Planarity Testing of Graphs

Swami Sarvottamananda

Ramakrishna Mission Vivekananda University

NITPatna-IGGA, 2011

Outline I

Introduction

Scope of the lecture

- *Characterisation of Planar Graphs:* First we introduce planar graphs, and give its characterisation alongwith some simple properties.

Scope of the lecture

- *Characterisation of Planar Graphs:* First we introduce planar graphs, and give its characterisation alongwith some simple properties.
- *Planarity Testing:* Next, we give an algorithm to test if a given graph is planar using the properties that we have uncovered.

Scope of the lecture

- *Characterisation of Planar Graphs:* First we introduce planar graphs, and give its characterisation alongwith some simple properties.
- *Planarity Testing:* Next, we give an algorithm to test if a given graph is planar using the properties that we have uncovered.
- *Planar Embedding:* Lastly we see how a given graph can be embedded in a plane using straight lines.

What are Planar Graphs—Drawings?

Definition (Drawing)

Given a graph $G = (V, E)$, a drawing maps each vertex $v \in V$ to a distinct point $\Gamma(v)$ in plane, and each edge $e \in E, e = (u, v)$ to a simple open jordan curve $\Gamma(u, v)$ with end points $\Gamma(u)$, $\Gamma(v)$.

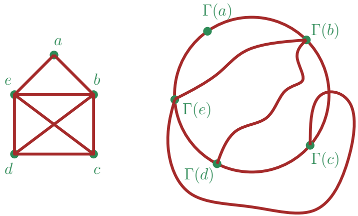


Figure: Drawing of a graph

What are Planar Graphs—Non-crossing Drawings?

Definition (Planar Graphs)

Given a graph $G = (V, E)$, G is planar if it admits a drawing such that no two distinct drawn edges intersect except at end points.

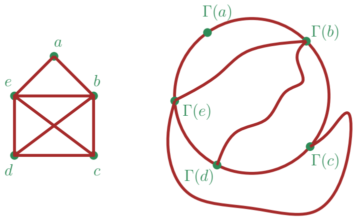


Figure: Planar drawing of a graph

Motivation

Properties of Planar Graph

There are number of interesting properties of planar graphs.

- They are sparse.

Properties of Planar Graph

There are number of interesting properties of planar graphs.

- They are sparse.
- They are 4-colourable.

Properties of Planar Graph

There are number of interesting properties of planar graphs.

- They are sparse.
- They are 4-colourable.
- A number of operations can be performed on them very efficiently.

Properties of Planar Graph

There are number of interesting properties of planar graphs.

- They are sparse.
- They are 4-colourable.
- A number of operations can be performed on them very efficiently.
- They can be efficiently stored (A data structure called $SPQR$-trees even allows $O(1)$ flipping of planar embeddings).

Properties of Planar Graph

There are number of interesting properties of planar graphs.

- They are sparse.
- They are 4-colourable.
- A number of operations can be performed on them very efficiently.
- They can be efficiently stored (A data structure called $SPQR$-trees even allows $O(1)$ flipping of planar embeddings).
- There size including faces, edges and vertices is $O(n)$.

Applications of Planar Graph

Planar graphs are extensively used in Electrical, Mechanical and Civil engineering.

- Easy to visualize. In fact, crossing of edges is the main culprit for reducing comprehensibility.

Applications of Planar Graph

Planar graphs are extensively used in Electrical, Mechanical and Civil engineering.

- Easy to visualize. In fact, crossing of edges is the main culprit for reducing comprehensibility.
- VLSI design, circuit needs to be on surface: lesser the crossings, better is the design.

Applications of Planar Graph

Planar graphs are extensively used in Electrical, Mechanical and Civil engineering.

- Easy to visualize. In fact, crossing of edges is the main culprit for reducing comprehensibility.
- VLSI design, circuit needs to be on surface: lesser the crossings, better is the design.
- Highspeed Highways/Railroads design, crossings are always problematic.

Applications of Planar Graph

Planar graphs are extensively used in Electrical, Mechanical and Civil engineering.

- Easy to visualize. In fact, crossing of edges is the main culprit for reducing comprehensibility.
- VLSI design, circuit needs to be on surface: lesser the crossings, better is the design.
- Highspeed Highways/Railroads design, crossings are always problematic.
- Irrigation canals, crossings simply not admissible.

Applications of Planar Graph

Planar graphs are extensively used in Electrical, Mechanical and Civil engineering.

