A Hypocoloring Model for Batch Scheduling
(Preliminary version)

D. de Werra *, M. Demange†, J. Monnot‡, V. Th. Paschos‡

Résumé

Nous considérons un problème de sous-coloration pondérée dans un graphe modélisant des problèmes d’ordonnancement par lots ; chaque sommet $v$ a un poids $w(v)$ ; chaque couleur $S$ est une collection de cliques disjointes deux à deux (au sens des sommets et des arêtes). Le poids $w(S)$ est défini comme étant $\max\{w(K) = \sum_{v \in K} w(v) | K$ clique $\in S\}$. Dans ce problème d’ordonnancement, le temps d’exécution est donné par $\sum_{i=1}^{k} w(S_i)$ où $S = (S_1, \ldots, S_k)$ est une partition des sommets du graphe $G$ dont chaque classe de couleur est définie comme précédemment. Nous présentons des propriétés de telles colorations concernant des classes particulières de graphes (line-graphes de cactus, block graphes) et nous exposons des résultats de complexité et d’approximabilité. Nous démontrons que le problème de décision associé est NP-complet pour deux classes de graphes : les graphes bipartis de degré maximum au plus 39 et les graphes planaires sans triangle de degré maximum au moins 3. Nous proposons également des algorithmes polynomiaux pour les graphes de degré maximum au plus 2 et pour les forêts de degré maximum au plus $k$, pour tout $k$ constant. Finalement, nous présentons un algorithme exponentiel basé sur un principe de séparations pour les graphes sans triangle.

Mots-clés : Ordonnancement par lots ; Coloration ; Sous-coloration ; Hypocoloration ; Coloration pondérée ; Approximation

Résumé

*Ecole Polytechnique Fédérale de Lausanne, Switzerland, dewerra@dma.epfl.ch
†ESSEC, Dept. SID, France, demange@essec.fr
‡LAMSADE, CNRS UMR 7024, Université Paris-Dauphine, 75775 Paris Cedex 16, France, {monnot,paschos}@lamsade.dauphine.fr
A Hypocoloring Model for Batch Scheduling
(Preliminary version)

Starting from a batch scheduling problem, we consider a weighted subcoloring in a graph $G$; each node $v$ has a weight $w(v)$; each color class $S$ is a subset of nodes which generates a collection of node disjoint cliques. The weight $w(S)$ is defined as $\max \{w(K) = \sum_{v \in K} w(v) | K \in S\}$. In the scheduling problem, the completion time is given by $\sum_{i=1}^{k} w(S_i)$ where $S = (S_1, \ldots, S_k)$ is a partition of the node set of graph $G$ into color classes as defined above. Properties of such colorings concerning special classes of graphs (line graphs of cacti, block graphs) are stated; complexity and approximability results are presented. The associated decision problem is shown to be NP-complete for bipartite graphs with maximum degree at most 39 and triangle-free planar graphs with maximum degree $k$ for any $k \geq 3$. Polynomial algorithms are given for graphs with maximum degree two and for the forests with maximum degree $k$. An (exponential) algorithm based on a simple separation principle is sketched for graphs without triangles.

Key words: Batch scheduling; graph coloring; subcolorings, hypocolorings; weighted colorings; approximability.

1 Introduction

Chromatic scheduling is the domain of scheduling problems which can be formulated in terms of graph coloring or more precisely of generalized graph coloring (i.e., coloring with a few additional requirements).

The development of chromatic scheduling have thus generated various extensions of graph coloring motivated by applications like course timetabling or processor scheduling problems or satellite communication.

In particular the concept of weighted coloring has been introduced to generalize classical coloring models and to handle situations where operations occur with possibly different processing times. In this paper, we shall generalize a weighted coloring model used in de Werra et al. [10] for studying some types of batch scheduling problems. Such a generalization of classical coloring appears in Fiala et al. [12], Albertson et al. [1] and Brown and Corneil [8] but simply as a variation of coloring problems. We also refer the reader to Broersma et al. [6]. To our knowledge the weighted case has not been studied specifically.

After motivating the use of such weighted coloring by means of batch scheduling, we will recall some complexity results related to these colorings and derive some complementary properties together with approximation results. We will also characterize a few solvable cases (graphs of degree 2) and an enumeration algorithm generalizing a classical coloring technique will be given for triangle-free graphs. The last section will present some possible extensions of these types of weighted colorings.
For all graph theoretical terms not defined here, the reader is referred to Berge [3] and, for all definitions related to complexity, to Garey and Johnson [13].

2 A chromatic scheduling model

In order to describe our generalized weighted coloring model, we shall consider an instance of batch scheduling problem which can be stated as follows:

We are given a finite set $V$ of operations $v$ to be processed on some identical processors whose properties will be stated later. Each operation has a positive (generally integral) processing time $w(v)$ which does not depend on the processor. No preemptions will be allowed during the processing of an operation. Each processor will handle one operation at a time. In addition, there are some incompatibilities between pairs of operations $u, v$; if the pair $u, v$ is incompatible then operations $u$ and $v$ cannot be processed simultaneously (on different processors).

At this stage we may associate to each operation $v$ a node $v$ of a graph $G = (V, E)$; the set $E$ of incompatible pairs of operations will be associated with the edge set of $G$. Each node $v$ will have a weight $w(v)$. Now a batch $S$ of operations is a collection of pairwise compatible operations; the operations in $S$ are assigned to different processors (assuming there is a large enough number of processors) and they are processed simultaneously. So all operations in $S$ are completed when the operation $v$ with the largest processing time $w(v)$ is completed. At this stage, the set $S$ corresponds to a stable set in $G$. It is then natural to define the weight $w(S)$ as $w(S) = \max\{w(v) | v \in S\}$. Assigning each operation to some batch corresponds then to partitioning the node set $V$ of graph $G$ into a number $k$ of stable sets. This is precisely the problem of finding a $k$-coloring $S = (S_1, \ldots, S_k)$ of $G$ such that $C(S) = w(S_1) + \ldots + w(S_k)$ is minimum.

There are many situations where operations have to be assigned to batches (of compatible operations) which are processed one after the other (see Boudhar and Finke [5] for some examples). Examples in satellite communication and in production have also been modeled as special cases of the above batch scheduling problem (see Rendl [17], Boudhar and Finke [5]).

In the above model all operations in a batch are assigned to different processors and processed simultaneously. The processing time of a batch $S$ is limited by the largest processing time of the operations in $S$. If the processing times may take different values, it may be worthwhile to assign two (or more) incompatible operations $v$ with small processing times $w(v)$ to the same batch; they will be processed consecutively on the same processor. This will not increase the processing time $w(S)$ of the batch $S$ as long as the sum of processing times of these operations does not exceed the longest processing time $w(v)$ in $S$. 
In order to allow this possibility in our model, we have to generalize the definition of a stable set in a graph $G$. We may view a stable set $S$ in a graph $G$ as a set of nodes which induces a collection of node disjoint cliques of size one (without any edges between them).

In a similar way, we shall say that a subset $S$ of nodes is hypostable set in $G$ if it induces a collection of node disjoint cliques (without any edges between them).

In our batch scheduling model, we shall in a natural way define the weight $w(K)$ of a clique $K$ as $w(K) = \sum_{v \in K} w(v)$. Since $K$ corresponds to incompatible operations (assigned to the same processor), the processing time of all operations in $K$ will be the sum of all processing times. As a consequence, the weight of a hypostable set $S$ will be $w(S) = \max\{w(K) | K \in S\}$.

In the case of stable sets, we have $|K| = 1$ for each clique $K$ in $S$.

Our batch scheduling problem now consists in finding a $k$-hypocoloring $S = (S_1, \ldots, S_k)$ of the nodes of $G$, i.e., a partition of the node set into hypostable sets such that:

$$\hat{K}(S) = \sum_{i=1}^{k} w(S_i) \text{ is minimum}$$

Observe that $k$ is generally not given; its value results from the minimization of $\hat{K}$.

Assume we have a collection of people who have expressed some mutual compatibility (represented by edges of a graph whose nodes are the people). Each person $v$ needs a certain number $w(v)$ of time units to tell his (her) stories. We want to invite each one of these people to a banquet where different tables are set for each banquet. The natural requirements are that we want to place at a same table people who are all compatible. At any table each person will tell successively his (her) stories. So for each table we will need an amount of time equal to the sum of the $w(v)$ of the persons $v$ sitting at this table. In addition in order to avoid frustration we would require that on any given banquet there are no two people sitting at different tables who would have liked to be together (no edge between their nodes). The duration of a banquet will be the maximum time needed for the tables set up for this banquet. We want to invite each person to one banquet exactly and we want to find an assignment of the people to banquets (batches) and more precisely to tables in this banquet so that the total duration of the banquets is minimum (it is proportional to the cost of hiring personal to serve meals in the banquets). This is precisely a weighted hypocoloring of a graph with a minimum cost.

The above model may also be used for representing some machine scheduling problems: we are for instance given a collection of jobs $v$ with processing times $w(v)$ in a flexible manufacturing system; we link the nodes representing two jobs if these share a certain number of tools; it will thus be interesting to assign these jobs to a same machine on which the appropriate tools (and some others as well) will be installed. A batch will
consist of an assignment of jobs to some machines in which we try to assign to a same
machine jobs which share some tools. Since there is only a limited number of tools of
each type, we will try to assign to different machines jobs which do not need the same
tools. Hence a batch will be represented by a hypostable set in the graph of compatibilities
(common tools) and the processing time of a batch will again be the maximum load of
a machine (maximum of the sums of processing times of jobs assigned to the same ma-
cine). We shall concentrate on this model of weighted hypocoloring which is motivated
in a natural way by the batch scheduling context. For clarifying purposes some results
will be derived for this special model and we will mention in the last section how these
ideas may be transposed to more general situations

3 Some special cases

Getting back to the definition of $k$-hypocolorings in a graph $G$ we may by analogy
define the hypochromatic number $\chi_h(G)$ of $G$ as the smallest $k$ for which $G$ has a $k$-
hypocoloring. Recognizing if $\chi_h(G) \leq 2$ for graph $G$ is difficult (see Fiala et al. [12]);
so, we will simply derive a few cases where the hypochromatic number can be obtained
easily. In Broersma et al. [6], it is shown that the problem is easy for complements of
bipartite graphs.

A cactus is a connected simple graph where any two elementary cycles have at most
one node in common. By definition a cactus will have neither loops nor multiple edges.

Let $G$ be the line graph $L(H)$ of a simple graph $H$; then any $k$-hypocoloring of the
nodes of $G$ corresponds to an "edge $k$-hypocoloring" in $H$, i.e., a partition of the edge set
of $H$ into $k$ subsets which may be called hypostable. There is a one-to-one correspondence
between hypostable sets of nodes in $G$ and hypostable sets of edges in $H$. It is not difficult
to verify that a subset $E'$ of edges in $H$ is hypostable set iff it consists of node disjoint
sets of stars (set of edges with exactly one common node) and triangles.

Before stating some results on the hypochromatic number of line graphs of cacti, let
us introduce an auxiliary graph which will be useful later.

We are given a cactus $H$; its blocks (2-connected components) will by definition be
elementary cycles and edges not contained in any cycle (these are cut-edges). We associate
with each block of $H$ a node $b$; then for each node $v$ of $H$ which is in at least two blocks
we introduce a node $v$. We link each $b$ to the nodes $v$ corresponding to nodes of $G$ which
block $b$ contains to obtain the auxiliary graph $G(H)$. It is easy to verify that if $H$ is a
cactus then $G(H)$ is a tree.

In $G(H)$ we can then choose an arbitrary node $b$ (corresponding to a block) as a root
and orient all its edges (which become arcs) away from $b$. We may next assign numbers
$n(b)$ to the nodes $b$ (corresponding to blocks) in such a way that whenever there is a path from $b_i$ to $b_j$ then $n(b_i) < n(b_j)$. This will define the coloring order to be used later.

A cactus is bipartite if it contains no (elementary) odd cycles.

Consider for instance the bipartite cactus given in Figure 1:

![Figure 1](image1)

**FIG. 1** – a bipartite cactus $H$ with $\chi_h(L(H)) = 2$

A bicoloring of the edges of $H$ (heavy edges, light edges) is shown. Since the edge set of $H$ is not a union of node disjoint stars and triangles, we have $\chi_h(L(H)) = 2$. Figure 2 represents a bipartite cactus obtained from the previous one by introducing a pendent edge $[e, f]$ at node $e$.

![Figure 2](image2)

**FIG. 2** – a bipartite cactus $H$ with $\chi_h(L(H)) > 2$

Let us try to color its edges with two colors; clearly in the cycle on nodes $a, b, c, d$ we must have two heavy and two light edges. Assume w.l.o.g. that $[a, b]$ and $[a, d]$ are heavy,
so that \([b, c]\) and \([c, d]\) are light; then \([d, g]\) cannot be colored (heavy or light). So, we may assume w.l.o.g. that \([a, b]\), \([c, d]\) are light and \([b, c]\), \([a, d]\) are heavy.

- case 1: \([c, e]\) is heavy. Then \([e, f], [e, h]\) are light, \([h, i]\) is heavy, \([c, i]\) is light.
   Hence, \([d, g]\) and \([d, j]\) are heavy. So, \([a, q]\) is light. And now \([b, l]\) cannot be light (since \([a, q]\) and \([a, b]\) are light) and it cannot be heavy (since \([c, e]\) and \([b, c]\) are heavy).

- case 2: \([c, e]\) is light. Then \([e, f], [e, h]\) are heavy, \([h, i]\) is light, and \([c, i]\) is heavy.
   Hence, \([d, j]\) is heavy. It follows that \([a, o]\) and \([b, m]\) must be light (otherwise, we could have three consecutive heavy edges). So, now \([a, o]\), \([a, b]\), \([b, m]\) are three consecutive light edges, which is not allowed.

