References

[1] R. Karp. Complexity of computer computations, 1972. In *Reducibility among Combinatorial Problems*, Plenum Press.

[2] B. Loerinc. Computing chromatic polynomials for special families of graphs. Technical Report Courant Computer Science Report 19, Courant Institute of Mathematical Sciences NYU, February 1980.

[3] R. C. Read. An introduction to chromatic polynomials. *Journal of Combinatorial Theory*, 4:52–71, 1968.

[4] S. Skiena. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica.* Addison-Wesley, 1990.

[5] A. Satyanarayana; R. Tindell. Chromatic polynomials and network reliability. *Discrete Mathematics*, 67:57–79, 1987.

[6] R. C. Read; W. T. Tutte. Chromatic polynomials, 1988. In *Selected Topics in Graph Theory v.3*, Beineke and Wilson eds.

[7] H. Whitney. A logical expansion in mathematics. *Bull. Am. Math. Soc.*, 38:572–579, 1932.

4 Preliminary Results

The algorithm in *Combinatorica* makes use of both approaches by deciding which is appropriate at every stage of the decomposition of a graph. By altering the routines so that each approach (dense and sparse) could be used exclusively in decomposing a graph we obtained simple timing data on small graphs which suggest that the optimal point at which to switch from a sparse algorithm to a dense one occurs sooner than $\binom{V}{2}/2$. In some cases the difference between the two algorithms was a factor of 6 in favor of the dense approach on a graph that supposedly favored the sparse algorithm. So far the experiments have been simple and have involved graphs of up to only 12 vertices.

5 Current Work

Due to the slowness of *Mathematica* we are rewriting sparse and dense routines in C++ which should be significantly faster. In addition to avoiding obviously unnecessary work in both approaches (e.g. making an empty G_i complete or a complete G_i empty) the routines will enable detailed profiling of the algorithms. The goal of profiling is to make possible more sophisticated comparisons of the two approaches. Understanding the two approaches in terms of their combinatorial complexity will better inform future implementation decisions (e.g. choice of data structure). A study of combinatorial complexity issues has an added advantage over trivial timings since the results are machine independent. Decisions about when best to use one approach or the other should be based on combinatorial complexity and not simply on time.

6 Future Work

The insight gained through experiments should help us in analysis which will lead to a deeper understanding of the trade-off point if not the exact value. Once the trade-off point is established the two approaches can be recombined optimally and further enhancements can be implemented. Ideally the increase in efficiency will allow calculation of chromatic polynomials for much larger graphs and hence afford a closer examination of their properties[6][2]. A published, flexibly indexed database of chromatic polynomials would be of value to chromatic polynomial theory specifically and graph theory generally. We are currently searching for more applications of chromatic polynomials besides that of T.V. channel allocation [3], and network reliability [5]. We are in the beginning stages of this project and would appreciate any suggestions or references relating to our work.

3 Example

In graph G (fig. 1) A and B are not connected and so there is ambiguity concerning their coloring. They may be colored the same or differently. To remove this ambiguity we decompose G into 2 new graphs by adding an edge and by contracting the corresponding vertices.

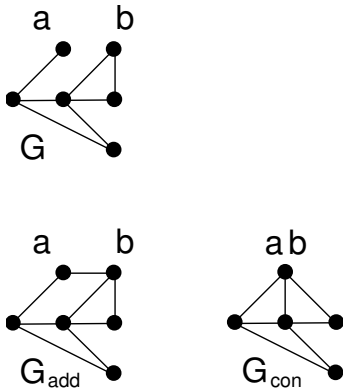

figure 1

In graph G_{add} vertices A and B must now be colored differently while in graph G_{con} vertex AB receives one color corresponding to A and B receiving the same color in G. Note that all other relationships in G are preserved in both new graphs. $P_G(k)$ is now equal to $P_{G_{add}} + P_{G_{con}}$. By repeating the decomposition process on the two new graphs, (fig. 2) and on their new graphs, *etc.*, the process eventually terminates with complete graphs whose $P(k)$'s are summed to yield $P_G(k)$.

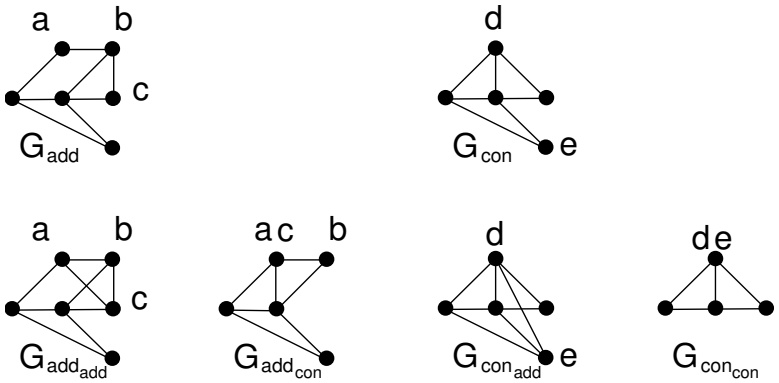

figure 2

Alternately an edge in G may be deleted and the corresponding vertices contracted. This process terminates in a series of empty graphs. Either method applied to G above yields the chromatic polynomial:

$$P(k) = k^6 - 7k^5 + 19k^4 - 25k^3 + 16k^2 - 4k$$

Calculating Chromatic Polynomials

Paul Kehle
pkehle@cs.indiana.edu

Greg Shannon

January 1993

Extended Abstract

1 Abstract

The chromatic polynomial P(k) of a graph G is a function which returns the number of different ways G can be colored given k colors. For an arbitrary graph G which is neither complete nor empty the calculation of $P(k)$ is NP-hard.[2][1] We conduct experiments to gain insight about the best point at which to switch from a "sparse algorithm" to a "dense algorithm". We seek next to generate and catalogue chromatic polynomials for all graphs up to 10 vertices (and other graph families) to support research into their properties, and to gain insight into the fine tuning of chromatic polynomial algorithms.

2 Background

$P(k)$ is very easy to calculate for both empty and complete graphs. For empty graphs $P(k) = k^n$ where n is the number of vertices in the graph. For complete graphs $P(k) = k(k-1)(k-2)\cdots(k-n+1)$. However, for an arbitrary graph G which is neither complete nor empty the problem is much worse. An arbitrary graph G must first be decomposed into a series of either complete or empty graphs G_i, $P_G(k)$ is then a linear combination of the $P_{G_i}(k)$'s.[7] The decomposition is recursive so the number of G_i's grows exponentially in the worst case. On dense graphs it is faster to decompose G into complete graphs, while on sparse graphs decomposition into empty graphs is more efficient. *Combinatorica* [4] (by Steven Skiena at SUNY Stony Brook) currently makes this decision based on the density of the graph. It decomposes G into complete graphs if:

$$E \geq \frac{1}{2}\begin{pmatrix} V \\ 2 \end{pmatrix}$$

where E is the number of edges in G, and V is the number of vertices. Otherwise the graph is decomposed into empty graphs. This switch-over value was suggested by Loerinc [2] on the basis of analysis of small graphs. Loerinc encourages experiments with larger graphs to verify the usefulness of this value.