RWTH Aachen | **WS 2018**
Lehrgebiet Theoretische Informatik | Problem Set 6
Dreier—Kuinke—Rossmanith | 26.11.2018

Parameterized Algorithms Tutorial

Tutorial Exercise T1

Design a fixed parameter algorithm for finding an $k \times k$-grid subgraph in a graph that is taken from a graph class with maximal degree d. The parameter is k. Analyse the running time.

Proposed Solution

For every vertex we calculate the $2k$ neighborhood. Fining one neighborhood takes at most time d^{2k}. which is also a bound on its size. We then check if this neighborhood is a $k \times k$ grid in time $d^{2k}\binom{d^{2k}}{k}k!$ (checking for every subset if it contains a grid). We repeat this for every vertex which adds a factor of n.

Tutorial Exercise T2

Give a parameterized algorithm that decides if a graph G contains k many vertex disjoint claws. A claw is a $K_{1,3}$.

Proposed Solution

We solve it by color coding: Give every vertex one of k colors independently uniformly at random. Let G_i be the subgraph of G induced on all vertices with color i. We can find a claw in G_i in $O(n^4)$ by trying all subsets of size 4. If every G_i contains a claw we answer yes, otherwise no.

The probability that k vertex disjoint claws are colored in a way such that the algorithm answers yes, that is one color per claw and all claws different colors, is $p = k!/k^{4k}$. The probability that the algorithm answers wrong $1/p$ times is $(1-p)^{1/p} \leq 1/e$, because $(1 + c/n)^n \leq e^c$ for all c. This means we have a constant failure probability after $O(k^{4k}/k!) = O^*(e^{4k})$ steps.

Tutorial Exercise T3

Design a fixed parameter algorithm for finding a cycle of length *at least* k in an arbitrary graph G.

Proposed Solution

If the treewidth of G is at most k we can solve this problem via Courcelle's Theorem. If the treewidth of G is larger, then it has to contain a large grid as a minor and is automatically a yes-instance.

Homework H1

The Triangle Packing problem is defined as follows: given a graph $G = (V, E)$ and an integer k, decide whether G has k vertex-disjoint 3-cycles. Use the idea of randomly coloring the vertices of G with k colors to enable easy detection of vertex-disjoint triangles. What is the expected running time of your algorithm?

Proposed Solution

Assume that the graph G has a set of k vertex-disjoint triangles. Call a random coloring $c\colon V(G) \to \{1, \ldots, k\}$ of the vertices *proper* if all vertices of a triangle receive the same color but different triangles receive different colors. Given a proper coloring of a yes-instance of the problem, one can find the triangles but searching for them in the graphs induced by the vertices of each color class, in turn. The probability that a coloring is proper is at least

$$\frac{k^{n-3k} \cdot k!}{k^n} = \frac{k!}{k^{3k}} > \frac{1}{k^{3k}} \cdot \left(\frac{k}{e}\right)^k = \left(\frac{1}{k^2 e}\right)^k .$$

The randomized algorithm repeats the following steps in an infinite loop: it randomly colors the vertices of the graph using colors $\{1, \ldots, k\}$; it then checks Since one obtains a proper coloring with probability $(k^2 e)^{-k}$, the expected running time is $O((k^2 e)^k \cdot \text{poly}(n))$.