All cases have been examined, so \(\chi_h(L(H)) > 2\).

The construction given in the proof of proposition 3.1 will show that for bipartite cactus of Fig. 2, we have \(\chi_h(L(H)) = 3\). It will be based on the auxiliary graph \(G(H)\).

**Proposition 3.1** If \(G\) is the line graph \(L(H)\) of a cactus, then \(\chi_h(L(H)) \leq 3\).

**Proof**: We shall color consecutively the edges in each two-connected component of \(H\) in the order \(n(b_1), n(b_2), \ldots\) defined by means of the auxiliary graph \(G(H)\).

We start from the component \(b_1\) and color its edges by using alternately colors \(a, b, c\) (if \(b_1\) is just one edge, we use just color \(a\)). So, we have colored the component \(b_1\) which corresponds to the root of the oriented tree in \(G(H)\).

At each node of \(b_1\), at most two colors appear on adjacent edges. We consider now every cut-node \(v\) of \(b_1\) and color all (uncolored yet) adjacent edges with one of the colors \(a, b, c\) which does not occur around \(v\). This is always possible.

Now, we consider all components \(b_i\) containing these edges and color their remaining edges alternately with the two colors not used yet in \(b_i\) (if \(b_i\) is a single edge, we do nothing).

At this stage, in each such \(b_i\) all nodes (except possibly the cut-node between \(b_1\) and \(b_i\)) have one color at least which does not occur on adjacent edges. So, we can continue coloring the component \(b_i\), one after the other in any order which follows the partial order defined by the oriented tree. At each cut-node \(v\), it will be possible to find an unused color among \(\{a, b, c\}\) to color the edges of the "adjacent" component \(b_i\) (such that \((v, b_i)\) is an arc in \(G(H)\)). This will finally give an edge 3-coloring of \(H\) where each color class contains no three consecutive edges, and hence it will correspond to a 3-hypocoloring in \(L(H)\). \(\square\)

It would be interesting to characterize the graphs \(H\) for which \(\chi_h(L(H)) \leq 2\). The example of bipartite cactus given above may suggest that such a characterization is not immediate. In this direction, we can state a property of block graphs: these are graphs where each block (two-connected component) is a clique.
Proposition 3.2 If \( G \) is a connected block graph, then \( \chi_h(G) \leq 2 \). In particular, if \( G = L(T) \) is the line graph of a tree, then \( \chi_h(L(T)) \leq 2 \).

Proof: A graph is the line graph \( L(T) \) of a tree if and only if it is a block graph where each node is contained in at most two blocks. So, the result for \( L(T) \) follows from the general case of block graphs.

Let \( G \) be a block graph, we may associate with it an auxiliary graph \( G' \) as follows:

- each block \( b \) (maximal clique) of \( G \) is associated with a node of \( G' \); each cut-node \( v \) (i.e., each node belonging to at least two blocks) becomes a node \( v \) of \( G' \).

We link each \( b \) to the node \( v \) corresponding to node \( v \) of \( G \) which are in the block \( b \).

It is again easy to verify that such a graph \( G' \) is a tree. We can choose a node \( b \) (corresponding to a block of \( G \)) as root and orient all edges from the root.

Then as before we will color the node of each block in any order respecting the orientations of the arcs of \( G' \). All nodes of the clique \( b_1 \) corresponding to the root are colored with color \( \alpha \). Then we consider consecutively all cut-nodes in \( b_1 \). We color the remaining nodes of all blocks having a node in common with \( b_1 \) with color \( \beta \) these are the blocks at distance 2 from \( b_1 \) in \( G' \). More generally having colored some blocks by following paths from \( b_1 \) in \( G' \), we consider the cut-nodes in these blocks. We color the remaining nodes of these blocks with the color not used for these cut-nodes.

We continue in this way until all nodes are colored with \( \alpha \) or \( \beta \). The construction will always be possible since any two blocks have at most one node in common. Each color class will consist of cliques of \( G \) which are node disjoint and furthermore there is no edge of \( G \) between two different cliques of the same color. So, we have obtained a 2-hypocoloring of \( G \). If \( G \) is simply a clique, then \( \chi_h(G) = 1 \).

4 Complexity of hypocoloring

Before presenting some complexity results for the hypocoloring problem, we will state a simple observation. It is a direct consequence of the fact that for any hypocoloring \( S \) and for any clique \( K \) of a graph \( G \), \( \hat{K}(C) \geq w(K) \).

Property 4.1 Let \( G \) be a complete \( k \)-partite weighted graph on sets \( L_1, \ldots, L_k \); then an optimum hypocoloring is \( S = (L_1, \ldots, L_k) \). Furthermore it satisfies: \( \hat{K}(S) = \max \{ w(K) : K \text{ is a clique of } G \} \).

More generally assume that a graph \( G \) has a \( k \)-coloring \( S = (S_1, \ldots, S_k) \) and let \( v_i \in S_i \) be a node of maximum weight in \( S_i (i = 1, \ldots, k) \).
Then if \( \{v_1, \ldots, v_k\} \) forms a clique in \( G \), \( S \) is an optimum hypocoloring and \( \hat{K}(S) = \max \{w(K) : K \text{ is a clique of } G\} = w(v_1) + \ldots + w(v_k) \).

We will now show that the hypocoloring problem is close to the coloring problem in some cases; more precisely, we prove that the hypocoloring problem is \textit{NP-hard} for a class \( \Psi \) of graphs as soon as the coloring problem is difficult enough in this class of graphs. On the other hand, this problem is also \textit{NP-hard} for bipartite graphs even though the coloring problem is polynomial for these graphs.

**Proposition 4.2** Let \( \Psi \) be a class of graphs. If the restriction of the coloring problem is \textit{NP-hard} for the \( \Psi \)-graphs, then the hypocoloring problem is strongly \textit{NP-hard} for the graphs of \( \Psi \).

**Proof**: Let \( \Psi \) be a class of graphs and assume that the coloring problem is \textit{NP-hard} for the restriction to \( \Psi \)-graphs. Let \( G \in \Psi \), and consider the instance \( I = (G, w_0) \) where the cost of each node is equal to one (i.e., \( w_0(v) = 1, \forall v \in V \)). Let \( S^* = (S_1, \ldots, S_k) \) be an optimum hypocoloring of \( G \); we will show that \( \chi(G) = \hat{K}(S^*) \).

We have \( \hat{K}(S^*) \leq \chi(G) \) since any coloring of \( G \) is also a hypocoloring of \( I \) with the same cost. Conversely, for each \( i \leq k \), \( S_i \) consists of node disjoint cliques \( K_1, \ldots, K_{j_i} \), and by choice of \( w_0 \), we have \( w(S_i) = \max_{1 \leq p \leq j_i} |K_p| \). So, the hypostable set \( S_i \) can be partitioned in at most \( \max_{1 \leq p \leq j_i} |K_p| \) stable sets \( S'_j \), just picking one node in each clique \( K_i \). We apply the same procedure for each hypostable and we obtain a coloring of \( G \) with \( \hat{K}(S^*) \) colors and the result follows. \( \square \)

It follows that the hypocoloring problem is \textit{NP-hard} for the graphs with maximum degree at most \( B \) for any \( B \geq 4 \), even if these graphs do not contain any triangle, for the planar graphs, etc. Moreover, when the maximum degree is at most 3 and the values of \( w \) are all one then, the previous proof and Brooks theorem [7] show that this case is polynomial. We will show in section (9.2), that the hypocoloring problem is polynomial when the maximum degree is at most 2. On the other hand, we now prove that when \( G \) is a triangle-free graph with maximum degree 3 and \( w \) can take only two different values, then the hypocoloring problem is strongly \textit{NP-hard}.

We start with \( 1 - IN \ 3SAT \) proved to be \textit{NP-complete} in Schaefer ([18]); this problem is defined as follows: Given a collection \( C \) of \( m \) clauses over the set \( X \) of \( n \) Boolean variables such that each clause has exactly three literals, is there a truth assignment \( f \) satisfying \( C \) such that each clause in \( C \) has exactly one true literal?

**Theorem 4.3** The hypocoloring problem is strongly \textit{NP-hard} for triangle-free graphs with maximum degree 3.

**Proof**: We shall reduce \( 1 - IN \ 3SAT \) to our problem. Let \( C = (C_1, \ldots, C_m) \) be a collection of \( m \) clauses with variable set \( X = \{x_1, \ldots, x_n\} \) such that every clause \( C_j \) of
A Hypocoloring Model for Batch Scheduling
(Preliminary version)

\[ C \text{ contains exactly three literals, } C_j = x \lor y \lor z \text{ where each literal } x \text{ or } y \text{ or } z \text{ is either } x_k \text{ or } \overline{x}_k \text{ for some suitable } k.\]

From instance \( I = (C, X) \) of \( 1 \text{-} IN \ 3SAT \), we construct an instance \( I' = (G, w) \) of hypocoloring such that the answer of \( I \) is yes if and only if there exists a hypocoloring \( S \) of \( I' \) with cost \( \bar{K}(S) \leq 3 \).

We use two gadgets : gadget clause and gadget variable. From the clause \( C_i = x \lor y \lor z \), we build the graph \( F_i \) described in Figure 3.

![Fig. 3 – a gadget clause \( F_i \)](image)

This graph has 11 nodes \( v_1(F_i), \ldots, v_5(F_i) \) and \( x(F_i), y(F_i), z(F_i), \overline{x(F_i)}, \overline{y(F_i)}, \overline{z(F_i)} \) and 12 edges \([v_1(F_i), v_2(F_i)], [v_1(F_i), v_3(F_i)], [v_2(F_i), v_4(F_i)], [v_3(F_i), v_5(F_i)], [v_4(F_i), x(F_i)], [v_4(F_i), z(F_i)], [v_5(F_i), y(F_i)]\) and the edges \([x(F_i), x(F_i)], [y(F_i), y(F_i)], [z(F_i), z(F_i)]\), \([x(F_i), y(F_i)], [y(F_i), z(F_i)]\). Moreover, the weights of nodes are one except for \( v_1(F_i) \) which is two (i.e., \( w(v_1(F_i)) = 2 \) and \( \forall v \in V(F_i) \setminus \{v_1(F_i)\}, w(v) = 1 \)).

Associated with variable \( x_j \), we build the graph \( H_j \) of Figure 4. It has \( 4m \) nodes : \( x_1(H_j), \ldots, x_m(H_j), x_1(H_j), \ldots, x_m(H_j), v_1(H_j), \ldots, v_{2m}(H_j) \), and \( 5m \) edges : for any \( k \leq m \), we have a box on nodes \( v_{2k-1}(H_j), v_{2k}(H_j), x_k(H_j), x_k(H_j), x_k(H_j), v_{2k-1}(H_j) \) described by the edges \( [v_{2k-1}(H_j), v_{2k}(H_j)], [v_{2k}(H_j), x_k(H_j)], [x_k(H_j), x_k(H_j)], [x_k(H_j), v_{2k-1}(H_j)] \). Moreover, there is a cycle of size \( 2m \) between the nodes \( \{v_1(H_j), v_2(H_j), \ldots, v_{2m}(H_j)\} \) described by the sequence \( \{v_1(H_j), v_2(H_j), \ldots, v_{2m-1}(H_j), v_{2m}(H_j), v_1(H_j)\} \). Finally, the weights of nodes are one (i.e., \( w(z) = 1, \forall z \in V(H_j) \)). In addition, we link these different graphs in the following way. If the variable \( x \) is in the clause \( C_i \) then : if \( x = x_j \) then we add the edge : \( [x_j(H_j), x(F_i)] \), else \( (x = \overline{x_j}) \) and then we add the edge \( [x_j(H_j), x(F_i)] \). This graph \( C \) has maximum degree equal to 3 and the instance \( I' = (G, w) \) is computable in polynomial time with respect to both \( n \) and \( m \).
Let $f$ be a truth assignment of $I = (\mathcal{C}, X)$ (i.e., $f(x_i) = 1$ if $x_i = true$ and $f(x_i) = 0$ if $\bar{x}_i = true$) The hypostable sets $S_1$ and $S_2$ of the hypocoloring $S = (S_1, S_2)$ of $I'$ verifying $\hat{K}(S) \leq 3$ are given by:

$$S_1 = \bigcup_{i=1}^{n} \{ v_1(F_i), v_4(F_i), v_5(F_i) \} \cup \{ x(F_i) : f(x) = 1 \} \cup \{ \bar{x}(F_i) : f(x) = 0 \}$$

$$\bigcup_{j=1}^{m} \bigcup_{k=1}^{m} \{ x_k(H_j), v_{2k-1}(H_j) : f(x) = 1 \}$$

$$S_2 = V(G) \setminus S_1$$

It is easy to verify that $S = (S_1, S_2)$ is a hypocoloring and $w(S_1) = 2$ and $w(S_2) = 1$; thus $\hat{K}(S) = w(S_1) + w(S_2) = 3$.

Conversely, let $S$ be a hypocoloring of $I'$ verifying $\hat{K}(S) \leq 3$; moreover, assume that $w(S_1) \geq w(S_i)$ for any hypostable set $S_i$ of $S$. We exhibit a truth assignment $f$ of $I$ by taking $f(x) = 1$ if and only if $x \in S_1$ (i.e., if $x = x_k$ is in the clause $C_i$, then we get $f(x) = 1$ else it is $\bar{x}_k$ which is in the clause $C_i$ and we get $f(x) = 1$).

