Approximating the Chromatic Polynomial of a Graph*

Nai-Wei Lin
Department of Computer Science
The University of Arizona
Tucson, AZ 85721
naiwei@cs.arizona.edu

Abstract

The problem of computing the chromatic polynomial of a graph is #P-hard. This paper presents an approximation algorithm for computing the chromatic polynomial of a graph. This algorithm has time complexity $O(n^2 \log n + nm^2)$ for a graph with n vertices and m edges. This paper also shows that the problem of computing the chromatic polynomial of a chordal graph can be solved in polynomial time. Knowledge about the chromatic polynomial of graphs can be employed to improve the performance of logic programs and deductive databases.

1 Introduction

This paper considers finite graphs without loops (i.e., edges joining a vertex to itself) and multiple edges between any pair of vertices. The *chromatic number* of a graph G, written as $\chi(G)$, is the minimum number of colors necessary to color G such that no adjacent vertices have the same color. The *chromatic polynomial* of a graph G, denoted by $C(G, x)$, is a polynomial in x representing the number of different ways in which G can be colored by using at most x colors.

The problem of k-colorability of a graph G is the one of deciding whether the value $C(G, k) > 0$. Since the problem of deciding whether the value $C(G, k) > 0$ is NP-complete [7], the problem of computing the value $C(G, k)$ is #P-complete [12]. The more general problem of computing the chromatic polynomial of a graph is therefore #P-hard.

This paper presents an approximation algorithm for computing the chromatic polynomial of a graph. This algorithm is based on the *greedy* method. We first determine an ordering on the vertices of the graph using some *heuristics*. According to the determined ordering, we

*This work was supported in part by the National Science Foundation under grant number CCR-8901283.

next derive an upper bound and a lower bound on how many different ways each vertex can be colored. The product of the upper bounds and the product of the lower bounds for all the vertices in the graph then give, respectively, an upper bound and a lower bound on the total number of different ways the entire graph can be colored. If the number of available colors is given as a symbolic variable, then the two products are polynomials in this variable. Finally, we take a mean of these two polynomials as an approximation of the chromatic polynomial of the graph. This algorithm has time complexity $O(n^2 \log n + nm^2)$ for a graph with n vertices and m edges.

A graph is called *chordal* if every cycle of length greater than 3 has an edge joining two nonconsecutive vertices of the cycle. Chordal graphs arise in many contexts and contain the following families of graphs: interval graphs, cactus graphs, adjoint graphs of cactus graphs, and so on [6]. Gavril has shown that problems of finding a minimum coloring, a minimum covering by cliques, a maximum clique, and a maximum independent set, of a chordal graph can be solved in polynomial time [5]. In this paper we show that the problem of computing the chromatic polynomial of a chordal graph can also be solved in polynomial time.

Many problems in areas such as operations research and artificial intelligence require enumerating all the solutions that satisfy a set of binary equality or disequality constraints on variables ranging over a finite domain of values. The constraints are of the form $x = y$ or $x \neq y$. These constraint satisfaction problems can be reduced to the graph coloring problem as follows. Each vertex in the graph corresponds to a variable in the constraints, each edge in the graph corresponds to a disequality constraint, and the edges of two vertices are merged if there is an equality constraint between the corresponding two variables. Hence, each coloring of the graph corresponds to a solution to the original constraint satisfaction problem. Accordingly, the associated counting problem for each such constraint satisfaction problem reduces to the computation of the chromatic polynomial of its corresponding graph.

This class of constraint satisfaction problems is frequently realized in the settings such as logic programs and deductive databases, where it is easy to generate multiple solutions of a problem if required. The ability to estimate the chromatic polynomial of a graph allows us to estimate the number of solutions generated by the procedures realizing these constraint satisfaction problems. Information about the number of solutions generated by procedures has been used to estimate the cost of executing procedures, and the cost information has been further used to optimize programs and queries in these settings. For example, this information has been employed for query optimization in deductive databases by appropriately rearranging the evaluation order of subgoals [10, 13], and for process granularity control in parallel logic programming systems by properly preventing small-grain processes from spawning [3].

