

Colorability of P_5 -free Graphs

Chính T. Hoàng* Joe Sawada† Zebin Wang‡

January 27, 2006

Abstract

This paper considers the question of whether or not a P_5 -free graph can be 4-colored in polynomial time. It is known that a connected P_5 -free graph G must have either a dominating clique or a dominating P_3 . Thus, when considering the 4-coloring question, we have three cases of interest: either G has a dominating K_4 , a dominating K_3 , or a dominating P_3 . In this paper we demonstrate a polynomial time approach for determining whether or not a P_5 -free graph G with a dominating K_4 can be 4-colored.

Keywords: P_5 -free, graph coloring, dominating clique

1 Introduction

Graph coloring is among the most important and applicable graph problems. The *k-colorability problem* is the question of whether or not the vertices of a graph can be colored with one of k colors so that no two adjacent vertices are assigned the same color. In general, the k -colorability problem is NP-complete [15]. Even for planar graphs with no vertex degree exceeding 4, the problem is NP-complete [10]. However, for other classes of graphs, like perfect graphs [13], the problem is polynomial-time solvable. For the following special class of perfect graphs, there are efficient polynomial time algorithms for finding optimal colorings: chordal graphs [11], weakly chordal graphs [14], and comparability graphs [8]. For more information on perfect graphs, see [1], [12] and [5].

Another interesting class of graphs are those that are P_t -free, that is, graphs with no chordless paths of length $t - 1$. If $t = 3$ or $t = 4$, then there exists efficient algorithms to answer the k -colorability question (see [5]). However, for $t = 5, 6$, or 7 , the complexity of the problem is unknown; for $t \geq 8$ the problem is NP-complete [19]. To handle the unknown cases, a natural first step is to consider what happens if the value of k is fixed. Taking this parameterization into account, a snapshot of the known complexities for the k -colorability problem of P_t -free graphs is given in Table 1.

Notice that when $t = 5$ and $k = 3$, the question can be answered in polynomial time (see [19]), but for $t = 5$ and $k = 4$ the complexity remains unknown. In this paper we focus on this boundary case and

*Physics and Computer Science, Wilfred Laurier University, Canada. Research supported by NSERC. email: choang@wlu.ca

†Computing and Information Science, University of Guelph, Canada. Research supported by NSERC. email: sawada@cis.uoguelph.ca

‡Computing and Information Science, University of Guelph, Canada.

$k \setminus l$	3	4	5	6	7	8	...	12	...
3	$O(m)$	$O(m)$	$O(n^\alpha)$	$O(mn^\alpha)$?	?	?	?	...
4	$O(m)$	$O(m)$?	?	?	?	?	NP_c	...
5	$O(m)$	$O(m)$?	?	?	NP_c	NP_c	NP_c	...
6	$O(m)$	$O(m)$?	?	?	NP_c	NP_c	NP_c	...
7	$O(m)$	$O(m)$?	?	?	NP_c	NP_c	NP_c	...
...

Table 1: Complexities of k -colorability of P_t -free graphs

attempt to determine the complexity of the 4-colorability problem for P_5 -free graphs. This is a natural place to continue the research since P_5 -free graphs have already been widely studied. In particular, Bacsó and Tuza [4] have shown that in every connected P_5 -free graph there exists a dominating clique or a dominating P_3 . Since we are considering the 4-colorability question, we need only consider P_5 -free graphs without a K_5 (since at least 5 colors are required for such graphs). If the graph is not connected, we can consider the connected components separately. Thus, for the remainder of this paper we will assume that we are dealing with a connected P_5 -free graph G with no K_5 . This gives rise to 3 cases: G has a dominating K_4 , G has a dominating K_3 , or G has a dominating P_3 .

For the case where there is a dominating K_4 we prove that the 4-colorability question can be answered in polynomial time. The remaining two cases remain open questions.

The remainder of the paper is presented as follows. We begin in Section 2 with relevant graph definitions, concepts, and notations. We then present a general strategy for attacking the 4-colorability question in Section 3. In Section 4 we apply the strategy to the case of a dominating K_4 . We conclude with a summary of the results in Section 5.

2 Background and Definitions

In this section we provide the necessary background and definitions used in the rest of the paper. For starters, we assume that $G = (V, E)$ is a simple undirected graph where $|V| = n$ and $|E| = m$. If A is a subset of V , then we let $G(A)$ denote the subgraph of G induced by A and let $N(A)$ denote the set of vertices in $V - A$ that are adjacent to some vertex in A .