In order to prove this, we shall establish properties of hypocoloring $S$:

(i) $S = (S_1, S_2)$ and $2 = w(S_1) \geq w(S_2)$.
(ii) $w(S_2) = 1$ and $S_2$ is a stable set.
A Hypocoloring Model for Batch Scheduling

(Proof of)

Proof: Assume that \( w(S_1) \geq w(S) \) for any hypostable \( S \) of \( S \); since \( w(S_1) \geq w(v_1(F_i)) = 2 \) and the graph \( G \) contains an induced \( P_3 \), then we necessarily have \( w(S_1) = 2 \) (otherwise \( \overline{\hat{K}(S)} > 3 \)). Moreover, since \( w(S_1) \geq 1 \), we have \( S_i = \emptyset \) for \( i \geq 3 \).

Proof of (ii): Since \( w(S_2) \geq 1 \) and \( \overline{\hat{K}(S)} \leq 3 \), we have \( w(S_2) = 1 \). If \( S_2 \) contains at least one edge \( e = [x, y] \), we have \( w(S_2) \geq w(x) + w(y) \geq 2 \), contradiction with the previous statement.

Proof of (iii): From (i), we know that for any \( i \leq m \), we have \( v_1(F_i) \in S_1 \) and \( \{v_2(F_i), v_3(F_i)\} \subset S_2 \); then, \( \{v_4(F_i), v_5(F_i)\} \subset S_1 \). Finally, at most one node of \( x(F_i), z(F_i) \) belongs to \( S_1 \) since otherwise we would have an induced \( P_3 \). If we have \( y(F_i) \in S_1 \), then \( \{x(F_i), z(F_i)\} \cap S_1 = \emptyset \) since otherwise \( S_2 \) contains an edge which is a contradiction with claim (ii). We conclude this claim by affirming that \( x(F_i), y(F_i), z(F_i) \) can not be simultaneously in \( S_2 \) since otherwise \( x(F_i), y(F_i), z(F_i) \) will be in \( S_1 \).

Proof of (iv): This claim affirm that the gadgets \( H_j \) for every \( j \leq n \) are always hypocolored as an simple 2-coloring; by claim (ii), we can observe that the box induce by the nodes \( \{v_{2k-1}(H_j), v_{2k}(H_j), x_k(H_j), x_k(H_j)\} \) is necessarily colored in a simple 2-coloring (otherwise \( S_2 \) contains an edge which is a contradiction with (ii)). Now, assume that an edge \( e = [v_{2k}(H_j), v_{2k+1}(H_j)] \) is in \( S_1 \); since any box is colored with a 2-coloring and that \( v_1(H_j), \ldots, v_{2m}(H_j) \) is an even cycle, we necessarily have an edge of this cycle in \( S_2 \), which is still impossible.

Proof of (v): If \( x(F_i) \in S_2 \), then by (ii) we have \( x_i(H_j) \in S_1 \) since we have supposed that \( e = [x_i(H_j), x(F_i)] \) is an edge in \( G \). Now, if \( x(F_i) \in S_1 \), then by the proof of (iii), we know that the nodes \( v_4(F_i), v_5(F_i) \) are in \( S_1 \) too and, then in \( F_i \), there is an edge adjacent to \( x(F_i) \) and belongs to \( S_1 \). Thus, since we have supposed that \( e = [x_i(H_j), x(F_i)] \) is an edge in \( G \), we have \( x_i(H_j) \in S_2 \).

Proof of (vi): Similar to the previous claim.

Finally, from (iii), we know that we have exactly one literal true in each clause. Moreover, the (iii), (iv) and (v) indicate that the function \( f \) is a truth assignment.

We now prove that, even if the graph \( G \) is triangle free and planar with a maximum degree equal to 3, the hypocoloring problem is still \textbf{NP-hard}.

**Theorem 4.4** The hypocoloring problem is strongly \textbf{NP-hard} for the triangle free and planar graphs with maximum degree 3.

**Proof:** In the previous theorem, all gadgets \( F_i \) and \( H_j \) are planar and then only the edges \([x_1(F_i), x_p(H_j)]\) may create some problems since they may cross each other. In this case,
we apply the *crossover* technique (see Garey and Johnson [13]) which consist in replacing each edge crossing by a planar gadget. Since they are only a polynomial number of edge crossings, we obtain a polynomial reduction.

First, we embed the graph \( G' \) of theorem 4.3 in the plane in such a way that every edge is a straight line and the crossing edge occurs only between two edges \([x_i(F_i), x_p(H_j)]\). This can be done in polynomial-time.

Second, we replace each crossing edge by the gadget \((L, w)\) described in Figure 5 and we obtain a new graph \( G'' \) which is planar, without triangle and with a maximum degree equal to 3.

Finally, we prove that there exists a hypocoloring \( S \) of \( G' \) satisfying \( \hat{K}(S) \leq 3 \) if and only if there exist a hypocoloring \( S' \) of \( G'' \) satisfying \( \hat{K}(S') \leq 3 \).

![Fig. 5 – the planar gadget \((L, w)\).](image-url)
A Hypocoloring Model for Batch Scheduling
(Preliminary version)

For sake of simplicity, we have not explicitly described the vertex set and the edge set of the graph \( F \). However, \( F \) contains 8 particular vertices \( x_1, x'_1, y_1, y'_1, x_2, x'_2, y_2, y'_2 \) and the weight of any vertex is one excepted \( x'_1, y'_1, x'_2, y'_2 \) for which is two (i.e., \( w(x'_1) = w(y'_1) = w(x'_2) = w(y'_2) = 2 \)).

Let \( S \) be an hypocoloring of \( G' \) (or \( G'' \)) satisfying \( \hat{K}(S) \leq 3 \); since \( G' \) (or \( G'' \)) contains the previous gadgets, we know that \( S = (S_1, S_2) \) with \( w(S_1) = 2 \) and \( w(S_2) = 1 \). Moreover, we have the following properties:

For any hypocoloring \( S = (S_1, S_2) \) satisfying \( \hat{K}(S) \leq 3 \), we have:

(i) \( x_1 \) and \( x_2 \) have not the same color.

(ii) \( y_1 \) and \( y_2 \) have not the same color.

(iii) \( \{x'_1, x'_2, y'_1, y'_2\} \subseteq S_1 \)

(iv) For any \( x = x_1, x_2, y_1, y_2 \), the neighbors outside of the gadget \( (L, w) \) have not the same color as \( x \).

These properties are not proved, but they can be easily shown by checking the four cases \( (x_1, y_1 \in S_1 \text{ or } x_1 \in S_2 \text{ and } y_1 \in S_2 \text{ or } x_1 \in S_2 \text{ and } y_1 \in S_1 \text{ or } x_1, y_1 \in S_2) \).

Finally, from these properties, we deduce that there exists a hypocoloring \( S \) of \( G' \) satisfying \( \hat{K}(S) \leq 3 \) if and only if there exists a hypocoloring \( S' \) of \( G'' \) satisfying \( \hat{K}(S') \leq 3 \).

\[ \square \]

Now, we deal with the case where the graph is bipartite and the maximum degree bounded above by 39.

**Theorem 4.5** The hypocoloring problem is strongly NP-hard for bipartite graphs with maximum degree at most 39.

**Proof:** We polynomially transform the pre-extension coloring problem \( 1 - PrExt \) (proved to be NP-complete in Bodlaender et al. [4]) into the hypocoloring problem in bipartite graphs; \( 1 - PrExt \) can be described as follows: given a bipartite graph \( G = (V, E) \) with \( |V| \geq 3 \) and maximum degree equal to 12 and three nodes \( v_1, v_2, v_3 \), does there exist a 3-coloring \( (S_1, S_2, S_3) \) of the nodes of \( G \) such that \( v_i \in S_i \) for \( i = 1, 2, 3 \)?

Let \( G = (L, R; E) \) be a bipartite graph where \( L \) (resp. \( R \)) is the "left set" (resp. "right set") of nodes and each edge has one endpoint in \( L \) and the other in \( R \) and let \( v_1, v_2, v_3 \) be three specific nodes (w.l.o.g. we may assume \( \{v_1, v_2, v_3\} \subseteq L \)).

We polynomially construct a new bipartite graph \( G' \) such that there exists a hypocoloring \( S \) of \( G' \) with \( \hat{K}(S) \leq 7 \) if and only if there exists a 3-coloring \( (S_1, S_2, S_3) \) of \( G \) with \( v_i \in S_i, \ i = 1, 2, 3 \). In order to do that, we use the two following gadgets:

- The bipartite graph \( H_0 = (L_0, R_0; E_0) \) where \( L_0 = \{l_0, l'_1, l_1, l'_2, l_2, l'_3, l_3\} \) and \( R_0 = \{r_1, r'_1, r_2, r'_2, r_3\} \) and the edges are for any \( 1 \leq i \leq 3 \), \( [l_i, y_i], [l'_i, y'_i] \) for \( y_i \in R_0 \setminus \{r_i, r'_i\} \). Finally, we get \( w(l_i) = w(l'_i) = w(r_i) = w(r'_i) = 23-i \) for \( i = 1, 2, 3 \). This graph is described in Figure 6.
The complete bipartite graph $K_{3,2}$ where one node $x$ of the left set and one node of the right set $y$ are specified. The weights are $w(x) = w(y) = 1$ and $w(v) = 2$ otherwise.

Now, the instance $I = (G', w)$ of hypocoloring is built in the following way: Starting from $G$, we add a copy of $H_0$ and we identify nodes $v_1, v_2, v_3$ of $G$ with nodes $l_1, l_2, l_3$ of $H_0$. Moreover, for each edge $e = [l, r]$ of $G$, we introduce a copy of $K_{3,2}$ and we identify nodes $l, r$ with nodes $x_e, y_e$ respectively. We call this graph $H_e$ where $L_e$ and $R_e$ denote respectively, the left and right sets. This gadget is described in Figure 7.
A Hypocoloring Model for Batch Scheduling

(Preliminary version)

Note that \( G' \) is still bipartite and its maximum degree is bounded above by \( 39 (\Delta(G')) \leq 3\Delta(G) + 3 = 39 \).

Let \((S_1, S_2, S_3)\) be a \(3\)-coloring of \( G \) with with \( v_i \in S_i, \ i = 1, 2, 3\); then we get a hypocoloring \( S' \) of the nodes of \( G' \) by applying the following process:

We start with \( S'_1 = (S_1 \setminus \{v_i\}) \cup \{l_i, r'_i\} \) for \( i = 1, 2, 3 \) and for each edge \( e = [l, r] \) of \( G \) with \( l \in L \) and \( r' \in R \), we do:

- If \( l \in S_j \) with \( j = 1, 2 \) then we take \( S'_j = S'_j \cup (L_e \setminus \{x_e\}) \)
- If \( r \in S_j \) with \( j = 1, 2 \) then we take \( S'_j = S'_j \cup (R_e \setminus \{y_e\}) \)
- If \( l \in S_3 \) and \( r \in S_j \) with \( j = 1, 2 \) then we take \( S'_{3-j} = S'_{3-j} \cup (L_e \setminus \{x_e\}) \)
- If \( r \in S_3 \) and \( l \in S_j \) with \( j = 1, 2 \) then we take \( S'_{3-j} = S'_{3-j} \cup (R_e \setminus \{y_e\}) \)

\( S' \) is a coloring of \( G' \) (thus a hypocoloring) and verifies \( \hat{K}(S') = w(S'_1) + w(S'_2) + w(S'_3) = 7 \).

Conversely assume \( G' \) has a \( k\)-hypocoloring \( S' = (S'_1, \ldots, S'_k) \) verifying \( \hat{K}(S') \leq 7 \). Moreover, suppose that we have ordered the hypostable sets in such way that \( w(S'_1) \geq \ldots \geq w(S'_k) \); we deduce the following properties:

(i) \( \{l_1, l'_1, r_1, r'_1\} \subseteq S'_1 \) and \( \{l_1, l'_1, r_1, r'_1\} \) is a stable set of \( S'_1 \) (i.e., in the subgraph induced by \( S'_1 \), they are isolated nodes).

(ii) \( \{l_2, l'_2, r_2, r'_2\} \subseteq S'_2 \) and \( \{l_2, l'_2, r_2, r'_2\} \) is a stable set of \( S'_2 \) (i.e., in the subgraph induced by \( S'_2 \), they are isolated nodes).

(iii) \( k = 3 \) and \( w(S'_1) = 4 \), \( w(S'_2) = 2 \), \( w(S'_3) = 1 \).

(iv) \( \{l_3, r_3\} \subseteq S'_3 \) and \( S'_3 \) is a stable set of \( G' \).

**Proof of (i)**: By construction, \( w(S'_1) \geq 4 \) and we have \( \{l_1, l'_1, r_1, r'_1\} \subseteq S'_1 \) since otherwise \( \hat{K}(S') \geq w(S'_1) + 4 \geq 8 \). Assume that there exists \( x \in S'_1 \) and \( y \in \{l_1, l'_1, r_1, r'_1\} \) such that \([x, y] \in E\). By symmetry of \( H_0 \), we can suppose \( y = l_1 \); since \( G' \) contains no triangle and the edge \([x, l_1] \in E\), we conclude that \( l_1 \notin S'_1 \), which gives a contradiction.

**Proof of (ii)**: From the definition of hypostable and for (i), we deduce \( \{l_2, l'_2, r_2, r'_2\} \cap S'_1 = \emptyset \). So, \( w(S'_2) \geq 2 \); if \( \{l_2, l'_2, r_2, r'_2\} \notin S'_2 \), then \( \hat{K}(S') \geq w(S'_1) + w(S'_2) + 2 \geq 8 \) which is impossible. Next, as for (i), we can prove that these nodes form a stable set in the subgraph induced by \( S'_2 \) and then, \( l_3 \notin S'_1 \cup S'_2 \) and \( r_3 \notin S'_1 \cup S'_2 \); we deduce that \( S'_3 \neq \emptyset \).

**Proof of (iii)**: From the previous remark, we have \( w(S'_3) \geq 1 \) and if \( S'_4 \neq \emptyset \) or \( w(S'_3) \neq 4 \) or \( w(S'_2) \neq 2 \) or \( w(S'_1) \neq 1 \), then we have \( \hat{K}(S') \geq 8 \) which is impossible.

