Colouring perfect graphs with bounded clique number

Maria Chudnovsky¹
Princeton University, Princeton, NJ 08544

Aurélie Lagoutte²
LIP, UMR 5668, ENS Lyon, CNRS, UCBL, INRIA, Université de Lyon, France

Paul Seymour³
Princeton University, Princeton, NJ 08544

Sophie Spirkl
Princeton University, Princeton, NJ 08544

November 4, 2015; revised December 13, 2015

¹Supported by NSF grant DMS-1550991.
²Partially supported by ANR Grant STINT:ANR-13-BS02-0007, and performed while visiting Princeton University.
³Supported by ONR grant N00014-14-1-0084 and NSF grant DMS-1265563.
Abstract

A graph is perfect if the chromatic number of every induced subgraph equals the size of its largest clique, and an algorithm of Grötschel, Lovász, and Schrijver [6] from 1988 finds an optimal colouring of a perfect graph in polynomial time. But this algorithm uses the ellipsoid method, and it is a well-known open question to construct a “combinatorial” polynomial-time algorithm that yields an optimal colouring of a perfect graph. Here we give a combinatorial algorithm that finds an optimal colouring of a perfect graph with clique number $k$, in time that is polynomial for fixed $k$. 
1 Introduction

All graphs in this paper are finite and have no loops or parallel edges. We denote the chromatic number of $G$ by $\chi(G)$, and the cardinality of the largest clique of $G$ by $\omega(G)$, and a colouring of $G$ with $\omega(G)$ colours is called an optimal colouring. If $X \subseteq V(G)$, we denote by $G[X]$ the subgraph of $G$ induced on $X$, that is, the subgraph with vertex set $X$ and edge set all edges of $G$ with both ends in $X$. A hole in $G$ is an induced cycle of length at least four, and an antihole in $G$ is a hole in $G^c$, the complement graph of $G$. A graph is Berge if it has no odd hole or odd antihole (a hole or antihole is odd if it has an odd number of vertices). This definition arose from a conjecture of Berge [1], now the strong perfect graph theorem [3], that every Berge graph is perfect — that is, that every induced subgraph admits an optimal colouring.

Grötschel, Lovász, and Schrijver [6] showed in 1988 that the ellipsoid method can be applied to find an $\omega(G)$-colouring of a perfect graph, and hence of a Berge graph in view of the result of [3]. Remarkably, however, we have not yet been able to find a “combinatorial” algorithm to do the same; and the result of this paper is a step in that direction. We give:

1.1 An algorithm that, with input an $n$-vertex Berge graph, outputs in time $O(n(\omega(G)+1)^2)$ an optimal colouring of $G$.

A skew partition in $G$ is a partition $(A, B)$ of $V(G)$ such that $G[A]$ is not connected and $G^c[B]$ is not connected. It is balanced if in addition:

- for all nonadjacent $u, v \in B$, every induced path of $G$ with ends $u, v$ and with interior in $A$ has even length, and

- for all adjacent $u, v \in A$, every antipath of $G$ with ends $u, v$ and with interior in $B$ has even length.

(The length of a path is the number of edges in it. An antipath in $G$ is an induced subgraph whose complement is a path, and its length is the length of this path.) A skew partition is unbalanced if it is not balanced.

To construct 1.1, we combine an algorithm of Chudnovsky, Trotignon, Trunck and Vušković [4], that optimally colours a Berge graph with no balanced skew partition, with an algorithm of the authors [2] that finds a balanced skew partition in a Berge graph if one exists. Let us state these more precisely:

1.2 [4] An algorithm that, with input an $n$-vertex Berge graph $G$ with no balanced skew partition, outputs in time $O(n^7)$ an optimal colouring of $G$.

1.3 [2] An algorithm that, with input an $n$-vertex Berge graph $G$, outputs in time $O(n^6)$ a balanced skew partition of $G$ if there is one.

