Strong Colorings of Hypergraphs

Geir Agnarsson[1] and Magnús M. Halldórsson[2]

[1] Department of Mathematical Sciences, George Mason University, MS 3F2, 4400 University Drive, Fairfax, VA 22030
geir@math.gmu.edu

[2] Department of Computer Science, University of Iceland, Dunhaga 3, IS-107 Rvk, Iceland
mmh@hi.is

Abstract. A *strong vertex coloring* of a hypergraph assigns distinct colors to vertices that are contained in a common hyperedge. This captures many previously studied graph coloring problems. We present nearly tight upper and lower bound on approximating general hypergraphs, both offline and online. We then consider various parameters that make coloring easier, and give a unified treatment. In particular, we give an algebraic scheme using integer programming to color graphs of bounded composition-width.

Keywords: hypergraph, strong coloring, approximation, composition width.

1 Introduction

The purpose of this article is to discuss the properties of a special kind of vertex coloring of hypergraphs, where we insist on vertices that are contained in a common hyperedge receiving distinct colors. Such *strong colorings* capture a number of graph coloring problems that have been treated separately before.

A *hypergraph* $H = (V, \mathcal{E})$ consists of a finite set $V = V(H)$ of vertices and a collection $\mathcal{E} = \mathcal{E}(H) \subseteq \mathbb{P}(V)$ of subsets of V. A *strong coloring* of H is a map $\Psi : V(H) \to \mathbb{N}$ such that whenever $u, v \in e$ for some $e \in \mathcal{E}(H)$, we have that $\Psi(u) \neq \Psi(v)$. The corresponding *strong chromatic number* $\chi_s(H)$ is the least number of colors for which H has a proper strong coloring.

Strong coloring can be viewed as a regular vertex coloring problem of the *clique graph* $G_c(H)$ (also known as *2-section graph* or *representing graph* [6]) of the hypergraph H, defined on the same set of vertices, with edge set $E(G_c(H)) = \{\{u, v\} : u, v \in e \text{ for some } e \in \mathcal{E}(H)\}$. In this way, $\chi_s(H) = \chi(G_c(H))$, the ordinary chromatic number of the clique graph.

We consider both online and offline coloring algorithms. We analyze them in terms of their *competitiveness* or *approximation factor*, respectively, which in both cases is the maximum ratio between the number of colors used by the algorithm on an instance to the chromatic number of the instance. In the standard online graph coloring problem, the graph is presented one vertex at a time

G. Persiano and R. Solis-Oba (Eds.): WAOA 2004, LNCS 3351, pp. 253–266, 2005.

along with edges only to the previous vertices. Each time the algorithm receives a vertex, it must make an irrevocable decision as to its color.

Since the hypergraph H is our original input, we would like to use all the associated parameters that come with it, and use those to either obtain an optimal strong coloring or a good approximation. This is relevant because it is often difficult to deduce the hypergraph from the graph representation, or to find the "best" such hypergraph. Finding the smallest clique hypergraph H that is equivalent to a graph G, i.e. such that $G_c(H) = G$, is equivalent to finding the smallest Clique Cover (GT 17: Covering by Cliques in [13]), which is hard to approximate within $n^{2-\epsilon}$ factor, for any $\epsilon > 0$ [12].

One of the main objectives of this research is in opening a new line of research by unifying several coloring problems as strong coloring appropriate types of hypergraphs. We have gathered a host of results on these problems by modifying and sometimes slightly extending previous results on graph coloring. Finally, we have made the first step into a systematic treatment of parameters of hypergraphs and their clique graphs that make solution or approximation easier.

1.1 Instances of Strong Colorings Problems

Down-Coloring DAGs. Our original motivation to study strong hypergraph colorings stems from a digraph coloring problem that occurs when bounding storage space in genetic databases. A *down-coloring* of a DAG (acyclic digraph) $\overline{G}$ is coloring of the vertices so that vertices that share a common ancestor receive different colors. One motivation for such a coloring (see [2]) is to provide an efficient structure for querying relational tables referencing the digraph, including the retrieval of rows in a given table that are conditioned based on sets of ancestors from $\overline{G}$.

For a DAG $\overline{G}$, the binary relation $\leq$ on $V(\overline{G})$ defined by $u \leq v \Leftrightarrow u = v$, or there is a directed path from v to u in $\overline{G}$, is reflexive, antisymmetric, and transitive, and is therefore a partial order on $V(\overline{G})$. We denote by $\max\{\overline{G}\}$ the set of source vertices of $\overline{G}$, i.e., the maximal vertices with respect to this partial order $\leq$. For vertices $u, v \in V(\overline{G})$ with $u \leq v$, we say that u is a *descendant* of v. The *down-set* $D[u]$ of a vertex $u \in V(\overline{G})$ is the set of all descendants of u in $\overline{G}$, that is, $D[u] = \{x \in V(\overline{G}) : x \leq u\}$. As defined in [3], the *down-hypergraph* $H_{\overline{G}}$ of a DAG $\overline{G}$ contains the same set of vertices with the down-sets $\mathcal{E}(H_{\overline{G}}) = \{D[u] : u \in \max\{\overline{G}\}\}$ of sources of $\overline{G}$ as hyperedges. A down-coloring of a digraph corresponds to a strong coloring of the corresponding down-hypergraph.

Note that not all hypergraphs are down-hypergraphs of a DAG, but they are easily recognized: We say that a hypergraph has the *unique element property* if each hyperedge contains a vertex not contained in any other hyperedge. We can observe that a hypergraph H is a down-hypergraph of some DAG iff H has the unique element property [3].

Further properties between graphs, hypergraphs and the posets yielding them can be found in [30], where the corresponding clique graph associated with the poset is called an *upper bound graph* of the poset.