- Easy to visualize. In fact, crossing of edges is the main culprit for reducing comprehensibility.
- VLSI design, circuit needs to be on surface: lesser the crossings, better is the design.
- Highspeed Highways/Railroads design, crossings are always problematic.
- Irrigation canals, crossings simply not admissible.
- Most of facility location problems on maps are actually problems of planar graphs.

Problem Definition

Problem Definition: Planarity Testing

Problem (Decision Problem)

Given a graph $G = (V, E)$, is G planar, i.e., can G be drawn in the plane without edge crossings?

Problem Definition: Planar Embedding

Problem (Computation Problem)

Given a graph $G = (V, E)$, if G is planar, how can G be drawn in the plane such that there are no edge crossings? I.e., compute a planar representation of the graph G.

Question: K_4

Is the following graph planar (K_4)?

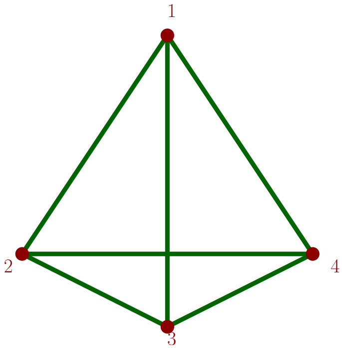


Figure: Graph K_4

Planarity of K_4

Yes K_4 is planar.

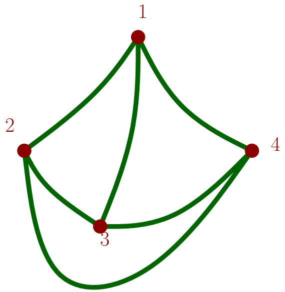
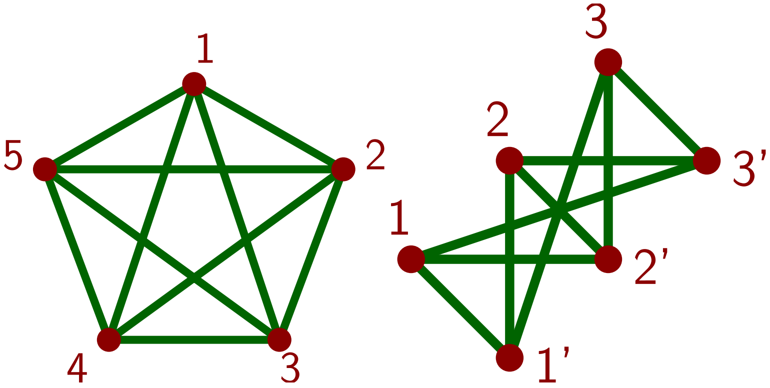


Figure: Planar graph K_4

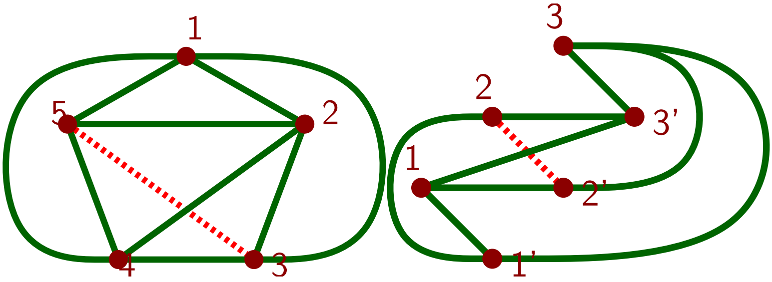
Question: K_5 and $K_{3,3}$

Are the following graphs planar?



Answer for K_5 and $K_{3,3}$

No, they aren't. There always will be at least one crossing.



Full proofs by Euler's celebrated theorem.

Question

Is the following graph planar?
There are so many crossings ($O(n^2)$).

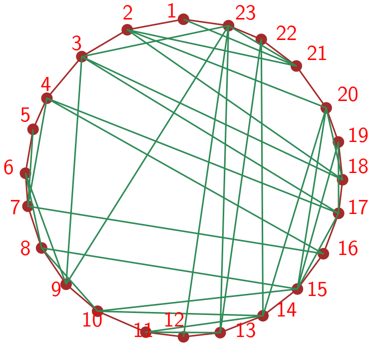


Figure: a hamiltonian graph

Answer

Yes! It is.