**Proof of (iv)**: We necessarily have \( \{l_3, r_3\} \subseteq S'_3 \); moreover, \( S'_3 \) contains only nodes of weight one. Now, assume that \( S'_3 \) is not a stable set, we will have \( w(S'_2) \geq 2 \) and it is in contradiction with (iii).

We want to prove that \((S_1, S_2, S_3) = (S'_1 \cap V, S'_2 \cap V, S'_3 \cap V)\) is a \(3\)-coloring of \( G \) with \( l_i \in S_i \); let us suppose the contrary, there will exist an edge \( e = [r', l] \in E \) with \( \{l, r\} \subseteq S'_j \) and \( j = 1 \) or \( 2 \) (since \( S'_3 \) is an stable set, see (iv)); Since the subgraph induced
by \( L_e \cup R_e \) is hypocolored with at most three colors, we have by construction of \( H_e \), \( S_j' \cap (L_e \setminus \{l\}) = \emptyset \) and \( S_j' \cap (R_e \setminus \{r\}) = \emptyset \). Then, since \( (L_e \cup R_e) \setminus \{l, r\} \) must be hypocolored with at least two colors and the cost of these nodes are two, we deduce that \( S_j' \) contains a node of weight two, which is impossible. Finally, (i), (ii) and (iv) indicate that \( l_i \in S_i \) which concludes the proof.

\[ \Box \]

# 5 Additional properties of optimal hypocolorings

We will derive here some properties which are based on the fact that hypocolorings are in some sense extensions of node colorings; the following is a simple extension of the observation that the number of colors used in an optimal weighted coloration of a simple graph does not exceed \( \Delta(G) + 1 \) (see Demange et al. [9]).

**Lemma 5.1** In a weighted graph \( I = (G, w) \) such that \( \forall v \in V, w(v) > 0 \), any \( k \)-hypocoloring \( S \) with minimum cost \( \hat{K}(S) \) satisfies \( k \leq \Delta(G) + 1 \)

**Proof:** We shall show that any \( l \)-hypocoloring \( S' = (S'_1, \ldots, S'_l) \) with \( l > \Delta(G) + 1 \) can be transformed into a \( k \)-hypocoloring \( S \) with \( k \leq \Delta(G) + 1 \) and \( \hat{K}(S) < \hat{K}(S') \). Moreover, this construction is done in polynomial time. As usual we assume \( w(S'_1) \geq w(S'_2) \geq \ldots \geq w(S'_l) \); assume \( S'_i \neq \emptyset \), so there is a node \( x \in S'_i \). It has at most \( \Delta(G) \) neighbors. Since, \( l > \Delta(G) + 1 \), there is at least one color say \( s \), which satisfies \( s \leq \Delta(G) + 1 < l \) and which does not occur in the neighborhood \( N(x) \). So, we can recolor \( x \) with color \( s \) and setting \( S'_s = S'_s \setminus \{x\} \), we have \( w(S'_s) = w(S'_s) \) since \( x \leq w(S'_l) \leq w(S'_s) \) and also \( w(S'_s \setminus \{x\}) \leq w(S'_s) \). Setting \( S'_s = S'_s \) \( (i \neq s, l) \) we get a \( l \)-hypocoloring \( S^* \) with \( |S'_s| < |S'_1| \) and \( \hat{K}(S^*) \leq \hat{K}(S') \). We repeat this until all nodes in \( S'_s \) have been recolored with a smaller color, then we continue until there are no more nodes with colors \( s > \Delta(G) + 1 \). At the end, the cost of the resulting hypocoloring verifies \( \hat{K}(S^*) < \hat{K}(S') \) since we have assumed \( \forall v \in V, w(v) > 0 \).

This bound is not the best possible; by analogy with the theorem of Brooks ([7]), we could try to get a bound of \( \Delta(G) \) instead of \( \Delta(G) + 1 \). The bound \( \Delta(G) + 1 \) is attained when \( G \) is a clique but in this case, we also have an optimal solution with exactly \( i \) colors for any \( 1 \leq i \leq \Delta(G) + 1 \). This motivates the next improvement of the bound.

We now state some results which hold for general graphs \( G \) with maximum degree \( \Delta(G) \)

**Proposition 5.2** If \( I = (G, w) \) is a weighted graph with maximum degree \( \Delta(G) \) then there exists a \( k \)-hypocoloring \( S = (S_1, \ldots, S_k) \) with minimum cost \( \hat{K}(S) \) satisfying the following:

(i) \( k \leq \Delta(G) \)
A Hypocoloring Model for Batch Scheduling

(Preliminary version)

(ii) \( \forall i \leq k, \forall v \in S_i, d_{G_i}(v) \geq i - 1 \) where \( G_i \) is the subgraph of \( G \) induced by \( S_1 \cup \ldots \cup S_i \).

(iii) \( \forall i \leq k, S_i \) contains no \( K_{\Delta(G)+3-i} \)

**Proof:** Let us consider an optimum cost of the hypocoloring. We repeat this as long as possible and the result follows.

Proof: Let us consider an optimum cost of the hypocoloring. We repeat this as long as possible and the result follows.

\( \Delta(G) \leq k \) and we get a new \( k \)-hypocoloring \( S' \) with \( \hat{K}(S') \leq \hat{K}(S) \) and \( |S'_k| < |S_k| \), a contradiction.

So, we can assume that all colors \( 1, \ldots, \Delta(G) \) occur in \( N(v) \). Let \( u \) be a neighbor of \( v \) in \( S_{\Delta(G)} \) (i.e., \( u \in N(v) \cap S_{\Delta(G)} \)); \( u \) has some color \( c \leq \Delta(G) \) missing in \( N(u) \) since \( v \in S_{\Delta(G)+1} \) is in \( N(u) \).

If \( c < \Delta(G) \), then we can recolor \( u \) with \( c \) and then \( v \) can be recolored with \( \Delta(G) \); we still have a \( k \)-hypocoloring and the cost has not increased. Again \( |S'_k| < |S_k| \), a contradiction.

So we must have color \( c = \Delta(G) \) missing in \( N(u) \). We recolor \( v \) with color \( c = \Delta(G) \) and we get a new \( k \)-hypocoloring (edge \( [v, u] \) is now in \( S'_{\Delta(G)} \)).

Repeat this as long as there are nodes with color \( k = \Delta(G) + 1 \), we will finally have a \( \Delta(G) \)-hypocoloring \( S' = (S'_1, \ldots, S'_{\Delta(G)}) \) with \( \hat{K}(S') \leq \hat{K}(S) \) because \( w(S'_{\Delta(G)}) \leq w(S_{\Delta(G)}) + w(S_{\Delta(G)+1}) \). This is a contradiction again. Since we have examined all cases, (i) is proven.

For (ii): The proof is exactly the same as previously. If \( d_{G_i}(v) < i - 1 \) for some \( v \in S_i \) then, we can recolor \( v \) with some color missing in \( \{1, \ldots, i - 1\} \) without increasing the cost of the hypocoloring. We repeat this as long as possible and the result follows.

For (iii): Assume that \( S_i \) contains a \( K_{\Delta(G)+3-i} \) and let \( v \) be a node of this clique. Since \( d_G(v) \leq \Delta(G) \), we deduce that \( d_{G_i}(v) \leq d_G(v) - (\Delta(G) + 2 - i) \leq i - 2 \) which gives a contradiction with (ii).

**Remark 5.3** We observe that it is always possible to find in polynomial time, a hypocoloring which verifies proposition (5.2). In other words, we can polynomially transform a hypocoloring \( S \) satisfying lemma (5.1) into a hypocoloring \( S' \) satisfying proposition (5.2) and verifying \( \hat{K}(S') \leq \hat{K}(S) \).

**Remark 5.4** The bound (i) is best possible: for every integer \( p \), there exists a tree \( G \) with \( \Delta(G) = p \) and weights \( w(v) \) for the nodes of \( G \) such that all optimal \( k \)-hypocolorings have \( k = p \) colors.
$G$ is constructed as follows: start from the tree $T_2$ for $p = 2$; it consists of a
chain $a, b, c$ where nodes have labels $l(a) = l(c) = 2$, $l(b) = 1$ and $w(a) = w(c) = 10^3$, $w(b) = 10^9$.

Generally having obtained tree $T_{i-1}$ (where nodes have labels in $1, 2, \ldots , i-1$ and
weights at most $10^{i-2}$) we construct $T_i$ by introducing at each node $v$ of $T_{i-1}$ a chain of
two additional nodes $v', v''$ where these new nodes have label $i$ and weights $(1/2)10^{i-1}$.
Now in $T_i$ we take one of these new chains $v', v''$ such that the node adjacent in $T_{i-1}$ has a
cost $10^{i-2}$ and we condense the edge $[v', v'']$ into the node $v_i$ (the weight of the new node
is $w(v_i) = w(v') + w(v'') = 10^{i-1}$).

One can verify that the graph $T_i$ is a tree with maximum degree $i$; one also observes
that the labels define a $p$-hypocoloring $S = (S_1, \ldots , S_p)$ where $S_i = \{v | l(v) = p+1-i\}$. Furthermore one can verify that $S$ is the unique optimum hypocoloring.

For bipartite graphs, we can obtain a bound of the number of different colors used in any
optimal coloring $S^*$ using the size of the weight function $w$ (i.e., the number of distincts
values of the weights $w$). We will denote by $|w|$ the size of $w$ :

**Proposition 5.5** If $I = BP(L, R, w)$ is a weighted bipartite graph such that $w(v) > 0$
for each $v \in V$, then any hypocoloring $S^* = (S_1, \ldots , S_k)$ with minimum cost $\hat{K}(S)$
satisfies : $k = |S^*| \leq |w| + 1$

**Proof :** Let $BP(L, R, w)$ be a weighted bipartite graph and $S^* = (S_1, \ldots , S_l)$ be an
optimal hypocoloring of $BP$ with $w(S_1) \geq \ldots \geq w(S_l)$. We show that $l = |S^*| \leq |w| + 1$
by an inductive proof on $|w|$ :

If $|w| = 1$ then from proposition (4.2), we know that the hypocoloring problem is
equivalent to the coloring problem. Hence, we have : $w_{\text{max}}|S^*| = \hat{K}(S^*) = w_{\text{max}}\chi(G) = 2w_{\text{max}} = w_{\text{max}}(|w| + 1)$ and the result follows.

Now, assume that the result is true when $|w| \leq k$ and prove the result when $|w| = k + 1$.

If $\{v : w(v) = w_{\text{max}}\}$ is not a stable set, then from property (4.1), $S = (L, R)$ is also
an optimal hypocoloring with two colors and any optimal hypocoloring has at most two
colors.

Now, assume that $V_{\text{max}} = \{v : w(v) = w_{\text{max}}\}$ is a stable set. If $V_{\text{max}} \not\subseteq S_1$, then $S^*$
has two colors and the result is true. Otherwise, $V_{\text{max}} \subseteq S_1$ and delete $S_1$ from $G$. This
new graph $BP'$ satisfies the inductive hypothesis and then, any optimal hypocoloring $S'^*$
of $G'$ verifies : $|S'^*| \leq k$. Finally, since $(S_2, \ldots , S_l)$ is an optimal hypocoloring of $G'$, we
have the expected result. \qed
6 Approximability of some cases of hypocoloring.

We shall present here approximation algorithms for hypocolorings in some special classes of graphs. More specifically, we are interested in a class of graphs for which the coloring problem is easy and the chromatic number is small; formally, we denote by \( \Psi_k \) a class of graphs verifying:

(i) \( \forall G \in \Psi_k, \quad \chi(G) \leq k \) and (ii) the coloring problem on \( \Psi_k \)-graphs is polynomial.

For instance, a \( \Psi_2 \)-class is the case where \( G \) is a tree.

Let \( \Psi_k \) be a class of graphs described above such that the hypocoloring problem is NP-hard (in particular, we have \( k \geq 2 \)). \( I = (G, w) \) will be an instance of hypocoloring where \( G \in \Psi_k \) and the nodes are sorted by non-increasing weights; so, \( w(v_1) \geq \ldots \geq w(v_n) \). Moreover, we can always suppose that \( w(v) > 0, \forall v \in V \) (delete the nodes \( v \) with \( w(v) = 0 \)).

We also denote by \( G_i \) the subgraph induced by the node set \( \{v_1, \ldots, v_i\} = V_i \) and by \( j_0 \) the smallest index \( i \) such that \( G_i \) contains an induced \( P_3 \). When \( G_n = G \) does not contain an induced \( P_3 \), we set \( j_0 = n + 1 \). Finally, we denote by \( C(V') \) an optimal coloring of the subgraph of \( G \) induced by \( V' \).

7 Standard approximation.

A trivial bound of the standard approximability on \( \Psi_k \)-graphs is \( k \) and consists of computing \( C(V) \) in a whole graph (in other words, we just compute an optimal coloring). We now propose a polynomial time approximation algorithm achieving a better constant standard approximation ratio for this class of graphs. This algorithm, denoted by \( \Psi_k \_HYPOCOLOR \) works as follows:

1. Sort the nodes of \( G \) in non-increasing weight order;
2. Compute \( j_0 = \min\{i : G_i \text{ contains an induced } P_3 \} \) and \( V_{j_0} = \{v_1, \ldots, v_{j_0}\} \);
3. For \( i = 1 \) to \( j_0 \) do
   (a) \( S_i^1 = V_{j_0} \setminus \{v_i, \ldots, v_{j_0}\} \);
   (b) Compute \( C(V \setminus S_i^1) \) (i.e. an optimal coloring on the unweighted graph induced by \( V \setminus S_i^1 \));
   (c) Define hypocoloring \( S_i = (S_i^1, C(V \setminus S_i^1)) \);
4. Compute \( S = \arg\min\{ \tilde{K}(S^i) : i = 1, \ldots, j_0 \} \);

We can easily observe that each hypocoloring \( S^i \) is feasible and that its number of colors is at most \( k + 1 \). So, the algorithm \( \Psi_k \_HYPOCOLOR \) is correct and its time-complexity is similar to the one of computing an optimal coloring on \( \Psi_k \)-graphs time \( n \).
**Theorem 7.1** Algorithm $\Psi_{k_{\text{HYPOCOLOR}}}$ polynomially solves hypocoloring in $\Psi_k$-graphs within standard approximation ratio bounded above by $\frac{k^2}{2k-1}$.