The remainder of this paper is organized as follows. Section 2 derives a number of upper and lower bounds on the chromatic polynomial of a graph. Section 3 discusses how the orderings on the vertices of a graph affect the bounds on its chromatic polynomial. Section 4 presents

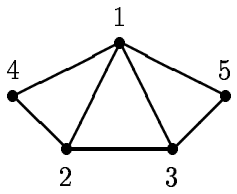

Figure 1: An example

an approximation algorithm for computing the chromatic polynomial of a graph based on an ordering proposed in Section 3. Section 5 shows some experimental results on the performance behavior of the algorithm. Finally, Section 6 concludes this paper.

2 Bounds on Chromatic Polynomials

Since the value of the chromatic polynomial $C(G,x)$ is always nonnegative, for any graph G and natural number x, we shall assume that any polynomial $P(x)$ really means the function $max(P(x),0)$. Furthermore, for any two polynomials $P_1(x)$ and $P_2(x)$, we define $P_1(x) \leq P_2(x)$ if and only if $max(P_1(x),0) \leq max(P_2(x),0)$, for all natural numbers x. We shall derive bounds on the chromatic polynomial of a graph based on the greedy method.

Let $G = (V,E)$ be a graph. The *order* of G, denoted by $|G|$, is the number of vertices in G. Let U be a subset of V. The subgraph of G *induced* by U is a graph $H = (U,F)$ such that F consists of all the edges in E both of whose vertices belong to U. The *neighbors*, $N_G(v) = \{w \in V | (v,w) \in E\}$, of a vertex v in G is the set of vertices adjacent to v. The *adjacency graph*, $Adj_G(v)$, of v is the subgraph of G induced by $N_G(v)$.

Let $G = (V,E)$ be a graph of order n and $\omega = v_1, \ldots, v_n$ be an ordering on V. We define two sequences of subgraphs of G according to ω. The first is a sequence of subgraphs $G_1(\omega), \ldots, G_n(\omega)$, called *accumulating subgraphs*, where $G_i(\omega)$ is the subgraph induced by $V_i = \{v_1, \ldots, v_i\}$, for $1 \leq i \leq n$. The second is a sequence of subgraphs $G'_2(\omega), \ldots, G'_n(\omega)$, called *interfacing subgraphs*, where $G'_i(\omega)$ is the adjacency graph $Adj_{G_{i-1}(\omega)}(v_i)$, for $1 < i \leq n$. For example, consider the graph shown in Figure 1. The imposed ordering is denoted by the labels of vertices. The corresponding accumulating subgraphs and interfacing subgraphs are shown in Figure 2.

The following proposition gives an upper bound and a lower bound on the chromatic polynomial of a graph in terms of, respectively, the chromatic number and the order of interfacing subgraphs.

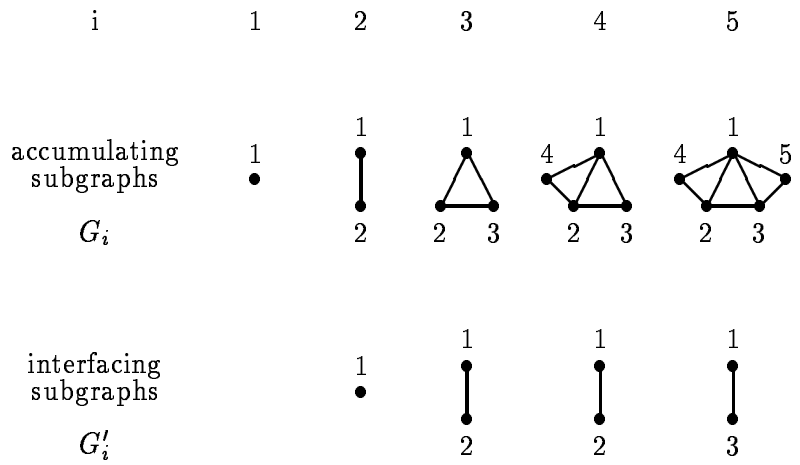

Figure 2: The accumulating and interfacing subgraphs of the graph in Figure 1

Proposition 2.1 *Let $G = (V, E)$ be a graph of order n and Ω be the set of all possible orderings on V. Suppose the interfacing subgraphs of G corresponding to an ordering $\omega \in \Omega$ are $G'_2(\omega), \ldots, G'_n(\omega)$. Then*