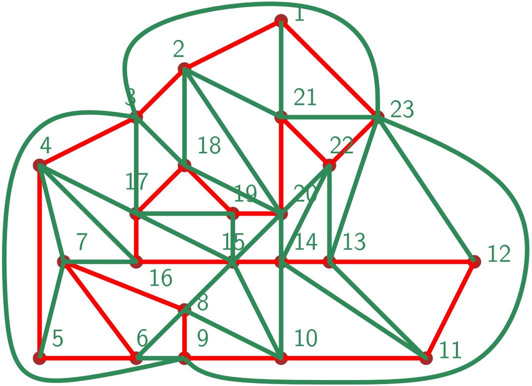


Figure: Planar embedding of the last graph

But how to arrive at the answer?

Characterisation of Planar Graphs

Basic Assumptions

We assume that our graphs are connected and there are no self loops and no multi-edges.
Disconnected graphs, 1-degree vertices, multi-edges can be easily dealt with.

Euler's Relation

Theorem (Euler's Relation)

Given a plane graph with n vertices, m edges and f faces, we have $n - m + f = 2$.

Fact

The exterior is also counted as a face. The above relation also applies to simple polyhedrons with no holes.

Euler formula gives the necessary condition for a graph to be planar[3].

Euler's Relation: Corollary 1

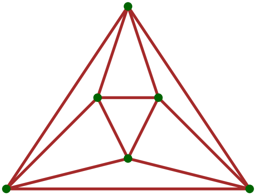


Figure: Octahedron; $n = 6, m = 12 \leq 3n - 6$

Corollary

For a maximal planar graph, where each face is a triangle, we have $m = 3n - 6$*, and therefore, for any graph with at least three vertices, we have* $m \leq 3n - 6$*.*

Proof: $\sum_{x \in F} e_x = 2m$ and therefore since $e_x \geq 3$, $2m \geq 3f$.

Non-planarity of K_5

Lemma

K_5 is non-planar.

Proof: $n = 5, m = 10 > 3n - 6 = 9.$

Euler's Relation: Corollary 2

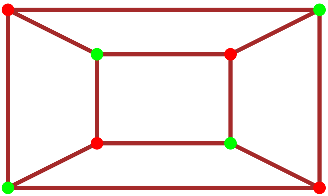


Figure: Cube; $n = 8, m = 12 \leq 2n - 4$

Corollary

For a planar graph, where no face is a triangle, we have $m \leq 2n - 4$.

Proof: $\sum_{x \in F} e_x = 2m$ and therefore since $e_x \geq 4$, $2m \geq 4f$. $m \leq 2n - 4$ follows.

Non-planarity of $K_{3,3}$

Lemma

$K_{3,3}$ is non-planar.

Proof: $K_{3,3}$ is bi-partite, therefore has no cycle of odd length, hence if it is planar then no face will be triangular.
Then, $n = 6, m = 9 > 2n - 4 = 8$.

Euler's Relation: Corollary 3

Corollary

Any planar graph is 6 colourable.

Proof: Since $m \leq 3n - 6$, there exists a vertex with degree less than 6 (otherwise $\sum_v d_v = 2m \Rightarrow 6n \leq 2m$).
By induction, if we remove this vertex, resulting graph is 6-colourable. Just give this vetex a colour other than the five colours of the neighbours.

Euler's Relation: Corollary 4

Corollary

Any planar graph is 5 colourable.

Proof: The neighbours of 5-degree vertex aren't all connected. Take one such pair and do the following:

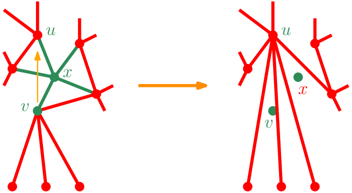


Figure: u and v are not connected

By induction, if we remove the vertex x and v, resulting graph is 5-colourable. Just give v same colour as u and x a colour other than the 4 colours of the neighbours.

Concepts relating to Kuratowski's and Wagner's Theorems

Euler's conditions are necessary but not sufficient, for example join K_5 and $K_{3,3}$ by an edge.
Next we look at Kuratowski's and Wagner's Theorems for conditions of sufficiency.
Before stating the theorems, we need to understand subdivisions and minors of a graph.

What is subdivision and minor of a graph?

