Planar Graphs Addendum Notes

Addendum I: Kuratowski's, and Wagner's Theorems

- Kuratowski provided a characterization of planar graphs in terms of forbidden graphs now known as Kuratowski's theorem:
    - A finite graph is planar if and only if it does not contain a subgraph K_5 that is a subdivision (where vertices can be added into edges zero or more times) of the complete graph $K_{3,3}$

      ^---- requires an algorithm that checks for subdivisions in the graph K_5

- Another method is Wagner's theorem which deals with minors (a type of graph that results from a subgraph and repeatedly contradicting an edge into a vertex, with each neighbor of the original end-vertices becoming a neighbor of the new vertex):
    - A finite graph K_5 is planar if and only if it does not have the complete graph $K_{3,3}$ as a minor

      ^---- requires an algorithm to contradict an edge into a vertex repeatedly

Summary:
Either of the above theorems can determine whether a graph is planar; however, either requires an algorithm to check the conditions of each respective theorem. The following addendum shows a list of algorithms that could be used with Kuratowski's theorem.

Addendum II: Other Planarity Criteria

Problem: for a graph *G* with *n* vertices, it is possible to determine whether the graph may be planar or not (shown in ***ii***).

i) Planarity Testing and the Fraysseix-Rosenstiehl Criterion

Definition: the planarity testing problem is the algorithmic problem of testing whether a given graph is a planar graph. Contains planarity criteria (***ii)*** that works with Kuratowski's theorem.

- The two previous theorems are indirect applications of planarity testing, while Fraysseix-Rosenstiehl's criterion is direct.
    - The criterion characterizes planar graphs in terms of a left-right ordering of edges in a depth-first search tree.

- There are four algorithms that can find whether a graph is planar or not: path, vertex, and edge addition methods, and the construction sequence method.
    - Out of all methods, the edge addition method is one of the best methods, as it was a modified, more efficient version of the vertex method.
    - Edge addition is used to compute planar embedding if possible; otherwise, a Kuratowski subdivision is computed.
    - The method is one of the two current state-of-the-art algorithms, the other being the Fraysseix criterion.

ii) Planarity Criteria

For a simple, connected, planar graph with v vertices and e edges and f faces, the following simple conditions hold for $v \geq 3$:

Theorem I: $e \leq 3v - 6$
Theorem II: if there are no cycles of length 3, then $e \leq 2v - 4$
Theorem III: $f \leq 2v - 4$

In this sense, planar graphs are sparse graphs in that they have only *O(v)* edges, smaller than the maximum $O(v^2)$

- If theorem I & II fail, other methods may be used

Other methods include:

- Mac Lane's planarity criterion gives an algebraic characterization of finite planar graphs via their cycle spaces.

- Hanani-Tulte's theorem states that a graph is planar if and only if it has a drawing in which each independent pair of edges crosses an even number of times; it can be used to characterize the planar graphs via a system of equations modulo 2

Summary: Since the project focuses on planar graphs with only edges and vertices, the two main methods of finding whether a graph is planar or not are:

1. Edge addition and Kuratowski subdivisions
2. Fraysseix criterion (preferable since it is a direct algorithm to planarity testing)

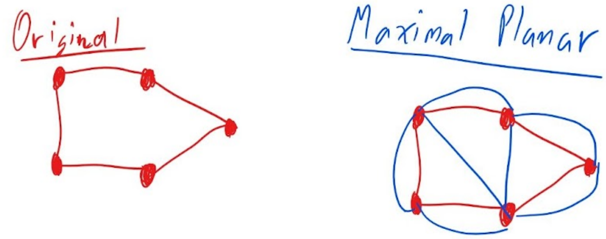
Addendum III: Related Families of Graphs

i) Maximal Planar Graphs

Definition: a simple graph is maximal planar if it is planar, but adding an edge on the given vertex set would destroy the graph's property of planarity.

- If a maximal planar graph has v vertices with $v > 2$, then it has precisely $3v - 6$ edges and $2v - 4$ faces

Example:



- If any more edges are added, the graph will lose its property of planarity.

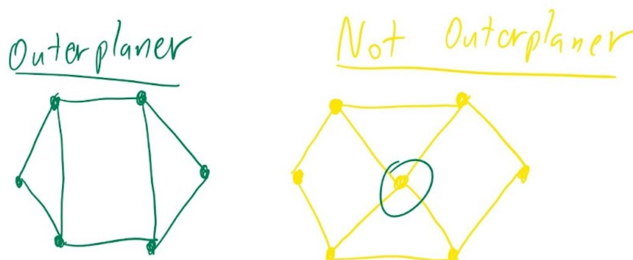
ii) Outerplanar Graphs

Definition: Outerplanar graphs are graphs with an embedding in the plane such that all vertices belong to the unbound face of the embedding.

- Every outerplanar graph is planar, but the converse is not true. K_4 is a planar graph but not outerplanar.
    - A similar theorem to Kuratowski's states that:

        A finite graph is outerplanar if and only if it does not contain a subdivision of K_4 or $K_{2,3}$.