$$max_{\omega\in\Omega}\{x\prod_{i=2}^{n}(x-|G'_i(\omega)|)\} \le C(G,x) \le min_{\omega\in\Omega}\{x\prod_{i=2}^{n}(x-\chi(G'_i(\omega)))\}. \quad (1)$$

Proof Suppose $G_1(\omega), \ldots, G_n(\omega)$ are the accumulating subgraphs of G corresponding to an ordering ω. The proof is by induction on $G_j(\omega)$, for $1 \le j \le n$. In base case, $G_1(\omega)$ is a graph consisting of one vertex v_1, so $C(G_1(\omega), x) = x$. Suppose $x\prod_{i=2}^{j}(x-|G'_i(\omega)|) \le C(G_j,x) \le x\prod_{i=2}^{j}(x-\chi(G'_i(\omega)))$ for some j, $1 \le j < n$. Then consider adding the vertex v_{j+1} and associated edges into $G_j(\omega)$ to form $G_{j+1}(\omega)$. For the lower bound, since $|G'_{j+1}(\omega)|$ is the degree of v_{j+1} in $G_{j+1}(\omega)$, we have $(x-|G'_{j+1}(\omega)|)\ C(G_j(\omega),x) \le C(G_{j+1}(\omega),x)$. From the hypothsis, we obtain

$$x\prod_{i=2}^{j+1}(x-|G'_i(\omega)|) \le (x-|G'_{j+1}(\omega)|)\ C(G_j(\omega),x) \le C(G_{j+1}(\omega),x).$$

For the upper bound, since $\chi(G'_{j+1}(\omega))$ is the minimum number of colors necessary for coloring $G'_{j+1}(\omega)$, we have $C(G_{j+1}(\omega),x) \le (x-\chi(G'_{j+1}(\omega)))\ C(G_j(\omega),x)$. From the hypothsis, we obtain

$$C(G_{j+1}(\omega),x) \le (x-\chi(G'_{j+1}(\omega)))\ C(G_j(\omega),x) \le x\prod_{i=2}^{j+1}(x-\chi(G'_i(\omega))).$$

Since $G_n(\omega) = G$, we have the following formula

$$x\prod_{i=2}^{n}(x - |G_i'(\omega)|) \leq C(G,x) \leq x\prod_{i=2}^{n}(x - \chi(G_i'(\omega))). \tag{2}$$

Finally, because Formula (2) holds for any ordering ω, Formula (1) follows. $\Box$

The upper bound in Formula (1) is in terms of the chromatic number of interfacing subgraphs. Unfortunately, the computation of the chromatic number of a graph is NP-complete. Nevertheless, it turns out that if each of the chromatic numbers in Formula (1) is replaced by any of its lower bounds, the resultant expression is still an upper bound on the chromatic polynomial. The following theorem by Bondy gives a lower bound on the chromatic number of a graph that can be computed efficiently [2].

Theorem 2.2 *Let G be a graph of order n, $d(1) \geq \cdots \geq d(n)$ be the degrees of nodes in G and σ_j be defined recursively by*

$$\sigma_j = n - d(\sum_{i=1}^{j-1}\sigma_i + 1).$$

Suppose $k \leq n$ is some integer satisfying

$$\sum_{j=1}^{k-1}\sigma_j < n. \tag{3}$$

Then $\chi(G) \geq k$. $\Box$

We define $\rho(G)$, called the *Bondy number* of a graph G, to be the largest integer $k \leq n$ satisfying Formula (3). Then an upper bound on the chromatic polynomial of a graph can be expressed in terms of the Bondy number of interfacing subgraphs.