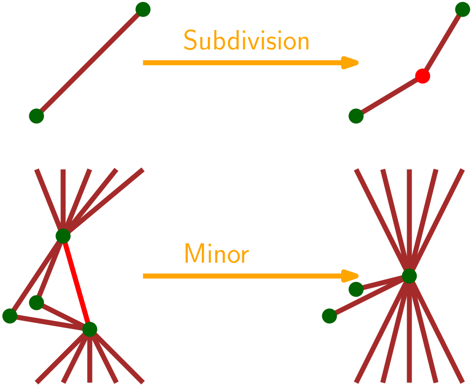


Figure: Subdivision and minors

Subdividing an edge in a planar graph does not make it non-planar. Shrinking an edge of a planar vertex to make a single vertex does not make it non-planar.

What is subdivision and minor of a graph?

In a way subdivision and minors are complementary. In subdivisions we add a vertex and in minors we remove a vertex.
Algorithmically, both are expensive for planarity testing.

Kuratowski's Theorem

Definition

Subdividing any edge means replacing the edge with a path of length 2.

Theorem (Kuratowski's Theorem[6])

G is planar iff G contains no sub-division of K_5 or $K_{3,3}$.

As noted earlier, subdividing an edge in a planar graph does not make it non-planar.

Wagner's Theorem

Theorem (Wagner's Theorem[10])

G is planar iff G contains no subgraph which has K_5 or $K_{3,3}$ as minor.

This is an alternate characterisation of planar graphs. See [4] for yet another characterisation.
As noted earlier, shrinking an edge of a planar graph to make a single vertex does not make it non-planar.

Need of Subdivisions and Minors

Why do we need subdivisions or minors? Isn't subgraphs sufficient? Then we need to show a non-planar graph which do not have K_5 or $K_{3,3}$ as subgraphs.

Peterson Graph

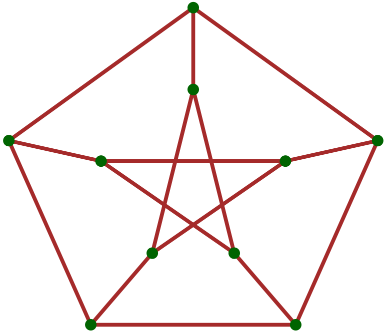


Figure: Peterson Graph

A graph which doesn't have K_5 or $K_{3,3}$ as a subgraph. However, it has a subdivision of $K_{3,3}$ and both K_5 and $K_{3,3}$ as minors.

Non-planarity of Peterson Graph

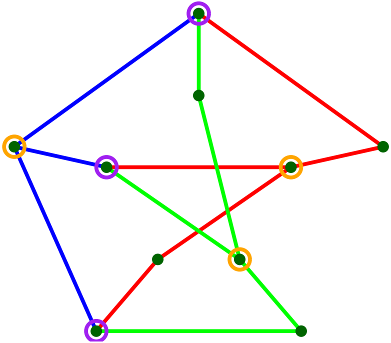


Figure: Peterson Graph

Peterson Graph has a subdivision of $K_{3,3}$. The same subgraph can also be used to get $K_{3,3}$ as minor.

Non-planarity of Peterson Graph

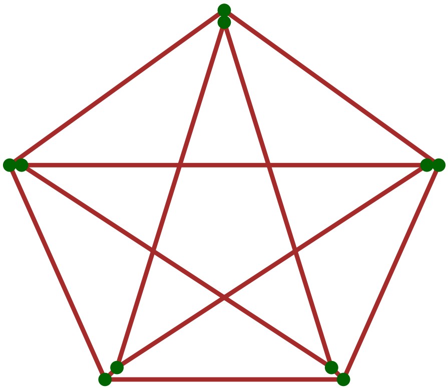
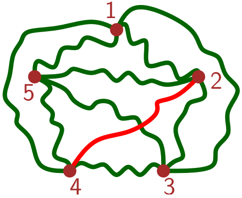


Figure: Peterson Graph

Peterson Graph has K_5 as a minor.

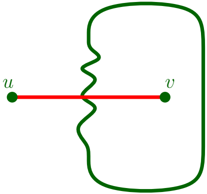
Sufficiency Proof of Kuratowski's Theorem



Proof of sufficiency: Suffuciency immediately follows from non-planarity of K_5 and $K_{3,3}$.
Any subdivison of K_5 and $K_{3,3}$ is also non-planar.
Sufficiency condition of Wagner's theorem can also be proved easily. Shrinking edges will not change planarity. So, if we get K_5 or $K_{3,3}$ by shrinking edges, then initial subgraph must be non-planar to start with.

