



Coloring Meyniel graphs in linear time

Benjamin Lévêque¹, Frédéric Maffray²

*Laboratoire Leibniz-IMAG
46 avenue Félix Viallet
38031 Grenoble Cedex, France*

Abstract

A Meyniel graph is a graph in which every odd cycle of length at least five has two chords. We present a linear-time algorithm that colors optimally the vertices of a Meyniel graph and finds a clique of maximum size.

Keywords: Perfect graphs, Meyniel graphs, coloring, algorithm

A *coloring* of a graph is a mapping that assigns one color to each vertex in such a way that any two adjacent vertices receive distinct colors. A coloring is *optimal* if it uses as few colors as possible. The *chromatic number* $\chi(G)$ of a graph G is the number of colors used by an optimal coloring. We consider the graphs in which every odd cycle of length at least five has at least two chords. Meyniel [9] and, independently, Markosyan and Karapetyan [8], proved that such graphs are perfect, and it has become customary to call them *Meyniel graphs*. We are interested in polynomial-time algorithms that color optimally the vertices of a Meyniel graph G with n vertices and m edges. Hoàng [7] gave such an algorithm, with complexity $\mathcal{O}(n^8)$, which uses the amalgam decomposition from Burlet and Fonlupt [2]. Hertz [5] devised a coloring algorithm that works for general graphs in time $\mathcal{O}(nm)$ and, when the input graph is

¹ Email: benjamin.leveque@imag.fr

² Email: frederic.maffray@imag.fr

Meyniel, produces an optimal coloring. The optimality is based on the concept of even pair [3]: at each step Hertz’s algorithm finds an even pair of vertices in the graph and contracts them. Roussel and Rusu [12] defined a coloring algorithm called LEXCOLOR (for Lexicographic Color) which works for general graphs in time $\mathcal{O}(n^2)$. In the case of a Meyniel graph, this algorithm colors the vertices optimally. It does not contract even pairs but “simulates” such a contraction, and its optimality follows from that of Hertz’s algorithm. LEXCOLOR is based on LEXBFS (for Lexicographic Breadth First Search), which was originally invented by Rose, Tarjan and Lueker [10] to find a simplicial elimination ordering in a chordal graph.

We propose an algorithm that we call MCCOLOR, which will color any graph in time $\mathcal{O}(n + m)$. This algorithm is a simplification of Roussel and Rusu’s LEXCOLOR. Just like LEXCOLOR, in the case of a Meyniel graph, MCCOLOR produces an optimal coloring by “simulating” the contraction of even pairs. MCCOLOR is based on the algorithm MCS (for Maximum Cardinality Search) due to Tarjan and Yannakakis, [13], which is a simplification of LEXBFS and can also be used to find a simplicial elimination ordering in a chordal graph. MCCOLOR can also be viewed as a simplified version of the coloring heuristic DSATUR (for Degree Saturation) due to Brélaz [1].

Our algorithm is a rather simple version of the greedy coloring algorithm. Colors are viewed as integers $1, 2, \dots$. At each step, the algorithm considers, for every uncolored vertex x , the number of colors that appear in the neighbourhood of x , selects an uncolored vertex for which this number is maximum, assigns to this vertex the smallest color not present in its neighbourhood, and iterates this procedure until every vertex is colored.

ALGORITHM MCCOLOR

Input: A graph G with n vertices.

Output: A coloring of the vertices of G .

Initialization: For every vertex x of G do $label(x) := \emptyset$;

General step: For $i = 1, \dots, n$ do:

1. Choose an uncolored vertex x that maximizes $|label(x)|$;
2. Color x with the smallest color in $\{1, 2, \dots, n\} \setminus label(x)$;
3. For every uncolored neighbour y of x , add $color(x)$ to $label(y)$.

Algorithm MCCOLOR can be implemented in time $\mathcal{O}(n + m)$.

An *even pair* [3] in a graph G is a pair of non-adjacent vertices such that every chordless path between them has even length (number of edges). Given two vertices x, y in G , the operation of *contracting* them means removing x and y and adding one vertex with an edge to each neighbour of x or y .

Fonlupt and Uhry [4] proved that the graph G' obtained from any graph G by contracting an even pair of G satisfies $\omega(G') = \omega(G)$ and $\chi(G') = \chi(G)$, where $\omega(G)$ is the maximum size of a clique in G . The optimality of our algorithm is based on the same results as for the algorithm of Roussel and Rusu [12]. It can be shown that each color class produced by the algorithm corresponds to the contraction of some even pairs. Thus Fonlupt and Uhry's theorem implies that Algorithm MCCOLOR colors every Meyniel graph G with $\omega(G)$ colors. This shows the optimality of the algorithm (and also the perfectness of G).

