Approximate Minimum Sum Colorings and Maximum k-Colorable Subgraphs of Chordal Graphs

Ian DeHaan*[1] and Zachary Friggstad**[2]

[1] Department of Combinatorics and Optimization, University of Waterloo, ijdehaan@uwaterloo.ca
[2] Department of Computing Science, University of Alberta, zacharyf@ualberta.ca

Abstract. We give a $(1.796+\epsilon)$-approximation for the minimum sum coloring problem on chordal graphs, improving over the previous 3.591-approximation by Gandhi et al. [2005]. To do so, we also design the first PTAS for the maximum k-colorable subgraph problem in chordal graphs.

1 Introduction

We consider a coloring/scheduling problem introduced by Kubicka in 1989 [14].

Definition 1. *In the* Minimum Sum Coloring *(*MSC*) problem, we are given an undirected graph $G = (V, E)$. The goal is to find a proper coloring $\phi : V \to \{1, 2, 3, \ldots\}$ of vertices with positive integers which minimizes $\sum_{v \in V} \phi(v)$. In weighted* MSC*, each vertex $v \in V$ additionally has a weight $w_v \geq 0$ and the goal is then to minimize $\sum_{v \in V} w_v \cdot \phi(v)$.*

Naturally, in saying ϕ is a proper coloring, we mean $\phi(u) \neq \phi(v)$ for any edge $uv \in E$. MSC is often used to model the scheduling of unit-length dependent jobs that utilize shared resources. Jobs that conflict for resources cannot be scheduled at the same time. The goal in MSC is then to minimize the average time it takes to complete a job.

In contrast with the standard graph coloring problem, where we are asked to minimize the number of colors used, sum coloring is NP-hard on many simple graph types. Even on bipartite and interval graphs, where there are linear time algorithms for graph coloring, MSC remains APX-hard [3,10].

In [2], it was shown that if one can compute a maximum independent set in any induced subgraph of G in polynomial time, then iteratively coloring G by greedily choosing a maximum independent set of the uncolored nodes each step yields a 4-approximation for MSC. A series of improved approximations for other graph classes followed, these are summarized in Table 1. Of particular relevance for this paper are results for perfect graphs and interval graphs. For MSC in perfect graphs, the best approximation is $\mu^\star \approx 3.591$, the solution to $\mu \ln \mu = \mu + 1$. For MSC in interval graphs, the best approximation is $\frac{\mu^\star}{2} \approx 1.796$.

* Supported by an NSERC Undergraduate Student Research Award held at the University of Alberta.
** Supported by an NSERC Discovery Grant and Accelerator Supplement.

Table 1. Known results for sum coloring. The $O^\star$-notation hides poly(log log n) factors. Our work appears in bold.

	u.b.	l.b.
General graphs	$O^\star(n/\log^3 n)$ [2,6]	$O(n^{1-\epsilon})$ [2,7]
Perfect graphs	3.591 [8]	APX-HARD [3]
Chordal graphs	$\mathbf{1.796} + \epsilon$	APX-HARD [10]
Interval graphs	1.796 [11]	APX-HARD [10]
Bipartite graphs	27/26 [15]	APX-HARD [3]
Planar graphs	PTAS [12]	NP-HARD [12]
Line graphs	1.8298 [13]	APX-HARD [16]

In this paper, we study MSC in chordal graphs. A graph is chordal if it does not contain a cycle of length at least 4 as an induced subgraph. Equivalently, every cycle of length at least 4 has a chord - an edge connecting two non-consecutive nodes on the cycle. Chordal graphs form a subclass of perfect graphs, so we can color them optimally in polynomial time. But MSC itself remains APX-HARD in chordal graphs [10], as they generalize interval graphs. The class of chordal graphs is well studied; linear-time algorithms have been designed to recognize them, to compute maximum independent sets, and to find minimum colorings, among other things. A comprehensive summary of many famous results pertaining to chordal graphs can be found in the excellent book by Golumbic [9]. Chordal graphs also appear often in practice; for example Pereira and Palsberg study register allocation problems (which can be viewed as a sort of graph coloring problem) and observe that the interference graphs for about 95% of the methods in the Java 1.5 library are chordal when compiled with a particular compiler [17].