Proof of Necessary Condition of Kuratowski's Theorem

Proof of necessity: Proof of necessity is a little difficult.

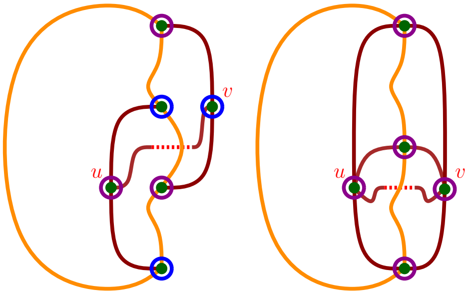


- Suppose G is non-planar.
- First we remove edges and vertices of a non-planar graph such that it becomes a minimal non-planar graph.
- I.e. removing any edge will make the resulting graph planar.
- How does removing an edge of a non-planar graph make it planar?
- Somehow we always need to join a vertex inside and outside of a cycle.

Necessary Condition of Kuratowski's Theorem

Why can't we move u or v to the other side?
There must be something which is stopping us doing that.

Necessary Condition of Kuratowski's Theorem



- Either there is no common joining vertices but which make conflicting ears. We get $K_{3,3}$'s subdivision.
- Or, there are three common joining vertices but which make conflicting tripods. We get K_5's subdivision.

Necessary Condition of Kuratowski's Theorem

Thus it can be shown that

Fact

If G is non-planar it must contain either a sub-division of $K_{3,3}$ or K_5.

Planarity Testing of Graphs

How to test Planarity

Question is—How to apply Kuratowski's theorem? Obviously an exponential method.

Planarity testing using Kuratowski's Theorem

To test for K_5's subdivision.

- Choose 5 vertices of G.
- Check if all 5 vertices are connected by $\binom{5}{4} = 10$ distinct paths as K_5.

To test for $K_{3,3}$'s subdivision.

- Choose 6 vertices of G.
- Check if are 6 vertices connected by $3 \times 3 = 9$ distinct paths as $K_{3,3}$.

As pointed out previously both are obviously exponential time algorithms.

Planarity testing using Wagner's Theorem

This is easier to understand. Do the following for every edge.

- Choose an edge of G (m choices).
- Shrink it.
- If 6 vertices are remaining check for $K_{3,3}$
- If 5 vertices are remaining check for K_5
- Repeat.

Worst case $O(m!)$.

Both are expensive

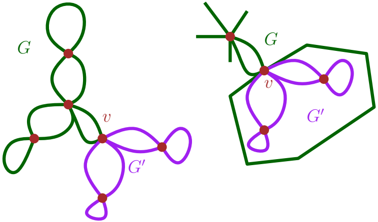
Conclusion: both are obviously exponential time algorithms.
How can we do it more efficiently?

Basic Assumptions-I

We assume the following.

- The given graph G does not have self loops or multi-edges.
- G is undirected and satisfies $m \leq 3n - 6$.
- G is connected. If it is disconnected, we can test planarity of disconnected components separately. This also means there won't be isolated vertices.
- No vertex is of degree 1.
- G is stored by adjacency lists. We also rename vertices v's to their $dfs(v)$'s.

Basic Assumptions-II



We further assume.

- G is bi-connected. If G is not, it is easy to find cut-vertices and test the planarity of each bi-connected component separately.
- We can embed the component in a face adjacent to the cut vertex.
- Actually we can make any face of a planar graph the outer face.

Outline of Planarity Testing

- First we find out bi-connected components of the given graph G.
- Test planarity of each bi-connected component individually. If all are planar then G is planar.
- Assume that bi-connected component itself is G.
- We do a depth-first-search and computer all tree edges and forward/backward edges.
- In the process we compute $dfs(v)$'s and two lowpoint arrays $L_1(v)$'s and $L_2(v)$'s.
- We sort the adjacency lists according to the criteria of lowpoints.
- There is one cycle, and we try to embed G branch by branch recursively.

