Planar Graphs: Coloring and Drawing

Graphs & Algorithms
Lecture 8
Coloring maps with 4 colors

**Four Color Theorem** (Appel, Haken 1976)
If $G$ is planar, then $\chi(G) \leq 4$.

**Idea of the proof**
- W.l.o.g. we can assume $G$ is a planar triangulation.
- A **configuration** is a separating cycle $C \subseteq G$ (the **ring**) together with the portion of $G$ inside $C$.
- For the Four Color Problem, a set of configurations is an **unavoidable set** if a minimum counterexample must contain some member of it.
- A configuration is **reducible** if a planar graph containing cannot be a minimal counterexample.
- Find an **unavoidable set** in which each configuration is reducible.
A problem with a long history

- Kempe's original attempt (1879) with a set of 3 unavoidable configurations was wrong.
- Appel and Haken, working with Koch, (1976) came up with a set of 1936 unavoidable configurations, each of which is reducible (1000 hours of computer time).
- Robertson, Sanders, Seymour, and Thomas (1996) used a set of 633 unavoidable configurations (3 hours of computer time).
Algorithms for coloring planar graphs

• Coloring a planar graph with $k \leq 2$ (if possible) or $k \geq 6$ colors is trivial.

• Heawood's proof of the 5-Coloring Theorem can be turned into a $O(n^2)$ algorithm for $k = 5$.

• There exist $O(n)$ algorithms for 5-coloring.

• The proof of Appel and Haken of the 4-Coloring Theorem can be turned into a $O(n^4)$ algorithm.

• Robertson, Sanders, Seymour, and Thomas [1996] provide a $O(n^2)$ algorithm for $k = 4$.

• Deciding whether a planar graph $G$ permits a coloring for $k = 3$ is NP-complete even if $\Delta(G) \leq 4$. 
Characterization of planar graphs

• Subdivision of $K_5$ and $K_{3,3}$:

Theorem (Kuratowski [1930])
A graph is planar if and only if it does not contain a subdivision of $K_5$ or $K_{3,3}$.

• Can we use this for planarity testing?
• Testing all subsets of the vertex set requires exponential time.
H-fragments and the conflict graph

• Let G and H be a graphs such that H ⊆ G. An H-fragment is either
  – an edge not H whose endpoints are in H, or
  – a component of G – V(H) together with the edges (and vertices of attachment) that connect it to H.

• Let C be a cycle in G. Two C-fragments A, B conflict
  – if they have three common vertices of attachment to C or
  – if there are four vertices v₁, v₂, v₃, v₄ in cyclic order on C such that v₁, v₃ are vertices of attachment of A and v₂, v₄ are vertices of attachment of B.

• The conflict graph of C is a graph whose vertices are the C-fragments of G with edges between conflicting C-fragments.
Another characterization

**Theorem** (Tutte [1958])

A graph is planar **if and only if** it the conflict graph of each cycle in G is bipartite.

• Examples
  – conflict graph of a spanning cycle of $K_5$ is $C_5$
  – conflict graph of a spanning cycle of $K_{3,3}$ is $C_3$

• Can we use this for planarity testing?
• Yields a quadratic algorithm.
A quadratic planarity testing algorithm

• Due to Demoucron, Malgrange, and Pertuiset [1964]

• Ideas
  – Suppose a planar embedding of $H$ can be extended to a planar embedding of $G$ (invariant).
  – Build increasingly large plane subgraphs $H \subseteq G$ that can be extended to an embedding of $G$ if $G$ is planar.
  – Make small decisions that do not lead into trouble.
  – Enlarge $H$ by choosing a face $F$ that can accept some $H$-fragment $B$ (boundary of $F$ contains attachment of $B$).
  – Add any path in $B$ between any two vertices of attachment to $H$ (there is just one way to embed the path into $F$).
Planarity testing algorithm

\textbf{PLANARITY TESTING}( 2-connected graph G)

1. \( H \leftarrow \text{any cycle } C \subseteq G \text{ embedded in the plane} \)
2. \( \mathcal{F} \leftarrow \text{inner and outer face of } C \)
3. \textbf{while } \( H \neq G \)
4. \textbf{do } \( \mathcal{B} \leftarrow \text{all } H\text{-fragments of } G \)
5. \textbf{for each } \( B \in \mathcal{B} \)
6. \textbf{do } \( \mathcal{F}_B \leftarrow \{ F \in \mathcal{F} : F \text{ contains all vertices of attachment of } B \} \)
7. \textbf{if } \exists B \in \mathcal{B} : |\mathcal{F}_B| = 0
8. \textbf{then return } NONPLANAR
9. \textbf{if } \exists B \in \mathcal{B} : |\mathcal{F}_B| = 1
10. \textbf{then } \( B_0 \leftarrow \text{any } B \in \mathcal{B} : |\mathcal{F}_B| = 1 \)
11. \textbf{else } \( B_0 \leftarrow \text{any } B \in \mathcal{B} \)
12. \( P \leftarrow \text{any path between two vertices of attachment of } B_0 \)
13. \( H \leftarrow H \cup P \text{ with } P \text{ embedded into some } F \in \mathcal{F}_{B_0} \)
14. \( \mathcal{F} \leftarrow \text{update faces of } H \)
15. \textbf{return } PLANAR

- \( G \) is planar iff every block of \( G \) is planar.
- check \( e(G) \leq 3n - 6 \)
- maintain appropriate lists for face boundaries
- each search takes linear time
The switching argument

Unless there is a conflict like this

Impossible!