Thus, given a Berge graph that we need to colour, we first apply 1.3, and if it tells us that there is no balanced skew partition, we just apply 1.2 and we are done. The question is, what do we do if 1.3 gives us a skew partition $(A, B)$? In this case, we partition $A$ into two nonempty sets $A_1, A_2$ such that there are no edges between $A_1$ and $A_2$, and work with the subgraphs induced on $A_1 \cup B$ and $A_2 \cup B$. With some sort of induction, we obtain optimal colourings of these two subgraphs; and
then we need to piece them together to obtain a colouring of the original graph $G$. So, there are two issues: given colourings of these two subgraphs, how do we fit them together? And how can we arrange an induction that will lead to a polynomial-time algorithm? At the moment we do not know how to handle either issue with no bound on clique number.

2 Making two colourings match

If $X, Y \subseteq V(G)$, we say that $X$ is complete to $Y$ if $X \cap Y = \emptyset$ and every vertex in $X$ is adjacent to every vertex in $Y$, and $X$ is anticomplete to $Y$ if $X$ is complete to $Y$ in the complement graph $\overline{G}$.

Fix $k \geq 0$. We assume that we have an algorithm $\mathcal{P}$ say, that will optimally colour any $n$-vertex Berge graph with clique number at most $k - 1$, in time $O(n^c)$, where $c \geq k$. Let $(A, B)$ be a balanced skew partition of an $n$-vertex Berge graph $G$ with clique number $\ell$. Let $(A_1, A_2)$ be a partition of $A$ with $A_1, A_2 \neq \emptyset$, such that $A_1$ is anticomplete to $A_2$, and for $i = 1, 2$ let $G_i = G[A_i \cup B]$. (Thus $k \geq 2$.) For $i = 1, 2$ let $\phi_i$ be an optimal colouring of $G_i$, mapping from $V(G_i)$ to the set of integers $\{1, \ldots, \omega(G_i)\}$. Let us describe how to obtain an optimal colouring of $G$.

1. Take a partition $(B_1, B_2)$ of $B$ into two nonempty sets, such that $B_1$ is complete to $B_2$, and compute $b_i = \omega(G[B_i])$ for $i = 1, 2$. Let $b = \omega(G[B])$. Exchange $B_1, B_2$ if necessary to arrange that $|B_1| - b_1 \leq |B_2| - b_2$.

2. For $i = 1, 2$, let $L_i = \{\phi_i(v) : v \in B_i\}$, and let $\ell_i = |L_i|$: by permuting the colours, arrange that $L_i = \{1, \ldots, \ell_i\}$. Let $S_i$ be the set of $v \in A_i \cup B$ with $\phi_i(v) \in L_i$. So $B_1 \subseteq S_i$ and $B_2 \cap S_i = \emptyset$.

3. For $i = 1, 2$, let $H_i$ be obtained from $G_i[S_i]$ by adding $\ell_i - b_1$ new vertices, each complete to $B_1$ and anticomplete to $S_i \setminus B_1$, and all adjacent to one another. Apply $\mathcal{P}$ to $H_i$ to obtain an $\ell_i$-colouring $\xi_i$ of $H_i$. By permuting colours arrange that the $\ell_i - b_1$ new vertices have colours $b_1 + 1, \ldots, \ell_i$. For each $v \in A_i \cup B$, let $\psi_i(v) = \phi_i(v)$ if $v \notin S_i$, and $\psi_i(v) = \xi_i(v)$ if $v \in S_i$.

4. For $i = 1, 2$, let $T_i$ be the set of vertices $v \in V(G_i)$ with $\psi_i(v) \in \{1, \ldots, b_i\}$. Apply $\mathcal{P}$ to $G[T_i \cup T_j]$ and to $G \setminus (T_i \cup T_j)$, to obtain a $b_1$-colouring of $G[T_i \cup T_j]$ and a $(k - b_1)$-colouring of $G \setminus (T_i \cup T_j)$. Combine them to make a $k$-colouring of $G$.

Let us fill in some more detail and explanation. In step 1, such a partition exists because $G[B]$ is not anticonnected, and we can find one in time $O(n^2)$. Computing $\omega(G[B])$ for $i = 1, 2$ takes time at most $O(n^k)$, by trying all subsets of size at most $k$.

