Computational results for Symmetric Chromatic Polynomial of Trees

Zane Smith, Isaac Smith, & Anthony Ciavarella
The Ohio State University

Let $G=(V,E)$ be a finite graph, and let $V=\{v_1,v_2,...,v_n\}$. The Chromatic Symmetric Polynomial is a function of countably many commuting indeterminates, and is defined as:

$$X_G(x_1,x_2,...)=\sum_{\kappa}\prod_{i=1}^{n}x_{\kappa(i)} \quad (1)$$

Where $\kappa:\{1,2,...,n\}\to\mathbb{N}$ is a coloring of G and the sum above is taken over all *proper* colorings. In [1], Stanley provides another form of X_G that we will primarily be using throughout our research. The power-sum functions, $p_m := \sum_{i=1}^{\infty}x_i^m$ provide a basis for the space of symmetric polynomials and X_G can be written as:

$$X_G=\sum_{S\subseteq E}(-1)^{\#S}p_{\pi(S)} \quad (2)$$

Where $\pi(S)=(\pi_1,\pi_2,...,\pi_r)$ is an integer partition of n with each π_j corresponding to the number of vertices in a disjoint component of G after removing every edge not in S. And where $p_{\pi(S)} := p_{\pi_1}p_{\pi_2}...p_{\pi_r}$ is a product of power sum functions.
Also in [1], Stanley both gives the definition above, and the following conjecture:

Conjecture. *X_G distinguishes trees*
That is, for any two non-isomorphic trees, G_1 and G_2, we have $X_{G_1}\neq X_{G_2}$.

It's already known that X_G does not distinguish graphs in general (see Fig. 0).

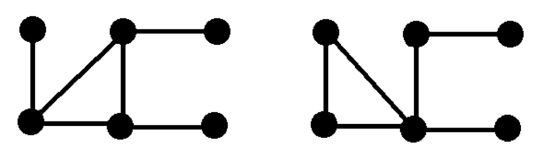

Figure 0: Two non-isomorphic graphs with the same Symmetric Chromatic Polynomial.

The representation given in (2) is much easier for computation, either by hand or by computer. From now on we'll be considering only trees, and we investigate the sets of partitions formed by taking subsets of edges as described above. The main focus of our research was to write a program to search through all trees up to 23 vertices and record the specific ones which have many matching terms of X_G. The number 23 was chosen due to limits in processor speed. Since the number of unique trees grows approximately as 3^n, each increment in size of trees we consider triples all computation times.

Considering the subsets consisting of single elements from E, one receives practically no information about a tree. There are $n-1$ such subsets, since all trees on n vertices have $n-1$ edges, and they all give the partition (2,1,1,...,1). In contrast, there are also $n-1$ subsets of size $n-2$, but they give much more information about the tree. For example, the number of partitions of the form $(1,n-1)$ is exactly the number of leaves, also the partition $(n/2,n/2)$ appears iff there are two centroids.

Subsets of this form can each be thought of as removing a single edge from G, hence we refer to such a subset as a "1-cut" and for convenience we refer to the set $\{\pi(S) : S\subseteq E$ and $|S|=n-2\}$ as the "1-cuts" of a given graph. In general, the "k-cuts" of a graph is the set $\{\pi(S) : S\subseteq E$ and $|S|=n-(k+1)\}$.

Instead of immediately computing the entire X_G for all trees, we approach the problem systematically by checking consecutively larger cuts. First we compute the 1-cuts of all trees, then organize identical trees into subsets. From there we compute 2-cuts just for these trees and once again organize all those with identical 2-cuts. We repeated this process once more with 3-cuts, etc. Finally, there are a few trees on 22 and 23 vertices with non-unique 4-cuts.

The following tables quantify precisely how many trees have non-unique k-cuts for k=1,2,3. The second column is the number of trees on that many vertices with non-unique k-cuts. The third column is the second column divided by the total number of trees. The 4th column is the number of distinct families of trees sharing a set of k-cuts.