Distance-2 Coloring Graphs. A *distance-2 coloring* of a graph is a vertex coloring where vertices at distance two or less must receive different colors. This problem has received attention for applications in frequency allocation [37, 26], where two stations must use a different frequency if they are both to be able to communicate with a common neighbor. Another application given in [29] relates to the partition of the columns of a matrix for parallel solution so that columns solved in the same iteration do not share a non-zero element in the same row.

The *neighborhood hypergraph* N_G of a graph G consists of the same vertex set with a hyperedge consisting of the closed neighborhood $N[v] = \{u : u = v \text{ or } \{u, v\} \in E(G)\}$ of each vertex $v \in V(G)$. A strong coloring of N_G is equivalent to a distance-2 coloring of G.

The distance-2 coloring problem is also equivalent to an ordinary coloring problem on the *square graph* G^2 of the graph G. The k-th power G^k of a graph G is a graph on the same vertex set, with an edge between any pair of vertices of distance at most k in G. The square graph is indeed the clique graph of the neighborhood hypergraph. While it is easy to compute the power graph G^k from G, Motwani and Sudan [34] showed that it is NP-hard to compute the k-th root G of a graph G^k, for any $k \geq 2$. On the other hand, it is not hard to deduce the original graph when given its neighborhood hypergraph.

McCormick [29] was the first to show that the problem of coloring the power of a graph is NP-complete, for any fixed power. He gave a greedy algorithm with a $O(\sqrt{n})$-approximation for squares of general graphs, which was matched by the NP-hardness of an $\Omega(n^{1/2-\epsilon})$-approximation, for any $\epsilon > 0$ [4]. Several recent papers have studied distance-2 coloring planar graphs [26, 5], for which the current best upper bound is $1.66\Delta(G) + O(1)$ colors due to Molloy and Salavatipour [33].

1.2 Related Coloring Results

The best current upper bound for approximating ordinary graph coloring is $O(n(\lg\lg n/\lg n)^3)$ [15], while it is hard to approximate within $n^{1-\epsilon}$ factor, for any $\epsilon > 0$ [7]. For a survey on graph coloring approximations, see [36].

The *weak* hypergraph coloring problem is an alternative generalization of the graph coloring problem, where the vertices are to be colored so that no hyperedge is monochromatic. Several results are known about such approximations, including a $\Omega(n^{1-\epsilon})$ hardness [25].

Each color class in a strong coloring is called a *strong independent set* (strong stable set). The k-set packing problem is equivalent to the strong independent set problem in degree-k hypergraphs, by looking at the dual graph. This is NP-hard to approximate within factor $O(k/\lg k)$ [20]. This suggests, but does not guarantee, that coloring degree-k hypergraphs is hard to do within an asymptotic factor much smaller than k.

Strong coloring a hypergraph H is also equivalent to *edge coloring* the dual hypergraph H^*. Kahn [23] showed that $\chi'(H) \leq \Delta + o(\Delta)$, if no two hyperedges share many vertices. Further improvements were obtained by Molloy and Reed [32].

1.3 Overview of Paper

In the following section, we introduce the general version of the problem, which involves multicolorings, where a set of colors is to be assigned to each vertex. This follows naturally from some preprocessing of the hypergraph instance. In Section 3, we consider several parameters of graphs and hypergraphs and analyze their effect on approximability. In Section 4, we give bounds for online and offline strong coloring algorithms on general graphs. Finally, in Section 5, we present the technically most involved part of the paper, with a polynomial time coloring algorithm for the class of k-composite graphs.

2 Hypergraph Contractions and Multicolorings

To describe the strong coloring problem in its full generality, we must introduce *multicolorings*.

Multicoloring. For a simple graph G let $\nu : V(G) \rightarrow \mathbb{N}$ be a natural weight. By a *multicoloring* of (G, ν), we mean an assignment $\tilde{c} : V(G) \rightarrow \mathbb{P}(\mathbb{N})$ to the power set of $\mathbb{N}$, such that (i) $|\tilde{c}(u)| = \nu(u)$ for each $u \in V(G)$, and (ii) $\{u, v\} \in E(G) \Rightarrow \tilde{c}(u) \cap \tilde{c}(v) = \emptyset$. The corresponding *multichromatic number* $\tilde{\chi}(G, \nu)$ is then the smallest k which allows a legitimate multicoloring $\tilde{c} : V(G) \rightarrow \mathbb{P}(\{1, \ldots, k\})$.

In general, we may thus be given a *weighted hypergraph*, for which we seek a *strong multicoloring*. This corresponds to a multicoloring of the clique graph, whose weight function is identical to its corresponding hypergraph.

Contractions. One reason why it may be natural to generalize the problem to multicolorings is to handle certain contractions, or operations that simplify the instance. We consider particularly contractions that involve vertices with identical neighborhoods.

A *hypermodule* is a set S of vertices that appear identical to vertices outside S, i.e., for $u \in V \setminus S$ and $v, w \in S$, then $\{u, v\} \in E(G_c(H))$ iff $\{u, w\} \in E(G_c(H))$. A *contraction* takes a weighted hypergraph (H, ν) and a hypermodule S and produces a smaller reduced hypergraph (H', ν') where S has been replaced by a single vertex of weight $\chi(G_c(H[S]), \nu)$. One now can show that $\chi_s(H, \nu) = \chi_s(H', \nu')$.

Furthermore, degrees in the reduced hypergraph are no greater than before: for any vertex v we have $d_{H'}(v) \leq d_H(v)$. Thus, any result for approximation or time complexity involving degrees, number of vertices, or number of edges, carries over for the reduced hypergraph.

We may want to limit the kind of contractions that we seek. In particular, within our context, it is natural to search for *clique contractions*, where the clique graph $G_c(H[S])$ induced by S is a clique. In this case, degrees remain unchanged.

