THE TUTTE POLYNOMIAL ON GRAPHS OF BOUNDED CLIQUE-WIDTH

Presenting a subexponential algorithm for a special case of a notoriously hard (#P-complete) graph invariant...

Petr Hliněný

Dept. of Computer Science
FEI, VŠB – TU Ostrava

e-mail: petr.hlineny@vsb.cz
http://www.cs.vsb.cz/hlineny

Joint work with **Omer Gimenez** and **Marc Noy**

Dept. of Applied Mathematics
UPC Barcelona

1 FORESTS IN COGRAPHS

. – the first (simplified) step towards our algorithm. . .

Definition. **Cograph** is a graph constructed from vertices using

- a *disjoint union* (no added edges), or
- a *"complete" union* (adding all edges across).

Cographs have quite long history of research. . .

Fact. (folklore)

- All cliques are cographs.
- Precisely those graphs without induced P_4.
- Cographs are closed on complements, contractions, induced subgraphs.
- Not closed on normal subgraphs / edge deletion.
- Recognizable in P.

1.1 Enumerating Forests

- Enumeration of spanning trees in P - a determinant evaluation.
- Enumeration of spanning forests #P-hard [Jaeger, Vertigan, Welsh, 90].
- Enumeration of spanning forests in P on graphs of bounded tree-width (cf. Tutte polynomial).

Theorem 1.1. *Spanning forests can be enumerated on cographs in time*

$$\exp\left(O(n^{2/3})\right).$$

Note: Subexponential algorithms – $2^{o(n)}$

For NP-complete problems, no better solutions than an exhaustive search are expected to exist.

Hence, for naturally defined problems like the SAT with n variables, no $2^{o(n)}$ algorithm (called often *subexponential*) is expected to exist.

1.2 Algorithm on Cographs

A **forest signature** $\boldsymbol{\alpha}$ – a multiset of component sizes (positive integers);

- represented by a *characteristic vector* $\boldsymbol{\alpha} = (a_1, a_2, \ldots, a_n)$,
- *size* $s_{\boldsymbol{\alpha}} = \sum_{i=1}^{n} i \cdot a_i$ (and cardinality as usual $|\boldsymbol{\alpha}| = \sum_{i=1}^{n} a_i$).

Lemma 1.2. (folklore) *There are* $2^{\Theta(\sqrt{n})}$ *signatures of size* n *(∼integer parts.).*

A **forest double-signature** $\boldsymbol{\beta}$ – a multiset of ordered pairs of integers, counting dual-labeled (nonempty) component sizes;

- a refinement of a forest signature,
- having a *characteristic vector* $\boldsymbol{\beta} = (b_{(0,1)}, b_{(0,2)}, \ldots, b_{(1,0)}, b_{(1,1)}, \ldots)$,
- *size* $s_{\boldsymbol{\beta}} = \sum_{(x,y)} (x+y) \cdot b_{(x,y)}$.

Lemma 1.3. *There are* $\exp\left(\Theta(n^{2/3})\right)$ *distinct double-signatures of size* n*.*

– Quite difficult to prove, but easy a slightly worse bound $\exp\left(\Theta(n^{2/3} \log n)\right)$.

We apply the following two $\exp(O(n^{2/3}))$ algorithms along the decomposition scheme of the given cograph:

Algorithm 1.4. *Combining the spanning forest signature tables of graphs F and G into the one of the disjoint union $H = F \dot\cup G$. (Simple.)*

`Input:` Graphs F, G, and their forest signature tables $\boldsymbol{T}_F, \boldsymbol{T}_G$.

`Output:` The forest signature table $\boldsymbol{T}_H$ of $H = F \dot\cup G$.

