Coloring

Algorithms and Networks

Universiteit Utrecht

Graph coloring

- Vertex coloring:
  - Function f: V $\rightarrow$ C, such that for all $\{v,w\} \in E$:
    - $f(v) \neq f(w)$
- Chromatic number of G: $\chi(G)$: minimum size of C such that there is a vertex coloring to C.
- Vertex coloring problem:
  - Given: graph G, integer k
  - Question: Is there a vertex coloring of G with k colors?

Set coloring

- Given: graph G=(V,E), set of colors $C(v) \subseteq C$ for all $v \in V$.
- Question: is there a vertex coloring f of G with also for all $v \in V$: $f(v) \in C(v)$?

Set coloring is NP-complete

- In NP: trivial.
- Transform from 3-sat. Take a vertex for each variable, and a vertex for each clause.
- $C(x_i) = \{x_i, \textit{not } x_i\}$
- C(clause) = variables in clause
- Coloring a vertex x_i means setting that variable to the value not the color
  - E.g.: setting x_i to true means coloring x_i with ($\textit{not } x_i$)

Coloring is NP-complete

- Transform from set coloring
- Take instance of set coloring. Let C be set of all colors.
- Add a clique with one vertex per color in C.
- Add an edge between vertex v in original graph and color vertex *c* if $c \notin C(v)$

Famous bounds

- Four color theorem:
  - The chromatic number of a planar graph is at most four.
- A graph is bipartite, if and only if it has chromatic number 2.
- The chromatic number of a graph is at most the maximum degree of a vertex plus 1.
  - Improvement: Brooks' theorem

Heuristics

Greedy coloring

- Take some permutation of the vertices.
- **for** $i = 1$ **to** n **do**
  - Color the ith vertex with a color different from its colored neighbors.

Independent sets

- $k = 1$
- **repeat until** all vertices are colored
  - Find a (maximal) independent set of uncolored vertices
  - Color these with k
  - k++

Simple lemma

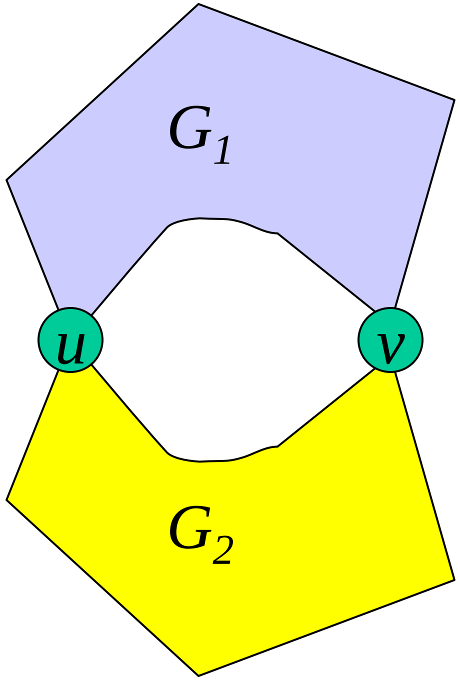
- The chromatic number of a graph is the maximum of the chromatic number of its biconnected components.

Brooks' theorem

- Suppose G is a connected graph, with the maximum vertex degree $\Delta(G)$. Suppose G is not a complete graph or a cycle of odd length. Then $\chi(G) \leq \Delta(G)$.
- Proof:
  - Suppose G is biconnected. (If not, see previous lemma; …)
  - Two cases.

Case 1: There are nonadjacent u and v with $G - u - v$ disconnected

- Write $k = \Delta(G)$.
- Assume $\Delta(G) > 2$.
  - $\Delta(G_1) \leq k$
  - $\Delta(G_2) \leq k$
- Induction: $\chi(G_1) \leq k$; $\chi(G_2) \leq k$.
- Again two cases:
  - …



Two cases

Case 1a. In G_1 and in G_2, we have a *k*-coloring with *u* and *w* different colors.

– Permute the colors in G_2 and form a coloring of G by taking `union' of colorings.

Case 1b. In G_1 every *k*-coloring has the same color for *u* and for *w*.

– *u* and *w* must have degree k-1 in G_1.
– *u* and *w* have degree 1 in G_2.
– There is a *k*-coloring of G_2 with *u* and *w* the same color.
– Permute the colors in G_2 and form coloring of G.

- Case 1c. In G_2 every *k*-coloring has the same color for *u* and for *w*. Similar.

Case 2: For all non-adjacent u, v, $G - u - v$ is connected

- Let v be vertex of maximum degree.
- v must have non-adjacent neighbors u, w. (Why?)
- $G - u - w$ is connected. Choose spanning tree T of $G - u - w$.
- Choose v as root of T.
- Color u and w by 1, and then color the vertices in T in postorder, greedily.
  - k colors are sufficient: vertices except v have an uncolored neighbor; v has two neighbors with same color.

QED

Coloring interval graphs

- Sort vertices with respect to non-decreasing right endpoints
- Greedy coloring with this ordering gives optimal coloring!
  - Number of colors equals maximum clique size

Non-approximability

- Lund, Yannakakis, 1994
- There is an $\varepsilon>0$: Unless P=NP, then no polynomial time algorithm with ratio n^{ε}

Chromatic index

- Edge coloring
  - Problem statement
  - Vizings theorem: G is edge-colorable with $\Delta(G)$ or $\Delta(G)+1$ colors
  - NP-complete to decide which holds of these two options.
  - Edge coloring equals vertex coloring of edge graph

Coloring and Sodoku

- Solving a soduku
  - Model as *precoloring extension problem*
  - Translate to set coloring, or coloring

Networks and graphs are everywhere

- *If you are a carpenter, everything looks like a hammer*

1								6
		6		2		7		
7	8	9	4	5		1		3
			8		7			4
				3				
	9				4	2		1
3	1	2	9	7			4	
	4			1	2		7	8
9		8						