Note that a set of vertices that are contained in exactly the same hyperedges of H is a hypermodule that induces a clique in $G_c(H)$. By viewing each such hypermodule S of H as a single vertex u_S, and connecting two such vertices if, and only if, they are both contained in a common hyperedge of H, we obtain the *reduced graph* $G_r(H)$. This reduced graph $G_r(H)$ further has a natural weight $\nu : V(G_r(H)) \to \mathbb{N}$ given by $\nu(u_S) = |S|$. Hence, H yields a corresponding weighted reduced graph $(G_r(H), |\cdot|)$, something we will use in Section 5.

Hypergraph contraction preserves both *chordality* and *perfectness*. A polynomial time algorithm for multicoloring perfect graphs was given by Grötschel, Lovász, and Schrijver [8], under the problem name of *weighted coloring*.

3 Parameters of Graphs and Hypergraphs

The largest cardinality of a hyperedge of H will be denoted by $\sigma(H)$ and the largest cardinality of a hypermodule corresponding to a vertex of $G_r(H)$ will be denoted by $\mu(H)$. Clearly we have $\mu(H) \leq \sigma(H)$.

Maximum Degree. For a vertex $u \in V(H)$ of a hypergraph H, its *degree* $d_H(u)$ is the number of hyperedges that contain the vertex u. Note that the degree of u is usually much smaller than the number of neighbors of u (that is, the number of vertices contained in a common hyperedge with u.) The minimum (maximum) degree of a vertex in H is denoted by $\delta(H)$ ($\Delta(H)$).

Hypergraphs of degree at most t have the property that their clique graphs are $(t+1)$-claw free, i.e. contain no induced star on $(t+2)$-vertices. This ensures that almost any coloring obtains a ratio of at most t, even online.

Call an online coloring algorithm *frugal* if it does not introduce a new color unless it is forced to do so, i.e. if the corresponding vertex is already adjacent to vertices of all other colors. The First-Fit algorithm is clearly frugal. An offline algorithm is frugal if each vertex assigned a color i is adjacent to vertices of each color $1, \ldots, i-1$.

Lemma 1. *Any frugal coloring algorithm is at most $\Delta(H)$-competitive for the clique graph of a hypergraph H.*

Clique graphs of degree-2 hypergraphs contain the class of line graphs, and thus strong coloring such hypergraphs subsumes the edge coloring problem of multigraphs. This is hard to approximate within an absolute ratio of less than 4/3, but can be done using at most $1.1\chi'(G) + 0.7$ colors [35], where χ' is the edge chromatic number.

We can obtain an incomparable bound in terms of the maximum degree of the clique graph, by extending an approach of [14] to multicolorings.

Theorem 1. *Multicoloring can be approximated within $\lceil(\Delta(G)+1)/3\rceil$.*

This uses the following specialization of a lemma of Lovász [27]. It can be implemented in linear time [14] by first assigning the vertices greedily in order to the color class to which they have the fewest neighbors, followed by local improvement steps that move a vertex to a class with fewer neighbors.

Lemma 2. *Let G be a graph and let $t = \lceil(\Delta+2)/3\rceil$. There is a partition of $V(G)$ into sets $V_1, \ldots, V_t$ such that the graph $G[V_i]$ induced by each V_i is of maximum degree at most 2.*

Given the graphs of maximum degree 2 promised by the lemma, we can color each of them optimally in linear time. (Details omitted.) Thus, we obtain a t-approximation to the multicoloring problem.

For unweighted graphs, a better approximation bound of $\lceil(\Delta+1)/4\rceil$ can be obtained [14], by using that graphs of maximum degree 3 can be colored optimally by way of Brooks theorem. It, however, does not apply for multicolorings.

Inductiveness. By the *inductiveness* (or the *degeneracy*) of H, denoted by $\mathrm{ind}(H)$, we mean the parameter defined by $\mathrm{ind}(H) = \max_{S\subseteq V(H)}\{\delta(H[S])\}$. Here, $H[S]$, for a vertex subset S denotes the *subhypergraph of H induced by S*, or the hypergraph with edge set $\mathcal{E}(H[S]) = \{X\cap S : X \in \mathcal{E}(H) \text{ and } |X\cap S| \geq 2\}$. Recall that the inductiveness naturally relates to a greedy coloring of a graph G that uses at most $\mathrm{ind}(G) + 1$ colors (see [5]). The degree of a vertex v in $G_c(H)$ is at most $(\sigma(H) - 1)$ times its degree in H. Thus, the observation of [3] that $\mathrm{ind}(G_c(H)) \leq \mathrm{ind}(H)(\sigma(H) - 1)$. Hence, we have an $\mathrm{ind}(H)$-factor approximation by the greedy algorithm. For ordinary graphs, we can obtain a simple contraction in the approximation by a factor of nearly 2. We observe here that this holds also for multicolorings. The following observation is from [16].

Theorem 2. *Suppose we are given a graph G with vertex weights w and a C-coloring of G (i.e., a partition of the vertex set into independent sets). Then we can approximate the multichromatic number within a factor of $\lceil C/2\rceil$.*

Corollary 1. *Multicoloring can be approximated within $\lceil(\mathrm{ind}(G)+1)/2\rceil$.*

The inductiveness measure is useful for bounding the performance of online algorithms. Irani [21] showed that the First-Fit coloring algorithm uses at most $O(\mathrm{ind}(G)\lg n)$ to color an n-vertex graph G.

Corollary 2. *The First-Fit coloring algorithm is $O(\mathrm{ind}(H)\lg n)$-competitive for coloring the clique graph of a hypergraph H.*

Composition Width. First we define a modular decomposition [22], which is also called *substitution decomposition* as in [31], and has been studied widely, since it is by many considered one of three most important hierarchical graph decomposition, the others being tree decomposition [38] and the graph decomposition upon which clique-width is defined [11].