vertices	1-cut	density	1-families
7	2	0.182	1
8	6	0.261	3
9	25	0.532	11
10	59	0.557	24
11	178	0.757	60
12	445	0.808	136
13	1154	0.887	274
14	2884	0.913	602
15	7425	0.959	1152
16	18650	0.965	2474
17	47824	0.983	4520
18	122328	0.988	9640
19	316032	0.994	17218
20	819370	0.996	36429
21	2140092	0.998	63813
22	5614634	0.998	135799
23	14817623	0.999	233970

vertices	2-cut	density	2-families
10	2	0.0189	1
11	2	0.00851	1
12	16	0.029	8
13	20	0.0154	10
14	145	0.0459	72
15	193	0.0249	96
16	1035	0.0536	498
17	1524	0.0313	742
18	7562	0.061	3571
19	11765	0.037	5643
20	54157	0.0658	25210
21	89294	0.0416	42363
22	387625	0.0689	178161

vertices	3-cut	density	3-families
16	2	0.000104	1
17	2	0.000041	1
18	22	0.000178	11
19	24	0.000076	12
20	186	0.000226	92
21	216	0.000101	107
22	1348	0.00024	667

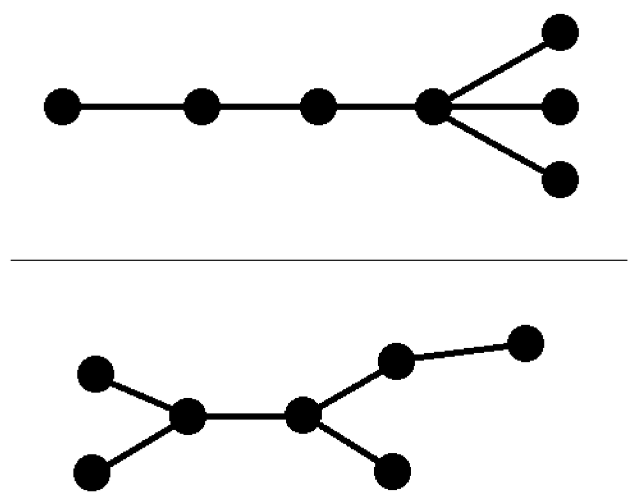

Figure 1: The smallest pair of trees with identical 1-cuts.

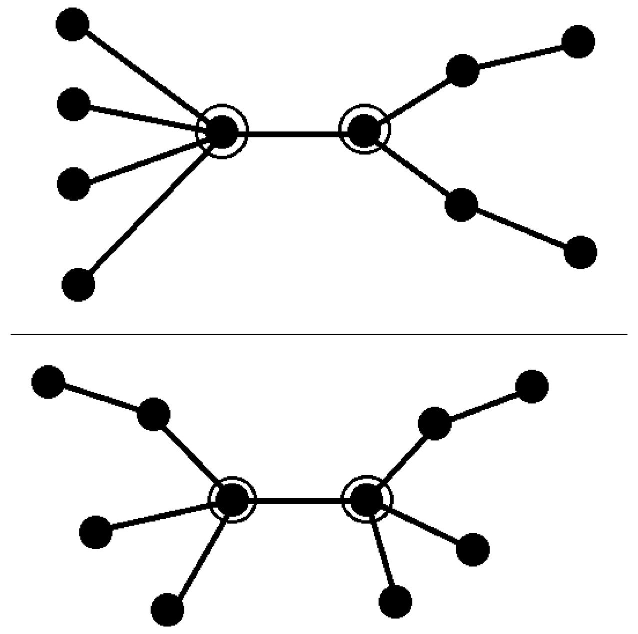

Figure 2: The smallest pair of trees with identical 2-cuts. Circled vertices are the centroids.

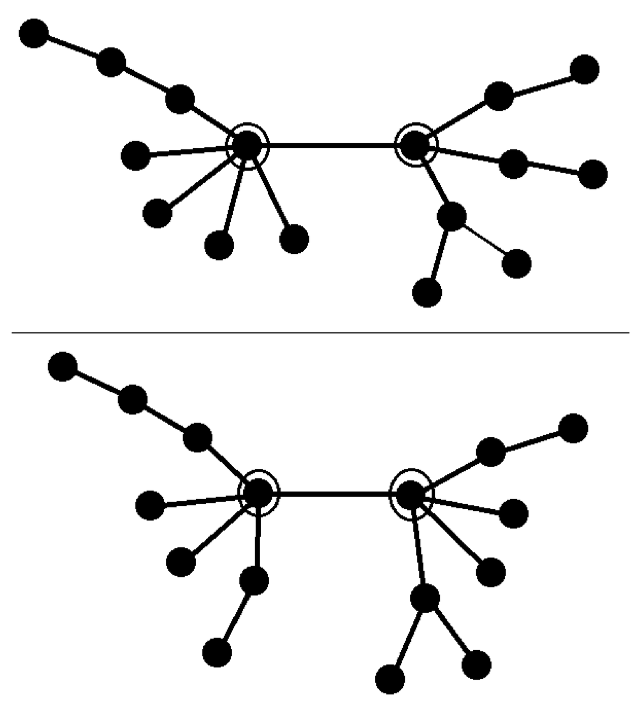

Figure 3: The smallest pair of trees with identical 3-cuts. Circled vertices are the centroids.