`create` *empty table* $\boldsymbol{T}_H$ *of forest signatures of size* $|V(H)|$;
`for` *all signatures* $\boldsymbol{\alpha}_F \in \Sigma_F$, $\boldsymbol{\alpha}_G \in \Sigma_G$ `do` $\qquad \exp(O(n^{2/3}))\times$
 `set` $\boldsymbol{\alpha} = \boldsymbol{\alpha}_F \uplus \boldsymbol{\alpha}_G$ *(a multiset union)*;
 `add` $\boldsymbol{T}_H[\boldsymbol{\alpha}] += \boldsymbol{T}_F[\boldsymbol{\alpha}_F] \cdot \boldsymbol{T}_G[\boldsymbol{\alpha}_G]$;
`done.`

Algorithm 1.5. *Combining the spanning forest signature tables of graphs F and G into the one of the complete union $H = F \oplus G$. (Difficult.)*

`Input:` Graphs F, G, and their forest signature tables $\boldsymbol{T}_F, \boldsymbol{T}_G$.

`Output:` The forest signature table $\boldsymbol{T}_H$ of $H = F \oplus G$.

`create` *empty table* $\boldsymbol{T}_H$ *of forest signatures of size* $|V(H)|$;

```
for all signatures α_F ∈ Σ_F, α_G ∈ Σ_G do                                  exp(O(n^{2/3}))×
   set z = |V(F)|;
   create empty table X of forest double-signatures of size z;
   set X[double-signature {(a,0) : a ∈ α_F}] = 1;
   for each c ∈ α_G (with repetition) do                                     O(n)×
      create empty table X' of forest double-signatures of size z+c;
      for all double signatures β of size z s.t. X[β] > 0 do                exp(O(n^{2/3}))×
(*)      for all submultisets γ ⊆ β (with repetition) do                    exp(O(n^{2/3}))×
            set d_1 = Σ_{(x,y)∈γ} x, d_2 = Σ_{(x,y)∈γ} y;
            set double-signature β' = (β − γ) ⊎ {(d_1, d_2 + c)};
            add X'[β'] += X[β] · Π_{(x,y)∈γ} cx;                            O(n)
         done
      done
      copy X = X', z = z + c; dispose X';
   done
   for all double-signatures β of size |V(H)| do                            exp(O(n^{2/3}))×
      set signature α_0 = {x + y : (x,y) ∈ β};
      add T_H[α_0] += X[β] · T_F[α_F] · T_G[α_G];
   done
done.
```

2 THE TUTTE POLYNOMIAL

Definition. For a graph $G = (V, E)$,

$$T(G; x, y) = \sum_{F \subseteq E} (x-1)^{r(E)-r(F)} (y-1)^{|F|-r(F)},$$

where $r(F) = |V| - k(F)$ and $k(F)$ is the num. of components induc. by (V, F).

Fact. (folklore)

- $T(G; 1, 1) = \#$ spanning trees,
- $T(G; 2, 1) = \#$ spanning forests,
- $T(G; 1-x, 0) \cdot * =$ the chromatic polynomial,
- $T(G; 0, 1-y) \cdot * =$ the flow polynomial.

Fact. Knowing $T(G; x, y)$ $\sim$ knowing the number of spanning subgraphs on edges F with $|F| = i$ and $k(F) = j$.

2.1 Computing the Tutte Polynomial

Theorem 2.1. (Jaeger, Vertigan, and Welsh, 1990)
Evaluating the Tutte polynomial $T(G; x, y)$ at $(x, y) = (a, b)$ is $\#P$-hard unless $(a-1)(b-1) = 1$ or $(a, b) \in \{(1, 1), (-1, -1), (0, -1), (-1, 0), (i, -i), (-i, i), (j, j^2), (j^2, j)\}$, where $i^2 = -1$ and $j = e^{2\pi i/3}$.

Theorem 2.2. (Andrzejak / Noble, 1998)
The Tutte polynomial $T(G; x, y)$ can be computed in polynomial time on a graph G of bounded tree-width.

(The version of Noble gives an FPT algorithm...)

Fact. A subexp. $2^{o(n)}$ algorithm for the Tutte polynomial on an n-vertex graph
$\longrightarrow$ a $2^{o(n)}$ algorithm for 3-colouring,
$\longrightarrow$ a $2^{o(n)}$ algorithm for 3-SAT – unexpected!