Definition 1. *Let G be a graph with $V(G) = \{u_1,\ldots,u_k\}$ If $G_1,\ldots,G_k$ are graphs, then let $G' = G\langle G_1,\ldots,G_k\rangle$ denote the graph obtained by replacing each vertex u_i in G by the graph G_i, and connect each vertex in G_i to each vertex in G_j if, and only if, u_i and u_j are connected in G'. In this case we say that G is a* modular decomposition *of $G_1,\ldots,G_k$. The induced subgraphs G_i of G' are called* modules *of G'.*

Definition 2. *We call a graph G'* k-composite *if it is a null graph, or recursively, if there is a graph G on $\ell \leq k$ vertices and k-composite graphs $G_1,\ldots,G_\ell$, such that $G' = G\langle G_1,\ldots,G_\ell\rangle$. The* composition-width *of G', denoted* $\mathrm{cow}(G')$, *is the least k for which G' is k-composite.*

REMARKS: (i) Every null graph is 1-composite and every clique is 2-composite. (ii) If G is k-composite, then G is k'-composite for each $k' \geq k$. (iii) Every graph on n vertices is n-composite.

Clique-Width. By a *labeled* graph G we mean a graph G provided with a labeling function $\iota : V(G) \to \mathbb{N}$. Consider the following four graph operations, introduced in [11]: (i) Create a new vertex u with a label i. (ii) Form the disjoint union of labeled graphs G_1 and G_2, denoted by $G_1 \oplus G_2$. (iii) Connect all i-labeled vertices with all the j-labeled vertices, where $i \neq j$ (iv) Relabel all vertices labeled i with the label j, where $i \neq j$. The *clique-width* of a graph G, denoted by $\mathrm{cw}(G)$ is the least number of labels needed so that G can be constructed by the above four graph operations.

Observation 3. *For a graph G we have $\mathrm{cw}(G) \leq \mathrm{cow}(G)$. If further $\mathrm{cw}(G) \in \{1,2\}$ then we have $\mathrm{cw}(G) = \mathrm{cow}(G)$.*

Clique-width was first defined and studied in [11] and it generalizes the notion of treewidth, introduced in [38], that is, a graph of bounded treewidth is necessarily also of bounded clique-width [10], but not conversely since a clique of arbitrary size has clique-width of two, while its treewidth is the number of vertices in the clique.

Problems definable in a certain variations of Monadic Second Order Logic, including maximum independent set, are solvable in polynomial time for graphs of bounded clique-width [9]. However, graph coloring has been shown to be not one of those (see [9]). Still, it has been shown that for a fixed $k \in \mathbb{N}$, the chromatic number of a graph G of clique-width of at most k, can be determined in time $O(2^{3k+1}k^2n^{2^{2k+1}+1})$ [24]. This however depends on that the expression that forms the graph using the four operations above is given.

4 Approximations for Down-Colorings

In this section we give bounds on the approximability of strong coloring general hypergraphs. The $O(\sqrt{m})$-approximation of strong independent sets of [18] leads to an equivalent approximation of strong coloring. However, we here obtain a bound on the inductiveness in terms of an arbitrary number m of edges, and obtain an approximation in terms of $\sigma(H)$, the largest hyperedge size.

Theorem 4. *For a hypergraph H with m edges, $\mathrm{ind}(G_c(H)) \leq \sqrt{m}\sigma(H)$.*

Proof. Let $k = \sqrt{m}$. Let S be a vertex subset inducing a subgraph of $G_c(H)$ of minimum degree $\mathrm{ind}(G_c(H))$. Let $H[S]$ be the subgraph of H induced by S, and let m_S be the number of hyperedges in it. If there is a vertex of degree at most k in $H[S]$, then its degree in $G_c(H[S])$ (which is at least $\mathrm{ind}(G_c(H))$ is at most $k\sigma(H)$, in which case the theorem follows. Otherwise, each vertex is of degree at least k and the number of edge-vertex incidences is at least $k|S|$. It follows that the average edge size in $H[S]$ is at least $k|S|/m_S$, and thus $\sigma(H) \geq k|S|/m_S \geq k|S|/m = |S|/k$.

Now, since the inductiveness of $G_c(H)$ is equal to the minimum degree of $G_c(H[S])$ which has at most $|S|$ vertices, we have that $\mathrm{ind}(G_c(H[S])) \leq |S|-1 < \sigma(H) \cdot k$ and the theorem follows.

Corollary 3. *There is a greedy algorithm that approximates the strong coloring of hypergraphs within a factor of $\sqrt{m}$. This yields a $\sqrt{M}$ approximation for the down-coloring of DAGs, where $M = |\max\{\overline{G}\}|$ is the number of source vertices in $\overline{G}$.*

Observe that we bounded the number of colors used by the algorithm in terms of the maximum edge size $\sigma(H)$. Thus, we have shown that the strong chromatic number of a hypergraph H differs from $\sigma(H)$ by a factor of at most $\sqrt{|\mathcal{E}(H)|} = \sqrt{m}$. In terms of down-graphs and hypergraphs, we obtain the following bound.

Corollary 4. $\chi(G_c(H_{\overline{G}})) = \chi_s(H_{\overline{G}}) \leq \sqrt{|\max\{\overline{G}\}|} \cdot \sigma(H_{\overline{G}}) \leq \sqrt{n} \cdot \sigma(H_{\overline{G}})$.

Compare the above corollary with [3, Obs. 2, p. 306], which shows that the bounds given can be obtained.