Our main result is an improved approximation algorithm for MSC in chordal graphs.

Theorem 1. *For any constant $\epsilon > 0$, there is a polynomial-time $\frac{\mu^\star}{2} + \epsilon \approx 1.796 + \epsilon$ approximation for weighted* MSC *on chordal graphs.*

That is, we can approximate MSC in chordal graphs essentially within the same guarantee as for interval graphs. Prior to our work, the best approximation in chordal graphs was the same as in perfect graphs: a 3.591-approximation by Gandhi et al. [8].

To attain this, we study yet another variant of the coloring problem.

Definition 2. *In the weighted* Maximum k-Colorable Subgraph *(*MkCS*) problem, we are given a graph $G = (V, E)$, vertex weights $w_v \geq 0$, and a positive integer k. The goal is to find a maximum-weight subset of nodes $S \subseteq V$ such that the induced subgraph $G[S]$ is k-colorable.*

We also design a polynomial-time approximation scheme (PTAS) for weighted MkCS in chordal graphs.

Theorem 2. *For any $\epsilon > 0$, there is a $(1-\epsilon)$-approximation for weighted* MkCS *in chordal graphs.*

Prior to our work, the best approximation recorded in literature was a 1/2-approximation by Chakaravarthy and Roy [4]. Although one could also get a $(1-1/e)$-approximation by greedily finding and removing a maximum-weight independent set of nodes for k iterations, i.e., the maximum coverage algorithm.

Organization
We begin with a high-level discussion of our techniques. Then, Section 2 presents the proof of Theorem 1 assuming one has a PTAS for MkCS in chordal graphs. Theorem 2 is proven in Section 3.

1.1 Our Techniques

Our work is inspired by the 1.796-approximation for MSC in interval graphs by Halldórsson, Kortsarz, and Shachnai [11]. They show that if one has an exact algorithm for MkCS, then by applying it to values of k from a carefully selected geometric sequence and "concatenating" these colorings, one gets a 1.796-approximation. In interval graphs, MkCS can be solved in polynomial time using a greedy algorithm. We show that a similar result holds: we show Theorem 1 holds in any family of graphs that admit a PTAS for MkCS and a polynomial time algorithm for the standard minimum coloring problem. However, we need to use linear programming techniques instead of a greedy algorithm since their approach seems to heavily rely on getting *exact* algorithms for MkCS.

MkCS in Chordal Graphs
In chordal graphs, MkCS is NP-COMPLETE, but it can be solved in $n^{O(k)}$ time [19]. We rely on this algorithm for constant values of k, so we briefly summarize how it works to give the reader a complete picture of our PTAS.

Their algorithm starts with the fact that chordal graphs have the following representation. For each chordal graph $G = (V, E)$ there is a tree T with $O(n)$ nodes of maximum degree 3 plus a collection of subtrees $\mathcal{T} = \{T_v : v \in V\}$, one for each $v \in V$. These subtrees satisfy the condition that $uv \in E$ if and only if subtrees T_u and T_v have at least one node in common. For a subset $S \subseteq V$, we have $G[S]$ is k-colorable if and only if each node in T lies in at most k subtrees from $\{T_v : v \in S\}$. The tree T and subtrees $\mathcal{T}$ are computed in polynomial time and then a straightforward dynamic programming procedure is used to find the maximum k-colorable subgraph. The states of the DP algorithm are characterized by a node a of T and subtrees $\mathcal{S} \subseteq \mathcal{T}$ with $|\mathcal{S}| \leq k$ where each subtree in $\mathcal{S}$ includes a.