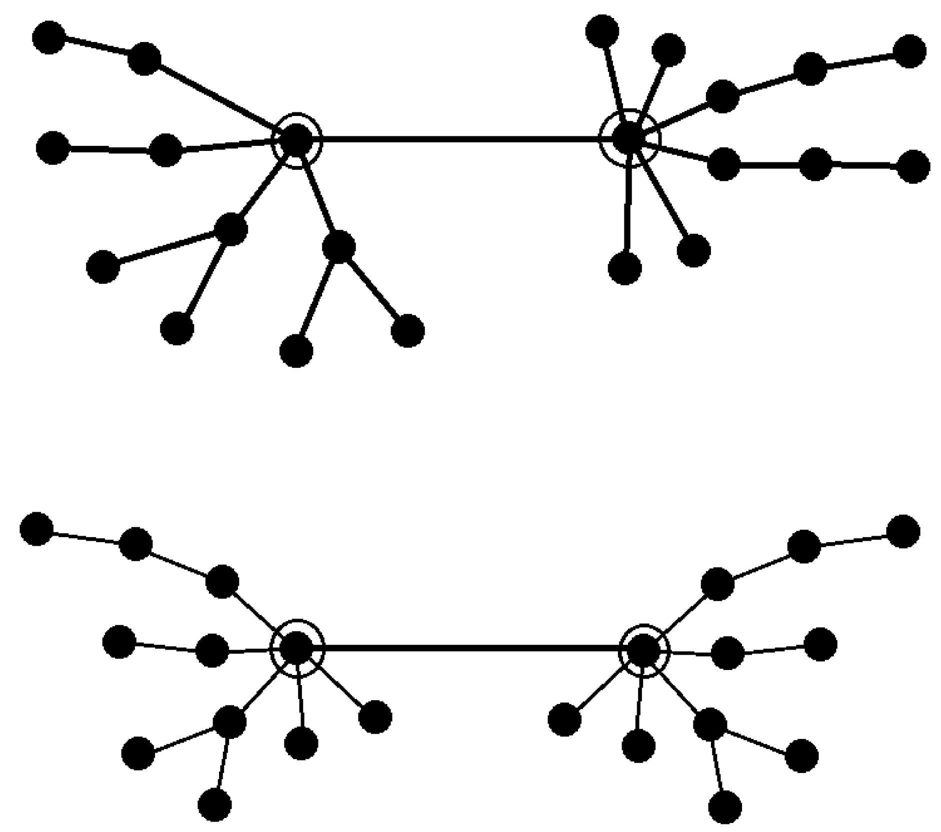

Figure 4: The smallest pair of trees with identical 4-cuts. Circled vertices are the centroids.

A Centroid of a tree is a vertex of minimal weight. The weight of a vertex is the number of vertices on its largest branch. For example, in fig. 3 the two centroids have weight 8. Generally speaking, when two trees have the same 2-cuts or 3-cuts one of the pair can be obtained by permuting some subset of branches about the centroids of the other. We don't yet have a method for determining which of these branch permutations (if any) will give a new tree with identical 2-cuts or 3-cuts.

Another interesting result we've gained from this data is the "belayed exponential growth" of the numbers of trees with non-unique 2(or 3)-cuts. Going from an even to odd order, the total number of non-unique trees grows relatively little. However, going from an odd to an even order there's a much larger increase. This is most likely related to the fact that only trees with an even order can have two centroids, which have a central edge connecting them. Such a central edge makes it more likely for there to be valid branch permutations as described above.

In addition, every pair of two-centroid trees on n vertices with identical cuts can be extended to a pair of n+1 vertex trees with identical cuts simply by splitting the central edge and placing a vertex in the middle. Hence, when going from an even to an odd order the only "new" trees with non-identical cuts are those whose branch permutations were only made valid through the addition of such a central vertex (since otherwise they'd have shown up among the previous trees with an even order).

[1] Richard Stanley, "A symmetric function generalization of the chromatic polynomial of a graph," Advances in Math. 111 (1995), 166-194.