Coping with Intractability

If an optimisation problem is NP-hard we generally regard it as intractable

We traditionally cope with this in the following ways

1. Restrict the input
2. Use heuristics
3. Use approximation algorithms
4. Random, fixed parameter, exact algorithms, average case etc

1 Restricting the input

A planar graph is a graph that can be drawn in the plane without edge crossings

Theorem 1 *Every planar graph is 4-colourable*

So 4 colouring is trivial for planar graphs, the answer is always yes

Theorem 2 *3 colouring is NP complete for planar graphs*

Theorem 3 *Every triangle free planar graph is 3 colourable*

Hence 3 colouring is trivial for triangle free planar graphs

2 Using Heuristics

A well know heuristic for colouring is first fit

- Order the vertices of an n-vertex graph G as $v_1, ..., v_n$
- Colour the vertices one by one in that order assigning the smallest available colour
- So, v_i gets the smallest colour x not used on $N(v_i) \cap \{v_1, ..., v_{i-1}\}$ where $N(v_i)$ denotes the neighbourhood of v_i in G
- Every tree is bipartite so can eb coloured with at most 2 colours

3 Approximation algorithms

- An algorithm is a k-approximation if it always finds a solution that is a factor of k within the optimum

3.1 Vertex cover

Here is an approximation algorithm for vertex cover

Listing 1 Approx-Vertex-Cover(G=(V,E))

```
1 C=∅
2 E'=E(G)
3 while E'≠ ∅ do
4     let (u,v) be an arbitrary edge of E'
5     C=C∪{u,v}
6     remove from E' every edge incident with either u or v
7 return C
```

Theorem 4 *The algorithm is a 2 approximation for vertex cover*

Proof

- Let $C = \{u_1, v_1, u_2, v_2, ..., u_p, v_p\}$ be the output, where the edges (u_i, v_i) were chosen in executions of step 3, so $|C| = 2p$
- By construction, $G - C$ has no edges so C is a vertex cover
- Let C^* be a minimum vertex cover of G. As (u_i, v_i) is an edge, at least one of u_i, v_i belongs to C^*. So $|C^*| \geqslant p$. Hence $|C| = 2p \leqslant 2|C^*|$