Our contribution is an approximation for large values of k. It is known that a graph G is chordal if and only if its vertices can be ordered as $v_1, v_2, \ldots, v_n$ such that for every $1 \leq i \leq n$, the set $N^{left}(v_i) := \{v_j : v_i v_j \in E \text{ and } j < i\}$ is a clique. Such an ordering is called a **perfect elimination ordering**. We consider the following LP relaxation based on a perfect elimination ordering. We have a variable x_v for every $v \in V$ indicating if we should include v in the subgraph.

$$\textbf{maximize} \left\{ \sum_{v \in V} w_v \cdot x_v : x_v + x(N^{left}(v)) \leq k \; \forall \; v \in V, x \in [0,1]^V \right\}.$$

The natural $\{0,1\}$ solution corresponding to a k-colorable induced subgraph $G[S]$ is feasible, so the optimum LP solution has value at least the size of the largest k-colorable subgraph of G.

We give an LP-rounding algorithm with the following guarantee.

Lemma 1. *Let x be a feasible LP solution. In $n^{O(1)}$ time, we can find a subset $S \subseteq V$ such that $G[S]$ is k-colorable and $\sum_{v \in S} w_v \geq \left(1 - \frac{2}{k^{1/3}}\right) \cdot \sum_{v \in V} w_v \cdot x_v$.*

Theorem 2 then follows easily. If $k \leq 8/\epsilon^3$, we use the algorithm from [19] which runs in polynomial time since k is bounded by a constant. Otherwise, we run our LP rounding procedure.

Linear Programming Techniques for MSC

We give a general framework for turning approximations for weighted MkCS into approximations for MSC.

Definition 3. *We say that an algorithm for weighted* MkCS *is a (ρ, γ) approximation if it always returns a $\gamma \cdot k$ colorable subgraph with vertex weight at least $\rho \cdot OPT$, where OPT is the maximum vertex weight of any k-colorable subgraph.*

For Theorem 1, we only need to consider the case $\rho = 1 - \epsilon$ and $\gamma = 1$. Still, we consider this more general concept since it is not any harder to describe and may have other applications.

We prove the following, where e denotes the base of the natural logarithm.

Lemma 2. *Suppose there is a (ρ, γ) approximation for weighted* MkCS *on graphs in a class of a graphs where minimum colorings can be found in polynomial time. Then, for any $1 < c < \min(e^2, \frac{1}{1-\rho})$, there is a $\frac{\rho \cdot \gamma \cdot (c+1)}{2 \cdot (1-(1-\rho) \cdot c) \cdot \ln c}$-approximation for* MSC *for graphs in the same graph class.*

Our main result follows by taking $\gamma = 1$ and $\rho = 1 - \epsilon$. For small enough ϵ, we then choose $c^* \approx 3.591$ to minimize the expression, resulting in an approximation guarantee of at most 1.796.

Roughly speaking, we prove Lemma 2 by considering a time-indexed configuration LP relaxation for latency-style problems. Configuration LPs have been considered for MSC in other graph classes, such as line graphs [13]. The configurations used in previous work have variables for each independent set. We use a stronger LP that has variables for each k-colorable subgraph for each $1 \leq k \leq n$.

Our configuration LP was inspired by one introduced by Chakrabarty and Swamy for the Minimum Latency Problem (a variant of the Travelling Salesperson Problem) [5], but is tailored for our setting. For each "time"

$k \geq 1$ we have a family of variables, one for each k-colorable subgraph, indicating if this is the set of nodes that should be colored with integers $\leq k$. This LP can be solved approximately using the (ρ, γ)-approximation for MkCS, and it can be rounded in a manner inspired by [5,18].

Note that Theorem 2 describes a $(1-\epsilon, 1)$-approximation for MkCS for any constant $\epsilon > 0$. If we had a $(1, 1+\epsilon)$-approximation then the techniques in [11] could be easily adapted to prove Theorem 1. But these techniques don't seem to apply when given MkCS approximations that are inexact on the number of nodes included in the solution.