DEFINITION 1 *A set of vertices A is said to dominate another set B , if every vertex in B is adjacent to at least one vertex in A .*

DEFINITION 2 *A set of vertices A is B -null if no vertex in B is adjacent to a vertex in A .*

DEFINITION 3 *P_t denotes the chordless path on t vertices and $t - 1$ edges. Such a path has length $t - 1$.*

DEFINITION 4 *Given a graph G , an integer k and for each vertex v , a list $l(v)$ of k colors, the k -list coloring problem asks whether or not there is a coloring of the vertices of G such that each vertex receives a color from its list.*

DEFINITION 5 *The restricted k -list coloring problem is the k -list coloring problem in which the lists $l(v)$ of colors are subsets of $\{1, 2, \dots, k\}$.*

DEFINITION 6 *In the k -list coloring problem (restricted or not), a vertex v is said to be t -listed if its list $l(v)$ has size t .*

The following definition of 2-SAT is important since we can map any 2-list coloring instance to the problem of 2-SAT.

DEFINITION 7 *Given a boolean formula E in conjunctive normal form such that each clause contains two literals (a literal is a variable or its negation), the 2-SAT problem asks whether there is a truth assignment to the variables that satisfy E .*

It is well known that 2-SAT can be solved in polynomial time (see [3, 7, 9]). In particular, [3] has the proof of the following theorem.

THEOREM 1 *Any 2-SAT problem can be solved in $O(n^2)$ time where n is the number of variables in the formula.*

It is well-known and easy to prove that the general 2-list coloring problem can be reduced to the 2-SAT problem.

We now prove a result that will be needed later.

THEOREM 2 *The restricted 3-list coloring problem for P_5 -free graphs is polynomially solvable.*

Proof of Theorem 2. Let G be a P_5 -free graph. We may assume that G is connected, for otherwise, we can recursively color each component of G (or decide that there is no 3-coloring). We also may assume that G has at least three vertices. Let Φ be an instance of the problem, that is, Φ is a collection of lists $l(v)$ for each vertex v .

The graph G has to be K_4 -free, for otherwise, Φ has no solution. By the result of [4], G has a dominating K_3 or P_3 . Consider the case when G has a dominating K_3 . There are at most $3!$ colorings of this K_3 . Once one such coloring is assigned to the K_3 , each of the remaining vertices v can have at most 2 colors, that is, the vertices of $G - K_3$ are 2-listed (for example, if vertex $x \in K_3$ gets color 1, then for each neighbor y of x , $l(y)$ can no longer contain 1). Each coloring of the K_3 gives rise to a 2-SAT formula. We only need to solve at most 6 such formulas to obtain a solution to the original instance Φ . The case that G has a dominating P_3 can be handled in the same manner. \square

A similar proof has been used by [18] for the 3-coloring problem on P_5 -free graphs. However for a given class C of graphs, it is unknown whether or not a solution to the k -colorability problem for C can be used to solve the restricted k -list coloring problem for C .

3 Coloring Strategy

In this section we will outline our general strategy for finding a polynomial-time algorithm to answer the 4-colorability question for P_5 -free graphs.

Our approach is to take an instance of a specific coloring problem Φ for a given graph and replace it with a polynomial number of instances $\phi_1, \phi_2, \phi_3, \dots$ such that the answer to Φ is “yes” if and only if there is some instance ϕ_k that also answers “yes”. In general, if each ϕ_k corresponds to one of the two aforementioned results, then we can answer the coloring question Φ in polynomial time.

For example, consider a complete graph with four vertices u, v, w, x where each vertex has color list $\{1, 2, 3, 4\}$. This 4-listing corresponds to our initial instance Φ . Now, since vertex u has four choices, we can consider each of the 4 ways to color u to obtain four new instances that together are equivalent to Φ :

ϕ_1 : $u = 1$ and v, w, x are 3-listed with $\{2, 3, 4\}$,

ϕ_2 : $u = 2$ and v, w, x are 3-listed with $\{1, 3, 4\}$,

ϕ_3 : $u = 3$ and v, w, x are 3-listed with $\{1, 2, 4\}$,

ϕ_4 : $u = 4$ and v, w, x are 3-listed with $\{1, 2, 3\}$.