Tree edges and Fronds

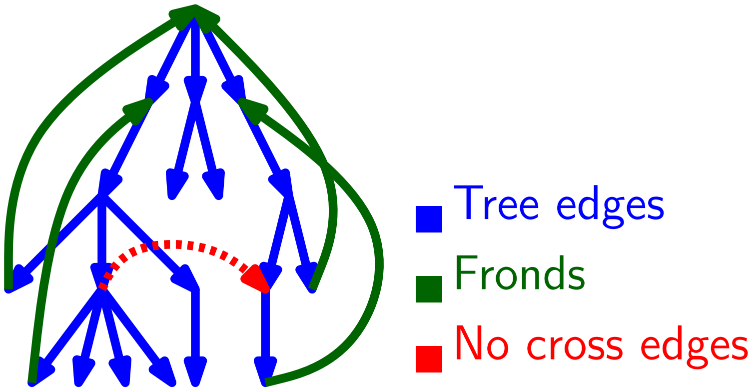


Figure: DFS tree and backward/forward edges

Lowpoints L_1 and L_2

We define lowpoints $L_1(u)$ and $L_2(u)$ for each vertex as follows. Let $S(u)$ be set of all descendants of u and $T(u)$ be set of all neighbours of $S(u)$.

Definition

$L_1(u) = \min T(u)$
$L_2(u) = \min(T(u) - L_1(u))$

Lowpoints L_1 and L_2

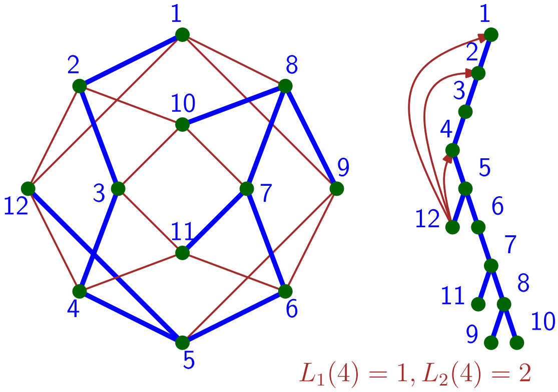


Figure: DFS tree and Lowpoints

Facts about Lowpoints L_1 and L_2

Fact

Since G is 2-connected graph, it has no cut-vertices, so we can conclude that $L_1(v) < u$ whenever u is parent of v and u is not the root node. It follows that $L_2(v) \leq u$.

Fact

Low-points are well-defined, since every vertex has atleast 2 neighbours.

Reordering of Edges

Next we reorder edges of a vertes so that components are added on the tree in increasing depth of where they are attached on the DFS tree.
Let $a(v)$ denote the parent of v. Then weight of edge uv is defined as

$$wt_u(v) = \begin{cases} 2v & \text{if } uv \text{ is a frond with } v < u \\ 2L_1(v) & \text{if } a(v) = u \text{ and } L_2(v) = u \\ 2L_1(v) + 1 & \text{if } a(v) = u \text{ and } L_2(v) < u \\ 2n + 1 & \text{otherwise} \end{cases}$$

Reordering of Edges: Explanation

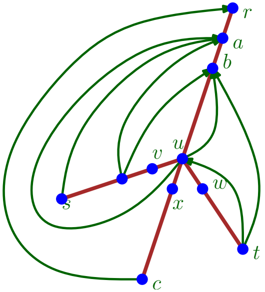


Figure: DFS tree and Lowpoints

At vertex u the branches are embedded in the order $B_u(x), ua, B_u(v), ub, B_u(w)$, and ut
Since $wt_u[x] = 2r < wt_u[a] = 2a < wt_u[v] = 2a + 1 < wt_u[b] = 2b = wt_u[w] < wt_u[t] = 2n + 1$

Two type of branches

A branch $B_u(v)$, where $u = a(v)$, with $L_2(v) = u$ is called a type I branch. If $L_2(v) < u$ it is called a type II branch.
Type I branches are easier.
We need to calculate branch points $b(v)$ where v separates first from the main trunk.

Fact

G is planar iff each brance can be embedded on one side of its stem (defined as strongly planar, as in Mehlhorn's book). Stem is the part from L_1 point to the beginning of the branch.

This enables us to check conflict only among the branches (called segments in the Mehlhorn's book).

Branch Points

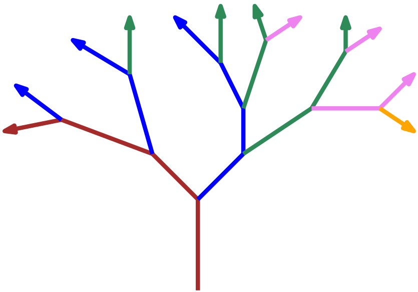


Figure: Branch Points of Each Branch

Fact

The way ordering of edges is done, the stem is always formed by the first children till a frond is encountered.

Conflicting branches and fronds

Ordering of Adj[u] by weights is not sufficient to guarantee that an embedding is possible without further refinement.