2 An LP-Based Approximation Algorithm for MSC

As mentioned earlier, our approach is inspired by a time-indexed LP relaxation for latency problems introduced by Chakrabarty and Swamy [5]. Our analysis follows ideas presented by Post and Swamy who, among other things, give a 3.591-approximation for the Minimum Latency Problem [18] using a configuration LP.

2.1 The Configuration LP

For a value $k \geq 0$ (perhaps non-integer), $\mathcal{C}_k$ denotes the vertex subsets $S \subseteq V$ such that $G[S]$ can be colored using at most k colors. For integers $1 \leq k \leq n$ and each $C \in \mathcal{C}_k$, we introduce a variable $z_{C,k}$ that indicates if C is the set of nodes colored with the first k integers. We also use variables $x_{v,k}$ to indicate vertex v should receive color k. We only need to consider n different colors since no color will be "skipped" in an optimal solution.

$$\textbf{minimize:} \quad \sum_{v \in V} \sum_{k=1}^{n} w_v \cdot k \cdot x_{v,k} \qquad (\textbf{LP-MSC})$$

$$\begin{aligned}
\textbf{subject to:} \quad \sum_{k=1}^{n} x_{v,k} &= 1 && \forall\, v \in V && (1)\\
\sum_{C \in \mathcal{C}_k} z_{C,k} &\leq 1 && \forall\, 1 \leq k \leq n && (2)\\
\sum_{C \in \mathcal{C}_k : v \in C} z_{C,k} &\geq \sum_{k' \leq k} x_{v,k'} && \forall\, v \in V, 1 \leq k \leq n && (3)\\
x, z &\geq 0
\end{aligned}$$

Constraint (1) says each vertex should receive one color, constraint (2) ensures we pick just one subset of vertices to use the first k colors on, and constraint (3) enforces that each vertex colored by a value less than or equal to k must be in the set we use the first k colors on.

Recall that this work is not the first time a configuration LP has been used for MSC. In [13], the authors consider one that has a variable $x_{C,k}$ for every independent set C, where the variable models that C is the independent set used at time t. Our approach allows us to prove better bounds via LP rounding, but it has the stronger requirement that in order to (approximately) solve our LP, one needs to (approximately) solve the MkCS problem, rather than just the maximum independent set problem.

Let OPT denote the optimal MSC cost of the given graph and OPT_{LP} denote the optimal cost of (**LP-MSC**). Then $OPT_{LP} \leq OPT$ simply because the natural $\{0,1\}$ solution corresponding to OPT is feasible for this LP.

At a high level, we give a method to solve this LP approximately by using the approximation for MkCS to approximately separate the constraints of the dual LP, which is given as follows.

$$\textbf{maximize:} \quad \sum_{v \in V} \alpha_v - \sum_{k=1}^{n} \beta_k \qquad (\textbf{DUAL-MSC})$$

$$\textbf{subject to:} \quad \alpha_v \leq w_v \cdot k + \sum_{\hat{k}=k}^{n} \theta_{v,\hat{k}} \quad \forall\, v \in V, 1 \leq k \leq n \tag{4}$$

$$\sum_{v \in C} \theta_{v,k} \leq \beta_k \quad \forall\, 1 \leq k \leq n, C \in \mathcal{C}_k \tag{5}$$

$$\beta, \theta \geq 0 \tag{6}$$

Note (**DUAL-MSC**) has polynomially-many variables. We approximately separate the constraints in the following way. For values $\nu \geq 0, \rho \leq 1, \gamma \geq 1$, let $\mathcal{D}(\nu; \rho, \gamma)$ denote the following polytope:

$$\left\{ (\alpha, \beta, \theta) : (4), (6), \sum_{v \in C} \theta_{v,k} \leq \beta_k \ \forall\, 1 \leq k \leq n\ \forall\, C \in \mathcal{C}_{\gamma \cdot k}, \sum_{v} \alpha_v - \frac{1}{\rho} \cdot \sum_{k} \beta_k \geq \nu \right\}$$

