The Complexity of Computing the Sign of the Tutte Polynomial

Leslie Ann Goldberg

(based on joint work with Mark Jerrum)

Oxford Algorithms Workshop, October 2012

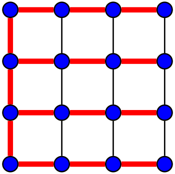
The Tutte polynomial of a graph $G = (V, E)$

$$T(G; x, y) = \sum_{A \subseteq E} (x-1)^{\kappa(V,A) - \kappa(V,E)} (y-1)^{|A| - |V| + \kappa(V,A)}$$

$\kappa(V, A) =$ number of connected components of the graph (V, A)

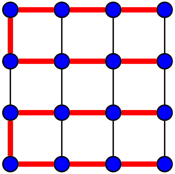
$$T(G;x,y) = \sum_{A\subseteq E} (x-1)^{\kappa(V,A)-\kappa(V,E)}(y-1)^{|A|-(|V|-\kappa(V,A))}$$

If G is connected, $T(G;1,1)$ counts spanning trees.

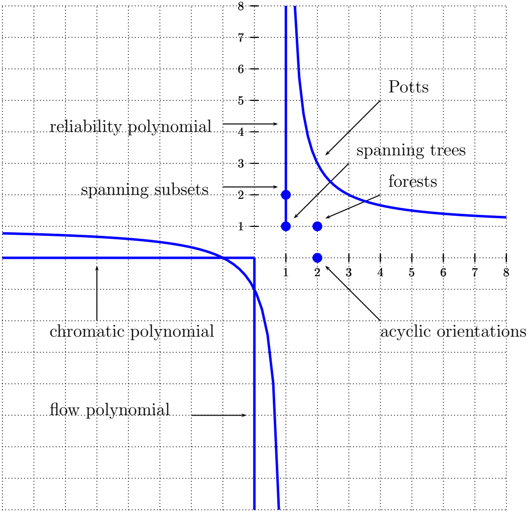


$$T(G;x,y) = \sum_{A \subseteq E} (x-1)^{\kappa(V,A)-\kappa(V,E)} (y-1)^{|A|-(|V|-\kappa(V,A))}$$

If G is connected, $T(G;2,1)$ counts forests.



Combinatorial interpretation of the Tutte polynomial



Partition function of the q-state Potts model at $(x-1)(y-1) = q$

Complexity of evaluating the Tutte polynomial

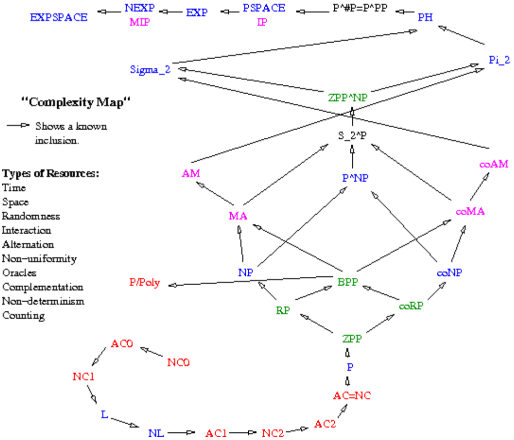
For fixed rationals x and y, Jaeger, Vertigan and Welsh (1990) studied the complexity of exactly evaluating $T(G; x, y)$, given an input graph G. They showed that for every pair (x, y), this problem is either in FP or #P-hard.

FP: There is a polynomial-time algorithm.

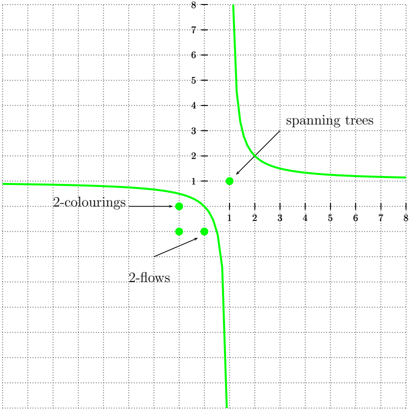
NP-hard: This problem is as difficult as determining whether a Boolean formula has a satisfying assignment.

#P-hard: This problem is as difficult as counting the satisfying assignments of a Boolean formula.