Proposition 2.3 *Let $G = (V,E)$ be a graph of order n and Ω be the set of all possible orderings on V. Suppose the interfacing subgraphs of G corresponding to an ordering $\omega \in \Omega$ are $G_2'(\omega), \ldots, G_n'(\omega)$. Then*

$$max_{\omega\in\Omega}\{x\prod_{i=2}^{n}(x - |G_i'(\omega)|)\} \leq C(G,x) \leq min_{\omega\in\Omega}\{x\prod_{i=2}^{n}(x - \rho(G_i'(\omega)))\}. \tag{4}$$

Proof By Theorem 2.2 and an argument parallel to the upper bound argument of Proposition 2.1. $\Box$

3 Ordering of Vertices

It is clear that carrying out the computation of the maximum and minimum among all the possible orderings in Formula (4) is impractical. As a result, we shall employ a representative ordering to compute upper and lower bounds on the chromatic polynomial.

A graph is a *clique* if every pair of vertices in the graph are adjacent. An ordering on the vertices of a graph is said to be a *perfect elimination ordering* if all the corresponding interfacing subgraphs are cliques [9]. Dirac [4] and Rose [8] have shown that a graph is chordal if and only if it has a perfect elimination ordering. The graph in Figure 1 is an example of a chordal graph, and the labels of vertices show a perfect elimination ordering. The following proposition states an appealing property of perfect elimination ordering.

Proposition 3.1 *Let $G = (V, E)$ be a graph of order n and ω be an ordering on V. If ω is a perfect elimination ordering, then $x \prod_{i=2}^{n}(x - |G_i'(\omega)|) = C(G, x) = x \prod_{i=2}^{n}(x - \rho(G_i'(\omega)))$.*

Proof By the fact, easily proved, that $\rho(K) = |K|$ for any clique K. □

One implication of Proposition 3.1 is that if a perfect elimination ordering of a graph can be generated efficiently, then the chromatic polynomial of that graph can be computed efficiently. Unfortunately, not every graph has a perfect elimination ordering. For example, no complete bipartite graph $K_{n,m}$ with $n > 1$ and $m > 1$ is chordal.

We now describe an ordering that will be used to compute bounds on the chromatic polynomials. For obvious reasons, we shall try to generate a perfect elimination ordering whenever it is possible. The ordering generation is an iterative graph reduction process and the ordering is generated in reverse order.

At each iteration, we search for a vertex such that its adjacency graph is a clique. If such a vertex v exists, it is chosen as the vertex to generate. It is clear that if the graph resulted from removing v has a perfect elimination ordering, then the original graph containing v also has a perfect elimination ordering. The latter ordering can be constructed by simply adding v at the rear of the former ordering. The process continues by removing v from the graph and proceeding to the next iteration.

On the other hand, if such a vertex v does not exist, then we choose a vertex w that has the smallest degree to generate. The basic idea behind this heuristic is that, to yield nontrivial lower bounds for larger number of values, we demand the maximum order of the interfacing subgraphs in Formula (2) to be as small as possible. Since the ordering generation is a graph reduction process, the degree of vertices or the order of the adjacency graph of vertices will become smaller when the process goes on. Therefore, when the generation of a perfect elimination ordering cannot continue, we greedily choose the vertex that has the smallest degree to generate. The process continues by removing the chosen vertex w from the

graph and proceeding to the next iteration. The whole process terminates when all the vertices are generated.

Notice also that the ordering among the vertices whose adjacency graph is a clique is not crucial. In the process, once a vertex has a clique as its adjacency graph, its later adjacency graphs will still remain as cliques. This is because the removal of vertices from a clique results in another clique.