In step 2, $S_i$ is the set of vertices of $G_i$ that have the same colour under $\phi_i$ as some vertex in $B_1$. Since $B_2$ is complete to $B_1$, no vertex in $B_2$ belongs to $S_i$.

In step 3, $H_i$ is Berge, since $(A, B)$ is balanced. Also, $\omega(H_i) \leq \ell_i$, since every clique of $H_i$ that contains no new vertex has already been coloured with $\ell_i$ colours, and every clique that contains a new vertex is disjoint from $A \cup B_2$ and has at most $b_1$ vertices in $B_1$. Also $\ell_i \leq k - 1$, because $B_2 \neq \emptyset$ and no colour appears under $\phi_i$ in both $B_1, B_2$. Consequently $\mathcal{P}$ can be applied to $H_i$, and it yields an $\ell_i$-colouring. We claim that $|V(H_i)| \leq n$. There are only $\ell_i - b_1$ vertices of $H_i$ that are not vertices of $G$, and there are at least $|B_2|$ vertices in $G$ that are not in $H_i$; and

$$\ell_i - b_1 \leq |B_1| - b_1 \leq |B_2| - b_2 \leq |B_2|.$$
This proves that $|V(H_i)| \leq n$, and so this application of $\mathcal{P}$ takes time $O(n^c)$. Since the new vertices are a clique and so all have different colours, we can arrange by permuting colours that the $\ell_i - b_1$ new vertices have colours $b_1 + 1, \ldots, \ell_i$. Consequently, only colours $1, \ldots, b_1$ appear in $B_1$. Since only colours $1, \ldots, \ell_i$ appear in $S_i$ under $\xi_i$, and only colours $\ell_i + 1, \ldots, k$ appear in $V(G_i) \setminus S_i$ under $\phi_i$, it follows that $\psi_i$ defined in step 3 is a $k$-colouring of $G_i$, and under it only colours $1, \ldots, b_1$ appear in $B_1$. This step takes time $O(n^c)$.

In step 4, since $G_i[T_i]$ is coloured with only $b_1$ colours under $\psi_i$, its clique number is at most $b_1$; and since every clique included in $T_1 \cup T_2$ is included in one of $T_1, T_2$, it follows that the clique number of $G[T_1 \cup T_2]$ is at most $b_1$ (in fact, exactly $b_1$). Similarly, the clique number of $G \setminus (T_1 \cup T_2)$ is at most $k - b_1$. Since $b_1, k - b_1 < k$, these applications of $\mathcal{P}$ are valid, and yield colourings as described. This step takes time $O(n^c)$.

Consequently, the whole algorithm takes time $O(n^c)$, since $c \geq k \geq 2$.

3 The induction

We need the following lemma, a result of Duchet and Meyniel [5] (we give a proof for the reader’s convenience, and since it is not explicitly proved in [5]). The stability number of $G$ is $\omega(G)$.

3.1 Let $G$ be a nonnull connected graph with stability number $\alpha$. Then there is a connected induced subgraph with at most $2\alpha - 1$ vertices and with stability number $\alpha$.

Proof. Choose a stable set $S$ with the following properties:

- there is a subset $T \subseteq V(G)$ with $|T| < |S|$ such that $G[S \cup T]$ is connected;
- there is a stable set of cardinality $\alpha$ including $S$;
- $S$ is maximal subject to these two conditions.

This is possible because setting $S$ to be a singleton subset of some largest stable set satisfies the first two bullets. We claim that $|S| = \alpha$; for suppose not, and let $S'$ be a stable set of cardinality $\alpha$ including $S$. Since $S'$ is a maximum stable set, every vertex of $G$ either belongs to $S'$ or has a neighbour in $S'$. Since $G$ is connected, there is a shortest path between $S$ and $S' \setminus S$, with vertices $p_1, \ldots, p_k$ in order say, where $p_1 \in S$ and $p_k \in S' \setminus S$. Since $S'$ is stable, $k \geq 3$; and since every vertex of $G$ belongs to $S'$ or has a neighbour in $S'$, $k \leq 4$. If $k = 3$, we could replace $S$ by $S \cup \{p_3\}$ and $T$ by $T \cup \{p_2\}$, contrary to the choice of $S$. So $k = 4$. If $p_3$ has more than one neighbour in $S' \setminus S$, say $p_4$ and $p'_4$, then we could replace $S$ by $S \cup \{p_4, p'_4\}$ and $T$ by $T \cup \{p_2, p_3\}$, again contrary to the maximality of $S$. So $p_4$ is the only neighbour of $p_3$ in $S' \setminus S$. But then $(S' \setminus \{p_4\}) \cup \{p_3\}$ is a stable set of cardinality $\alpha$, and we could replace $S$ by $S \cup \{p_3\}$ and $T$ by $T \cup \{p_2\}$, again a contradiction. This proves that $|S| = \alpha$, and so proves 3.1.