**Proof**: Let $G = (V, E)$ be a weighted $\Psi_k$-graph and $S^* = (S_1^*, \ldots, S_t^*)$ be an optimal hypocoloring of $I = (G, w)$ with $w(S_1^*) \geq \ldots \geq w(S_t^*)$. If $l = 1$, then $j_0 = n + 1$ and the solution $S$ computed by $\Psi_{k_{\text{HYPOCOLOR}}}$ verifies $\hat{K}(S) \leq \hat{K}(S^{i_0}) = \hat{K}(S^*)$. So, assume that $j_0 \leq n$; denote by $S_i$ the set $S_i^1 \cap V_i$ for any $i \leq j_0$.

If $S_1 = \emptyset$ then $\hat{K}(S^*) \geq 2w_{\text{max}}$ and $\hat{K}(S) \leq \hat{K}(S^1) \leq k w_{\text{max}}$. Thus, we have $\hat{K}(S) \leq \frac{k}{2} \hat{K}(S^*)$.

If $S_1 \neq \emptyset$ then we obtain $S_1 = V_1$; on the other hand, we also have $S_{j_0} \neq V_{j_0}$ since $V_{j_0}$ is not a hypostable set. Thus, the index $i_0 = \max \{i : S_i = V_i\}$ exists and verifies $i_0 < j_0 \leq n$. Now, we are interested in the solution $S^{i_0} = (S_1^{i_0}, \ldots, S_{k+1}^{i_0})$; by construction, we have: $w(S_1^{i_0}) \geq \ldots \geq w(S_{k+1}^{i_0})$.

For $r \geq 3$,

$$w(S_r^{i_0}) \leq \frac{1}{2}(w(S_1^{i_0}) + w(S_2^{i_0})) \leq \frac{k-1}{2k-1} (w(S_1^{i_0}) + w(S_2^{i_0}))$$

So

$$w(S_3^{i_0}) + \ldots + w(S_{k+1}^{i_0}) \leq \frac{(k-1)^2}{2k-1} (w(S_1^{i_0}) + w(S_2^{i_0}))$$

Hence

$$\hat{K}(S^{i_0}) \leq \left(\frac{(k-1)^2}{2k-1} + 1\right) (w(S_1^{i_0}) + w(S_2^{i_0})) \leq \frac{k^2}{2k-1} (w(S_1^{i_0}) + w(S_2^{i_0})) \quad (2)$$

Since $\hat{K}(S^*) \geq w(S_1^{i_0}) + w(S_2^{i_0})$ from the choice of $i_0$ and from the fact that $S_2^{i_0}$ is a stable set ($w(S_2^{i_0})$ is the maximum weight of the nodes in $V \setminus S_1^{i_0}$) the result follows. $\square$

**Corollary 7.2**

- There exists a $\frac{16}{9}$-standard approximation for the hypocoloring problem in graphs with maximum degree at most $3$.
- There exists a $\frac{3}{2}$-standard approximation for the hypocoloring problem in $K_4$-free graphs with maximum degree at most $3$.

**Proof**: we know that $\gamma(G) \leq 4$ when $G$ has maximum degree bounded above by $3$ and $\gamma(G) \leq 3$ when $G$ is also $K_4$-free (by Brooks theorem). Moreover we can compute in polynomial-time an optimal coloring, when the graphs have a maximum degree bounded above by $3$. Thus, in the first case, these graphs are a $\Psi_4$-class whereas in the second case, these graphs are a $\Psi_3$-class. In any case, we can apply the theorem 7.1. $\square$

**Corollary 7.3** There exists a $\frac{9}{5}$-standard approximation for the hypocoloring problem in triangle free planar graphs.
Proof: From Grotzsch theorem [14], we know that the coloring problem is easy for triangle free planar graphs. Thus, since $\gamma(G) \leq 3$ when $G$ is triangle free planar, we deduce that triangle free planar graphs is a $\Psi_3$-class, and then we can apply theorem 7.1.

The bipartite graphs are also a special case of $\Psi_2$-class and then, we have:

**Corollary 7.4** There exists a $\frac{4}{3}$-standard approximation for the hypocoloring problem in bipartite graphs.

On the other hand, we can also establish a bound on the standard approximability of these two types of graphs.

Examine first the proof of theorem 4.4, consider a triangle-free planar graph $G$. Remember that $\hat{K}(S^*) \leq 3$ iff the answer for $1 - IN\ 3SAT$ is yes. Assume now that there exists a polynomial time approximation algorithm $A$ achieving, for some $\epsilon_0 > 0$, an approximation ratio $(4/3) - \epsilon_0$.

- If $\hat{K}(S^*) \leq 3$ (the answer for $1 - IN\ 3SAT$ in $I = (C, X)$ is yes), then $\text{val}_A(G) = 4 - 3\epsilon_0$ and, since val has to be integer, $\text{val}_A(G) = 3$;
- on the other hand, if $\hat{K}(S^*) \geq 4$, i.e., the answer for $1 - IN\ 3SAT$ in $I = (C, X)$ is no, then $\text{val}_A(G) \geq \hat{K}(S^*) \geq 4$.

Consequently, with the hypothesis that a polynomial time approximation algorithm $A$ achieves, for some $\epsilon_0 > 0$, approximation ratio $(4/3) - \epsilon_0$ for hypocoloring, one can in polynomial time decide on the right answer for $1 - IN\ 3SAT$ by simply reading the value of the solution computed by $A$. The following proposition summarizes the above discussion.

**Proposition 7.5** Unless $P = NP$, for any $\epsilon > 0$ no polynomial time algorithm achieves an approximation ratio bounded above by $(4/3) - \epsilon$ for the hypocoloring problem in triangle-free planar graphs with maximum degree bounded above by 3.

By the same type of proof and from theorem 4.5, we can also show:

**Proposition 7.6** Unless $P = NP$, for any $\epsilon > 0$ no polynomial time algorithm achieves an approximation ratio bounded above by $(7/8) - \epsilon$ for the hypocoloring problem in bipartite graphs with maximum degree bounded above by 39.

**8 Differential approximation.**

We now show that in the $\Psi_k$-graphs, there is a differential approximation scheme. Assume that $G = (V, E) \in \Psi_k$ and consider then the following algorithm, called $\Psi_k$-Hypo_SCHEME in what follows and run it with parameters $G$ and a fixed constant $\epsilon > 0$:
1. Sort the nodes of $G$ in non-increasing weight order;
2. Set $\eta = \lceil 1/\epsilon \rceil$;
3. Compute an optimal hypocoloring $\hat{S}$ of the graph induced by $V_{2k\eta+k}$;
4. Compute an optimal coloring $C(V_{2k\eta+k})$ of the subgraph induced by $V \setminus V_{2k\eta+k}$;
5. Set $S = (\hat{S}, C(V_{2k\eta+k}))$;

Since $\eta$ and $k$ are fixed constants, the set $\hat{S}$ of step 3 of algorithm $\Psi_k\text{-Hypo\_SCHEME}$ can be computed by an exhaustive search in constant time. Consequently, the whole complexity of $\Psi_k\text{-Hypo\_SCHEME}$ is linear in $n$.

**Theorem 8.1** For any fixed $\epsilon > 0$, the differential approximation ratio of algorithm $\Psi_k\text{-Hypo\_SCHEME}$ when called with inputs $G \in \Psi_k$ and $\epsilon$, is bounded below by $1 - \epsilon$.

**Proof** : Let $G = (V, E)$ be a weighted $\Psi_k$-graph and $S^* = (S_1^*, \ldots, S_l^*)$ be an optimal hypocoloring of $I = (G, w)$ with $w(S_1^*) \geq \ldots \geq w(S_l^*)$. We first show that the number of hypostable sets of $\hat{S}$ is not very large and that $\hat{K}(S^*) \geq \hat{K}(\hat{S})$.

*claim (i)* : $|\hat{S}| \leq k(\eta + 1)$

*Proof of claim (i)* : We denote by $\hat{V}_1$ the node set formed by the union of hypostable sets of $\hat{S}$ having their number of connected components equal to one; so, the graph $\hat{G}_1$ induced by $\hat{V}_1$ is an union of disjoint cliques plus eventually some edges between these cliques. We affirm that $\hat{G}_1$ has at most $k$ nodes; by construction, the weight of these hypostable sets is $w(\hat{V}_1) = \sum_{v \in \hat{V}_1} w(v)$ and if $|\hat{V}_1| > k$, then an optimal coloring of $\hat{G}_1$ gives a better value (since $\hat{G}_1 \in \Psi_k$) which is impossible; thus, we have : $|\hat{S}| \leq (|V_{2k\eta+k} \setminus \hat{V}_1|)/2 + |\hat{V}_1| = (2k\eta)/2 + k$.

*claim (ii)* : $\hat{K}(S^*) \geq \hat{K}(\hat{S})$

*Proof of claim (ii)* : We take the hypocoloring $S^\prime$ of the subgraph induced by $V_{2k\eta+k}$ which is the restriction of the hypocoloring $S^*$ (i.e., $S^\prime = (S_1^* \cap V_{2k\eta+k}, \ldots, S_l^* \cap V_{2k\eta+k})$).

Since the criterion is maximum, we have : $\hat{K}(S^*) \geq \hat{K}(S^\prime)$; on the other hand, since the property of hypocoloring is hereditary, we also have $\hat{K}(\hat{S}) \leq \hat{K}(S^\prime)$ and the result follows.

Now, since $\text{WOR}(G_{2k\eta+k}) = \sum_{v \in V_{2k\eta+k}} w(v)$ and by inequality of claim (i) and the fact that the nodes of $G$ are sorted in non-increasing weight order, we have : $\text{WOR}(G_{2k\eta+k}) - \hat{K}(\hat{S}) \leq (k\eta)w(v_{2k\eta+k})$. Moreover, since $\eta \geq 1/\epsilon$, we obtain :

$$\epsilon \left( \text{WOR}(G_{2k\eta+k}) - \hat{K}(\hat{S}) \right) \geq kw(v_{2k\eta+k}) \tag{3}$$

Finally, by claim (ii) and since $\text{WOR}(G_{2k\eta+k}) \leq \text{WOR}(G)$, the solution $\hat{S}$ produced by the algorithm $\Psi_k\text{-Hypo\_SCHEME}$ verifies $w(C(V_{2\eta+k})) \leq kw(v_{\eta+k}) \chi(G) \leq k$ since
A Hypocoloring Model for Batch Scheduling
(Preliminary version)

\( G \in \Psi_k \) and by inequality (3), we have:

\[
\hat{K}(S) = \hat{K}(\hat{S}) + w(C(V_{4\eta+k})) \leq (1-\epsilon)\hat{K}(\hat{S}) + \epsilon \text{WOR}(G_{2k\eta+k}) \\
\leq (1-\epsilon)\hat{K}(S^*) + \epsilon \text{WOR}(G)
\]

We note finally that with exactly the same arguments as for the proofs of Proposition 7.6 and Proposition 7.5 and taking into account that \( \text{WOR}(G) - \hat{K}(S^*) \leq P(n) \) for some polynomial \( P \), one can conclude that, unless \( P = \text{NP} \), no polynomial time algorithm can achieve a differential approximation ratio greater than \( 1 - \frac{1}{P(n)} \), and the following holds.

**Proposition 8.2** Unless \( P = \text{NP} \), if the hypocoloring problem is strongly \( \text{NP-hard} \) on a \( \Psi_k \)-class, then the hypocoloring problem cannot be solved in a \( \Psi_k \)-class by a fully polynomial time differential approximation scheme.

This is the case for the triangle-free graphs with maximum degree equal to 3 and for the bipartite graphs.

**9 Polynomial cases**

In this section, we consider two polynomial cases of the weighted hypocoloring problem: when the input is a collection of disjoint trees with maximum degree at most \( \Delta \) and when the input is a collection of disjoint cycles. The first case is equivalent to solve hypocoloring in trees with degree at most \( \Delta \) whereas the second case is equivalent to solve hypocoloring in graphs with degree at most 2, as we will be shown later. Thus, since a tree is a particular bipartite graph, we have a frontier for the hardness of the hypocoloring problem between trees with maximum degree at most 39 and bipartite graphs with maximum degree at most 39; the first case will be shown to be polynomial in the next subsection whereas the bipartite case has been shown to be strongly \( \text{NP-hard} \) (see proposition 4.5). Finally, we will see that there is also another hardness gap for general graphs between graphs with maximum degree at least 3 and graphs with maximum degree at most 2.

Before establishing these results, we shall give some results on hypocolorings in \((r + 1)\)-clique free graphs. For a hypostable set \( S \), the **characteristic value** will be the integer number \( q \) such that \( q = w(S) \). Since we are in \((r + 1)\)-clique free graphs, there are at most \( O(n^r) \) possible characteristic values of the different hypostable sets. More generally, for a hypocoloring \( S = (S_1, \ldots, S_k) \) with \( w(S_1) \geq \ldots \geq w(S_k) \) we call **vector of characteristic values**, the vector \((q_1, \ldots, q_k)\) such that for any \( i \leq k, q_i = w(S_i) \). In \((r + 1)\)-clique free graphs, given a vector \((q_1, \ldots, q_k)\) with \( q_1 \geq \ldots \geq q_k \), the problem of
deciding if there is a hypocoloring \( S' \) whose vector of characteristic values \((q_1', \ldots, q_k')\) verifies for any \( i, q_i' \leq q_i \) can be polynomially reduced to the list-hypocoloring, problem as will be shown below. This latter problem is formally described as follows :

\[ \text{list-hypocoloring}_{r'} : \]

**Instance:** a graph \( G = (V, E) \), a set \( C \) of colors and, for every clique \( K \) with size at most \( r \), \( C_K \subseteq C \) is a set of colors such that each one of them may occur on some nodes of the clique \( K \) but not on all nodes at a time.