Let $G = (V, E)$ be a graph and U be a subset of V. Then the graph $G - U$ denotes the subgraph induced by $V - U$. We define an ordering ω_0, called a *perfect-smallest-last ordering*, as follows. Let $G_1, \ldots, G_n$ be a sequence of subgraphs of a graph G of order n, with

1. $G_1 = (V_1, E_1) = G$ and $G_{i+1} = (V_{i+1}, E_{i+1}) = G_i - \{v_i\}$,
2. $v_i = \begin{cases} v & \text{if there is a vertex } v \in V_i \text{ such that} \\ & Adj_{G_i}(v) \text{ is a clique} \\ min_{v \in V_i}\{d_{G_i}(v)\} & \text{otherwise,} \end{cases}$

for $1 \leq i \leq n$, then $\omega_0 = v_n, \ldots, v_1$. Since a perfect-smallest-last ordering always chooses a vertex whose adjacency graph is a clique if such a vertex exists. It has the following property:

Proposition 3.2 *If a graph G is chordal, then a perfect-smallest-last ordering on the vertices of G is a perfect elimination ordering.* □

4 An Approximation Algorithm

We are now ready to present an algorithm for computing an upper bound and a lower bound on the chromatic polynomial of a graph based on Formula (4) and the perfect-smallest-last ordering. The algorithm is shown in Figure 3. Apart from the bounds, this algorithm also computes a mean of the bounds. Let $L(G, x)$ denote the lower bound $x \prod_{i=2}^{n}(x - |G_i'(\omega_0)|)$ and $U(G, x)$ denote the upper bound $x \prod_{i=2}^{n}(x - \rho(G_i'(\omega_0)))$ in Formula (4), respectively, with ω_0 being a perfect-smallest-last ordering on the vertices of G. We estimate the chromatic polynomial of a graph G as

$$\widehat{C}(G, x) = \frac{2 \times U(G, x) \times L(G, x)}{U(G, x) + L(G, x)}, \tag{5}$$

the *harmonic mean* of $U(G, x)$ and $L(G, x)$. Notice that although $\widehat{C}(G, x)$ is an estimate of $C(G, x)$, $\widehat{C}(G, x)$ itself may be a rational function, but not a polynomial.

We now consider the complexity of Algorithm CP. Let n and m be, respectively, the number of vertices and edges in the graph. We first consider the test in the **if** statement. Detecting

Algorithm CP(G)
begin
$G_1 \equiv (V_1, E_1) := G$;
$U(G, x) := L(G, x) := 1$;
for $i := 1$ to n **do**
if there is a vertex $v \in V_i$ such that $Adj_{G_i}(v)$ is a clique **then**
$v_i := v$
else
$v_i := min_{v \in V_i}\{d_{G_i}(v)\}$
fi
$U(G, x) := U(G, x) \times (x - \rho(G_i(v_i)))$;
$L(G, x) := L(G, x) \times (x - |G_i(v_i)|)$;
$G_{i+1} \equiv (V_{i+1}, E_{i+1}) := G_i - \{v_i\}$
od
$\widehat{C}(G, x) :=$ the harmonic mean of $U(G, x)$ and $L(G, x)$
end

Figure 3: An algorithm for estimating the chromatic polynomial of a graph

if a graph of order k is a clique can be performed in time $O(k^2)$. At the worst case, when no adjacency graph is a clique and the detection has to be performed for every vertex, the time required is $O(n + \sum_{j=1}^{n} d^2_{G_i}(v_j)) = O(n + (\sum_{j=1}^{n} d_{G_i}(v_j))^2) = O(n + m^2)$. The minimum operation in the **else** branch can be performed in time $O(n)$. Thus, altogether, the complexity of the **if** statement is $O(n + m^2)$.

The symbolic multiplications of polynomials for $U(G, x)$ and $L(G, x)$ can be performed in time $O(n)$ since the order of $U(G, x)$ and $L(G, x)$ is at most n. The computation of $\rho(G_i(v_i))$ demands a sorting step, so it requires time $O(n \log n)$. The updating from G_i to G_{i+1} can be performed in time $O(n + m)$. Put together, the complexity for each iteration of the **for** statement is $O(n \log n + m^2)$. Taking the number of iterations into account, the time demanded for the **for** statement is $O(n^2 \log n + nm^2)$.

