Jagiellonian University
Department of Theoretical Computer Science

Adrian Siwiec

Perfect Graph Recognition and Coloring

Master Thesis

Supervisor: dr inż. Krzysztof Turowski

September 2020

Abstract

TODO

Contents

1 Perfect Graphs

All graphs in this paper are finite, undirected and have no loops or parallel edges. We denote the chromatic number of graph G by $\chi(G)$ and the cardinality of the largest clique of G by $\omega(G)$. *Coloring* of a graph means assigning every node of a graph a color. A coloring is *valid* iff every two nodes sharing an edge have different colors. An *optimal* coloring (if exists) is a valid coloring using only $\omega(G)$ colors.

Given a graph $G = (V, E)$, sometimes by $V(G)$ and $E(G)$ we will denote a set of nodes and edges of G. Given a set $X \subseteq V$ by $G[X]$ we will denote a graph induced on X. A graph G is *perfect* iff for all $X \subseteq V(G)$ we have $\chi(G[X]) = \omega(G[X])$.

Editorial notes By solid lines we will mark edges, by dashed lines we will mark nonedges, when significant. Sometimes nonedges will not be marked in order not to clutter the image.

Where should we put it?

Give some examples why are these interesting, some subclasses, and problems that are solvable for perfect graphs, including recognition and coloring

Given a graph G, its *complement* $\overline{G}$ is a graph with the same vertex set and in which two distinct nodes u, v are connected in $\overline{G}$ iff they are not connected in G. For example a clique in a graph becomes an independent set in its complement. A perfect graph theorem, first conjured by Berge in 1961 [**CB61**] and then proven by Lovász in 1972 [**LL72**] states that a graph is perfect iff its complement graph is also perfect.

Should we give some proof of that here? Maybe based on proof in [**GC03**]

A *hole* is an induced chordless cycle of length at least 4. An *antihole* is an induced subgraph whose complement is a hole. A *Berge* graph is a graph with no holes or antiholes of odd length.

In 1961 Berge conjured that a graph is perfect iff it is Berge in what has become known as a strong perfect graph conjecture. In 2001 Chudnovsky et al. have proven it and published the proof in an over 150 pages long paper **MC06** [**MC06**]. The following overview of the proof will be based on this paper and on an article withe the same name by Cornuéjols [**GC03**].

1.1 Strong Perfect Graph Theorem

Odd holes are not perfect, since their chromatic number is 3 and their largest cliques are of size 2. It is easy to see, that an odd antihole of size n has a chromatic number of $\frac{n+1}{2}$ and largest cliques of size $\frac{n-1}{2}$. It is therefore clear, that if a graph is not Berge it is not perfect. To prove that every Berge graph is perfect is the proper part of the strong perfect graph theorem.

How long and detailed overview of the proof should we provide?

2 Recognizing Berge Graphs

Cite the paper and tell this is only a short overview

2.1 Recognition algorithm Overview

Main ideas of the algorithm.

First we check all on G, then on $\overline{G}$

2.1.1 Simple structures

Pyramids A *path* in G is an induced subgraph that is connected, with at least one node, no cycle and no node of degree larger than 2 (sometimes called chordless path). The *length* of a path or a cycle is the number of edges in it. A *triangle* in a graph is a set of three pairwise adjacent nodes.

Should we move these definitions elsewhere?

A *pyramid* in G is an induced subgraph formed by the union of a triangle $\{b_1, b_2, b_3\}$, three paths $\{P_1, P_2, P_3\}$ and another node a, so that:

- $\forall_{1 \le i \le 3}$ P_i is a path between a and b_i
- $\forall_{1 \le i < j \le 3}$ a is the only node in both P_i and P_j and $b_i b_j$ is the only edge between $V(P_i) \setminus \{a\}$ and $V(P_j) \setminus \{a\}$.
- a is adjacent to at most one of $\{b_1, b_2, b_3\}$.

It is easy to see that every graph containing a pyramid contains an odd hole – at least two of the paths P_1, P_2, P_3 will have the same parity.

On recognition of pyramids. Lemma 2.1 from the paper.

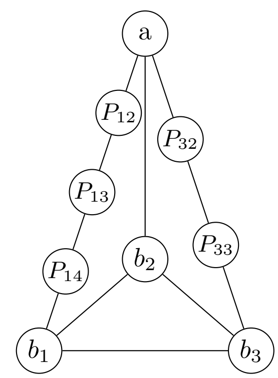

Figure 1: An example of a pyramid.

Jewels Five nodes $v_1, \ldots, v_5$ and a path P is a *jewel* iff:

- $v_1, \ldots, v_5$ are distinct nodes.
- $v_1v_2, v_2v_3, v_3v_4, v_4v_5, v_5v_1$ are edges.
- v_1v_3, v_2v_4, v_1, v_4 are nonedges.
- P is a path between v_1 and v_4, such that v_2, v_3, v_5 have no neighbors in its inside.

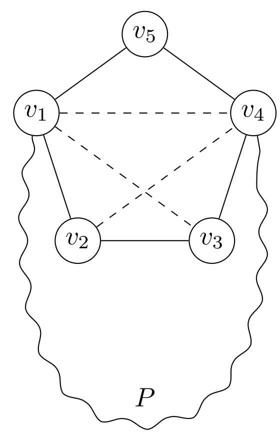

Figure 2: An example of a jewel.

Recognizing simple structures (Pyramids, Jewels, T1, T2, T3).

Finding and Using Half-Cleaners.

Overview of proof of why algorithm using Half-Cleaners is correct.

2.2 Implementation

Anything interesting about algo/data structure?

Optimizations - Bottlenecks in performance (next path, are vectors distinct etc).

Validity tests - unit tests, tests of bigger parts, testing vs known answer and vs naive.

2.3 Parallelism with CUDA (?)

TODO

2.4 Experiments

Naive algorithm - brief description, bottlenecks optimizations (makes huge difference).

Description of tests used.

Results and Corollary - almost usable algorithm.

3 Coloring Berge Graphs

3.1 Ellipsoid method

Desctiption.

Implementation.

Experiments and results.

3.2 Combinatorial Method

Cite the paper.

On its complexity - point to appendix for pseudo-code.

Appendices

A Perfect Graph Coloring algorithm

TODO