4.1 Approximation Hardness

We now give a reduction from the ordinary coloring problem that shows that the approximation of the greedy algorithm is close to best possible. The $\Omega(n^{1/2-\epsilon})$-hardness result for distance-2 coloring of [4] already yields the same hardness for strong coloring hypergraphs with n vertices and n edges. We show here similar result for down-hypergraphs and restricted types of DAGs.

Given a graph G_0, we construct a DAG $\overline{G}$ of height two by letting $V(\overline{G}) = E(G_0) \cup V(G_0)$ and $E(\overline{G}) = \{(e, v) : v \in e, v \in V(G_0), e \in E(G_0)\}$. The digraph has a source vertex for each edge in G_0, a leaf vertex for each node in G_0, and an edge from a source to a leaf if the leaf corresponds to a vertex incident on the edge corresponding to the source vertex.

Let $H = H_{\overline{G}}$ be the corresponding down-hypergraph. Note that the subhypergraph $H[V(G_0)]$ induced by the leaves is a graph and is exactly the graph G_0. The source nodes of H induce an independent set. Thus, $\chi(G_0) \leq \chi_s(H) \leq \chi(G_0)+1$. In fact, $\chi_s(H) = \chi(G_0)$ if $\chi(G_0) \geq 3$. By the results of Feige and Kilian [7], the chromatic number problem cannot be approximated within a factor of $|V(G)|^{1-\epsilon}$, for any $\epsilon > 0$, unless NP $\subseteq$ ZPP, i.e. unless there exist polynomial-time randomized algorithms for NP-hard problems. Here, we have $|V(G_0)| = \Omega(\sqrt{|V(H)|})$ and hence the following.

Theorem 5. *It is hard to approximate the down-coloring of DAGs within a factor of $n^{1/2-\epsilon}$, for any $\epsilon > 0$. This holds even for digraphs of height two.*

We may now ask if it is possible to give a better approximation for important special cases of the down-coloring problem. In particular, digraphs arising from pedigrees (i.e. records of ancestry for people) have some special properties; in particular, each vertex has in-degree at most 2, and normally a fairly small out-degree. We can show that even in this case, we cannot do better. (Proof omitted.)

Theorem 6. *It is hard to approximate the down-coloring of DAGs within a factor of $n^{1/2-\epsilon}$, for any $\epsilon > 0$, even when restricted to DAGs of in-degree and out-degree two.*

4.2 Online Coloring

In the standard online graph coloring problem, the graph is presented one vertex at a time along with edges only to the previous vertices. Each time the algorithm receives a vertex, it must make an irrevocable decision as to its color [19, 17].

The hypergraph model might lead to a different model, e.g. where all the vertices contained in edges incident on previous vertices are given. It can be inferred from the arguments below that this does not produce a great advantage.

Applying the result of Irani cited earlier, we obtain the following upper bound for online coloring.

Corollary 5. *First-Fit is $O(\sqrt{m} \lg n)$-competitive for the clique graph of a hypergraph H with m edges and n vertices. More generally, it is $O(\sqrt{m} \lg n)$-competitive for graphs that can be covered with m cliques.*

In the case of distance-2 coloring, one can argue a better bound. Namely, let G be the underlying graph and H_G be its neighborhood hypergraph. Then, $\Delta(H) = \Delta(G) + 1$ and $\sigma(H) = \Delta(G)$. Thus, the competitive ratio of any any frugal online coloring algorithm is at most $\min(\Delta(H), n/\sigma(H)) \leq \sqrt{n}$.

Proposition 1. *Any frugal online coloring algorithm is $\sqrt{n}$-competitive for the distance-2 coloring graphs.*

On the hardness side, lower bounds for online graph coloring carry over for clique graphs, simply by viewing the graphs as hypergraphs. Halldórsson and Szegedy [19] showed that for any online algorithm, there is a $\lg n$-colorable graph on n vertices for which the algorithm uses $\Omega(n/\lg n)$ colors. This holds also for randomized algorithms against an oblivious adversary. This was later extended to a *known graph* model, where a graph isomorphic to the (fixed) input graph is given in advance [17]. Thus, there is one particular graph that is hard for any online coloring algorithm. By padding this graphs with isolated vertices (or small cliques), we can have this hold for graphs of any density.

Lemma 3. *For any n and m, there is a particular graph with n vertices and at most m cliques such that* any *online coloring algorithm is at least $\Omega(\sqrt{m}/\lg^2 m)$ competitive.*

5 Multicoloring, an Algebraic Approach

In this section we will consider an algebraic approach to determine the strong chromatic number of a given hypergraph by using integer programming. We then show how the same method can recursively yield an improved poly-time algorithm to obtain an optimal coloring for k-composite graphs.

Observation 7. *For a hypergraph H and its weighted reduced graph $(G_r(H), |\cdot|)$, we have $\chi_s(H) = \tilde{\chi}(G_r(H), |\cdot|)$.*

Here we view a given hypergraph H as weighted graph $(G_r(H), |\cdot|)$, since by Observation 7 we have $\chi_s(H) = \tilde{\chi}(G_r(H), |\cdot|)$. Hence, we will here consider multicolorings of a weighted graph (G, ν), where $\nu : V(G) \to \mathbb{N}$ is a natural weight.