The computation of the harmonic mean needs a symbolic multiplication and a symbolic addition. It can be performed in time $O(n^2)$. Therefore, the complexity of the entire algorithm is $O(n^2 \log n + nm^2)$. This complexity analysis leads to the following theorem:

Theorem 4.1 *The problem of computing the chromatic polynomial of a chordal graph is solvable in polynomial time.*

Proof By Propositions 3.1, 3.2 and the complexity analysis of Algorithm CP. □

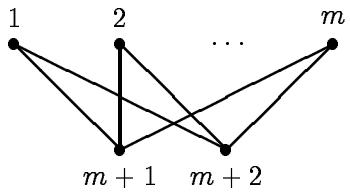

Figure 4: Graph $K_{m,2}$

5 Performance Analysis and Measurements

This section investigates the performance behavior of Algorithm CP. Since the values of $C(G,x)$ are usually very large, we shall consider the relative error $\Delta = |\widehat{C}(G,x) - C(G,x)|/C(G,x)$ in performance analysis.

The worst-case relative error of an approximation algorithm occurs at the cases where $C(G,x) = U(G,x)$ or $C(G,x) = L(G,x)$; namely, $C(G,x)$ is equal to one of the two extremes. The worst-case relative error is minimized when the relative errors for $C(G,x) = U(G,x)$ and $C(G,x) = L(G,x)$ are equal. The harmonic mean provides such an optimum because

$$\Delta \leq \frac{U(G,x) - \widehat{C}(G,x)}{U(G,x)} = \frac{\widehat{C}(G,x) - L(G,x)}{L(G,x)} = \frac{U(G,x) - L(G,x)}{U(G,x) + L(G,x)}. \tag{6}$$

Since both $U(G,x)$ and $L(G,x)$ are nonnegative, we have

$$\Delta \leq \frac{U(G,x) - L(G,x)}{U(G,x) + L(G,x)} \leq 1, \tag{7}$$

for any nonnegative integer x. This implies that we always have $0 \leq \widehat{C}(G,x) \leq 2C(G,x)$. This statement is always true if we can compute both a lower bound and an upper bound on the measure we are interested in. On the other hand, we should also realize that because the problem of k-colorability of a graph is NP-complete, there is no polynomial time algorithm for approximating the chromatic polynomial of a graph that has a relative error less than 1 unless P = NP [11]. Therefore, if P $\neq$ NP, 1 is the best worst-case relative error we can expect. We now give an example that has a relative error of 1. Consider the complete bipartite graph $K_{m,2}$ shown in Figure 4. According to Formula (7), we have the relative error

$$\frac{x^2(x-1)^{m-1} - x^2(x-2)^{m-1}}{x^2(x-1)^{m-1} + x^2(x-2)^{m-1}},$$

which gives 1 when $x = 2$. In general, $\widehat{C}(K_{m,n},x) = 0$ for $x \leq min(m,n)$, while $C(K_{m,n},x) > 0$ for $x \geq 2$.

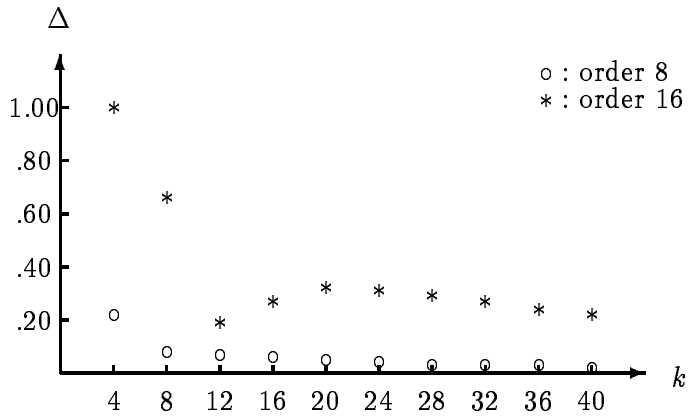

Figure 5: The average values of the relative errors of the approximate chromatic polynomials for random graphs of order 16 and 8

Another reason for choosing the harmonic mean is as follows. Since $U(G,x)$ becomes more significant than $L(G,x)$ as x increases, we usually have $\widehat{C}(G,x) > C(G,x)$ except for some small x. At the mean time, the harmonic mean is always less than or equal to the arithmetic mean [1]. Therefore, in most cases the harmonic mean gives a better upper bound than the arithmetic mean.