**Question:** does \( G \) admit a hypocoloring such that for any clique \( K \), not all the nodes of \( K \) have the same color \( i \) with \( i \in C_K \)?

**Remark 9.1** Clearly, we must have \( C_K \) contained in \( C_K' \) when \( K \subseteq K' \). Moreover, the problem list-hypocoloring, polynomially reduces to the problem list-hypocoloring, when \( r \leq r' \).

**Remark 9.2** Note that when list-hypocoloring, is polynomial for a class \( \Psi \) of \((r+1)\)-clique free graphs then, the related problem of constructing such a hypocoloring is polynomial. Let \( I = (G, (C_K) \) be an instance of list-hypocoloring,. In order to do that, we find a \( r \)-clique \( K \) of \( G \) such that \( C_K \neq C \). Let \( i \notin C_K \) and denote by \( G' \) the subgraph of \( G \) induced by \( V \setminus V(K) \); we color \( K \) with color \( i \) and for each node \( v \) of \( G' \) which is adjacent to a node of \( K \), we set \( C_v = C_v \cup \{ i \} \) and more generally for each clique \( K \) of \( G' \) containing \( v \), we set (see remark 9.1) \( C_K = C_K \cup \{ i \} \). Let \( I' \) be the resulting instance. If the answer of list-hypocoloring, on instance \( I' \) is yes, we apply again this procedure on \( I' \) else we set \( C_K = C_K \cup \{ i \} \) and we apply again this procedure on \( I \). Finally, when \( C_K = C \) for every \( r \)-clique \( K \) of \( G \), we apply the same procedure with \((r-1)\)-cliques and so on.

For every clique \( K \) of \( G \), we set \( w(K) = \sum_{v \in K} w(v) \). Let \((q_1, \ldots, q_k) \) with \( q_1 \geq \ldots \geq q_k \) be a vector. Since \( G \) is \((r+1)\)-clique free, each hypotable set is a union of disjoint cliques with size at most \( r \), we can polynomially construct an instance of list-hypocoloring, in the following way : the graph \( G \) is the same, \( C = \{1, \ldots, k\} \) and for any clique \( K \) of \( G \), \( C_K = \{i \in C : q_i < w(K)\} \). Assume that the answer of the list-hypocoloring, instance is yes and let \( S' = (S'_1, \ldots, S'_k) \) with \( w(S'_1) \geq \ldots, w(S'_k) \) (some \( S'_i \) can be empty) be such a hypocoloring. Let \( K \) be a clique of \( S'_i \); by construction \( w(K) \leq q_i \) since \( i \notin C_K \) and then, \( q_i' = w(S'_i) \leq q_i \). Conversely, if such a hypocoloring exists then, the answer of the list-hypocoloring, instance is yes. So, we have proved the result claimed.

In the graphs with maximum degree \( \Delta \) (which is a subclass of \((\Delta + 2)\)-clique free graphs), we now prove that hypocoloring polynomially reduces to list-hypocoloring,\(_{\Delta+1}\).

**Proposition 9.3** The hypocoloring problem polynomially reduces to the list-hypocoloring,\(_{\Delta+1}\) problem for graphs of degree bounded above by \( \Delta \).
Proof: Let \((G, w)\) be a weighted graph with maximum degree \(\Delta\), a minimum hypocoloring can be computed by the following algorithm:

1. For every vector \((q_1, \ldots, q_\Delta)\) with \(q_1 \geq \ldots \geq q_\Delta\) and such that \(q_i = w(K_i)\) for some clique \(K_i\) of \(G\) do:
   - (a) Solve the related list-hypocoloring \(\Delta+1\) instance;
   - (b) If the answer is yes, construct such a hypocoloring;
2. Select a minimum weight hypocoloring among feasible hypocolorings computed during an execution of step (1b);

The complexity of this algorithm is \(O(n\Delta^2)\) times of the complexity of solving the list-hypocoloring \(\Delta+1\) problem. Since \(G\) admits a minimum hypocoloring with at most \(\Delta\) colors (see proposition 5.2), let \(S^* = (S^*_1, \ldots, S^*_\Delta)\) with \(w(S^*_1) \geq \ldots, w(S^*_\Delta)\) be an optimal hypocoloring (with possibly some \(S^*_i = \emptyset\)) and denote by \((q^*_1, \ldots, q^*_\Delta)\) its vector of characteristic values. Let us examine the step of algorithm with the vector \((q^*_1, \ldots, q^*_\Delta)\). By construction, the answer of the list-hypocoloring \(\Delta+1\) instance is yes and the algorithm finds a hypocoloring \(S' = (S'_1, \ldots, S'_\Delta)\) verifying \(w(S'_i) \leq q^*_i\) and then, \(w(S'_i) = q^*_i\) since \((q^*_1, \ldots, q^*_\Delta)\) is a vector of characteristic values of an optimal hypocoloring. \(\square\)

Corollary 9.4 Let us consider a class \(\Psi\) of graphs of degree bounded above by \(\Delta\) such that list-hypocoloring \(\Delta+1\) is polynomial on \(\Psi\). Then, the minimum hypocoloring problem is also polynomial on \(\Psi\).

Corollary 9.5 For any \(r \geq 40\), list-hypocoloring \(r\) in bipartite graphs of degree 39 is \(NP\)-complete.

Corollary 9.6 For any \(r \geq 4\), list-hypocoloring \(r\) in planar triangle-free graphs of degree 3 is \(NP\)-complete.

9.1 Trees with maximum degree \(\Delta\)

In trees, there are at most \(2n-1\) characteristic values for the different hyphostable sets. Thus, the algorithm of proposition 9.3 is in this case in \(O(n^2)\) times the complexity-time of the list-hypocoloring \(\Delta+1\). We now show that we can solve the list-hypocoloring \(\Delta+1\) problem in trees by using dynamic programming.

Let \(C = \{1, \ldots, \Delta\}\) be the set of colors and consider \((T = (V, E); (C_K)_{K \in V \cup E})\) an instance of list-hypocoloring \(\Delta+1\) where \(T\) is a tree. Given a node \(v\), we respectively denote
by $H_v(T)$ and $H'_v(T)$ the sets of colors defined by: $h \in H_v(T)$ (resp. $H'_v(T)$) if and only if there is a feasible hypocoloring for which $v$ is colored by $h$ and no (resp. exactly one) neighbor of $v$ is colored by $h$. We denote by $v_1, \ldots, v_d$ the neighbors of $v$. The deletion of $v$ induces a forest with $d$ connected components; let $T_i, i = 1, \ldots, d$ denote the subtree containing $v_i$. The following lemma can then be easily shown:

**Lemma 9.7** For $h \in C$, we have:

$$h \in H_v(T) \iff \begin{cases} h \notin C_v, \\ \forall j, (H_{v_j}(T_j) \cup H'_{v_j}(T_j)) \setminus [(H_{v_j}(T_j) \cup H'_{v_j}(T_j)) \cap \{h\}] \neq \emptyset \\ h \notin C_v, \\ \forall j, (H_{v_j}(T_j) \cup H'_{v_j}(T_j)) \setminus [(H_{v_j}(T_j) \cup H'_{v_j}(T_j)) \cap \{h\}] \neq \emptyset \end{cases}$$

$$h \in H'_v(T) \iff \begin{cases} \exists j \in \{1, \ldots, d\}, h \in H_{v_j}(T_j) \text{ and } h \notin C_{[v,v_j]}, \\ \forall j' \neq j, (H_{v_{j'}}(T_{j'}) \cup H'_{v_{j'}}(T_{j'})) \setminus [(H_{v_{j'}}(T_{j'}) \cup H'_{v_{j'}}(T_{j'})) \cap \{h\}] \neq \emptyset \end{cases}$$

**Proposition 9.8** For any $t \geq 2$, List-hypocoloring in trees is polynomial.

**Proof**: Let us consider the following polynomial-time algorithm:

1. Choose a root $r \in V$ and orient the tree from $r$ to leaves ($T_v$ denotes the subtree induced by $v$ and its descendants);
2. Compute, for every node $v$ and from leaves to the root, sets $H_v(T_v)$ and $H'_v(T_v)$ (by using lemma 9.7);
3. For every color in $H_r(T_v) \cup H'_r(T_v)$ compute a feasible hypocoloring by using lemma 9.7 (from the root to leaves);

The related complexity is $O(n\Delta^2)$. 

**Corollary 9.9** The hypocoloring problem is polynomial in trees with bounded degree. The related complexity is $O(\Delta^2n^{\Delta+1})$.

**Remark 9.10** We can polynomially transform a forest $(T_1, \ldots, T_p)$ with maximum degree $\Delta$ into a tree $T$ with maximum degree $\Delta$ such that an optimal hypocoloring of the tree is an optimal hypocoloring of the forest. In order to do that, we iteratively take two subtrees of the forest that we link by the leaves with an additional node with cost 0.

When the number of different weights of $w$ is constant, we are able to give a polynomial algorithm on the tree. Indeed, by proposition 5.5, we know that the size of any optimal hypocoloring is bounded by $|w| + 1$, and then, the previous algorithm also works in polynomial-time:

**Corollary 9.11** The hypocoloring problem is polynomial in trees when the number $|w|$ of different weights is bounded. The related complexity is $O(n|w| + 2)$. 

9.2 Graphs with maximum degree two

We shall examine here the special situation where the graph $G$ has maximum degree $\Delta = 2$ (the case $\Delta = 1$ is trivial: $G$ consists of isolated edges and nodes, so $\chi_h(G) = 1$). We prove by a similar technique of previously that the case of maximum degree two is also polynomial. However, the method presented here is slightly more involved than the previous one.

In our situation, the connected components of $G$ are chains and cycles and all hypostable sets will consist of nodes and of edges (cliques of size 2). From proposition (5.2), there exists an optimal hypocoloring $S = (S_1, S_2)$ of $G$ with $w(S_1) \geq w(S_2)$ since $\Delta(G) = 2$. The case $S_2 = \emptyset$ is trivial and can be solved in linear-time. Thus, we will suppose $S_1 \neq \emptyset$ for $i = 1, 2$ and more generally that $\chi_h(G) \geq 2$.

**Lemma 9.12** Let $G$ consist of a collection of node disjoint chains $C_1, \ldots, C_p$ where each node $v$ has weight $w(v) \geq 0$ and $\chi_h(G) \geq 2$; there exists a graph $G'$ consisting of a single cycle $C$ such that for any non negative integer $r$, $G'$ has a 2-hypocoloring $S'$ with $\tilde{K}(S') \leq r$ if and only if $G$ has a 2-hypocoloring $S$ with $\tilde{K}(S) \leq r$.

**Proof**: Let $a_i, b_i$ be the end nodes of chain $C_i$ in $G$ for $i = 1, \ldots, p$. We introduce into $G$ new nodes $c_i, d_i$ with weight $w(c_i) = w(d_i) = 0$ and new edges $[c_i, d_i], [d_i, a_i], [a_i, c_i+1]$ for $i = 1, \ldots, p$ (the indices are taken modulo $p$, which implies that $[b_{p+1}, c_{p+1}]$ is $[b_p, c_p]$).

Let $G'$ be the graph obtained in this way. Clearly any $k$-hypocoloring ($k \geq 2$ since $\chi_h(G) \geq 2$ if $G'$ is not a triangle) $S$ of $G$ can be extended to a $k$-hypocoloring $S'$ of $G'$ with $\tilde{K}(S) = \tilde{K}(S')$: consider w.l.o.g. nodes $c_i, d_i$; if $b_{i-1} \in S_c$ and $a_i \in S_d$ ($c \neq d$) then introduce $c_i$ into $S_d$ and $d_i$ into $S_c$. If $\{b_{i-1}, a_i\} \subseteq S_c$ then introduce $[c_i, d_i]$ into $S_d$. Conversely, the restriction of a 2-hypocoloring $S'$ of $G'$ to the nodes of $G$ gives a 2-hypocoloring $S'$ of $G'$ with $\tilde{K}(S) = \tilde{K}(S')$. $\square$

As a consequence, we can restrict our attention to graphs $G = (V, E)$ whose connected components are cycles and let $n = |V| = |E|$. We will define the weight $w(e)$ of an edge $e = [x, y]$ as the sum $w(x) + w(y)$. Then, we notice that there are at most $n + t$ possible values for $w(S_1)$ where $t$ is the number of triangle of $G$ and $2n$ possible values for $w(S_2)$. It is important to notice that we cannot solve separately the problem in each connected component as we can done in the unweighted version: an optimum hypocoloring is not necessarily formed by optimum hypocoloring on each connected components (see for instance the instance given by two chains of length two $(x_1, x_2, x_3)$ and $(y_1, y_2, y_3)$ with $w(x_1) = 8, w(y_1) = w(y_2) = 5, w(x_2) = 3, w(x_3) = w(y_3) = 1$). From (iii) of proposition (5.2), we know that $S_2$ does not contain any $K_3$. Thus, if there is a triangle in a hypostable set, it must be in $S_1$.

