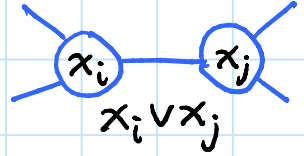


Vertex cover: [L7] [Karp 1972]

- choose k vertices to hit all edges in a graph
- \equiv positive 2SAT with k true variables:

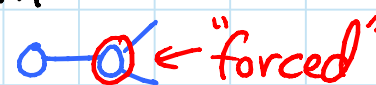
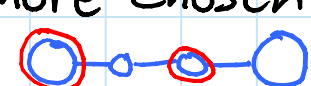



- NP-hard even for planar max-deg.-3 graphs
[Garey & Johnson - SIJAM 1977] [L7] [Lichtenstein 1982]

Polynomial:

- Exact vertex cover: each edge hit exactly once
- Edge cover: choose edges to hit all vertices
(\approx matching)

Connected vertex cover: [Garey & Johnson - SIJAM 1977]

- vertex cover must induce connected graph
- NP-hard for planar max-deg.-4 graphs
- reduction from previous problem
 - never need to choose leaf \Rightarrow  \leftarrow "forced"
 - add forced polygon in each face
 - increase each original vertex degree up to 4
 - subdivided edge needs exactly 1 more chosen (if otherwise a vertex cover) 
 - never useful to choose both
- added exactly $5 \cdot \# \text{ edges} +$  to cover
- these additions induce connected graph & every original vertex is adjacent

Rectilinear Steiner tree:

- given n points in the plane
- connect via horizontal & vertical segments of minimum total length (\Rightarrow tree)
- reduction from previous problem:
 - [Garey & Johnson - SIJAM 1977]
 - draw (max-deg.-4) graph rectilinearly on grid
 - scale by $4n^2$
 - points at all integer points along edges EXCEPT within radius 1 of vertices
 - connections may as well go through vertices
 - every edge must connect to a vertex $\rightarrow 2|E|$
 - must also connect other end of edges in a spanning tree of $G[VC]$ $\rightarrow 2(|V|-1)$

k-coloring: (vertex) AKA Chromatic Number

- given graph & positive integer k
- find color assignment $c: V \rightarrow \{1, 2, \dots, k\}$ such that no edge $\{v, w\}$ is monochromatic
 $c(v) = c(w)$
- like XOR 2SAT but with k -valued logic
 $x_i \neq x_j$
- NP-hard [Karp 1972]
- 2-coloring (bipartiteness) is polynomial
- 3-coloring NP-hard [Garey, Johnson, Stockmeyer - TCS 1976]
 - reduction from 3SAT:
 - colors gadget (3 distinct colors)
 - variable gadget: red & blue
 - clause gadget: red x_i 's force red forward in Δ ; else can put red back
 - reduction to planar 3-coloring:
 - crossover gadget: $x = x'$ & $y = y'$ (center alternates)
 - reduction to planar max. degree 4:
 - high-degree gadget: $x = x' = \dots$ (Δ s force copying)
- polynomial for max. degree 3:
possible \Leftrightarrow not K_4 [Brooks 1941]

Push-1X: can't revisit a square

- reduction from Planar 3-coloring max.-deg-4
[Demaine, Demaine, Hoffmann, O'Rourke - CGTA 2003]
- planar Euler tour (construct by induction)
- choose color whenever visiting a vertex
 - via fork gadget
 - one-ways to "forget" color for next vertex
- equal gadget to force same at a vertex
- nonequal gadget to prevent monochromatic edge
- both reduce to
 - XOR crossover: single-use crossover
→ or ↓
 - NAND: two adjacent paths,
each preventing the other

Push-1G: blocks feel gravity [Friedman 2002]

- mimic same reduction
- one way, fork, XOR crossover
- NAND reduces to XOR crossover!

Graph orientation: [Horiyama, Ito, Nakatsuka, Suzuki, Uehara

- given an undirected 3-regular graph - CCG 2012]
- with 3 vertex types, find a valid orientation
 - 1-in-3: exactly 1 incoming edge, 2 outgoing
 - 2-in-3: exactly 2 incoming edges, 1 incoming
 - 0-or-3: exactly 0 or 3 incoming/outgoing edges
- NP-complete by reduction from 1-in-3 SAT
- in plane, also need crossover gadget

Packing L trominoes: [Horiyama, Ito, Nakatsuka, Suzuki, Uehara] & earlier by [Moore & Robson - DCG 2001]

- given grid polygon, can we pack k \square 's inside?
- exact packing: $k = \text{area}/3$
- NP-hard by reduction from Graph Orientation
 - "double 0-or-3" \approx left 2 or right 2



Packing I trominoes: similar \square

& earlier by [Beaquier, Nivat, Remila, Robson - CGTA 1995]

Linear layout of graph = bijection $f: V \rightarrow \{1, 2, \dots, |V|\}$
- maps edges to segments in 1D
(see survey by Díaz, Petit, Serna 2002)

Bandwidth = minimize length of longest edge

Minimum linear arrangement = minimize total edge length

- e.g. motivated by VLSI layout

Cutwidth = minimize maximum # edges cut by a vertical line

- sum version \equiv min. linear arrangement

Vertex separation = minimize maximum (over x coords.) # vertices on left with edges on right

Sum cut = minimize sum of \uparrow

Edge bisection = minimize # edges crossing middle x

Vertex bisection = minimize # vertices in left half with edges to right half

Betweenness: given triples of the form
"y is between x & z"
 $x < y < z$ or $x > y > z$
find valid linear ordering

[Opatrný -
SICOMP
1979]

Bipartite crossing number: [Garey & Johnson - SIADM 1983]

minimum # crossings in bipartite graph drawing with 2 sides on 2 parallel lines

- reduction from Minimum Linear Arrangement

- $|E|^2$ edges between top_i & bottom_i

⇒ top order = bottom order if $< |E|^4$ crossings

- edge $\{v_i, v_j\}$ → edge $\{\text{top}_i, \text{bottom}_j\}$

- # crossings = $|E|^2 (k - |E| + 1) - 1$

OPT MLA ↑ ↑ -1 per edge → $< |E|^2$ for edge crossings

Crossing number:

[Garey & Johnson - SIADM 1983]

draw graph with min. # crossings

- reduction from previous problem

- $3k+1$ copies of infrastructure

↳ bipartite # crossings

Rubik's Cube: [Demaine, Demaine, Eisenstat, Lubiw, Winslow - ESA 2011]

- min # moves for $n \times n \times 1$ Rubik "square"

- $\Theta(n^2 / \lg n)$ in worst case

- NP-hard with "don't cares" (missing stickers)

by reduction from Betweenness

- OPEN: all stickers

[Erickson 2010]