On the edge-colouring of split graphs

Celina M. H. de Figueiredo *João Meidanis*
Célia Picinin de Mello†

Relatório Técnico IC–96-04

Junho de 1996

On the edge-colouring of split graphs

Celina M. H. de Figueiredo* João Meidanis† Célia Picinin de Mello†

Abstract. We consider the following question: can split graphs with odd maximum degree be edge-coloured with maximum degree colours? We show that any odd maximum degree split graph can be transformed into a special split graph. For this special split graph, we were able to solve the question, in case the graph has a quasi-universal vertex.

Sumário. Consideramos o seguinte problema: grafos *split* com grau máximo ímpar admitem uma coloração de arestas com grau máximo cores? Mostramos como qualquer grafo *split* com grau máximo ímpar pode ser transformado em um grafo *split* especial. Para este grafo *split* especial, resolvemos o problema proposto no caso do grafo admitir um vértice quase-universal.

1 Introduction

An *edge-colouring* of a graph is an assignment of colours to its edges such that no adjacent edges have the same colour. The *chromatic index* of a graph is the minimum number of colours required to produce an edge-colouring for that graph.

An easy lower bound for the chromatic index is the maximum vertex degree Δ. A celebrated theorem by Vizing states that these two quantities differ by at most one [1]. Graphs whose chromatic index equals the maximum degree are said to be *Class* 1; graphs whose chromatic index exceeds the maximum degree by one are said to be *Class* 2.

Very little is known about the complexity of computing the chromatic index in general. It is known that complete graphs and chordless cycles are Class 1 if and only if the number of vertices is even. Bipartite graphs are all Class 1. Planthold [2, 3] proved that graphs with a universal vertex or with a quasi-universal vertex can be classified in polynomial time. Hoffman and Rodger [4] derived a similar result for complete multipartite graphs. The four-colour problem is equivalent to saying that all 2-connected 3-regular planar graphs are Class 1. In previous papers, we settled the question for doubly chordal graphs with odd maximum degree [5] and for indifference graphs with at most three maximal cliques [6]. Doubly chordal graphs include interval graphs and strongly chordal graphs. These results summarize all that is known about edge-colouring specific classes of graphs.

In this note, we consider the following question: can split graphs with odd maximum degree be edge-coloured with Δ colours? Edge-colouring of split graphs was first considered

*Instituto de Matemática, Universidade Federal do Rio de Janeiro, Caixa Postal 68530, 21944-970 Rio de Janeiro, RJ, Brasil. `celina@cos.ufrj.br`. Partially supported by CNPq.

†Instituto de Computação, Universidade Estadual de Campinas, Caixa Postal 6065, 13081-970 Campinas, SP, Brasil. `{meidanis,celia}@dcc.unicamp.br`. Partially supported by FAPESP and CNPq.

in [7]. In that work, we discussed local sufficient conditions for a graph to be Class 2. We considered subclasses of chordal graphs, such as split graphs and interval graphs.

We begin this note characterizing two subclasses of split graphs, which we call EASY and DIFFICULT. The EASY ones can be easily coloured by a pullback technique [5]. We essentially "borrow" the colour from a complete graph. The DIFFICULT ones, as the name indicates, are hard to colour, but this class has the important property that if all its members are Class 1, then *all* odd maximum degree split graphs are Class 1.

Next, we show that every DIFFICULT graph can be transformed into an EASY one by repeated applications of a subdivision operation on vertices. The idea is to use this transformation to transfer the colouring of the EASY graph to the DIFFICULT one. The transformation increases the number of vertices but keeps the number of edges constant. We were able to colour some DIFFICULT graphs with this technique, and we show these results in the sequel. Specifically, we decided the cases where one subdivision operation is enough.

2 Definitions and notations

General terms

In this paper, G denotes a simple, undirected, finite, connected graph. $V(G)$ and $E(G)$ are the vertex and edge sets of G. A *stable* set is a set of vertices pairwise non-adjacent in G. A *clique* is a set of vertices pairwise adjacent in G. A *maximal* clique of G is a clique not properly contained in any other clique. A *maximum* clique of G is a clique of maximum size, i.e., a clique with the largest possible number of vertices.

A *subgraph* of G is a graph H with $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. For $X \subseteq V(G)$, we denote by $G[X]$ the *subgraph induced by* X, that is, $V(G[X]) = X$ and $E(G[X])$ consists of those edges of $E(G)$ having both ends in X.