Consider our weighted graph (G, ν) where $V(G) = \{u_1, \ldots, u_k\}$. Let $G' = G\langle Q_1, \ldots, Q_k\rangle$ be the modular decomposition of the cliques $Q_1, \ldots, Q_k$, where each Q_i has $\nu(u_i)$ vertices. Clearly we have that $\tilde{\chi}(G, \nu) = \chi(G\langle Q_1, \ldots, Q_k\rangle)$, and so the computation of $\tilde{\chi}(G, \nu)$ can be trivially reduced to the computation of the chromatic number of a graph G'. However, taking further into the account the structure of $G\langle Q_1, \ldots, Q_k\rangle$, we can shorten the computations considerably, especially when the $\nu(u_i)$'s are large compared to k. We proceed as follows:

For each proper (i.e. nonempty) independent set $I \subseteq V(G)$ we form a variable x_I. We denote by $\mathcal{I}(G)$ the set of all independent sets of G, and for each $u \in V(G)$ we denote by $\mathcal{I}(G; u) \subseteq \mathcal{I}(G)$ the set of all independent sets of G that contain the vertex u. For each $u \in V(G)$ we form the following constraint

$$\sum_{I \in \mathcal{I}(G;u)} x_I = \nu(u). \tag{1}$$

Let us fix a listing of the elements of $\mathcal{I}(G)$: For an ordering $V(G) = \{u_1, \ldots, u_k\}$, note that each $U = \{u_{i_1}, \ldots, u_{i_\ell}\}$ yields a word $U \mapsto \mathrm{word}(U) = u_{i_1} \cdots u_{i_\ell}$, and hence the sets of $V(G)$ can be ordered lexicographically, viewing $u_1, \ldots, u_k$ as an ordered alphabet. We now can list the elements of $\mathbb{P}(V(G))$ *degree lexicographically*, or by *deglex* in short, in the following way [1]:

$$U_1 < U_2 \Leftrightarrow \begin{cases} |U_l| < |U_2| & \text{or} \\ |U_l| = |U_2| & \text{and } \mathrm{word}(U_1) < \mathrm{word}(U_2) \text{ lexicographically.} \end{cases} \tag{2}$$

With the deglex ordering (2), we can form the $|\mathcal{I}(G)|$-tuple $\mathbf{x}$ of the variables x_I, and the constraints from (1), determined by the k vertices of G, can be written collectively as $\mathsf{A}(G) \cdot \mathbf{x} = \mathbf{n}$, where $\mathbf{n} = (\nu(u_1), \ldots, \nu(u_k))$ and $\mathsf{A}(G)$ is a uniquely determined $|\mathcal{I}(G)| \times k$ matrix with only 0 or 1 as entries. Note that the sum Σ of all the variables x_I can be given by the dot-product $\Sigma = \mathbf{1} \cdot \mathbf{x}$, where $\mathbf{1}$ is the $|\mathcal{I}(G)|$-tuple with 1 in each of its entry.

Theorem 8. *For an integer weighted graph (G, ν) the multichromatic number $\tilde{\chi}(G, \nu)$ is given by the integer program*

$$\tilde{\chi}(G, \nu) = \min\{\mathbf{1} \cdot \mathbf{x} \ : \ \mathsf{A}(G) \cdot \mathbf{x} = \mathbf{n},\ \mathbf{x} \in (\mathbb{N} \cup \{0\})^{|\mathcal{I}(G)|}\}, \tag{3}$$

where $\mathsf{A}(G)$ is uniquely determined by (1) and (2), and $\mathbf{n} = (\nu(u_1), \ldots, \nu(u_k))$.

It is well-known that the problem of solving an integer programming problem as (3) is NP-complete. However, considering the complexity in terms of $n = \sum_{u \in V(G)} \nu(u)$ (which corresponds to the number of vertices in the original hypergraph H) and assuming that $k = |V(G)|$ is fixed and "small" compared to n, it is worthwhile to discuss complexity analysis.

Lemma 4. *A connected graph G on k vertices has at most 2^{k-1} proper independent sets.*

By Lemma 4 the number of variables x_I in $\mathbf{x}$ is at most 2^{k-1}. Note also that by our deglex ordering, $\mathsf{A}(G)$ is already in reduced row echelon form with its first $k \times k$ submatrix being the $k \times k$ identity matrix I_k. If $N = \max_{u \in V(G)} \nu(u)$, then clearly each optimal solution $\mathbf{x}$ must satisfy $0 \leq x_I \leq N$ for each $I \in \mathcal{I}(G)$. Hence, we have at most $2^{k-1} - k$ free variables x_I in $\mathbf{x}$, namely those x_I with $|I| \geq 2$, each taking value in $\{0, 1, \ldots, N\}$. We now make some rudimentary computational observations, which are asymptotically tight for general G:

To check whether a set of i vertices is independent or not, we need $\binom{i}{2}$ edge-comparisons. Hence, $\mathcal{I}(G)$ can be obtained by at most $\sum_{i=1}^{k} \binom{k}{i}\binom{i}{2} = k(k-1)2^{k-3}$ comparisons. Having $\mathcal{I}(G)$, to determine $\mathcal{I}(G; u)$ for each $u \in V(G)$ we need at most $k2^{k-1}$ operations. In all, determining the linear program (3) we need at most $k(k-1)2^{k-3} + k^2 2^{k-1} < k^2 2^k$ operations. Further, for each given value of $\mathbf{x}$, the expressions $\mathsf{A}(G) \cdot \mathbf{x}$ and $\mathbf{1} \cdot \mathbf{x}$ can be evaluated in at most $k2^{k-1} + 2^{k-1} < k2^k$ steps. Hence, we have the following.

Observation 9. *For a connected integer weighted graph (G, ν) with $|V(G)| = k$ and $N = \max_{u \in V(G)} \nu(u)$, an optimal $\tilde{\chi}(G, \nu)$-multicoloring can be obtained by $k2^k(k + (N+1)^{2^{k-1}-k})$ operations, or in $O(k2^k(N+1)^{2^{k-1}-k})$ time for k fixed.*

With the notation from previous Section 2 we therefore have the following for a hypergraph.