Notice that each of the vertices v, w, x became three listed since they are all adjacent to u and thus cannot be assigned the same color as u . Of course we cannot simply continue to perform this type of reduction on all vertices since this will lead to 4^n new instances which is not polynomial in number. However, in this case we can apply Theorem 2 to each of these 4 new instances. By solving them independently we can obtain an answer to the original problem in polynomial time.

To apply this strategy to the three cases of a dominating K_4, K_3 or P_3 , we begin by coloring these specific vertices. For example, in the case of the dominating K_4 , we color each vertex in the K_4 a unique color from $\{1, 2, 3, 4\}$. We need only consider one of these colorings, since the other $4! - 1$ possible colorings are equivalent. Since every other vertex is adjacent to at least one of these vertices, the colors allowed by these vertices become restricted which means that these vertices become either 3-listed, 2-listed, or 1-listed. If there are no 3-listed vertices then we can immediately apply Theorem 1 to answer the 4-colorability question. Thus, we need only focus on those vertices that are 3-listed.

4 Dominating K_4

In this section we assume that connected P_5 -free graph G has a dominating $K_4 = \{v_1, v_2, v_3, v_4\}$. Without loss of generality, we assign a unique color to each of the four vertices: $v_1 = 1, v_2 = 2, v_3 = 3$, and $v_4 = 4$. The remaining vertices get partitioned into sets depending on their adjacencies to the K_4 . Since no vertex can be adjacent to all four vertices in the K_4 (since G is assumed to be K_5 -free), we obtain the following sets:

- S_i : the set of all vertices adjacent to exactly v_i in the K_4 where $1 \leq i \leq 4$.
- S_{ij} : the set of all vertices adjacent to exactly v_i and v_j in the K_4 , where $1 \leq i < j \leq 4$
- S_{ijk} : the set of all vertices adjacent to exactly v_i, v_j, v_k in the K_4 , where $1 \leq i < j < k \leq 4$

A visualization of these sets is given in Figure 1.

The sets S_{ijk} are called the *fixed color sets*, since they have only once choice of color if G is 4-colorable. Each vertex in the sets S_{ij} is 2-listed: it has the choice of two colors. Each vertex in a set S_i is 3-listed:

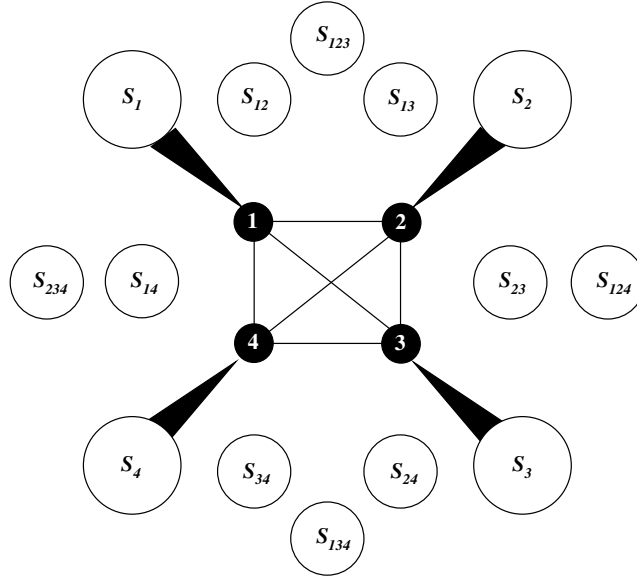


Figure 1: Visualization of a P_5 -free graph with a dominating K_4 with many edges missing.

it has the choice of 3 colors if G is 4-colorable. Observe that since there is no K_5 in G , any two vertices in the same fixed color set must not be adjacent. Additionally, for each vertex in a fixed color set, we can remove its fixed color from the lists of each of its neighbors. The resulting color lists for the vertices becomes our original list coloring problem instance Φ .

To apply the strategy outlined in the previous section, we will replace this problem instance Φ with an equivalent set of new problem instances $\phi_1, \phi_2, \phi_3, \dots$, such that each of the ϕ_k can be solved in polynomial time. To obtain a polynomial time algorithm, the number of new instances must be polynomial in number.

The following subsection outlines two helpful lemmas that will aid us in achieving this goal.

4.1 Helpful Lemmas

For each lemma, $G = (V, E)$ is a P_5 -free graph with no K_5 . Additionally, X and Y are disjoint subsets of V where every vertex in X is adjacent to some vertex in Y and every vertex in Y is adjacent to some vertex in X . Illustrations of the structure for each lemma are shown in Figure 2.