For each vertex v of a graph G, $Adj(v)$ denotes the set of vertices that are adjacent to v. In addition, $N(v)$ denotes the *neighbourhood* of v, that is, $N(v) = Adj(v) \cup \{v\}$. The *degree* of a vertex v is $\deg(v) = |\mathrm{Adj}(v)|$. The maximum degree of a graph G is then $\Delta(G) = \max_{v \in V(G)} \deg(v)$. A vertex u is *universal* if $\deg(u) = |V(G)| - 1$. A vertex u is *quasi-universal* if $\deg(u) = |V(G)| - 2$. Given a graph G and $k \geq 1$, we denote by G^k the graph having $V(G^k) = V(G)$ and satisfying $xy \in E(G^k)$ if and only if x and y are distinct and their distance in G is at most k. The *diameter* of G is $diam(G) = \max_{v,w \in V(G)} dist(v, w)$.

K_n denotes the complete graph on $n \geq 1$ vertices.

A *split graph* is a graph whose vertex set admits a partition into a stable set and a clique.

Colouring

An *assignment of colours to the vertices* of G is a function $\lambda \colon V(G) \to S$. The elements of the set S are called *colours*. A *conflict* in an assignment of colours is the existence of two adjacent vertices with the same colour. A *vertex-colouring* of a graph is an assignment of colours such that there are no conflicts. The *chromatic number* of a graph G is the minimum number of colours used among all vertex-colourings of G and is denoted by $\chi(G)$.

An *assignment of colours to the edges* of G is a function $\kappa: E(G) \rightarrow S$. Again, the elements of the set S are called *colours*. A *conflict* in an assignment of colours is the existence of two edges with the same colour incident to a common vertex. A vertex u is said to be *satisfied* when $\kappa(uv) = \kappa(uw)$ implies $v = w$, for all neighbours v, w of u. An *edge-colouring* of a graph is an assignment of colours such that every vertex is satisfied or, equivalently, such that there are no conflicts. The *chromatic index* of a graph G is the minimum number of colours used among all edge-colourings of G and is denoted by $\chi'(G)$.

A graph G is said to be *Class* 1 if $\chi'(G) = \Delta(G)$ and *Class* 2 if $\chi'(G) = \Delta(G) + 1$. Vizing's theorem [1] states that there are no other possibilities: all graphs are either Class 1 or Class 2. A constructive proof of Vizing's theorem appeared in [8].

3 EASY and DIFFICULT

Recall that *split graph* is a graph G for which $V(G)$ can be partitioned into two sets A and B such that A is a clique and B is a stable set. A *splitting clique* A of G is a maximal clique such that $V(G) \setminus A$ is a stable set of G. Note that every splitting clique is a clique of maximum size in G.

We concentrate on split graphs of odd maximum degree Δ, as mentioned in the introduction. We now define a subclass of odd maximum degree split graphs called EASY, which contains all graphs we consider easy to colour with Δ colours. An odd maximum degree split graph G is said to be EASY if $\chi(G^2) = \Delta(G) + 1$.

Theorem 1 EASY $\subseteq$ *Class 1.*

The proof is based on the concept of a pullback function. A *pullback* from graph G to graph H is a function $f: V(G) \rightarrow V(H)$ such that

- f is a homomorphism, i.e., f takes adjacent vertices of G into adjacent vertices of H;
- f is injective when restricted to each neighbourhood.

The following results relate edge-colouring to pullbacks. They are proved in [5].

Theorem 2 *If f is a pullback from G to H and H is k-edge-colourable, then G is k-edge-colourable.*

Theorem 3 *Composition of pullbacks is a pullback.*

Any graph satisfying $\chi(G^2) = \Delta(G) + 1$ admits a pullback into the complete graph $K_{\Delta+1}$. In particular, EASY graphs can be coloured with Δ colours because they admit a pullback into the graph $K_{\Delta+1}$, a complete graph with an even number of vertices.

We now define the subclass of DIFFICULT graphs. An odd maximum degree split graph G is said to be DIFFICULT when:

- all vertices in a splitting clique A of G have degree $\Delta(G)$;
- $diam(G) \leq 2$.

The importance of this class is justified by Theorem 4 below. Together with Theorem 2 they imply that if we prove that DIFFICULT $\subseteq$ Class 1, then all odd maximum degree split graphs are Class 1.