**Property 9.13** $\max_{e \in E} w(e) \leq w(S_1) \leq \max \{\max_{e \in E} w(e); \max_{K_3} w(K_3)\}$
For given values $p, q$ ($p \geq q$) we shall determine whether a 2-hypocoloring $S = (S_1, S_2)$ of $G$ exists such that $w(S_1) = p$ and $w(S_2) = q$. If such a hypocoloring can be found, we store the current solution $S = (S_1, S_2)$ with $\tilde{K}(S) = p + q$ if it is better than the best solution found so far.

We shall use in $A(p, q)$ the following properties:

**Property 9.14** If $w(v) > q$, then $v \in S_1$; if $x, y, z$ are three consecutive nodes on an induced $P_3$ with $x, y \in S_i$ then $z \in S_{3-i}$ for $i = 1, 2$.

**Property 9.15** If for some edge $e = [x, y]$, we have $w(e) > p$, then $x, y$ are neither both in $S_1$ nor both in $S_2$; if $w(e) > q$, then $x, y$ are neither both in $S_2$. In such situations, we shall simply say that the color $i$ is not feasible for edge $e = [x, y]$.

Starting from $G$ with given values $p, q$ we will apply the above properties as long as possible to derive consequences on the colors to be assigned to the nodes and to the edges of $G$. For instance, if $e = [x, y]$ verifies $w(x) > q$ and $w(y) > p - q$, then $x \in S_1$ and $y \in S_2$.

We will then arrive to a situation where no solution exists (this may occur for instance if 3 consecutive nodes must get the same color or if there exists an odd cycle where color 1, 2 are forbidden for all edges) or where some nodes have obtained some fixed color and we are left with a collection of chains whose end nodes have some fixed color (but no intermediate node is colored) and with some cycles which have no fixed color; in addition some edges may have forbidden colors. Observe in particular that a 2-hypocoloring may exist even if colors 1 and 2 are forbidden for some edge $e = [x, y]$; this simply means that $x$ and $y$ must have different colors.

In the remaining cycles without fixed colors, we have two possible situations: if $C_i$ has even length, then we alternate colors 1 and 2 (in other words, we produce a coloring). If $C_i$ has odd length, then by the previous remark, we know that there exists an edge $e$ for which color 1 is feasible. So, we introduce this edge into $S_1$ and we alternate colors 1 and 2 for the remaining nodes.

Now, each remaining cycle $C_i$ has at least one node with a fixed color. We can describe $C_i$ by the sequence $(F_1, D_1, F_2, \ldots, F_k, D_k)$ where $F_i$ and $D_i$ are chains. Moreover for any $i$, all nodes of $F_i$ have a fixed color and each $D_i$ has two endpoints with a fixed color and all intermediate nodes are uncolored. Note that some $F_i$ may be empty (in this case, i.e., $F_i = \emptyset$, we identify the terminal endpoint of $D_{i-1}$ with the initial endpoint of $D_i$) and each $D_i$ has at least 3 nodes. Thus for instance, if in a cycle $C_i$, property (9.14) determines the color of exactly one node, say $a_i$, then the sequence consists of chain $D_1$ and its two endpoints are the same node $a_i$.

Let $a_1, \ldots, a_s$ be the nodes of the chain $D_1$ where $a_1$ and $a_s$ have fixed colors. If $a_1 \in S_1$, then for $e = [a_1, a_2]$ color 1 cannot be forbidden because this would force
A Hypocoloring Model for Batch Scheduling
(Preliminary version)

$a_2 \in S_2$. If $a_1 \in S_2$, then similarly for $e = [a_1, a_2]$ color 1 cannot be forbidden. So, we have $w(a_1) + w(a_2) \leq p$ for $a_1 \in S_1$ or $w(a_1) + w(a_2) \leq q$ for $a_1 \in S_2$; in addition for each intermediate node $a_i$, we have $w(a_i) \leq q$. For $a_s$ we have the same relations as for $a_1$.

**Property 9.16** Let $D_i$ with nodes $a_1, \ldots, a_s$ be a chain such that $a_1, a_s \in S_j$ with $s$ odd or such that $a_1 \in S_j$, $a_s \in S_{3-j}$ with $s$ even for some $j = 1, 2$. Then there exists a 2-hypocoloring $S = (S_1, S_2)$ where the colors 1 and 2 alternate in $D_i$.

**Proof:** This follows immediately from the observation that for all intermediate nodes $a_i$, we have $w(a_i) \leq q$. □

**Property 9.17** Let $D_i$ with nodes $a_1, \ldots, a_s$ be a chain such that $a_1, a_s \in S_j$ with $s$ even or such that $a_1 \in S_j$, $a_s \in S_{3-j}$ with $s$ odd for some $j = 1, 2$. Then there exists a 2-hypocoloring $S = (S_1, S_2)$ such that $[a_1, a_2]$ gets one of its feasible colors.

**Proof:** Assume $a_1 \in S_1$; we have observed that we have in this case $w(a_1) + w(a_2) \leq p$, so we can introduce $a_2$ into $S_1$ and we are back to the case of property (9.16).

Similarly if $a_1 \in S_2$, we have $w(a_1) + w(a_2) \leq q$, so we introduce $a_2$ into $S_2$ and we are in a similar situation. The result follows from property (9.16). □

Finally, for each chain $D_i$, we apply properties (9.16) and (9.17) and we color properly the remaining cycles. Now, when a value of $p$ is fixed, we observe that the consequence of properties (9.14) and (9.15) can be obtained in $O(n^2)$ steps and gives a feasible value of $q$ (if there exists). Then again in $O(n)$ steps, we can apply properties (9.16) and (9.17) to determine a 2-hypocoloring. It should be observed that cases where no solution can be found occur only when consequence of properties (9.14) and (9.15) are drawn.

The whole procedure will be the following:

Starting with the smallest possible value of $p$ and the smallest possible value of $q \leq p$, we apply $A(p, q)$ to get the smallest $q$ for which a solution $(S_1, S_2)$ exists; Whenever such a solution has been found, we increase $p$ to the next possible value and we start again with the minimum $q$.

An optimal hypocoloring $(S_1, S_2)$ will be given by the solution stored.

**Proposition 9.18** There exists a polynomial time algorithm in $O(n^3)$ time to determine an optimum hypocoloring in a graph $G$ with $\Delta(G) \leq 2$.

**Proof:** Let $S^* = (S^*_1, S^*_2)$ be an optimal hypocoloring of $G$ with $w(S^*_1) \geq w(S^*_2)$. Let study the situation where $p$ is equals to $w(S^*_1)$ : by construction, the consequence of properties (9.14) and (9.15) give a partial feasible solution. Then, the procedure $A(p, q)$ finds a value $q$ such that the consequences of properties (9.16) and (9.17) also yield a complete
feasible solution \( S = (S_1, S_2) \) of \( G \) with \( p = w(S_1) > q \geq w(S_2) \). We necessarily have \( w(S_2) \geq q \) since \( q \) is a minimum value such that the partial solution exists; if \( w(S_2) \neq q \), then we obtain a contradiction since \( \tilde{K}(S^*) > p + q \geq \tilde{K}(S) \); thus, we have \( w(S_2) = q \) and the proposition follows.

10 An (exponential) algorithm for triangle-free graphs

We shall now consider graphs containing no induced triangles; these are precisely the graphs \( G \) for which the largest size \( \omega(G) \) of a clique is two. In such graphs, hypostable sets consist of nodes and of edges (cliques of size 2). Such graphs can have arbitrarily large chromatic numbers; it follows that they can also have arbitrarily large hypochromatic number: indeed a triangle-free graph \( G \) with \( \chi(G) \geq 2k \) has \( \chi_h(G) \geq k \). If we had \( \chi_h(G) < k \), then we could take a minimum hypocoloring \( S = (S_1, \ldots, S_r) \) of \( G \) with \( r < k \). Each \( S_i \) could be decomposed into two stable sets \( S'_i, S''_i \) (since it consists of nodes and of edges) and we would get a \( 2r \)-coloring \( (S' = (S'_1, \ldots, S'_r, S''_1, \ldots, S''_r) \) of \( G \) with \( 2r < 2k \), which is a contradiction. So, triangle-free graphs are far from being trivial with respect to "hypochromaticity".

We shall now show that, based on the separation principle (link two nodes or merge them) described for instance in Berge, chap. 15 [3] for usual colorings of graphs, we can develop a "light version" procedure for determining a hypocoloring \( S \) with minimum cost \( \tilde{K}(S) \) in a weighted triangle-free graph. This procedure will in addition exhibit in a striking way the symmetry between usual colorings and hypocolorings.

For usual colorings, one separates the possible colorings of a graph \( G \) into two classes by repeatedly choosing a pair of non adjacent nodes \( x, y \). There is a one-to-one correspondence between the colorings of \( G \) where \( x \) and \( y \) have the same color and the colorings of the graph \( G_1 \) obtained from \( G \) by merging nodes \( x, y \) into a single node \( x' \) (linked to all neighbors of \( x \) and to all those of \( y \) in \( G \)).

In the same way there is a one-to-one correspondence between the colorings of \( G \) where \( x \) and \( y \) have different colors and the colorings of the graph \( G_2 \) obtained from \( G \) by introducing an edge \([x, y]\).

So, \( G \) can be replaced by \( G_1 \) and \( G_2 \). We repeat this operation for each one of \( G_1 \) and \( G_2 \) as long as possible, i.e., until we obtain graphs containing no more pair of non adjacent nodes, i.e., graphs which are cliques. The size of the smallest clique obtained in this way is the chromatic number of \( G \). This idea can be extended to weighted colorings of \( G \) : whenever we merge nodes \( x, y \) into \( xy \), we set \( w(xy) = \max\{w(x), w(y)\} \). The above algorithm can be applied as before. The clique with minimum weight will give the optimal cost of a weighted coloring.

Our purpose is to give an (exponential) algorithm for finding a hypocoloring \( S \) with
minimum cost $\hat{K}(S)$ in a weighted triangle-free graphs by using an additional separation principle. Since, we are handling hypostable sets (i.e., sets of nodes and edges), we will have to introduce some blocking mechanism which will prevent us from introducing some edges into hypostable sets: if $e$ is in some $S_i$, then no adjacent edge can be introduced into the same hypostable set $S_i$. The algorithm will be based on a "Contract or Connect" principle; we will call it the \textit{Contract or Connect Algorithm} or shortly COCA. It is described in table 10.

Clearly such a procedure may be exponential. It enumerates in an implicit way all hypocolorings of $G$ and finds the minimum value of the cost $\hat{K}(S)$ of such hypocolorings $S$.

Edges which are no longer allowed to be introduced into a hypostable set are \textit{blocked}. While $I I (H)$ separates the usual colorings of $H$ as described above, procedure $I I (H)$ separates the hypocolorings into two classes: the hypocolorings where $x$ and $y$ (linked) are in the same hypostable set and the hypocolorings where $x$ and $y$ are in different sets.

At the final stage, a graph $H$ will be a clique with all edges blocked; the corresponding colorings can be reconstructed by looking at the name of each node which is obtained by concatenation of the names of the nodes which have been sequentially merged. The "light" version of the COCA algorithm is just the enumeration algorithm for usual (unweighted) colorings. Notice that even if $G$ contains no triangles by assumption, the auxiliary graphs constructed by the separation procedure $I$ and $I I$ may contain triangles.

\textbf{Data :} Weighted triangle free graph $G$
\textbf{Output :} A hypocoloring $S$ with minimum cost $\hat{K}(S)$

1. \textbf{Initialisation :} All edges are free; $L=\text{list of graphs to examine}; s(G)=\text{best solution so far}; w^* = +\infty$
2. While $L \neq \emptyset$ do
   (a) Choose a graph $H$ in $L$;
   (b) If $H$ has at least one free edge then apply procedure $I(H)$, else (all edges are blocked)
      i. If $H$ is not a clique then apply procedure $I I (H)$, else ($H$ is a clique with all edges blocked)
         A. $w = \sum_{v \in V(H)} w(v)$;
         B. If $w < w^*$ then $s(G) = H$ and $w^* = w$;
3. Remove $H$ from $L$ and introduce the graphs $H_1, H_2$ (produced by separation procedure) if they exist into $L$;

\textbf{Table 1. The COCA Algorithm}
We now describe the two separation procedures:

**separation procedure I(\(H\))**
1. Choose a free edge \([x, y]\) in \(H\);
2. Let \(H_1\) be obtained from \(H\) by condensing \([x, y]\) into a node \(xy\) with \(w(xy) = w(x) + w(y)\); all edges adjacent to \(xy\) are blocked;
3. Let \(H_2\) be obtained from \(H\) by blocking \([x, y]\).

**separation procedure II(\(H\))**
1. Choose two non-adjacent nodes \(x, y\) in \(H\);
2. Let \(H_1\) be obtained from \(H\) by condensing \(x, y\) into a node \(xy\) with \(w(xy) = \max\{w(x); w(y)\}\); all edges adjacent to \(xy\) remain blocked;
3. Let \(H_2\) be obtained from \(H\) by introducing the edge \([x, y]\); \([x, y]\) is blocked;

**Remark 10.1** In the case where the hypostable sets are redefined as disjoint sets of cliques of size at most two (i.e., non-adjacent nodes and edges) then the COCA algorithm can be used to find optimal "hypocolorings" in arbitrary graphs.

An illustration of the COCA algorithm is given in Figure 8 for a small example.

Notice that the COCA algorithm could be extended to handle the case of hypostable sets containing no clique of size greater than a given \(r\), but its formulation would not be as elegant as the above one.