Complexity Class Inclusions (courtesy of Jin-Yi Cai's Theory Reading Group

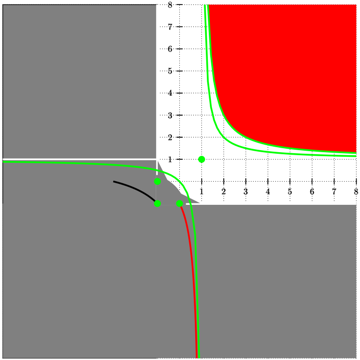


Complexity of evaluating the Tutte polynomial: Jaeger, Vertigan and Welsh



$(x-1)(y-1) = 1.$

Approximate computation



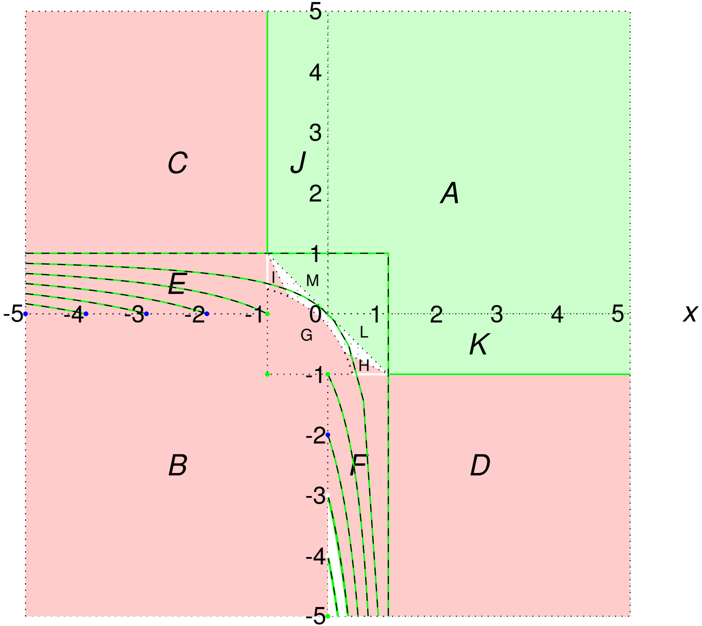
grey points: Approximate evaluation is NP-hard.

red points: Approximate evaluation is hard subject to stronger complexity assumptions.

on the black hyperbola segment: Approximate evaluation is #P-hard.

- For most of these NP-hard points (and more), approximate evaluation is #P-hard. (red points on the next slide)
- It is #P-hard for a very simple reason — determining the sign of the polynomial (whether the evaluation of the polynomial is positive, negative, or zero) is #P-hard.
- The sign of the polynomial is nearly a decision problem (there are only three possible outcomes)

$y = \gamma + 1$



The sign of the chromatic polynomial

- G: an n-vertex graph.
- $P(G; q)$: the unique degree-n polynomial in q such that $P(G; q)$ is the number of proper q-colourings of G.
- For $q \leq 32/27$, the sign of $P(G; q)$ depends upon G in an essentially trivial way.

  Jackson: Suppose $q \in (1, 32/27]$. For every connected graph with $n \geq 2$ vertices and b blocks, $P(G; q)$ is non-zero with sign $(-1)^{n+b-1}$.
- Conjecture (Jackson and Sokal): For any fixed $q > 32/27$, and all sufficiently large n and m, there are 2-connected graphs G with n vertices and m edges that make $P(G; q)$ non-zero with either sign.

How it turns out: computing the sign of the chromatic polynomial

- For $q \leq 32/27$, the sign of $P(G; q)$ is a trivial function of G, which is easily computed.
- At $q = 2$, $P(G; q)$ is the number of 2-colourings of G. The sign of $P(G; q)$ is positive if G is bipartite, and is 0 otherwise. (Not trivial, but easily computed.)
- However, for **every** other fixed $q > 32/27$, computing the sign of $P(G; q)$ is NP-hard.

The picture in more detail (for $q > 32/27$)

- For every fixed non-integer $q > 32/27$, the complexity of computing the sign of $P(G; q)$ is #P-hard.
- For every fixed integer $q > 2$, the problem of computing the sign of $P(G; q)$ is merely NP-complete.

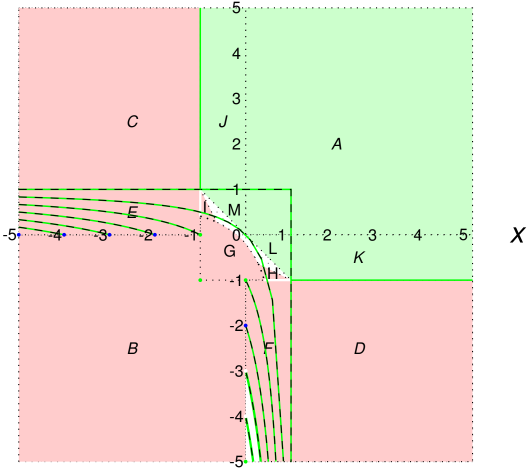
Ramifications for approximate evaluation (for $q > 32/27$)

- If q is not an integer, then an algorithm for approximating $P(G; q)$ would enable one to exactly solve every problem in #P.
- If q is an integer, then $P(G; q)$ can be approximated in polynomial time using an oracle for an NP-predicate.

  This follows from the fact that evaluating $P(G; q)$ is in $\#\mathrm{P}_{\mathbb{Q}}$. (A function in $\#\mathrm{P}_{\mathbb{Q}}$ can be written as a #P-function divided by a polynomial-time computable function.)

The Tutte polynomial

$y = \gamma + 1$



- Computing the sign of the Tutte polynomial is #P-hard at red points
- Computing the sign is in FP at green points.
- Computing the sign is NP-complete at blue points.
- At red points, approximating the Tutte polynomial is also #P-hard.
- At blue and green points, approximation can be done in polynomial time with an NP oracle.