Theorem 4 *If G is any odd maximum degree split graph, then there is a* DIFFICULT *graph H with $\Delta(G) = \Delta(H)$ and a pullback function $f\colon V(G) \to V(H)$.*

Proof: Suppose G is a split graph with $diam(G) > 2$. Let $V(G)$ be partitioned into a splitting clique A and a stable set B. If there is a vertex v in A with $\deg(v) < \Delta$, then we add an edge between v and any vertex u not adjacent to v. This addition does not change the maximum degree because $\deg(u) < \Delta$. If u now sees every vertex of A, then $A' = A \cup \{u\}$ is now a splitting clique with corresponding stable set $B' = B \setminus \{u\}$. By repeating this operation as many times as necessary, we eventually obtain a graph satisfying the first condition in the definition of DIFFICULT.

If at this point the diameter is greater than 2, we take any two vertices x and y with $dist(x, y) = 3$ and join them into a single vertex z. If vertex z sees every vertex of A, then $A' = A \cup \{z\}$ is now a splitting clique with corresponding stable set $B' = B \setminus \{z\}$. If necessary, we add edges incident to z until we obtain a graph satisfying the first condition in the definition of DIFFICULT.

Furthermore, there is a pullback function from the old graph into the new one that maps x and y into z and maps the other vertices into themselves. By repeating this operation as many times as necessary, we eventually obtain a graph with diameter at most 2. The final pullback is just the composition of all the previous pullbacks. ■

4 Transforming DIFFICULT into EASY

Given a graph G and an integer k, we define $remain(G, k)$ as the least number of vertices of G whose removal leaves a graph that admits a vertex-colouring with k colours. Note that if G is an EASY graph, then $remain(G^2, \Delta + 1) = 0$. On the other hand, given a DIFFICULT graph G such that $r = remain(G^2, \Delta + 1) > 0$, the removal of any stable set of size r from G leaves a graph H with $remain(H^2, \Delta + 1) = 0$.

Given a graph G, a *vertex subdivision* operation transforms G into a graph $s(G)$ and is defined as follows. Let $v \in V(G)$ and let $Adj(v)$ be partitioned into sets $V_1, \cdots, V_t$. In $s(G)$, we have one new vertex v_i, for each set V_i. The vertex and edge sets are defined as: $V(s(G)) = V(G) \setminus \{v\} \cup \{v_1, \cdots, v_t\}$ and $E(s(G)) = (E(G) \setminus \{vw : w \in Adj(v)\}) \cup \{v_i u_i : u_i \in V_i\}$. We say that vertex v is *subdivided* into vertices $v_1, \ldots, v_t$.

The main result of this section shows how to transform a DIFFICULT graph into an EASY graph.

Theorem 5 *For every* DIFFICULT *graph, there is a series of subdivision operations that transforms it into an* EASY *graph.*

Proof: Let G be a DIFFICULT graph with $remain(G^2, \Delta+1) > 0$. We shall prove that there is a subdivision operation that transforms G into $s(G)$ such that: $remain(s(G)^2, \Delta+1) < remain(G^2, \Delta+1)$.

Let $r = remain(G^2, \Delta+1) > 0$. By definition, there exists a set R of r vertices of G whose removal leaves a graph G' such that $(G')^2$ admits a vertex-colouring C with $\Delta+1$ colours. Let $v \in R$. Define the following subsets of $Adj(v)$:

$$PreV_i = \{n \in Adj(v) : C(n) \neq i; n \in Adj(w) \Rightarrow C(w) \neq i\}.$$

We note that $Adj(v) = \cup PreV_i$. Indeed, if $x \in Adj(v)$ but $x \notin \cup PreV_i$, then $N(x) \cap C_i \neq \emptyset$, for $1 \leq i \leq \Delta+1$, and $Adj(x) \cap R \neq \emptyset$. This implies $deg(x) \geq \Delta+1$, a contradiction.

Some sets $PreV_i$ are possibly empty and the sets $PreV_i$ are not necessarily disjoint. Now there exists a collection of sets $V_1, \ldots, V_t$ that partitions $Adj(v)$ and such that $V_j \subseteq PreV_i$, for $1 \leq i \leq \Delta+1$. Moreover, there are at least two distinct indices i and j with V_i and V_j not empty. If only one V_i is not empty, then it is possible to colour v. Then, this partition defines a subdivision of vertex v which in turn defines the required graph $s(G)$. ■

5 When the graph is almost EASY

Let G be a DIFFICULT graph, with the usual partition of $V(G)$ into sets A and B. Let $|A| = a$ and $|B| = b$. In this section, we solve the case $remain(G^2, \Delta+1) = 1$. We show how to use a subdivision operation together with a pullback to get the required edge-colouring of G with Δ colours.