LEMMA 1 *If there exists vertices $u, v, w \in V - X - Y$ where $(u, w), (v, w) \in E$ such that:*

- *u dominates X and is Y -null,*
- *v dominates Y and is X -null, and*
- *w is both X -null and Y -null,*

then there is a vertex $y \in Y$ that dominates X .

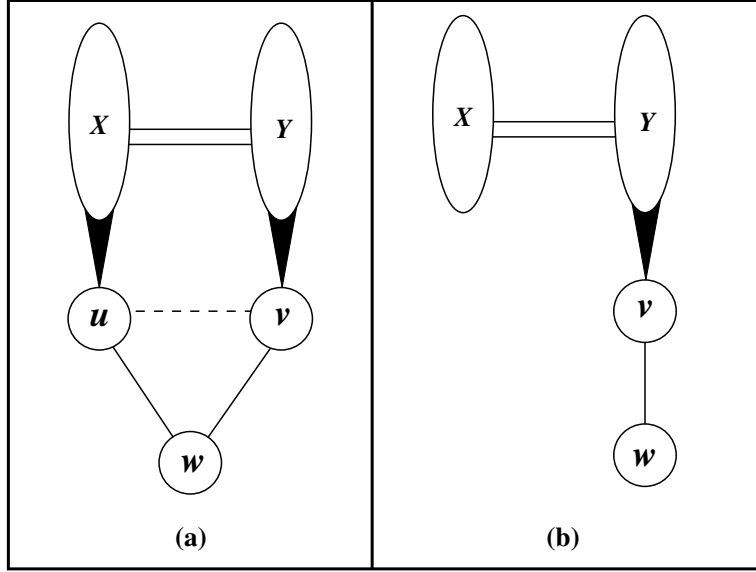


Figure 2: (a) Illustration for Lemma 1. The dotted line indicates that the edge may or may not be present. (b) Illustration for Lemma 2.

PROOF: Consider a vertex $y \in Y$ that is adjacent to a maximal number of vertices in X . Suppose that y does not dominate X . Then there must exist a vertex $x' \in X$ that is not adjacent to y but is adjacent to some $y' \in Y$. Now, since y is adjacent to a maximal number of vertices in X , it must be adjacent to some vertex $x \in X$ that is not adjacent to y' .

Now, if neither (x, x') nor (y, y') is an edge in E , then we obtain the $P_5 = x, y, v, y', x'$. However, if $(x, x') \in E$ we obtain the $P_5 = x, x', y', v, w$ and if $(y, y') \in E$ we obtain the $P_5 = y, y', x', u, w$. Thus, by contradiction, y must dominate X . \square

LEMMA 2 *If there exist adjacent vertices $v, w \in V - X - Y$ such that*

- *v dominates Y and is X -null, and*
- *w is both X -null and Y -null,*

then there there is a complete subgraph K_1, K_2 , or K_3 in $G(Y)$ that dominates X .

PROOF: Consider a vertex $y \in Y$ that is adjacent to a maximal number of vertices in X . If y does not dominate X , then there must exist a vertex $y' \in Y$ that is adjacent to a maximal number of vertices in X that are not adjacent to y . This means that there exists vertices $x, x' \in X$ such that (x, y) and (x', y') are in E , but (x, y') and (x', y) are not.

If neither (x, x') nor (y, y') is an edge, then we obtain the $P_5 = x, y, v, y', x'$. If $(x, x') \in E$, then we obtain the $P_5 = x, x', y', v, w$. In both cases we have a contradiction to G being P_5 -free. Therefore we must have the edge $(y, y') \in E$. Now if this edge does not dominate X then there must exist some vertex $y'' \in Y$ that is adjacent to a maximal number of vertices in X , including x'' that are not dominated by $\{y, y'\}$. Now again by the maximality of y , we know there is some vertex in X that is adjacent to y but not

y'' . Using the same argument as with y and y' , we must have the edge $(y, y'') \in E$. The same argument can also be used between y' and y'' (by the maximality decision used for selecting y'). Thus, we must have the edge $(y', y'') \in E$.