(© D. West 2001)
Other planarity testing algorithms

- The proof of Kuratowski's Theorem by Klotz [1989] can be turned into a quadratic algorithm. It finds a subdivision of $K_5$ or $K_{3,3}$ when $G$ is not planar.
Drawing planar graphs

• What is an aesthetically appealing drawing?

Theorem (Fáry [1948])
Each planar graph has an embedding in which all edges are straight line segments.

• We call an embedding of a planar graph in which all edges are straight line segments a Fáry embedding.
Constructing Fáry embeddings

**Theorem** (Fraysseix, Pach, and Pollack [1990])

Any plane graph with \( n \) vertices has a Fáry embedding on the \( 2n - 4 \) by \( n - 2 \) grid.

- The proof is constructive and yields a quadratic algorithm, which can be improved to \( O(n \log n) \).
- Before this work, it was not even known whether there always exists a Fáry embedding into a grid of polynomial size.
- The grid size is asymptotically optimal.
- W.l.o.g. assume that \( G \) is a triangulation, or add dummy edges otherwise.
Canonical ordering

**Lemma** (Canonical representation for plane graphs)

Let $G$ be a maximal planar graph embedded in the plane with exterior face $u$, $v$, $w$. Then there exists a labeling of the vertices $v_1 = u$, $v_2 = v$, $v_3$, ..., $v_n = w$ such that, for every $4 \leq k \leq n$, we have:

- The subgraph $G_{k-1} \subseteq G$ induced by $v_1$, $v_2$, ..., $v_{k-1}$ is 2-connected, and the boundary of its exterior face is a cycle $C_{k-1}$ containing the edge $\{u, v\}$.

- $v_k$ is in the exterior face of $G_{k-1}$, and its neighbors in $G_{k-1}$ form an (at least 2-element) subinterval of the path $C_{k-1} - \{u, v\}$.

**Proof**

Construct the labeling recursively (easy in $O(n^2)$).
Finding Fáry embeddings iteratively

• Suppose we are given a Fáry embedding of $G_{k-1} = G[v_1, \ldots, v_{k-1}]$. Let $u = w_1, w_2, \ldots, w_m = v$ be the vertices on the outer border of $G_{k-1}$.

• Let $w_l$ and $w_r$ denote the leftmost and rightmost neighbors on the outer border of $G_{k-1}$, respectively.

Idea: move
– every vertex to the right of $w_l$ by one unit,
– every vertex to the right of $w_r$ (including $w_r$) by another unit.
The Fáry embedding algorithm

\textsc{FáryEmbedding( canonical ordering }v_1, \ldots, v_n)\text{ )}

1 \ x[v_1] \leftarrow 0 : y[v_1] \leftarrow 0
2 \ x[v_2] \leftarrow 2 : y[v_2] \leftarrow 0
3 \ x[v_3] \leftarrow 1 : y[v_3] \leftarrow 1
4 \ \textbf{for } k = 4 \ \textbf{to } n
5 \ \qquad \textbf{do } w_l \leftarrow \text{leftmost neighbor of } v_k \text{ in } G_{k-1}
6 \ \qquad w_r \leftarrow \text{rightmost neighbor of } v_k \text{ in } G_{k-1}
7 \ \qquad \text{move every vertex } v \in \{v_1, \ldots, v_{k-1}\}
8 \ \qquad \quad \text{with } x[v] > x[w_l] \text{ to the right by 1}
9 \ \qquad \text{move every vertex } v \in \{v_1, \ldots, v_{k-1}\}
10 \ \qquad \quad \text{with } x[v] \geq x[w_r] \text{ to the right by 1}
11 \ \qquad \text{set } x[v_k] \text{ and } y[v_k] \text{ according to the crossing of the lines}
12 \ \qquad \text{through } w_l \text{ with slope } +1 \text{ and through } w_r \text{ with slope } -1
13 \ \textbf{return } x, y
Correctness of the algorithm

• The FÁRYEMBEDDING algorithm maintains the following invariants for all $3 \leq k \leq n$:
  1) $x[u] = y[u] = 0$
  2) $x[v] = 2k - 4$ and $y[v] = 0$
  3) For the vertices $u = w_1, w_2, ..., w_m = v$ on the outer border of $G_{k-1}$, we have $w_1 < w_2 < ... < w_m$.
  4) Each edge $\{w_i, w_{i+1}\}$, $1 \leq i < m$, has slope +1 or -1.

• It is possible to compute a canonical ordering in linear time.

• Using a clever updating strategy and appropriate data structures yields running time is $O(n \log n)$. 