Theorem *Algorithm MCCOLOR colors a Meyniel graph G with exactly $\omega(G)$ colors.*

We can extend our algorithm with the following :

ALGORITHM Q:

Input: A graph G and a coloring of its vertices using l colors.

Output: A set Q of l vertices of G .

Initialization: Set $Q := \emptyset$, $c := l$, and $q(x) := 0$ for every vertex x ;

General step: While $c \neq 0$ do:

Pick a vertex x of color c that maximizes $q(x)$, do $Q := Q \cup \{x\}$,

for every neighbour y of x do $q(y) := q(y) + 1$, and do $c := c - 1$.

Algorithm Q can be implemented in time $\mathcal{O}(m + n)$. Moreover, when the input consists of a Meyniel graph G with the coloring produced by MCCOLOR, the output is a clique of size l .

The coloring produced by Algorithm MCCOLOR has a few properties worth pointing out. (a) Every vertex of any color c lies in a clique of size c . More generally, every clique whose smallest color is c is included in a clique that contains a vertex of each color $1, \dots, c$. This is called a *strongly canonical* coloring in [6]. (b) The vertices of color 1 form a stable set that intersects all maximal cliques of G . Such a set is called a *good stable set*. Moreover, since any vertex can be used as the first vertex in MCCOLOR, we obtain that every vertex of a Meyniel graph lies in a good stable set; this is one of the main theoretical results in [7], where an $\mathcal{O}(n^7)$ algorithm was given to find such a set, while our algorithm produces a good stable set in time $\mathcal{O}(m + n)$.

Our algorithm is “robust” in the sense that if the input graph is not Meyniel and the output coloring is not optimal, it can detect this fault. To do this we apply MCCOLOR followed by Algorithm Q and we check (in linear time) whether Q is a clique. If Q is a clique, the coloring is optimal since it uses l colors and Q has size l . If Q is not a clique, the input graph is not Meyniel since our algorithm is optimal on Meyniel graphs.

References

- [1] Brélaz, D., *New methods to color the vertices of a graph*, *Comm. ACM* **22**(4) (1979), 251–256.
- [2] Burlet, M., J. Fonlupt, *Polynomial algorithm to recognize a Meyniel graph*, *Ann. Disc. Math.* **21** (1984), 225–252.
- [3] Everett, H., C.M.H. de Figueiredo, C. Linhares Sales, F. Maffray, O. Porto, B.A. Reed, *Even pairs*, In: “Perfect Graphs”, J.L. Ramírez-Alfonsín and B.A. Reed, eds. Wiley Interscience, 2001, 67–92.
- [4] Fonlupt, J., J.P. Uhry, *Transformations which preserve perfectness and h -perfectness of graphs*, *Ann. Disc. Math.* **16** (1982), 83–85.
- [5] Hertz, A., *A fast algorithm for coloring Meyniel graphs*, *J. Comb. Th. B* **50** (1990), 231–240.
- [6] Hertz, A., D. de Werra, *Connected sequential colorings*, *Disc. Math.* **74** (1989), 51–59.
- [7] Hoàng, C.T., *On a conjecture of Meyniel*, *J. Comb. Th. B* **42** (1987), 302–312.
- [8] Markosyan, S.E., I.A. Karapetyan, *Perfect graphs*, *Akad. Nauk Armjan. SSR Dokl.* **63** (1976), 292–296.
- [9] Meyniel, H., *The graphs whose odd cycles have at least two chords*, *Ann. Disc. Math.* **21** (1984), 115–119.
- [10] Rose, D.J., R.E. Tarjan, G.S. Lueker, *Algorithmic aspects of vertex elimination of graphs*, *SIAM J. Comput.* **5** (1976), 266–283.
- [11] Roussel, F., I. Rusu, *Holes and dominoes in Meyniel graphs*, *Int. J. Found. Comput. Sci.* **10** (1999), 127–146.
- [12] Roussel, F., I. Rusu, *An $O(n^2)$ algorithm to color Meyniel graphs*, *Disc. Math.* **235** (2001), 107–123.
- [13] Tarjan, R.E., M. Yannakakis, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, *SIAM J. Comput.* **13** (1984), 566–579.