Now, if the $K_3 = \{y, y', y''\}$ still does not dominate X , then there must exist a fourth vertex y''' adjacent to some vertex x''' that is not dominated by the K_3 . However, by the same arguments as before y''' must be adjacent to every vertex in the K_3 , which would mean that $\{y, y', y'', y''', v\}$ form a K_5 in G - a contradiction. \square

4.2 Applying the Lemmas

Using the two lemmas presented in the previous subsection, we now focus on *reducing* the original restricted 4-list coloring problem Φ into an equivalent set of simpler problems that we can solve in polynomial time. Recall, that after coloring the dominating K_4 we partitioned the vertices into the sets depending on their adjacencies to the K_4 . The vertices in all these sets became at most 2-listed except for S_1, S_2, S_3 and S_4 which are 3-listed. We will describe how to *reduce* vertices in S_1 so they become 2-listed. The other cases will follow by symmetry.

To reduce the vertices in S_1 , we must consider their dependencies (adjacencies) with the other vertices in the graph. Recall, that such dependencies with the fixed colored vertices have already been factored into the original list coloring problem. Thus, we need only focus on the other sets of 3-listed vertices along with the sets of 2-listed vertices. The following three methods describe the reduction process for three different cases that account for all of these sets:

Method 1: S_i and S_{ij} : Let X denote the set of all vertices in S_i adjacent to some vertex in S_{ij} and let Y denote the set of all vertices in S_{ij} adjacent to some vertex in S_i . By Lemma 2, there is a set of vertices K_1, K_2 , or K_3 in Y that dominate X . However we color such a dominating set of vertices, the vertices in X will become 2-listed. For example, if a vertex y in the dominating set gets color t , then the neighbors of y in X must drop t from their lists. Since it is impossible for there to be a K_3 in S_{ij} (otherwise there would be a K_5 in G) this means there are at most two ways to color K_1 or K_2 . \square

Method 2: S_i and S_{jk} : Let X denote the set of all vertices in S_i adjacent to some vertex in S_{jk} and let Y denote the set of all vertices in S_{jk} adjacent to some vertex in S_i . By Lemma 1, there is a vertex $y \in Y$ that dominates X . For each color choice of y we get a new instance. If we color y with the fourth color (not equal to i, j, k), then each vertex in X becomes two listed (we are done). If we color y with color i then the vertices of X remain 3-listed. However, now we can remove y from the graph and recursively apply the procedure. This will result in at most $2n$ colorings. \square

Method 3: S_i and S_j : Let X denote the set of all vertices in S_i adjacent to some vertex in S_j and let Y denote the set of all vertices in S_j adjacent to some vertex in S_i . By Lemma 1, there is a vertex $y \in Y$ that dominates X . For each color choice of y we get a new instance. If we color y one of the two colors not equal to i , then each vertex in X becomes two listed (we are done). If we color y with color i then the colors from X remain 3-listed. However, now we can remove y from the graph and recursively apply the procedure. This will result in at most $3n$ colorings. \square

Focusing on S_1 , we apply Method 1 to the sets S_{12}, S_{13}, S_{14} followed by Method 2 on the sets S_{23}, S_{24}, S_{34} followed by Method 3 on the sets S_2, S_3, S_4 . Let A denote the set of all vertices in S_1 that are adjacent to at least one vertex in $S_{23} \cup S_{24} \cup S_{34} \cup S_{12} \cup S_{13} \cup S_{14} \cup S_2 \cup S_3 \cup S_4$. After applying these methods, the number of new coloring instances will be polynomial in number and all vertices in A will be at most

2-listed. The vertices in $B = S_1 - A$, however, remain 3-listed. Fortunately, it is not difficult to handle these vertices as we discuss next.

Observe that if $a \in A$ and $b \in B$ then a can not be adjacent to b , since it would imply the existence of a P_5 . For example, if a is adjacent to a vertex x in S_{12} and a and b were adjacent, then b, a, x, v_2, v_3 would be a P_5 in G . Similar P_5 's can be found for the other cases. Thus, it follows that $N(B)$ consists only of fixed-colored vertices. Since v_1 is adjacent to all of B , the lists of vertices in B are subsets of $\{2, 3, 4\}$. Thus, we can apply Theorem 2 to determine if there is a valid 3-coloring of B . If such a coloring exists, then any 4-coloring of $G - B$ will be compatible with such a 3-coloring of B . This completes the reduction for S_1 .

By repeating this procedure for S_2, S_3 and S_4 , we end up with a polynomial number of instances where the vertices are all 2-listed. By applying Theorem 1, we can determine whether or not these instances have a valid coloring in polynomial time.

