved by a *greedy algorithm.* A greedy algorithm is one that proceeds with a step that seems most optimal immediately.

5 Prim's algorithm (Ch. 5.6 in [Gui18])

Prim's algorithm was developed in 1930 by Vojtěch Jarník, then it was rediscovered by Robert C. Prim in 1957.

Given a weighted connected graph G, we construct a minimum weight spanning tree T as follows.

Choose any vertex v_0 in G and include it in T . If vertices $S = \{v_0, v_1, \ldots, v_k\}$ have been chosen, choose an edge with one endpoint in S and one endpoint not in S and with smallest weight among all such edges. Let v_{k+1} be the endpoint of this edge not in S, and add it and the associated edge to T. Continue until all vertices of G are in T.

Theorem 5.1 (Theorem 5.6.2 in [Gui18])**.** *Prim's algorithm produces a minimum weight spanning tree*

Proof. Suppose G is connected on n vertices. Let T be the spanning tree produced by the algorithm, and T_{min} a minimum cost spanning tree. We prove that $w(T) = w(T_{min})$.

Let $e_1, e_2, \ldots, e_{n-1}$ be the edges of T in the order in which they were added to T; we label the vertices in a way that one endpoint of e_i is v_i, the other is in $\{v_0, \ldots, v_{i-1}\}$. We construct a sequence of trees $T_{min} = T_0, T_1, \ldots, T_{n-1} = T$ such that for each i, $w(T_i) = w(T_{i+1})$.

Suppose we have constructed tree T_i. If e_{i+1} is in T_i, let $T_{i+1} = T_i$. Otherwise, add edge e_{i+1} to T_i. This creates a cycle, one of whose edges, call it f_{i+1}, is not in $e_1, e_2, \ldots, e_i$ and has exactly one endpoint in $\{v_0, \ldots, v_i\}$. Remove f_{i+1} to create T_{i+1}. Since the algorithm added e_{i+1}, $w(e_{i+1}) \leq w(f_{i+1})$. If $w(e_{i+1}) < w(f_{i+1})$, then $w(T_{i+1}) < w(T_i) = w(T_m)$, a contradiction, so $w(e_{i+1}) = w(f_{i+1})$ and $w(T_{i+1}) = w(T_i)$.

Therefore $w(T) = w(T_{n-1}) = w(T_0) = w(T_{min})$.

Q.E.D.

Remark 5.2. *There is another commonly used algorithm, known as Kruskal's algorithm. See [KT17], Ch. 12.2.1 for details.*

References

[Gui18] David Guichard. *Combinatorics and Graph Theory*. Open access, 2018. Available at `https://www.whitman.edu/mathematics/cgt_online/book/`. 4

[KT17] Mitchel T. Keller and William T. Trotter. *Applied Combinatorics*. Open access, 2017. Available at `http://www.rellek.net/appcomb/`. 1, 3, 4, 5

[Mor17] Joy Morris. *Combinatorics*. Open access, 2017. Available at `http://www.cs.uleth.ca/~morris/Combinatorics/Combinatorics.html`. 3