Corollary 6. *For a hypergraph H with its reduced graph $G_r(H)$ on k vertices, the complexity of obtaining an optimal strong $\chi_s(H)$-coloring is given by $O(k2^k(\mu(H)+1)^{2^{k-1}-k})$.*

Note that our complexities are polynomial expressions, only because we assume k to be fixed here.

We conclude this section by considering the complexity of obtaining an optimal coloring of a k-composite graph by using the integer program (3) recursively. The following is clear.

Proposition 2. *Let $k \in \mathbb{N}$ be given. The chromatic number $\chi(G')$ of a k-composite graph $G' = G\langle G_1, \ldots, G_\ell\rangle$ where $\ell \leq k$, is given by $\chi(G') = \tilde{\chi}(G, \nu)$, where the integer weight $\nu : V(G) \to \mathbb{N}$ is given by $\nu(u_i) = \chi(G_i)$ for each $i \in \{1, \ldots, \ell\}$.*

By Observation 9 we can obtain a $\chi(G)$-coloring by at most $k2^k(k + (N + 1)^{2^{k-1}-k})$ operations, or in $O(k2^k(N+1)^{2^{k-1}-k})$ time, where $N = \max_i \chi(G_i)$, provided that we have an optimal coloring for each of the G_i's. If that is not the case however, we proceed recursively, but with more care, since we need at this point to keep track of the actual upper bound of arithmetic operations when applying the recursion. For a rooted tree (T, r) let T_u be the subtree rooted at $u \in V(T)$.

Lemma 5. *Let (T, r) be a rooted tree with n leaves, where each internal node has at least two children. If $f : \mathbb{N} \to \mathbb{R}$ is positive and non-decreasing, then*

$$\sum_{u \in V(T)} f(|V(T_u)|) \leq S_f(n) := (n-1)f(1) + \sum_{i=1}^{n} f(i).$$

Remark: The bound of $S_f(n)$ from Lemma 5 is tight: Consider the degenerate binary tree T on $2n-1$ vertices and n leaves, where each internal vertex has a leaf as a left child. In this case we have for any f that $\sum_{u \in V(T)} f(|V(T_u)|) = (n-1)f(1) + \sum_{i=1}^{n} f(i) = S_f(n)$.

An upper bound of the number of arithmetic operations need to obtain an optimal $\chi(G')$-coloring of a k-composite graph G', can now be obtained from the weighted rooted module tree $(T_{G'}, |\cdot|)$ of G' by $\sum_{u \in V(T_{G'})} f(w(u_{G''}))$, where each vertex $u_{G''}$ corresponds to an induced subgraph (strong module) G'' of G', and $w(u_{G''}) = |V(G'')|$, and the function f from Observation 9 is $f(n) = k2^k(k + (n+1)^{2^{k-1}-k})$, since $N = \max_{1 \leq i \leq k} \chi(G_i) \leq \sum_{i=1}^{k} |V(G_i)| = n$.

Theorem 10. *Provided the modular decompositions defining the k-composite connected graph G on n vertices are given, the number of arithmetic operations needed to obtain an optimal $\chi(G)$-coloring is $k2^{2^{k-1}+1}n + 4k(n+2)^{2^{k-1}-k+1}$ and hence can be obtained in $O(kn^{2^{k-1}-k+1})$ time, for k fixed.*

As mention above, the chromatic number of a connected graph G on n vertices of clique-width at most k, can be computed in $O(2^{3k+1}n^{2^{2k+1}+1})$ time [24], provided that the corresponding k-expression for G is given. By Observation 3 that also holds for k-composite graphs as well. However, the bound given in Theorem 10 is considerably better than the mentioned bound from [24]. In addition, it is not known, for $k \geq 4$, whether there exists a polynomial time algorithm to obtain a k-expression for a graph G of clique-width at most k.

For a graph G' known to have a module decomposition, a strong modular decomposition (where the modules do not have a nonempty intersection) which is unique, can be computed in $O(n^2)$ time, since such a decomposition of a graph is a special case of a modular decomposition a 2-structure (a slightly more general concept than a graph) [22]. However, in such a strong module decomposition $G' = G\langle G_1, \ldots, G_\ell \rangle$ it could be that $\ell > k$. But, if it is known that G' is k-composite, then the union of some of the strong modules $G_1, \ldots, G_\ell$ will make a single module of G'. To check this is the same as to check for modules in G, which will take at most $O(n^2)$ time as well. Hence, the module decomposition of G' constituting a k-decomposition will take at most $O(n^2)$ time at each step, which is much less than $f(n)$ from above, right before Theorem 10. Therefore, unlike for graphs of clique-width of k or less, we have the following corollary of Theorem 10, where we do not assume the modular k-decomposition.

Corollary 7. *The complexity of obtaining an optimal coloring for a k-composite graph on n vertices is $O(kn^{2^{k-1}-k+1})$.*

Note: A few words of warning are in order. If we do not have the modular decompositions, then we do need to compute the k-decomposition at each step of the recursive definition of a k-composite graph. A priori it could look as if the constant hidden in the $O(n^2)$ term might affect the overall complexity. However, a rooted tree with n leaves and no degree-2 internal vertex has at most $2n-1$ vertices. So, the overall computation is at most $(2n-1)O(n^2) = O(n^3)$ which is dominated by the expression in Theorem 10.

Acknowledgments

The authors are grateful to Walter D. Morris Jr. for many helpful discussions regarding integer programming.