Let us say a $k$-pellet in a graph $G$ is a subset $P \subseteq V(G)$ with $|P| = 2k$ such that $G[P]$ is anticonnected and $\omega(G[P]) \geq k$. By 3.1 applied in the complement, every anticonnected graph $G$ with at least $2\omega(G)$ vertices has at least one $\omega(G)$-pellet; and only polynomially many (for fixed $\omega(G)$), since $\omega(G)$-pellets have bounded size. Essentially, we are going to prove inductively that the running time of our algorithm is at most proportional to the number of $\omega(G)$-pellets in $G$ (actually,
proportional to that number plus one). But to make the argument clearer, let us replace the inductive step by the following tree structure.

Let us say a decomposition tree of $G$ is a rooted tree $T$, together with a choice of a subset $X_t \subseteq V(G)$ for each $t \in V(T)$, satisfying the following conditions:

- every vertex of $T$ has two or zero children (the children of a vertex $v$ are its neighbours that are not on the path between $v$ and the root);
- $X_w = V(G)$ if $w$ is the root, and for all $s \in V(T)$, if $r$ is a child of $s$ then $X_r \subseteq X_s$;
- if $s$ has children $r, t$ then $(X_s \setminus (X_r \cap X_t), X_r \cap X_t)$ is a balanced skew partition of $G[X_s]$;
- if $s \in V(T)$ has no children then either $G[X_s]$ has no balanced skew partition, or its clique number is less than $\omega(G)$, or it is not anticonnected, or it has at most $2\omega(G) - 1$ vertices.

By “processing” a subset $X \subseteq V(G)$, we mean doing the following:

- check that $|X| \geq 2\omega(G)$ (and if not, stop);
- check that $\omega(G[X]) = \omega(G)$ (and if not, stop);
- check that $G$ is anticonnected (and if not, stop);
- apply 1.3 to $G[X]$ (and if there is no balanced skew partition, stop);
- let $(A, B)$ be the output of 1.3; partition $A$ into two nonempty subsets $A_1, A_2$ such that $A_1$ is anticomplete to $A_2$, and output $A_1 \cup B, A_2 \cup B$.

Thus, processing a set $X$ takes time $O(n^{\max(k,6)})$, where $k = \omega(G)$. We can construct a decomposition tree by initially setting $V(T) = \{t\}$ and $X_t = V(G)$, and recursively processing each new $X_s$; if processing $X_s$ gives us two new sets $X_r, X_t$ say we add two children $r, t$ of $s$ to $T$. The time to find a decomposition tree thus depends on the number of vertices in the tree, and that is not so easy to estimate, because the various sets $X_t \ (t \in V(T))$ can intersect. But pellets give us a way to find a bound. We need:

**3.2** Let $k = \omega(G)$, and let $T, (X_t : t \in V(T))$ form a decomposition tree for $G$. Then at most $n^{2k} - 1$ vertices of $T$ have children, and at most $n^{2k}$ vertices have no children.

**Proof.** For each $t \in V(T)$, let $f(t)$ be the number of $k$-pellets included in $X_t$. We need:

1. $f(r) + f(t) + 1 \leq f(s)$ for each $s \in V(T)$ with children $r, t$.

