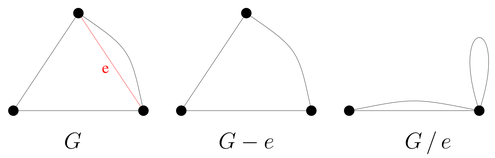
Identities and heuristics for computing the Tutte polynomial.

Michael Monagan Alan Wong

Introduction

Given an undirected graph $G = (V, E)$, we can associate with it a bivariate polynomial $T(G, x, y)$ called the Tutte polynomial which gives information about certain connectivity properties of the graph. It is a graph invariant that is a generalization of the chromatic polynomial, reliability polynomial, and other related graph polynomials. The Tutte polynomial has applications in knot theory, theoretical computer science, statistical physics, and chemistry.

Definition (Tutte [3]). Let G be an undirected graph. Let $e = (u, v)$ be an edge in G. Let $G - e$ denote the graph obtained by deleting e and let $G \,/\, e$ denote the graph obtained by contracting e, that is, first deleting e then joining vertices u and v.



The Tutte polynomial, denoted $T(G, x, y)$, is defined by

$$T(G) = \begin{cases} 1 & \text{if } G \text{ has no edges,} \\ x\,T(G \,/\, e) & \text{if } e \text{ is a cut-edge in } G, \\ y\,T(G - e) & \text{if } e \text{ is a loop in } G \\ T(G - e) + T(G/e) & \text{otherwise.} \end{cases}$$

This definition yields a recursive algorithm to compute $T(G, x, y)$. In general the algorithm will make an exponential amount of calls, as computing the Tutte polynomial is NP-hard. Haggard, Pearce, and Royle [1] developed a main strategy for reducing the computation time of the Tutte polynomial, which is to remember graphs that have already been handled in the computation tree, and using that information if an identical and/or isomorphic graph is encountered later. The question then becomes finding ways to select the edges for deletion/contraction to maximize the isomorphic graphs seen, as high in the computation tree as possible.

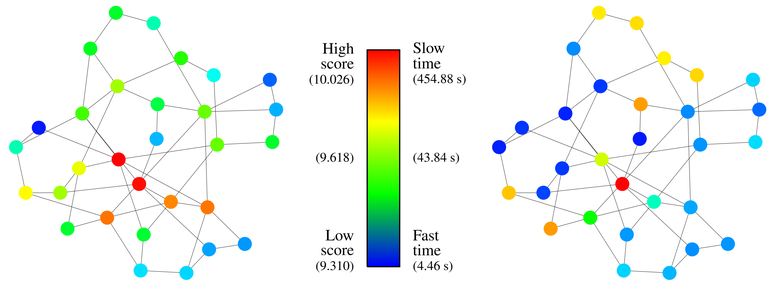
In [2], the first author found heuristics that significantly improved the time needed to compute the Tutte polynomial of sparse graphs. These heuristics include finding a short-arc (SHARC) ordering of the vertices that the algorithm follows to choose the next edge, an edge contraction scheme called VORDER-push which lowers the average degree of the next vertex in the ordering, and using a canonical graph representation that eliminates the need for an explicit isomorphism test. Implementing these features in Maple, he was able to find polynomial time algorithms for certain families of graphs. The research presented here is an extension of his work.

Maplesoft
Mathematics • Modeling • Simulation

cecm Centre for experimental & constructive mathematics

Heuristics

We noticed that graphs with higher average degree and higher girth seem to be more difficult to compute, so we modified the ordering to start in a region of the graph that is relatively sparse and has relatively short cycles. For each vertex, we ran a SHARC ordering starting at that vertex, and then computed a score for that vertex based on the degree sequence and nearby cycle/arc lengths of its ordering. Demonstrated below is a comparison of vertex score with the computation time of the Tutte polynomial given a SHARC ordering starting at that vertex, for a random biconnected graph on 30 vertices and 50 edges.