To consider the average performance behavior, we also run some experiments on randomly generated graphs. The edges in the graph are chosen independently and with probability 0.5. Figure 5 displays the results of the average values of the relative errors over 5 graphs of order 16 and over 5 graphs of order 8. The results show that the value of the relative error Δ increases as the order of graphs increases. This is because the higher the order of a graph, the higher the degree of the estimated bounds on its chromatic polynomial, and the higher the accumulated error. The results also show that the value of the relative error decreases as the number of colors k increases except for transient fluctuation for small values of k. This is because for each graph G of order n, both $U(G,x)$ and $L(G,x)$ are polynomial of degree n with the coefficient of x^n being 1. Hence, the degree of the numerator is smaller than the degree of the denominator in Formula (7) and $\lim_{x\to\infty}\Delta = 0$. The smallest-degree heuristic employed in the perfect-smallest-last ordering aims to restrict the transient fluctuation to very small values. This allows us to have a good approximation of the chromatic polynomial of a graph in most cases.

6 Conclusions

The problem of computing the chromatic polynomial of a graph is #P-hard. This paper has presented an approximation algorithm for computing the chromatic polynomial of a graph. This algorithm has time complexity $O(n^2 \log n + nm^2)$ for a graph with n vertices and m edges. This paper has also shown that the problem of computing the chromatic polynomial of a chordal graph can be solved in polynomial time. Knowledge about the chromatic polynomial of graphs can be employed to improve the performance of logic programs and deductive databases.

Acknowledgments

Special thanks to S. Debray, P. Downey and U. Manber for many valuable comments on this work.

References

[1] A. O. Allen, *Probability, Statistics, and Queuing Theory With Computer Science Applications*, Academic Press, Inc, 1990.

[2] J. A. Bondy, "Bounds for the Chromatic Number of a Graph," *Journal of Combinatorial Theory* 7, 1 (Jan. 1969), pp. 96–98.

[3] S. K. Debray and N.-W. Lin, "Cost Analysis of Logic Programs," to appear in *ACM Transactions on Programming Languages and Systems.*

[4] G. A. Dirac, "On Rigid Circuit Graphs," Abhandlungen aus dem Mathematischen Seminar der Universitat Hamburg, 25 (1961), pp. 71–76.

[5] F. Gavril, "Algorithms for Minimum Coloring, Maximum Clique, Minimum Covering by Cliques, and Maximum Independent Set of a Chordal Graph," *SIAM Journal of Computing* 1, 2 (June 1972), pp. 180–187.

[6] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[7] R. M. Karp, "Reducibility Among Combinatorial Problems," *Complexity of Computer Computations*, edited by R. E. Miller and J. W. Thatcher, Plenum Press, New York, 1972, pp. 85–103.

[8] D. J. Rose, "Triangulated Graphs and the Elimination Process," *Journal of Mathematical Analysis and Applications* 32, (1970), pp. 597–609.

[9] D. J. Rose, R. E. Tarjan and G. S. Lueker, "Algorithmic Aspects of Vertex Elimination," *SIAM Journal of Computing* 5, 2 (June 1976), pp. 266–283.

[10] D. E. Smith and M. R. Genesereth, "Ordering Conjunctive Queries," *Artificial Intelligence* 26 (1985), pp. 171–215.

[11] L. Stockmeyer, "On Approximation Algorithms for #P," *SIAM Journal on Computing* 14, 4 (November 1985), pp. 849–861.

[12] L. G. Valiant, "The Complexity of Computing the Permanent," *Theoretical Computer Science* 8, (1979), pp. 189–201.

[13] D. H. D. Warren, "Efficient Processing of Interactive Relational Database Queries Expressed in Logic", *Proc. Seventh International Conference on Very Large Data Bases*, 1981, pp. 272–281.