### 11 Some extensions and variations

In the above batch scheduling model described in section 2, we have required that each connected component of the subgraph induced by a hypostable set \(S\) be a clique (of size one or more). This may be too strong. Each connected component of the subgraph induced by \(S\) is a collection of operations assigned to the same processor; they will hence be processed consecutively. But they need not be all pairwise incompatible. Such operations could simply be required to form a connected component of the subgraph induced by \(S\) or, in other contexts, a subgraph verifying a hereditary property \(P\); we will call such colorings conditional subcolorings (or \(P\)-subcolorings). These notions are linked (in their unweighted versions) to the concept of conditional colorings of \(G\) with respect to a graph theoretical property \(P\); the conditional chromatic number \(\chi(G, P)\) is the minimum integer \(k\) such that there is a partition of the nodes into \(k\) sets such that the subgraph induced by each set has the property \(P\) (see Albertson et al. [1], Dillon [11] or Harary [15]). An important application of conditional coloring is the circuit manufacturing problem and is defined by \(P(V') = true\) iff the subgraph induced by \(V'\) is planar;
A Hypocoloring Model for Batch Scheduling
(Preliminary version)

the number $\chi(G, \text{Planar})$ is also known as the node thickness of a graph (see Beineke and White [2] and Mutzel et al. [16] for a survey).

So, we could consider more general hypostable sets and define their weights in an appropriate way. If for instance $S$ is an arbitrary subset of nodes in a weighted graph $G$ such that each connected component $C$ of $S$ verifies property $P$, let $C(S)$ be the collection of connected components $C$ in the subgraph induced by $S$. Let $f$ be a function giving the cost of connected component $C$ (if takes into account the cost of operations in $C$); so we can have $f(C) = \sum_{v \in V(C)} w(v)$ or $f(C) = \max_{v \in V(C)} w(v)$ for each connected component $C \in C(S)$ where $V(C)$ is the node set of $C$. In any case, we suppose that the function $f$ on $\mathcal{P}(V)$ verifies:

$$\forall v \in V, f(\{v\}) = w(v)$$

(4)
Then, as before, \( w(S) = \max\{f(C) : C \in \mathcal{C}(S)\} \). Hence, a \( P \)-generalized hypocoloring of the weighted graph \( G = (V, E) \) with respect to \( f \) will be a partition \( S \) of the node set \( V \) into \( k \) (disjoint) such subsets \( S_1, \ldots, S_k \); the cost \( \tilde{K}(S) \) will be defined as:

\[
\tilde{K}(S) = \sum_{i=1}^{k} w(S_i)
\]

where for each \( i, w(S_i) = \max\{f(C) : C \in \mathcal{C}(S_i)\} \).

- In this paper, we mainly refer to the case \( f(C) = \sum_{v \in V(C)} w(v) \) and \( P \) is the property to be a clique. As in the previous sections, an optimum value will be indifferently denoted by \( \tilde{K}(S) \) or \( \Pi(G, \text{Clique, sum}) \).

- In the more general situation corresponding to the case without constraints, we have \( f(C) = \sum_{v \in V(C)} w(v) \) and \( P \) is the trivial property (always equals to true). Note that if all weights are equal to one, then \( w(C) = |V(C)| \) and \( w(C) \) is the number of nodes in the largest connected component of the subgraph induced by \( C \). If each \( C \) is constrained to be a single node, then we have the classical node \( k \)-coloring concept.

- When \( f(C) = \max_{v \in V(C)} w(v) \) and \( P \) is the property to be a clique (i.e., we want to study the quantity \( \Pi(G, \text{Clique, max}) \)), we obtain another generalization of the coloring problem which has been studied under the name of subcoloring (see Brown and Corneil [8], Albertson et al. [1]). The subchromatic number of \( G \) is the smallest \( k \) for which there exists a partition of the node set into \( k \) hypostable sets (called substable sets). Basic complexity results related to subcolorings are given in Fiala et al. [12], namely proving that, if a graph \( G \) has subchromatic number equal to \( k \), the associated problem is NP-complete in the graphs with maximum degree bounded above by \( k^2 \) for any \( k \geq 2 \).

Observe finally that \( P \)-generalized hypocolorings of a weighted graph \( G \) with respect to \( f \) exist for any hereditary property \( P \). So the following quantity can be computed:

\[
\Pi(G, P, f) = \min\{\tilde{K}(S) : S \text{ is a } P\text{-hypocoloring of } G \text{ with respect to } f\}
\]

In this case, clearly we have \( \Pi(G, P, f) \leq \Pi(G, \text{max}) \) where in \( \Pi(G, \text{max}) \), the predicate is defined by: \( \forall V' \subseteq V, \ P(V') = \text{true} \ \text{iff} \ |V'| = 1 \) (the quantity \( \Pi(G, \text{max}) \) has been studied in de Werra et al. [10]) since \( P \) is a hereditary property and \( f \) verifies property (4). We also derive an upper bound on these quantities:

**Proposition 11.1** Let \( G = (V, E, w) \) be a weighted graph and \( S = (S_1, \ldots, S_k) \) a partition of node set \( V \) of \( G \). If \( V(C) \) denotes the node set of a connected subgraph \( C \) in \( G \) and \( w_{\max}(V') = \max_{v \in V'} w(v) \) for any \( V' \subseteq V \), we have for any hereditary property \( P \) and function \( f \) verifying (4):

\[
\Pi(G, P, f) \leq \sum_{i=1}^{k} w_{\max}(S_i) \max\{|V(C)| : C \in \mathcal{C}(S_i)\}
\]

(6)
A Hypocoloring Model for Batch Scheduling
(Preliminary version)

**Proof**: Assume a partition \( S = (S_1, \ldots, S_k) \) of the node set of \( G \), a hereditary property \( P \) and a function \( f \) verifying (4) such that: \( \sum_{i=1}^{k} w_{max}(S_i) \max\{|C| : C \in C(S_i)\} < \Pi(G, P, f) \leq \Pi(G, \max) \); then each \( S_i \) can be partitioned into at most \( \max\{|C| : C \in C(S_i)\} \) stable sets; since the value of each such stable set in \( S_i \) is at most \( w_{max}(S_i) \), this would give a weighted coloring of \( G \) with a value strictly smaller than \( \Pi(G, \max) \), which is impossible. \( \square \)

In fact, we could define the weighted chromatic number of a weighted graph \( G \) by:

\[
\Pi(G, \max) = \min_{S=(S_1, \ldots, S_p)} \text{partition of } V \sum_{i=1}^{p} w_{max}(S_i) \max\{|C| : C \in C(S_i)\}
\]

Indeed \( \Pi(G, \max) \) never exceeds the right hand side. Consider an optimal weighted coloring \( S^* = (S^*_1, \ldots, S^*_k) \) (i.e., with \( \Pi(G, \max) = \sum_{i=1}^{k} w(S^*_i) \)); for the corresponding partition we have \( |V(C)| = 1 \) for every connected component of every \( S^*_i \); hence, we have equality for this specific partition. Thus, when all weights are one, we get an alternative definition of the chromatic number; if we are given an arbitrary partition of the node set, it may also provide an upper bound on \( \chi(G) \).

On the other hand when we restrict to the graphs without \((k+1)\)-clique, we have for our problem the following inequality:

**Lemma 11.2** If \( G \) is \((k+1)\)-clique free then we have:

\[
\Pi(G, \text{clique, sum}) \leq \Pi(G, \max) \leq H_k \Pi(G, \text{clique, sum}) \tag{7}
\]

where \( H_k \) is the \( k \)-th harmonic number (i.e., \( H_k = \sum_{i=1}^{k} 1/i \)).

**Proof**: The first inequality has been proved. Let \( S^* = (S^*_1, \ldots, S^*_k) \) be an optimum hypocoloring of \( G \). We denote by \( C_j \) for \( j = 1, \ldots, r \) the connected components of \( S^*_i \). We split the hypostable set \( S^*_i \) into \( S^*_{1,1}, \ldots, S^*_{1,k} \) stable sets by the following procedure: \( v \in S^*_{1,1} \) if and only if there exist \( j \leq r \) such that \( v \in C_j \) and \( w(v) = \max_{x \in C_j} w(x) \). In other words, \( S^*_{1,1} \) consists of the set of elements of each connected component which have the largest weight. Then, we delete \( S^*_{1,1} \) from \( S^*_i \) and we repeat the procedure until \( S^*_i \) is empty. Then, we repeat the procedure on \( S^*_2 \) until we obtain a complete coloring. We know by construction that there exists at most \( k \) such stable sets from \( S^*_1 \) since \( G \) is \((k+1)\)-clique free (we can also have \( S^*_{i,k} = \emptyset \) for some \( i \leq r \)). We show that we have for \( i = 1, \ldots, r \) and for \( j = 1, \ldots, k \):

\[
w(S^*_{i,j}) \leq \frac{w(S^*_i)}{j} \tag{8}
\]

We only prove the result for \( i = 1 \) since it suffices to delete \( S^*_1 \) from \( G \) to obtain the result for \( i = 2 \) and so on. If \( j = 1 \), the result is trivial. Let \( j \leq k \) be an integer such that
w(S^*_1j) \neq 0 \text{ (otherwise, the result is again trivial);} denote by v_j a node of S^*_1j verifying w(v_j) = w(S^*_1j). By construction, there exists in the hypostable set S^*_1 a j-clique K_j such that \( v_j \in K_j \) and for any other node \( v \) of \( K_j \) we have \( w(v) \geq w(v_j) \). Thus, we obtain \( j w(v_j) \leq \sum_{v \in K_j} w(v) \leq w(S^*_1) \).

Finally, we sum the inequality (8) over all \( i, j \) and we obtain the expected result. \( \square \)

**Remark 11.3** This bound is best possible. Consider \( G = \bigcup_{i=1}^{k} K_i \) where \( K_i \) is a complete graph on \( i \) nodes and for any node \( v \) of \( K_i \), we set \( w(v) = k! / i \). We have exactly \( \Pi(G, \text{max}) = H_k \Pi(G, \text{clique, sum}). \)

We conclude this subsection with an inequality on the size of any optimal \( P \)-hypocoloring \( S^* \) of \( G \) with respect to criterion \( \text{max} \) (i.e., \( \hat{K}(S^*) = \Pi(G, P, \text{max}) \)); thus, in this case the cost of a \( P \)-hypostable set \( S \) is \( w(S) = \max_{x \in S} w(x) \). We will need some additional notations: we denote by \( q = \chi_P(G) \) the quantity \( \Pi(G, P, \text{max}) \) when all weights of \( G \) are equal to one. We also respectively denote by \( |S^*| \) and \( |w| \) the size of the solution \( S^* \) and the size of the weight function \( w \) (i.e., the number of different values take by \( w \)). Then, we have:

**Proposition 11.4** Let \( G = (V, E, w) \) be a weighted graph with \( w(v) > 0 \) for any node \( v \); every optimal \( P \)-hypocoloring \( S^* = (S_1, \ldots, S_k) \) of \( G \) with respect to criterion \( \text{max} \) such that \( w(S_1) \geq \ldots \geq w(S_k) \) satisfies:

\[
|S^*| \leq 1 + |w|(\chi_P(G) - 1)
\]

**Proof:** Let \( S^*(S_1, \ldots, S_k) \) be an optimal \( P \)-hypocoloring of \( G \) with respect to criterion \( \text{max} \). Assume now that \( q = \chi_P(G) \geq 2 \); choose the smallest \( i \) such that \( w(S_i) = \ldots = w(S_{i+q-1}) \geq w(S_k) \). We have \( i \leq k - q \) by assumption. Now \( S_i \cup S_{i+1} \cup \ldots \cup S_k \) generates a subgraph \( G' \) of \( G \); we have therefore \( \chi_P(G') \leq \chi_P(G) = q \) since the property is hereditary, so there exists a \( P \)-coloring \( (S'_1, \ldots, S'_{i+q-1}) \) of \( G' \) (with \( i + q - 1 < k \)). Assuming \( w(S'_i) \geq w(S'_{i+1}) \geq \ldots \geq w(S'_{i+q-1}) \) we have \( w(S'_i) = w(S_i) \) and \( w(S'_{i+s}) \leq w(S'_i) = w(S_{i+s}) \) for \( s = 1, \ldots, q - 1 \) since the criterion is \( \text{max} \). Setting \( S'_{j} = S_j \) for \( j = 1, \ldots, i - 1 \) we get an \( (i + q - 1) \)-coloring \( S' = (S'_1, \ldots, S'_{i+q-1}) \) of \( G \) with \( \hat{K}(S') < \Pi(G, P, \text{max}) \) which is a contradiction. \( \square \)

Most of the results established in previous sections can be transposed to \( \Pi(G, P, \text{sum}) \) and sometimes to \( \Pi(G, P, \text{max}) \). Namely the approximation results and the properties of the size (or the value) of optimal solutions.
12 Conclusion

Further research will be needed in several directions to explore the field of hypocolo-
rings. Batch scheduling is a motivation for this purpose, there are also several theoretical
aspects which are of interest. For instance, finding in a bipartite graph a 2-hypocoloring
with minimum cost (among those which use at most two colors) is an open problem to our
knowledge (an optimum hypocoloring in such a graph may have more than two colors) or
establishing complexity results for trees without restriction on the maximum degree.

In addition, heuristics for general graphs should be designed and tested. We hope that
the initial results derived in this paper will be a first step along this line of development.

Acknowledgments. This work was partially carried out during a visit of the first author
to the LAMSADE at Université de Paris-Dauphine in June 2002; the support of this insti-
tution is gratefully acknowledged.

Références

subchromatic number of a graph, Discrete Math. 74 (1989), 33–49.
[4] H. L. BODLAENDER, K. JANSEN, and G. J. WOEZINGER, Scheduling with incom-
[5] M. BOUDHAR and G. FINKE, Scheduling on a batch machine with job compatibil-
(1941), 194–197.
[8] J. L. BROWN and D. G. CORNEIL, On generalized graph colorings, J. Graph Theory
coloring : when stable sets are expensive (extended abstract), Proceeding W.G.2002,
L.N.C.S. (2002), –.
[10] D. DE WERRA, M. DEMANGE, J. MONNOT, and V. Th. PASCHOS, Time slot sche-