Vertex scores

Computation times for SHARC order starting at vertex

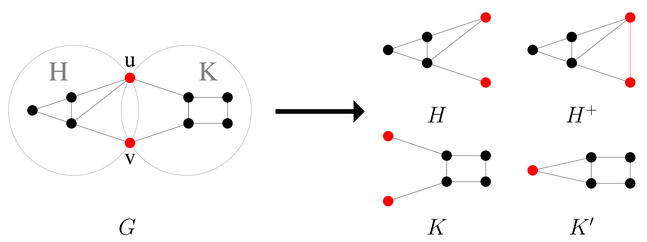
After selecting the vertex with the lowest score, we run the SHARC order again, with the modification that if at one step we must choose between arcs of the same length, we will pick the one with the lowest average score. We call this new ordering ModSHARC.

Identities

Further improvements can be made by finding identities that relate the Tutte polynomial of a graph and its subgraphs, giving us a natural way to parallelize the algorithm. For example, consider this theorem from Tutte:

Theorem ([3]). Let G be a graph with with m blocks $B_1, B_2, \ldots B_m$. Then $T(G, x, y) = \prod_{i=1}^{m} T(B_i, x, y)$.

We developed a similar idea for triconnected components. Shown below is an undirected, biconnected graph G with a 2-separation (H, K), separation pair u, v such that H, K are connected. Let H^+ be the graph obtained from H by adding the edge uv to H, and let K' be the graph obtained from K by identifying u, v.



Theorem 1. Let $A(H) = ((1-x)T(H^+, x, y) + xT(H, x, y))/(x + y - xy)$ and $B(H) = (T(H^+, x, y) - yA(H))/x$. Then $T(G, x, y) = A(H)T(K', x, y) + B(H)T(K, x, y)$.

Benchmarks

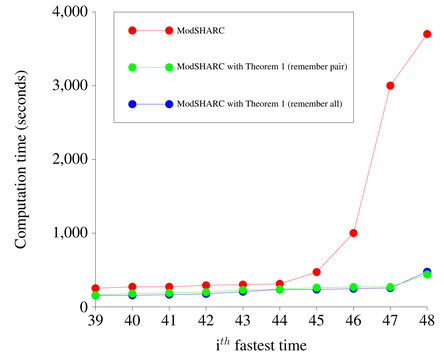
We implemented the new ordering scheme in Maple, and computed the Tutte polynomial of 200 random biconnected graphs for each pair n, m (number of vertices/edges) in the table below. The data was collected in 8 batches of 25 graphs (simultaneously run unless indicated by parentheses) on a 2 x 8 core 2.2/3.0 GHz Intel Xeon computer with 64 GB RAM.

The first two columns are timings using ModSHARC, one version using only degrees to calculate the vertex score, the other using a combination of degrees and cycle lengths. The last column uses the SHARC ordering as implemented in the current TuttePolynomial function in Maple.

		DegScore Only			CycleDegScore			SHARC		
n	m	avg	med	max	avg	med	max	avg	med	max
24	40	0.89	0.73	3.08	0.91	0.73	4.48	1.95	1.18	14.28
27	45	2.64	1.75	33.26	2.32	1.67	14.85	6.72	4.34	72.09
30	50	7.92	4.08	98.48	8.54	4.89	85.46	32.84	15.09	576.25
33	55	23	11	393	27	12	511	(692	49	58284)
36	60	(956	49	144790)	227	53	9365	(4210	239	340320)
39	65				(2193	182	55095)			

Timings (in seconds) for random graphs with n vertices and m edges using SHARC and ModSHARC orderings.

To demonstrate where Theorem 1 would potentially have the most benefit, we built 48 graphs with a 2-separation between a pair of random biconnected graphs of equal size and computed the Tutte polynomial on a 3.2 GHz Intel Core i7 computer with 64 GB RAM. Shown below are the ten slowest times.



Timings (in seconds) of biconnected graphs with a 2-separation (H, K) where $|V(H)| = |V(K)| = 32, |E(H)| = |E(K)| = 54$.

References

[1] Gary Haggard, David Pearce, and Gordon Royle. Computing Tutte Polynomials. *Transactions on Mathematical Software* **37**:3 (2011) article 24.

[2] Michael Monagan. A new edge selection heuristic for computing the Tutte polynomial of an undirected graph. *Proc. of FPSAC'12*, DTMCS, pp. 839−850.

[3] William Tutte. A contribution to the theory of chromatic polynomials. *Can. J. Math.* **6** (1954) pp. 80−91.