Treewidth and Courcelle's Theorem

A minor of a graph G is a graph obtained from G by contraction of edges and removal of nodes and edges.

Lemma

Any planar graph is a minor of a grid.

Proof

Simple. For example by Induction.

Treewidth and Courcelle's Theorem

Theorem

Let H be a finite, planar graph and $\mathcal{G}$ a class of graphs, not containing H as a minor.
Then there is a constant c_H, such that the treewidth of any graph in $\mathcal{G}$ is at most c_H.

Proof

very difficult and long

Treewidth and Courcelle's Theorem

Theorem

Let H be a finite, planar graph and $\mathcal{G}$ a class of graphs, not containing H as a minor.
Then there is a constant c_H, such that the treewidth of any graph in $\mathcal{G}$ is at most c_H.

Proof

very difficult and long

Treewidth and Courcelle's Theorem

Corollary

"A graph with large treewidth contains a large grid":

If $tw(G) > t$ then G contains the grid $Q_f(t)$ as minor, where f is a monotone, unbounded function.

Proof

direct consequence of the last theorem

Treewidth and Courcelle's Theorem

Example

Input: A planar graph G and k pairs (s_i, t_i) of nodes from G.
(Parameter is k)

Question: Are there edge disjoint paths connecting each s_i with t_i?

The problem belongs to FPT:

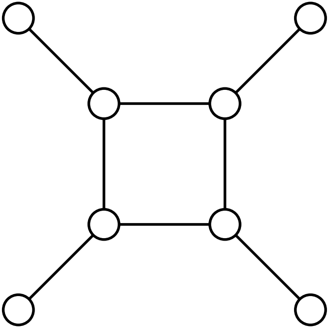
If the treewidth is small, we can apply Courcelle's Theorem.

If the treewidth is large, a large grid is contained as a minor.
Removing a node from this grid does not "harm" this structure.

Color Coding

Problem: Does a given graph contain a cycle of length k?

This problem is NP-complete, because Hamilton Cycle is a special case.



Question: Is it fixed parameter tractable?

Color Coding

Algorithm:

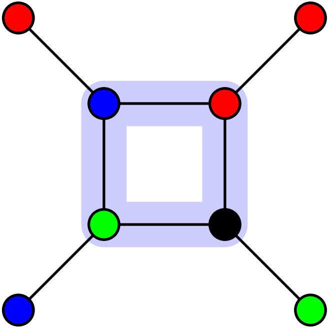
1. Randomly color each node in one of k colors
2. Check for a colorful cycle of length k, i.e., a cycle in which no two nodes have the same color

Analysis:

The probability that a cycle of length k becomes colorful is

$$k!/k^k \sim \sqrt{2\pi k}\, e^{-k}.$$

Color Coding



The cycle is colorful with probability $4!/4^4 = 3/32$.

Color Coding

After using the above algorithm to find a cycle of length k for N times, the probability that it failed to detect a cycle every time is

$$\left(1 - \frac{k!}{k^k}\right)^N.$$

Letting $N = Mk^k/k! \sim Me^k/\sqrt{k}$ yields

$$\left(1 - \frac{M}{N}\right)^N \sim e^{-M}.$$

This failure probability can be made arbitrarily small by the choice of M.

Color Coding

A question remains:

How do you check for a colorful cycle?

Color Coding

Answer:

Create a table $P(u, v, l)$.

$P(u, v, l)$ contains all sets of pairwise distinct nodes that constitute a path from u to v of length l.

$P(u, v, l)$ can be computed from $P(u, v, l - 1)$.

Time required: $2^k \cdot poly(n)$

Color Coding

Definition

A k-perfect family of hash functions is a family $\mathcal{F}$ of functions $\{1, \ldots, n\} \to \{1, \ldots, k\}$ such that for every $S \subseteq \{1, \ldots, n\}$ with $|S| = k$ there exists an $f \in \mathcal{F}$ that is bijective when restricted to S.

Let us first assume we had such a family of perfect hash functions...

Color Coding

Deterministic algorithm:

- Color the graph using each $f \in \mathcal{F}$.
- For each coloring, check for a colorful cycle of length k.

This algorithm works if we can construct a k-perfect family of hash functions.

This algorithm is fast if the family is small, can be expressed in little space, and its functions can be evaluated quickly.

Color Coding

Fortunately, there are k-perfect families of hash functions consisting of no more than $O(1)^k \log n$ functions.

They can be stored compactly.

They can be evaluated quickly: Each $f(i)$ can be computed fast.

That is, there is a deterministic FPT algorithm for finding cycles of length k.