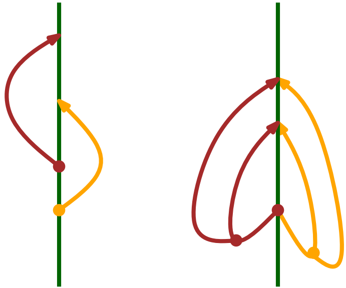


Figure: Conflicting Fronds and Branches

If there is a conflict then we must embed on the opposite sides of the current path from root.

Outline of Algorithm I

- We embed the whole graph using the DFS tree T branch by branch.
- Since G is bi-connected, initially we will have a cycle.
- A frond uv, where $v < u$, is embedded either on the left side of T or on the right side.
- So we have an embedding of T in the plane, with the fronds arranged around T giving an embedding of G.
- We first determine an ordering of $adj(u)$ so that the branch points are guaranteed to permit an embedding when G is planar.
- Then, following Hopcroft and Tarjan, we keep two linked lists of fronds, LF and RF containing the fronds and branches embedded on the left of T and on the right of T , respectively.

Outline of Algorithm II

- LF and RF can be viewed as stacks, whose tops contain the number of the vertex currently marking the upper limit in the tree to which fronds may be embedded.
- We also construct the conflict graph of branches in LF and RF as IG which is a bipartite graph.
- Each branch in turn is embedded recursively.

K_5 Planarity Testing

How does K_5 fare in our algorithm?

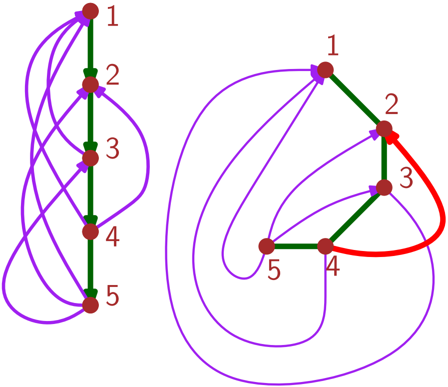


Figure: Testing of Planarity for K_5

$K_{3,3}$ Planarity Testing

How does $K_{3,3}$ fare in our algorithm?

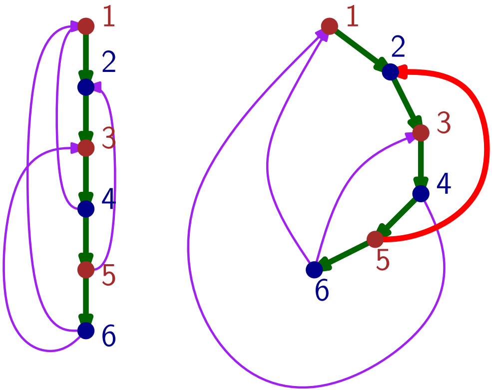


Figure: Testing of Planarity for $K_{3,3}$

Algorithm: EmbedBranch I

```
EmbedBranch(u: vertex)
  for each v in Adj[u] do
     NonPlanar := true
     if a(v) = u then { uv is a tree edge }
        if b(v) = u then { uv begins a new branch at u }
           if L1(v) is too small to permit an embedding the
           place Bu (v) either on LF or RF
        EmbedBranch(v)
        if NonPlanar then Exit
     else if v < u then { uv is a frond }
        EmbedFrond(u, v)
        if EmbedFrond is unsuccessful then Exit
  NonPlanar := false { no conflicting fronds were found
```

Analysis and Correctness of Planarity Algorithm

EmbedBranch is called repeatedly in the order of DFS algorithm for every branch that is encountered.

It is hard to see the algorithm works. Harder still to see it can be made to work in linear time[7].

Planar Embedding of Graphs

Planar Embedding Theorem

Theorem

Let G be a 2-connected planar graph. Then we can embed G in the plane in linear time[4, 8]

Other embeddings

The embedding that we have obtained has curved edges.
Sometimes we need different criteria for embedding, such as:

- Edges might be needed to be straight.
- Further, vertices might be needed on the grid.
- Furthermore, area might be needed to be minimized.
- Or we need edges made of orthogonal segments (then additionally vertices will need to be rectangular regions, for degree > 4).