THEOREM 3 *The restricted 4-list coloring problem for P_5 -free graphs with a dominating K_4 can be answered in polynomial time.*

5 Summary

It is an open question as to whether or not there exists a polynomial time algorithm for determining whether or not a P_5 -free graph can be 4-colored. It is known that a connected P_5 -free graph has either a dominating clique or a dominating P_3 . If such a graph can be 4-colored (and has three or more vertices), then it must have either a dominating K_4, K_3 , or P_3 . In this paper we have shown that the 4-coloring question for P_5 -free graphs with a dominating K_4 can be answered in polynomial time. The authors also believe that the general strategy outline in Section 3 can be extended, with some modifications, to handle the remaining two cases. We now discuss the case of the dominating K_3 . We can still apply Methods 1 and 3 to reduce S_i with respect to S_{ij} and S_j . However, we cannot apply Method 2 to reduce S_i with respect to S_{jk} because the hypothesis of Lemma 1 fails, the vertex w might not exist. It is conceivable that some variations of Lemma 1 and Method 2 would solve the problem.

Another interesting question encountered in this research is: "When can the solution to the k -colorability problem for a given class of graphs be used to answer the restricted k -list coloring question problem for the same class?"

References

- [1] Jorge L. Ramirez Alfonsin, Bruce A. Reed, *Perfect Graphs*, John Wiley & Sons, LTD, 2001
- [2] K. Appel, W. Haken and J. Koch, Every planar map is four colorable. Part II. Reducibility, Illinois J. Math. 21 (1977), 491–567.
- [3] B. Aspvall, M. Plass, and R. Tarjan, A linear-time algorithm for testing the truth of certain quantified boolean formulas, Info. Proc. Letters, 8:121-123, 1979.
- [4] G. Bacsó and Z. Tuza, Dominating cliques in P_5 -free graphs, Period. Math. Hungar. Vol. 21 No. 4 (1990) 303-308.

- [5] C. Berge and V. Chvátal (eds.), *Topics on perfect graphs*, North-Holland, Amsterdam, 1984.
- [6] A. Brandstädt, V.B. L and J.P. Spinrad, *Graph Classes: A Survey*, SIAM, 1999.
- [7] S. Even, A. Itai and A. Shamir, On the complexity of the timetable and multicommodity flow problems, *SIAM J. Comput.* 5, 691-703, 1976.
- [8] S. Even, A. Pnueli and A. Lempel, Permutation graphs and transitive graphs, *J. Assoc. Comput. Mach.* 19, 400-410, 1972.
- [9] M. R. Garey and David S. Johnson, *Computers and Intractability: A Guild to the Theory of NP-Completeness*, W.H.Freeman and Company, 1979.
- [10] M. R. Garey, D. S. Johnson and L. Stockmeyer, Some simplified NP-complete problems, *Theo. Comput. Sci.* 1, 237-267, 1976.
- [11] F. Gavril, Algorithms for minimum coloring, maximum clique, minimum coloring by cliques, and maximum independent set of a chordal graph, *SIAM J. Comput.* 1, 180-187, 1972.
- [12] M. C. Golumbic, *Algorithmic graph theory and perfect graphs*, Academic Press, New York, 1980.
- [13] M. Grötschel, L. Lovász and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* 1, 169-197, 1981.
- [14] R. Hayward, C. T. Hoàng and F. Maffray, Optimizing weakly triangulated graphs, *Graphs and Combinatorics* 5, 339-349, 1989.
- [15] R. M. Karp, Reducibility among combinatorial problems. In: R. E. Miller and J. W. Thatcher (eds), *Complexity of Computer Computations*, Plenum Press, New York, 85-103, 1972.
- [16] S. Mellin, Polynomial algorithm for P_k -free graph, *Dipomarbeit am Institut für Informatik, Universität zu Köln*, 2002.
- [17] B. Randerath and I. Schiermeyer, Vertex colouring and forbidden subgraphs - a survey, *Graphs and Combinatorics* 20 (2004) 1-40.
- [18] B. Randerath and I. Schiermeyer, 3 -Colorability $\in P$ for P_6 -free graphs, *Vol 136, No 2-3, (2004)* 299-313.
- [19] J. Sgall, G.J. Woeginger, The complexity of coloring graphs without long induced paths, *Acta Cybernet.* 15(1)(2001) 107-117.