Discrete Optimization 21 (2016) 1–13

Contents lists available at ScienceDirect

Discrete Optimization

www.elsevier.com/locate/disopt

Polyhedral studies of vertex coloring problems: The standard formulation

Diego Delle Donne[a,b,c,*], Javier Marenco[a,b]

[a] Sciences Institute, National University of General Sarmiento, Buenos Aires, Argentina
[b] Computer Sc. Department, FCEN, University of Buenos Aires, Buenos Aires, Argentina
[c] LIMOS, Blaise Pascal University, Clermont Ferrand, France

A R T I C L E I N F O

Article history:
Received 2 June 2015
Received in revised form 30 April 2016
Accepted 2 May 2016
Available online 30 May 2016

Keywords:
Vertex coloring
Standard formulation
Polyhedral characterization

A B S T R A C T

Despite the fact that many vertex coloring problems are polynomially solvable on certain graph classes, most of these problems are not "under control" from a polyhedral point of view. The equivalence between optimization and separation suggests the existence of integer programming formulations for these problems whose associated polytopes admit elegant characterizations. In this work we address this issue. As a starting point, we focus our attention on the well-known *standard formulation* for the classical vertex coloring problem. We present some general results about this formulation and we show that the vertex coloring polytope associated to this formulation for a graph G and a set of colors C corresponds to a face of the *stable set polytope* of a particular graph $\mathcal{S}_G^C$. We further study the perfectness of $\mathcal{S}_G^C$ showing that when $|C| > 2$, this graph is perfect if and only if G is a block graph, from which we deduce a complete characterization of the associated coloring polytopes for block graphs. We also derive a new family of valid inequalities generalizing several known families from the literature and we conjecture that this family is sufficient to completely describe the vertex coloring polytope associated to cacti graphs.

1. Introduction

Given a graph $G = (V, E)$, a *coloring* of G is an assignment $c : V \to \mathbb{N}$ of "colors" to vertices of G, such that $c(v) \neq c(w)$ for each edge $vw \in E$. The *vertex coloring problem* consists in finding a coloring of G minimizing the number of used colors. This parameter is widely known as the *chromatic number* of G, and is denoted as $\chi(G)$. There are many variants of the graph coloring problem, motivated by real-life constraints; the following are some examples:

* Corresponding author at: Sciences Institute, National University of General Sarmiento, Buenos Aires, Argentina.
E-mail addresses: ddelledo@ungs.edu.ar (D. Delle Donne), jmarenco@ungs.edu.ar (J. Marenco).

http://dx.doi.org/10.1016/j.disopt.2016.05.001

Table 1
Known complexities on some graph classes [3,7].

Class	Coloring	Precol	μ-col	(γ, μ)-col	List-col
Bipartite	P	NP-C	NP-C	NP-C	NP-C
Distance-hereditary	P	NP-C	NP-C	NP-C	NP-C
Interval	P	NP-C	NP-C	NP-C	NP-C
Unit interval	P	NP-C	NP-C	NP-C	NP-C
Line of $K_{n,n}$	P	NP-C	NP-C	NP-C	NP-C
Line of K_n	P	NP-C	NP-C	NP-C	NP-C
Split	P	P	NP-C	NP-C	NP-C
Complements of bipartites	P	P	?	?	NP-C
Cographs	P	P	P	?	NP-C
Complete bipartite	P	P	P	P	NP-C
Complete split	P	P	P	P	NP-C
Trees	P	P	P	P	P
Block	P	P	P	P	P
Cacti	P	P	P	P	P

"NP-C": NP-complete problem. "P": polynomial problem. "?": open problem.

Precoloring extension [1]: Given a graph $G = (V, E)$ and a partial assignment $\rho : V \to \mathbb{N}$, this generalization of the classical vertex coloring problem asks for a coloring c with the smallest number of used colors such that $c(v) = \rho(v)$ for every vertex v in the domain of ρ. In other words, a subset of vertices from G is already colored and the problem is to extend this coloring in a minimum fashion.

μ-coloring [2]: This generalization of the classical problem takes as additional input a function $\mu : V \to \mathbb{N}$ defining an upper bound $\mu(v)$ for the color assigned to each vertex v, i.e., the obtained coloring $c : V \to \mathbb{N}$ must satisfy $c(v) \leq \mu(v)$, for every $v \in V$.

(γ, μ)-coloring [3]: Besides the upper-bounding function $\mu : V \to \mathbb{N}$, this generalization of the μ-coloring problem considers also a function $\gamma : V \to \mathbb{N}$ which establishes lower bounds on the assignments for the vertices of G. Now, a coloring $c : V \to \mathbb{N}$ is asked for in such a way that $\gamma(v) \leq c(v) \leq \mu(v)$ holds for every $v \in V$. Note that this problem is also a generalization of precoloring extension.

List coloring [4]: This problem considers a set $L(v)$ of valid colors for each $v \in V$ and asks for a coloring c such that $c(v) \in L(v)$ for all $v \in V$. This version generalizes all the problems mentioned above.

There exist in the literature many other variants of the classical vertex coloring problem considering local constraints (see, for example, [4]). Although the classical vertex coloring problem is NP-hard [5], there are many graph classes for which this problem can be solved in polynomial time, one of the most important classes being perfect graphs [6]. A graph G is said to be *perfect* if $\chi(H) = \omega(H)$ for every induced subgraph H of G, where $\omega(H)$ represents the size of the maximum clique of H. However, the variants of the coloring problem mentioned above may not be polynomially solvable for perfect graphs, therefore it is interesting to study the computational complexities of these variants on subclasses of perfect graphs. In [3,7], the complexity boundary between coloring and list-coloring is studied for several subclasses of perfect graphs. Table 1 shows a summary of known complexities for the graph classes studied in these previous works.

Integer linear programming (ILP) has proved to be a very suitable tool for solving combinatorial optimization problems [8], and in the last decade ILP has been successfully applied to graph coloring problems, by resorting to several formulations for the classical vertex coloring problem. The following are some examples of ILP formulations for this problem:

Standard model [9–11]: This model includes a binary variable x_{ic} for each vertex $i \in V$ and each color $c \in C$, where C represents the set of available colors, asserting whether vertex i is assigned color c or not. This formulation may be extended with variables w_c for each color $c \in C$ specifying whether this color is used or not; the minimum coloring is found by minimizing the sum of these variables.