Let v be any vertex of B. Label the vertices of $B \setminus v$ with labels $1, \ldots, s$. We note that we have the following relations: $s = b-1$, $\Delta = a-1+b-1 = a+b-2$.

As in the proof of Theorem 5, subdivide vertex v into vertices $v_1, \ldots, v_s$. The set $Adj(v)$ is partitioned into $V_1, \ldots, V_s$, accordingly. We note that graphs G and $s(G)$ have the same number of edges. In order to obtain the required edge-colouring of G, we shall first obtain an edge-colouring of $s(G)$ with Δ colours.

Now we proceed to exhibit a pullback from $s(G)$ to $K_{\Delta+1}$.

Vertices of $B \setminus v$ continue to have labels $1, \ldots, s$. Now label v_i with label i $(1 \leq i \leq s)$, and label vertices in $V_1, \ldots, V_s$, with labels $s+1, s+2, \ldots, s+deg(v)$, following the ordering of the subscripts of the sets V_i.

Note that $s \leq b-1$, because $v \in B$. On the other hand, $deg(v) \leq a-1$, because no vertex of B sees all vertices of A. Hence $s + deg(v) \leq a+b-2 = \Delta$. Finally, to finish the pullback, label the vertices of $A \setminus Adj(v)$ with labels not used so far.

To see that this edge-colouring of $s(G)$ is in fact an edge-colouring of G, consider the edge joining vertex $x \in V_i$ to vertex v_i. Let us compute the size of the interval of the sums $label(x) + i$, where $x \in V_i$ and $1 \leq i \leq s$. The minimum possible is $1+s+1 = s+2$. The maximum possible is $s+s+deg(v) = 2s+deg(v)$. Hence this interval has size: $2s+deg(v)-(s+2)+1 = s+deg(v)-1$. Now $s+deg(v) \leq \Delta$ implies that the size of this interval is at most $\Delta-1$. In particular, we get that these values belong to distinct classes modulo Δ.

Therefore, we have the following theorem:

Theorem 6 *Let G be a split graph with odd maximum degree. Suppose that G satisfies $remain(G^2, \Delta + 1) \leq 1$. Then G is Class 1.*

6 Conclusions

We have reported our progress on edge-colouring odd maximum degree graphs. Since it is NP-complete to decide [9] whether an odd maximum degree graph is Class 1, we have been solving the problem for special classes: doubly chordal graphs [5] and indifference graphs [6]. We conjectured in [7] that every odd maximum degree split graph is Class 1.

In this note we have given positive evidence for this conjecture by proving it for special cases. Our method defines two functions: subdivision, which maps a split graph into a particular Class 1 split graph; and pullback, a special kind of homomorphism.

Acknowledgements. This work was done while the authors were visiting the Department of Combinatorics and Optimization at the University of Waterloo, Canada. The first author holds a post-doctoral grant from CNPq Proc. 204383/88-9. The second and third authors acknowledge support from CNPq/ProTeM-CC-II, Project ProComb - Proc. 680065/94-6 and FAPESP.

References

[1] V. G. Vizing. On an estimate of the chromatic class of a p-graph. *Diket. Analiz.*, 3:25–30, 1964. In russian.

[2] M. Plantholt. The chromatic index of graphs with a spanning star. *J. of Graph Theory*, 5:45–53, 1981.

[3] M. Plantholt. The chromatic index of graphs with large maximum degree. *Discrete Math.*, 47:91–96, 1983.

[4] D. G. Hoffman and C. A. Rodger. The chromatic index of complete multipartite graphs. *J. of Graph Theory*, 16:159–163, 1992.

[5] C. M. H. de Figueiredo, J. Meidanis, and C. P. de Mello. A greedy method for edge-colouring odd maximum degree doubly chordal graphs. *Congressus Numerantium*, 111:170–176, 1995.

[6] C. M. H. de Figueiredo, J. Meidanis, and C. P. de Mello. On edge-colouring indifference graphs. *Lecture Notes in Computer Science*, 911:286–299, 1995.

[7] C. M. H. de Figueiredo, J. Meidanis, and C. P. de Mello. Local conditions for edge-coloring. Technical report, DCC 95-17, UNICAMP, 1995. Anais da II Oficina Nacional em Problemas Combinatórios.

[8] J. Misra and D. Gries. A constructive proof of Vizing's theorem. *Inf. Proc. Lett.*, 41:131–133, 1992.

[9] I. Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10:718–720, 1981.

