Can you beat treewidth?

by *Dániel Marx*

1 CSP, graphs, minors, embeddings and treewidth

We say that CSP is *binary* if every constraint is between two variables. For an instance I of binary CSP, we can define the *primal graph* G of instance I as a graph with variables of I as vertices and we connect two variables x and y in G if there is a constraint in I between x and y.

An *embedding* of H into G is a mapping Ψ from $V(H)$ to connected subsets of G such that if $u, v \in V(H)$ are adjacent, then either $\Psi(u) \cap \Psi(v) \neq \emptyset$ or there is an edge connecting a vertex of $\Psi(u)$ and a vertex of $\Psi(v)$. The *depth of the embedding* is at most q if every vertex of G appears in the images of at most q vertices of H. Thus H has an embedding of depth 1 into G if and only if H is a minor of G.

Theorem 1. *There are computable functions $f_1(G), f_2(G)$ and a universal constant c such that for every $k \geq 1$, if G is a graph with $\mathrm{tw}(G) \geq k$ and H is a graph with $|E(H)| = m \geq f_1(G)$ and no isolatedvertices, then H has an embedding into G with depth at most $\lceil cm \log k/k \rceil$. Furthermore, such an embedding can be found in time $f_2(G)m^{O(1)}$.*

Given a graph G and an integer q, we denote by $G^{(q)}$ the graph obtained by replacing every vertex with a clique of size q and replacing every edge with a complete bipartite graph on $q + q$ vertices. It is easy to see that H has an embedding of depth q into G if and only if H is a minor of $G^{(q)}$. We call $G^{(q)}$ a *blow-up* of the graph G.

Lemma 1. *Let G be a graph with $\mathrm{tw}(G) \geq k$. There are universal constants $c_1, c_2 > 0$ such that $L(K_k)^{(\lceil c_1 \log n \rceil)}$ is a minor of $G^{(\lceil c_2 \log n \cdot k \log k \rceil)}$, where n is the number of vertices of G.*

Lemma 2. *For every $k > 1$ there is a constant $n_k = O(k^4)$ such that for every G with $|E(G)| > n_k$ and no isolated vertices, the graph G is a minor of $L(K_k)^{(q)}$ for $q = \lceil 130|E(G)|/k^2 \rceil$. Furthermore, a minor mapping can be found in time polynomial in q and the size of G.*

2 Complexity of binary CSP

Theorem 2. *If there is a $2^{o(m)}$-time algorithm for m-clause 3SAT, then there is a $2^{o(n)}$-time algorithm for n-variable 3SAT.*

Lemma 3. *Given an instance of 3SAT with n variables and m clauses, it is possible to construct in polynomial time an equivalent CSP instance with $n + m$ variables, $3m$ binary constraints, and domain size* 3.

Lemma 4. *Assume that G_1 is a minor of G_2. Given a binary CSP instance I_1 with primal graph G_1 and a minor mapping Ψ from G_1 to G_2, it is possible to construct in polynomial time an equivalent instance I_2 with primal graph G_2 and the same domain.*

Lemma 5. *Given a binary CSP instance $I_1 = (V_1, D_1, C_1)$ with primal graph $G^{(q)}$ (where G has no isolated vertices), it is possible to construct (in time polynomial in the size of the output) an equivalent instance $I_2 = (V_2, D_2, C_2)$ with primal graph G and $|D_2| = |D_1|^q$.*

Theorem 3. *If there is a recursively enumerable class $\mathcal{G}$ of graphs with unbounded treewidth and a function f such that binary CSP(G) can be solved in time $f(G)\|I\|^{o(\mathrm{tw}(G)/\log \mathrm{tw}(G))}$ for instance I with primal graph $G \in \mathcal{G}$, then ETH fails.*