Example:



- Outerplaner graphs have boundury vertices, meaning they are touching the infinte face

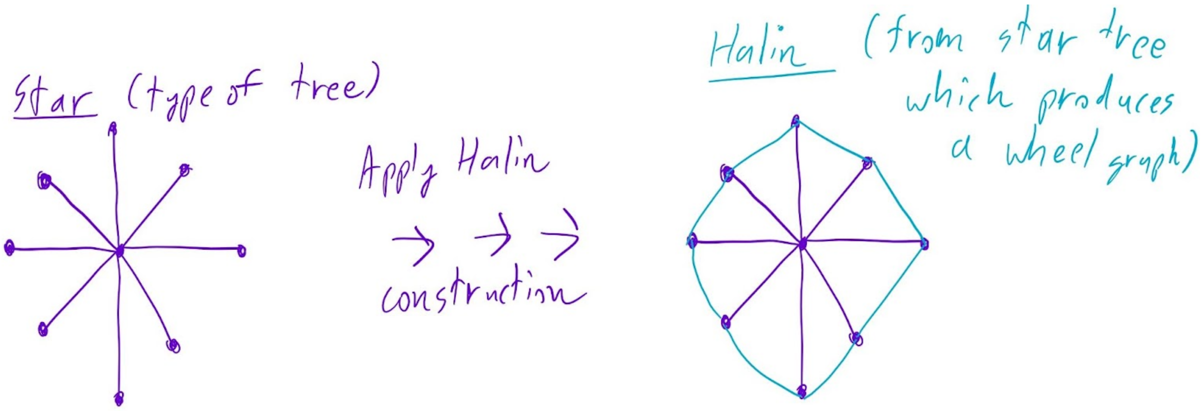
- The vertex circled is enclosed in the graph, not touching the infinite face

iii) Halin Graphs

Definition: A Halin graph is a graph formed from an undirected plane tree (with no degree-two nodes) by connecting its leaves into a cycle, in the order given by the plane embedding of the tree.

- Like outerplanar graphs, Halin graphs have low treewidth, making many algorithmic problems more easily solved than in unrestricted planar graphs.

Example:



- Halin graphs are constructed by creating a cycle by connecting the leaves of the tree

iv) Other Related Families

- An apex graph is a graph that may be made planar by removing one vertex, and a k-apex graph is a graph that may be made planar by removing at most k vertices.

- A 1-planar graph is a graph that may be drawn in the plane with at most one simple crossing per edge, and a k-planar graph is a graph that may be drawn with at most k simple crossings per edge.

- A map graph is a graph formed from a set of finitely many simply-connected interior-disjoint regions in the plane by connecting two regions when they share at least one boundary point. When at most three regions meet at a point, the result is a planar graph, but when four or more regions meet at a point, the result can be nonplanar.

- A toroidal graph is a graph that can be embedded without crossings on the torus.

- An upward planar graph is a directed acyclic graph that can be drawn in the plane with its edges as non-crossing curves that are consistently oriented in an upward direction. Not every planar directed acyclic graph is upward planar, and it is NP-complete to test whether a given graph is upward planar.

Mention: Starts/Stops and 3D Printing

- The paper discusses one method of printing parts without starting or stopping the extruder on a 3D printer.
    - Starts and stops are problematic for efficiency and reliability for 3D printed parts, as the finished product could have holes when the printer pauses and travels to another section of the part.
    - The higher the number of distinct edges and contours a geometry has, the more starts and stops are introduced into the toolpath.
    - Toolpaths are generated specific to the user-defined settings, then closed loops are generated, and infills fill the remaining void in the 3D printed part.

- The current solution towards starts and stops would be a technique called spiralizing in which an algorithm would remove independent closed-loop paths to make a continuous loop path
    - This is done by taking the layer polygon and offsetting the outer boundary inward by one bead length. In other words, a graph would have an edge close enough to an edge that the beads merge even though the toolpath is still a continuous loop.
    - For the beads to not intersect when a layer is complete, slowly incrementing the Z-height to the next bead layer is changed.
    - However, this method does not work for dense structures, so another solution is to make sure the path does not cross other edges (using the planarity property of planar graphs)
        - If the extruder got to a corner and then turned around to travel back on itself, it could return to the starting point while tracing all and paths while never starting/stopping.

- Printing with optimization causes starts and stops with spiralizing, which is not preferable with designers. Quality of printing can overweight optimization, so larger, and thicker prints should not have optimization available

<u>Conclusion:</u>

There are two options to determine whether a graph is planar or not:

- The Fraysseix criterion can be used since it is a direct algorithm for planarity testing. However, this method requires finding T-alike and T-opposite pairs in the depth-first search tree to make the procedure efficient. The tree pairs are initially present in the graph, which means it can be used to double existing edges.

- Edge addition and Kuratowski subdivisions, which involves finding embedding while maintaining planarity.
  However, this method is an indirect solution and requires the Kuratowski subdivisions to solve the problem if edge addition fails, resulting in possible inefficiency. We also do not want to add edges that were not initially present in the graph.

Therefore, the Fraysseix method will be used for the project. Using the spiraling method with the Fraysseix criterion will help avoid edges crossing on a 3D printed part and using a desirable path with Euler circuits.

Plan for the algorithm (draft):

1. Check planarity using the Fraysseix criterion, so we do not have crossed edges to find a desirable path such that there do not exist travel paths and start/stops.
2. Determine an Euler circuit and find the desirable path.