Relatórios Técnicos – 1995

95-01 **Paradigmas de algoritmos na solução de problemas de busca multidimensional,** *Pedro J. de Rezende, Renato Fileto*

95-02 **Adaptive enumeration of implicit surfaces with affine arithmetic,** *Luiz Henrique de Figueiredo, Jorge Stolfi*

95-03 **W3 no Ensino de Graduação?,** *Hans Liesenberg*

95-04 **A greedy method for edge-colouring odd maximum degree doubly chordal graphs,** *Celina M. H. de Figueiredo, João Meidanis, Célia Picinin de Mello*

95-05 **Protocols for Maintaining Consistency of Replicated Data,** *Ricardo Anido, N. C. Mendonça*

95-06 **Guaranteeing Full Fault Coverage for UIO-Based Methods,** *Ricardo Anido and Ana Cavalli*

95-07 **Xchart-Based Complex Dialogue Development,** *Fábio Nogueira de Lucena, Hans K.E. Liesenberg*

95-08 **A Direct Manipulation User Interface for Querying Geographic Databases,** *Juliano Lopes de Oliveira, Claudia Bauzer Medeiros*

95-09 **Bases for the Matching Lattice of Matching Covered Graphs,** *Cláudio L. Lucchesi, Marcelo H. Carvalho*

95-10 **A Highly Reconfigurable Neighborhood Image Processor based on Functional Programming,** *Neucimar J. Leite, Marcelo A. de Barros*

95-11 **Processador de Vizinhança para Filtragem Morfológica,** *Ilka Marinho Barros, Roberto de Alencar Lotufo, Neucimar Jerônimo Leite*

95-12 **Modelos Computacionais para Processamento Digital de Imagens em Arquiteturas Paralelas,** *Neucimar Jerônimo Leite*

95-13 **Modelos de Computação Paralela e Projeto de Algoritmos,** *Ronaldo Parente de Menezes e João Carlos Setubal*

95-14 **Vertex Splitting and Tension-Free Layout,** *P. Eades, C. F. X. de Mendonça N.*

95-15 **NP-Hardness Results for Tension-Free Layout,** *C. F. X. de Mendonça N., P. Eades, C. L. Lucchesi, J. Meidanis*

95-16 **Agentes Replicantes e Algoritmos de Eco,** *Marcos J. C. Euzébio*

95-17 **Anais da II Oficina Nacional em Problemas Combinatórios: Teoria, Algoritmos e Aplicações,** *Editores: Marcus Vinicius S. Poggi de Aragão, Cid Carvalho de Souza*

95-18 **Asynchronous Teams: A Multi-Algorithm Approach for Solving Combinatorial Multiobjective Optimization Problems,** *Rosiane de Freitas Rodrigues, Pedro Sérgio de Souza*

95-19 **wxWindows: Uma Introdução,** *Carlos Neves Júnior, Tallys Hoover Yunes, Fábio Nogueira de Lucena, Hans Kurt E. Liesenberg*

95-20 **John von Neumann: Suas Contribuições à Computação,** *Tomasz Kowaltowski*

95-21 **A Linear Time Algorithm for Binary Phylogeny using PQ-Trees,** *J. Meidanis and E. G. Munuera*

95-22 **Text Structure Aiming at Machine Translation,** *Horacio Saggion and Ariadne Carvalho*

95-23 **Cálculo de la Estructura de un Texto en un Sistema de Procesamiento de Lenguaje Natural,** *Horacio Saggion and Ariadne Carvalho*

95-24 **ATIFS: Um Ambiente de Testes baseado em Inje,c ao de Falhas por Software,** *Eliane Martins*

95-25 **Multiware Plataform: Some Issues About the Middleware Layer,** *Edmundo Roberto Mauro Madeira*

95-26 **WorkFlow Systems: a few definitions and a few suggestions,** *Paulo Barthelmess and Jacques Wainer*

95-27 **Workflow Modeling,** *Paulo Barthelmess and Jacques Wainer*

Relatórios Técnicos – 1996

96-01 **Construção de Interfaces Homem-Computador: Uma Proposta Revisada de Disciplina de Graduação,** *F'abio Nogueira Lucena and Hans K.E. Liesenberg*

96Abs **DCC-IMECC-UNICAMP Technical Reports 1992–1996 Abstracts,** *C. L. Lucchesi and P. J. de Rezende and J.Stolfi*

96-02 **Automatic visualization of two-dimensional cellular complexes,** *Rober Marcone Rosi and Jorge Stolfi*

Instituto de Computação
Universidade Estadual de Campinas
13081-970 – Campinas – SP
BRASIL

reltec@dcc.unicamp.br