Straight Line embedding: Basic Idea

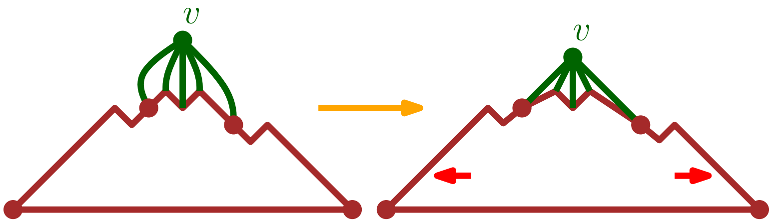


Figure: Straight Line Embedding

First we triangulate the planar graph. Start with an outer edge then add vertex after vertex.
We choose v such that it is connected to all consecutve nodes on the chain.
Then we shift the other vertices on the chain either right or left.

Straight Line Planar Embedding Theorem

Theorem

Any planar graph with n nodes has a straight line embedding into the $2n-4$ by $n-2$ grid such that edges are mapped into straight-line segments. Also such an embedding can be constructed in $O(n \log n)$ time.[8, 1, 9]

Tuttes Theorem

Theorem (Tuttes Theorem)

If G is a 3-connected planar graph, then G has a convex embedding in the plane.

This is a stronger result that Kuratowskis theorem.

Conclusion

Open Problems and Generalisations

- Planar graphs which admit straight-line grid drawings on grids of linear area.
- Planar graphs which admit unit length straight-line edges.
- Planar graphs can be generalised to higher dimension where we have hyper-faces.
- If we allow crossings, then sometimes it makes sense to minimise crossings.
- There is a concept of book thickness of graphs. We embed the graph such that vertices are in spine and edges can be in the pages. We have to minimise number of pages.

Summary

- We studied the concept of graph planarity.

Summary

- We studied the concept of graph planarity.
- We gave a short proof of Kuratowski's theorem.

Summary

- We studied the concept of graph planarity.
- We gave a short proof of Kuratowski's theorem.
- Next we saw how we can answer queries about planarity of graph.

Summary

- We studied the concept of graph planarity.
- We gave a short proof of Kuratowski's theorem.
- Next we saw how we can answer queries about planarity of graph.
- We looked into calculation of embeddings.

Summary

- We studied the concept of graph planarity.
- We gave a short proof of Kuratowski's theorem.
- Next we saw how we can answer queries about planarity of graph.
- We looked into calculation of embeddings.
- Now we are for the final concluding remarks.

You may read

- Harary's book for basic graph theory[3].
- Hopcroft and Tarjan's paper[5] for linear planarity testing.
- Mehlhorn and Mutzel's paper[8] for linear planar embeddings.
- S. Even, A. Lempel, and I. Cederbaum's work[2] for a simpler embedding algorithm.
- Chapter 4 of Mehlhorn's book[7] for details.

References I

H. de Fraysseix, J. Pach, and R. Pollack.
How to draw a planar graph on a grid.
Combinatorica, 10, 1990.

S. Even, A. Lempel, and I. Cederbaum.
An algorithm for planarity testing of graphs.
In *Theory of Graphs: Internat. Symposium (Rome 1966)*, pages 215–232. New York, Gordon and Breach, 1967.

F. Harary.
Graph Theory.
Addison-Wesley, Reading, Mass., 1969.

F. Haray and W. T. Tutte.
A dual form of kuratowskis theorem.
Bull. Amer. Math. Soc., 21(1):168, 1965.

References II

J. Hopcroft and R. E. Tarjan.
Efficient planarity testing.
Journal of ACM, 21(4):549–568, 1974.

Kazimierz Kuratowski.
Sur le problème des courbes gauches en topologie.
Fund. Math., 15:271–283, 1930.

Kurt Mehlhorn.
Data Structures and Algorithms 2: Graph Algorithms and NP-Completeness, volume 2 of *Monographs in Theoretical Computer Science. An EATCS Series.*
Springer-Verlag, 1984.

References III

Kurt Mehlhorn and Petra Mutzel.
On the embedding phase of the hopcroft and tarjan planarity testing algorithm.
Algorithmica, 16:233–242, 1996.

W. Schnyder.
Embedding planar graphs on the grid.
In *Proc., First ACM-SIAM Symp. on Discrete Algorithms*, pages 138–148, San Francisco, 1990.

K. Wagner.
Uber eine erweiterung eines satzes von kuratowski.
Deutsche Mathematik, 2:280–285, 1937.

At Last . . .

Thank You

shreesh@rkmvu.ac.in
sarvottamananda@gmail.com