References

1. W. W. Adams, P. Loustaunau. An Introduction to Gröbner Bases. *American Mathematical Society, Graduate Studies in Mathematics*, Volume 3, (1994).
2. G. Agnarsson and Á. Egilsson. On vertex coloring simple genetic digraphs. *Congressus Numerantium*, **161**, 117 – 127, (2004).
3. G. Agnarsson, Á. Egilsson and M. M. Halldórsson. Proper down-coloring simple acyclic digraphs. AGTIVE-2003, LNCS 3062, 299 – 312, Springer Verlag, (2004).
4. G. Agnarsson, R. Greenlaw, M. M. Halldórsson. On Powers of Chordal Graphs and Their Colorings. *Congressus Numerantium*, **142–147**, 2000.
5. G. Agnarsson and M. M. Halldórsson. Coloring Powers of Planar Graphs. *SIAM Journal of Discrete Mathematics*, **16**, No. 4, 651 – 662, (2003).
6. A. Brandstädt, V. B. Le, and J. P. Spinrad. Graph Classes: A Survey. *SIAM Monographs on Discrete Mathematics and Applications*, (1999).
7. U. Feige and J. Kilian. Zero Knowledge and the Chromatic number. *Journal of Computer and System Sciences*, **57**, no 2, 187–199, (1998).
8. M. Grötschel, L. Lovász, A. Schrijver. Polynomial algorithms for perfect graphs. *North-Holland Math. Stud., 88*, North-Holland, Amsterdam, (1984).
9. B. Courcelle, J.A. Makowsky, and U. Rotics. Linear Time Solvable Optimization Problems on Graphs of Bounded Clique Width. In WG '99.
10. B. Courcelle, S. Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, **101**, 77 – 114, (2000).
11. B. Courcelle, J. Engelfriet, G. Rozenberg. Handle-rewriting hypergraph grammars. *J. Comput. System Sci.*, **46**, 218 – 270 , (1993).
12. P. Creszenci and V. Kann. A compendium of NP-optimization problems.
13. M. R. Garey and D. S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, CA, 1979.
14. M. M. Halldórsson and H.-C. Lau. Low-degree Graph Partitioning via Local Search with Applications to Constraint Satisfaction, Max Cut, and 3-Coloring. *Journal of Graph Algorithms and Applications*, **1**(3):1–13, Nov. 1997.
15. M. M. Halldórsson. A still better performance guarantee for approximate graph coloring. *Information Processing Letters*, 45:19–23, 25 January 1993.
16. M. M. Halldórsson. Approximation via Partitioning. JAIST Research Report IS-RR-95-0003F, March 1995. Available at `www.hi.is/~mmh/Publications.html`
17. M. M. Halldórsson. Online coloring known graphs. *Electronic Journal of Combinatorics*, Feb 2000. `www.combinatorics.org`

18. M. M. Halldórsson, J. Kratochvíl, and J. A. Telle. Independent sets with domination constraints. *Discrete Applied Mathematics* **99**(1-3), 39–54, 17 Dec 1999. `www.elsevier.nl/locate/jnlnr/05267`
19. M. M. Halldórsson and Mario Szegedy. Lower bounds for on-line graph coloring. *Theoretical Computer Science*, 130:163–174, August 1994.
20. E. Hazan, S. Safra, and O. Schwartz. "On the Hardness of Approximating k-Dimensional Matching". Electronic Colloquium on Computational Complexity, TR03-020, 2003.
21. S. Irani. On-Line Coloring Inductive Graphs. *Algorithmica*, **11**:53–72, 1994.
22. Ö. Johansson. Graph Decomposition Using Node Labels. *Doctoral Dissertation, Royal Institute of Technology, Stockholm*, (2001).
23. J. Kahn. Asymptotically good list-colorings. *J. Combinatorial Th. (A)*, **73**:1–59, 1996.
24. D. Kobler, U. Rotics. Edge dominating set and colorings on graphs with fixed clique-width. *Discrete Applied Mathematics*, **126**, 197 – 221, (2003).
25. M. Krivelevich and B. Sudakov. Approximate coloring of uniform hypergraphs. *ESA* '98.
26. S. O. Krumke, M. V. Marathe, and S. S. Ravi. Approximation algorithms for channel assignment in radio networks. *2nd Dial M for Mobility*, 1998.
27. L. Lovász. On Decomposition of Graphs. *Stud. Sci. Math. Hung.*, **1**:237–238, 1966.
28. M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi and D. J. Rosenkrantz. Simple Heuristics for Unit Disk Graphs. *Networks*, **25**:59–68, 1995.
29. S. T. McCormick. Optimal approximation of sparse Hessians and its equivalence to a graph coloring problem. *Math. Programming*, 26(2):153–171, 1983.
30. F. R. McMorris, T. Zaslavsky. Bound graphs of a partially ordered set. *J. Combin. Inform. System Sci.*, **7**, 134–138, (1982).
31. R. H. Möhring, F. J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization. *North-Holland Math. Stud.*, **95**, North-Holland, Amsterdam, 257 – 355, (1984).
32. M. Molloy and B. Reed. Near-optimal list colourings. *Random Structures and Algorithms*, **17**:376–402, 2000.
33. M. Molloy and M.R. Salavatipour. Frequency Channel Assignment on Planar Networks. In *ESA 2002*, LNCS 2461, pp. 736-747.
34. R. Motwani and M. Sudan. Computing roots of graphs is hard. *Discrete Appl. Math.*, 54:81–88, 1994.
35. T. Nishizeki and K. Kashiwagi. On the 1.1 edge-coloring of multigraphs. *SIAM Journal on Discrete Mathematics*, **3**(3):391–410, 1990.
36. V. Th. Paschos. Polynomial approximation and graph-coloring. *Computing*, **70**, no. 1, 41–86, (2003).
37. S. Ramanathan and E. L. Lloyd. Scheduling algorithms for multi-hop radio networks. *IEEE/ACM Trans. on Networking*, 1(2):166–172, April 1993.
38. N. Robertson, P. Seymour. Graph minors V, Excluding a planar graph. *Journal of Combinatorial Theory (B)*, **41**, 92 – 114, (1986).