The random cluster formulation

$$q = (x-1)(y-1)$$
$$\gamma = y - 1$$

$$T(G; x, y) = \text{easy-to-compute factors} \times Z(G; q, \gamma)$$

$$Z(G; q, \gamma) = \sum_{A \subseteq E} q^{\kappa(V,A)} \gamma^{|A|}$$

$$P(G; q) = Z(G; q, -1)$$

Name SIGNTUTTE(q, γ).

Instance A graph $G = (V, E)$.

Output Determine whether the sign of $Z(G; q, \gamma)$ is positive, negative, or 0.

The multivariate version

Weight function $\boldsymbol{\gamma} = \{\gamma_e \mid e \in E\}$

$$Z(G; q, \boldsymbol{\gamma}) = \sum_{A \subseteq E} q^{\kappa(V,A)} \prod_{e \in A} \gamma_e.$$

Name $\textsc{SignTutte}(q; \gamma_1, \ldots, \gamma_k)$.

Instance A graph $G = (V, E)$ and a weight function $\boldsymbol{\gamma} : E \to \{\gamma_1, \ldots, \gamma_k\}$.

Output Determine whether the sign of $Z(G; q, \boldsymbol{\gamma})$ is positive, negative, or 0.

A glimpse at the hardness results

Lemma. *Suppose* $q > 1$ *and that* $\gamma_1 \in (-2, -1)$ *and* $\gamma_2 \notin [-2, 0]$. *Then* $\textsc{SignTutte}(q; \gamma_1, \gamma_2)$ *is #P-hard.*

Using the lemma:

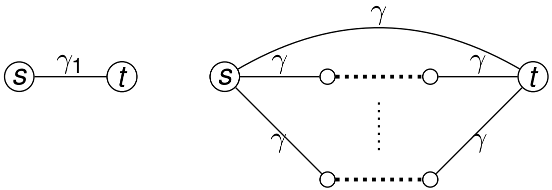
- Suppose that we can "implement" γ_1 and γ_2 from γ.
- Then $\textsc{SignTutte}(q; \gamma)$ is #P-hard.

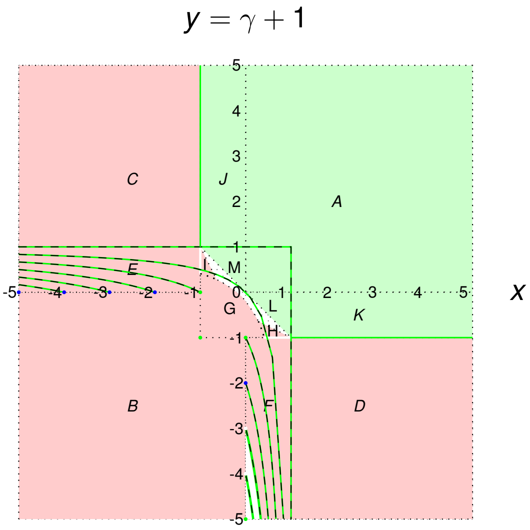
An easy consequence

Lemma. *(Main Lemma) Suppose $q > 1$ and that $\gamma_1 \in (-2, -1)$ and $\gamma_2 \notin [-2, 0]$. Then* SIGNTUTTE$(q; \gamma_1, \gamma_2)$ *is #P-hard.*

Lemma. *(Easy consequence) Suppose (x, y) is a point with $x < -1$ and $y < -1$. Let $q = (x-1)(y-1)$ and $\gamma = y - 1$. Then* SIGNTUTTE$(q; \gamma)$ *is #P-hard.*

Construction: Take $\gamma_2 = \gamma$. Implement γ_1 by taking the parallel composition of γ with lots of copies of a long (odd) series of γs.





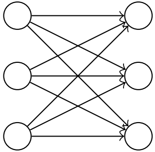
Region E (non-integer q)

- most significant challenge is implementing a γ' with $\gamma' < -1$ when $q > 2$
- implement using K_n minus an edge, where $n = \lfloor q \rfloor + 2$ with edge weights that are very close to -1
- analysis studies chromatic polynomial of K_n and K_n minus an edge.

Region F: Nowhere-zero flows. . .

Nowhere-zero q-flows of a graph $G = (V, E)$

Choose an arbitrary direction for each edge. A nowhere-zero q-flow is a mapping $\psi : E \to \{1, \ldots, q-1\}$ such that the flow into each vertex is equal to the flow out (doing arithmetic mod q).



To see that $K_{3,3}$ has a nowhere-zero 3-flow, direct edges from left to right. Consider the flow in which every edge has label 1.

If q is a positive integer and all edge weights are $-q$, then the Tutte polynomial counts the nowhere-zero q-flows of a graph.

Region F (non-integer q)

key challenge: implement a γ' with $-q < \gamma' < 0$ when $q > 2$.

- Construction for $q \in (3, 4)$: analyse the flow polynomial of the Petersen graph.
- This is zero at $q = 3$ and $q = 4$, since this graph has no nowhere-zero 3-flow or 4-flow.
- This is positive for $q > 4$ (hence negative between 3 and 4).
- On the other hand, the graph obtained by removing an edge has a positive flow polynomial for $q > 3$.
- The fact that the signs of these polynomials are different is key to the construction