Let $B = X_r \cap X_t$ and $A = X_s \setminus B$; so $(A, B)$ is a balanced skew partition of $G[X_s]$. Every $k$-pellet of $G[X_r]$ is also a $k$-pellet of $G[X_s]$ and the same for $G[X_t]$. We claim that no $k$-pellet is counted twice. For let $P$ be a $k$-pellet of $G[X_s]$. If it is a $k$-pellet of both $G[X_r]$ and $G[X_t]$, then $P \subseteq X_r \cap X_t = B$, and since $P$ is anticonnected, it is a subset of some anticomponent of $G[B]$. Since $B$ has at least two anticomponents, it follows that $\omega(G[B]) > \omega(G[P])$, which is impossible since $\omega(G[P]) = k = \omega(G)$. This proves that no $k$-pellet contributes to both $f(r), f(t)$, and so $f(r) + f(t) \leq f(s)$. We need to prove strict inequality, however; so we need to show that some
$k$-pellet of $G[X_s]$ is a subset of neither of $X_r, X_t$. To show this, by 3.1 there is a set $Y$ of cardinality $2k - 1$ such that $G[Y]$ is anticonnected and $Y$ includes a $k$-clique. By the same argument as before, $Y \not\subseteq B$, and so we may assume that $Y \cap A_1 \neq \emptyset$. Choose $a_2 \in A_2$; then since $a_2$ is nonadjacent to the vertices of $Y$ in $A_1$, $Y \cup \{a_2\}$ is anticonnected and is thus a $k$-pellet of $G[X_s]$, and not a $k$-pellet of either of $G[X_r], G[X_t]$. This proves (1).

Thus, if we sum $f(s) - f(r) - f(t) - 1$ over all vertices $s$ that have children $r, t$ say, then the answer is nonnegative. Let $q$ be the root, and $L$ the set of vertices with no children, and $I$ the set that have children. Then the sum can be rewritten as $f(q) - |I| - \sum_{r \in L} f(r)$, and hence is at most $f(q) - |I|$; and so $|I| \leq f(q)$. Since $f(q) < n^{2k}$, and $|L| = |I| + 1$, this proves 3.2.

Let us put these pieces together to prove 1.1. We prove the following.

3.3 For all $k \geq 1$, there is an algorithm $P_k$ that, with input a Berge graph $G$ with $\omega(G) = k$, outputs in time $O(n^{(k+1)^2})$ a $k$-colouring of $G$.

Proof. We proceed by induction on $k$. The result is true for $k = 1, 2$, so we assume that $k \geq 3$ and the result holds for $k - 1$. Here is the algorithm:

- Construct a decomposition tree $T,(X_t : t \in V(T))$ for $G$. Since $T$ has at most $n^{2k}$ internal vertices by 3.2, and $2k + \max(k, 6) \leq (k + 1)^2$, this takes time $O(n^{(k+1)^2})$.

- For each vertex $t \in V(T)$ such that $t$ has no children, compute an optimal colouring for $G[X_t]$ as follows. Since $t$ has no children, either $G[X_s]$ has no balanced skew partition, or its clique number is less than $k$, or it is not anticonnected, or it has at most $2k - 1$ vertices. In the first case we apply 1.2, in the second case we apply $P_{k-1}$, in the third we apply $P_{k-1}$ to its anticomponents, and in the fourth we can find an optimal colouring in constant time (depending only on $k$). A call to $P_{k-1}$ takes time $O(n^{k^2})$ from the inductive hypothesis, and since $k^2 \geq 7$, in each case we can find an optimal colouring in time $O(n^{k^2})$. There are at most $n^{2k}$ such vertices $t$, and since $2k + k^2 \leq (k + 1)^2$, this step takes time $O(n^{(k+1)^2})$ altogether.

- Combine these colourings to obtain an optimal colouring of $G$ (starting from the leaves and working inwards), using the algorithm of section 2 (with $P_{k-1}$) at most $n^{2k}$ times. Each call of the algorithm from section 2 takes time $O(n^{k^2})$, and since $2k + k^2 \leq (k + 1)^2$, altogether this step takes time $O(n^{(k+1)^2})$.

The total running time is thus $O(n^{(k+1)^2})$. This proves 3.3.

4 Acknowledgements

The authors would like to thanks Frédéric Maffray, Nicolas Trotignon and Kristina Vušković for many helpful dicussions.
References