So it is very unlikely to have a subexponential algorithm for the Tutte polynomial on general graphs...

Theorem 2.3. *The Tutte polynomial of a cograph can be computed in time*

$$\exp\left(O(n^{2/3})\right).$$

2.2 Extending the Algorithm

Extending Algorithms 1.4,1.5 for the Tutte polynomial is not difficult...

Extensions:

- Enumerate edge-subsets (spanning subgraphs) instead of forests.
- *Subgraph signatures* analogously record the component sizes.

  Moreover, we record the total number of edges.
- When joining components, we may add many (≥ 1) edges between two components, $\rightarrow$ computing "cellular selections".

Definition. *Cellular selection* from $C_1, \ldots, C_k$:
Selecting an ℓ-element subset $L \subseteq C_1 \cup \ldots C_k$, st. $L \cap C_i \neq \emptyset$ for all i.

A nice exercise:

Let $d_i = |C_i|$, and $u_{i,j}$ be the number of partial selections of j elements from the first i cells. Then

$$u_{i,j} = \sum_{s=1}^{r} u_{i-1,j-s} \cdot \binom{d_i}{s}.$$

3 CLIQUE-WIDTH

- Formal definition [Courcelle, Olariu, 00] (implicit [Courcelle et al, 93]).

Definition. Constructing a vertex-labeled graph G using the operations

- a new labeled vertex,
- a disjoint union of two graphs
- $\rho_{i \to j}$ relabeling of all i's to j's,
- η_{i-j} adding all edges between labels i and j.

(Called a *k-expression*.)

Clique-width $= \min$ number of labels needed to construct (unlabeled) G.

- Cographs have clique-width $= 2$, paths ≤ 3, cycles ≤ 4.
- Bounding the clique-width of a graph allows to efficiently solve all problems expressed in the MSO logic of adjacency graphs (MS$_1$) – quantifying over vertices and their sets. [Courcelle, Makowsky, Rotics, 00]

  (Bounding the tree-width allows to efficiently solve all problems in MS$_2$.)
- The chromatic number (and the chromatic polynomial) is polynomial time (not FPT) for graphs of bounded clique-width. [Kobler, Rotics, 03]

3.1 Algorithm on Bounded Clique-Width

A **subgraph k-signature $\boldsymbol{\beta}$** – a multiset of ordered k-tuples of integers, counting k-labeled (nonempty) component sizes.

(Analogous to double-signatures. . .)

Lemma 3.1. *There are $\exp\left(\Theta(n^{k/(k+1)})\right)$ distinct k-signatures of size n.*

Extending the algorithm – processing the η_{i-j} operation:

- Using only one signature table for the whole graph.
- Thus need an artificial new label 0 for iterative processing of components intersecting label j (corresp. to the sign. table of the second graph).
- A new (easy) point of adding edges inside a component.

Our main result:

Theorem 3.2. *Let G be a graph with n vertices of clique-width $\leq k$ along with a k-expression for G as an input. Then the Tutte polynomial of G can be computed in time*

$$\exp\left(O(n^{1-\frac{1}{k+2}})\right) .$$

3.2 Final Remarks

- Our signature table actually gives more – the so called *U polynomial* of G.
- Do we need a k-expression for G?

  Clique-width is difficult to compute.
  However, it is approximable by *rank-width*. [Oum, Seymour, 03]
- Computing rank-width (with an approx. decomposition) is FPT. [Oum]
  Best asympt. $O(n^3)$ for fixed k. [Oum, 05] via matroid branch-width [PH,02]

Questions

- Is the Tutte polynomial on graphs of bounded clique-width in P, or #P-hard, or between?

  (#P-hardness is not yet excluded by a subexp. algorithm!)
- Is the chromatic number FPT wrt. clique-width?
  (i.e. polynomial with a fixed exponent?)