Lemma 3. *If there is a (ρ, γ)-approximation for* MkCS*, there is also a polynomial-time algorithm $\mathcal{A}$ that takes a single value ν plus values (α, β, θ) for the variables of* (**DUAL-MSC**) *and always returns one of two things:*

- *A (correct) declaration that $(\alpha, \beta/\rho, \theta) \in \mathcal{D}(\nu; 1, 1)$.*
- *A hyperplane separating (α, β, θ) from $\mathcal{D}(\nu; \rho, \gamma)$.*

Proof. First, check that (4), (6), and $\sum_v \alpha_v - \frac{1}{\rho} \cdot \sum_k \beta_k \geq \nu$ hold. If not, the violated constraint gives a hyperplane separating (α, β, θ) from $\mathcal{D}(\nu; \rho, \gamma)$. Then, for each k, we run the MkCS (ρ, γ)-approximation on the instance with vertex weights $\theta_{v,k}, v \in V$. If this finds a solution with weight exceeding β_k, we return the corresponding constraint as a separating hyperplane. Otherwise, we know that the maximum possible weight of a k-colorable subgraph is at most β_k/ρ. If the latter holds for all k, then $(\alpha, \beta/\rho, \theta) \in \mathcal{D}(\nu; 1, 1)$.

Lemma 3.3 from [5] takes such a routine and turns it into an approximate LP solver. The following is proven in the exact same manner where we let $\mathbf{LP}^{(\rho,\gamma)}$ be the same as (**LP-MSC**), except $\mathcal{C}_k$ is replaced by $\mathcal{C}_{\gamma \cdot k}$ in both (2) and (3) and the right-hand side of (2) is replaced by $1/\rho$. For the sake of space, we provide only a quick overview of the proof below the statement.

Lemma 4. *Given a* (ρ, γ)*-approximation for* MkCS*, we can find a feasible solution* (x, z) *to* $\mathbf{LP}^{(\rho,\gamma)}$ *with cost at most* OPT_{LP} *in polynomial time.*

As a reminder, here OPT_{LP} is the optimum value of (**LP-MSC**) itself.

Roughly speaking, Lemma 4 is obtained as follows. First, for any ν, we can run the ellipsoid method using the approximate separation oracle. It will either generate enough constraints to certify $\mathcal{D}(\nu; \rho, \gamma) = \emptyset$ or it will eventually find a point $(\alpha, \beta, \theta) \in \mathcal{D}(\nu; 1, 1)$. A binary search can be used to find the largest ν for which $\mathcal{D}(\nu; \rho, \gamma)$ is not certified to be empty. The constraints produced by the ellipsoid method can be used to determine a polynomial-size set of variables that need to be considered in $\mathbf{LP}^{(\rho,\gamma)}$ to get a solution with value $\leq OPT$. See [5] for details.

2.2 The Rounding Algorithm and Analysis

The rounding algorithm is much like that in [11] in that it samples k-colorable subgraphs for various values of k in a geometric sequence and concatenates these colorings to get a coloring of all nodes. For convenience, let $z_{C,k} = z_{C,\lfloor k \rfloor}$ for any real value $k \geq 0$.

Algorithm 1 MSCRound(G)

find a solution (x, z) to $\mathbf{LP}^{(\rho,\gamma)}$ with value $\leq OPT$ using Lemma 4
if necessary, increase $z_{\emptyset,k}$ until $\sum_{C \in \mathcal{C}_{\gamma \cdot k}} z_{C,k} = 1/\rho$ for each k
let $1 < c < \min(e^2, 1/(1-\rho))$ be a constant we will optimize later
let $h = c^{\Gamma}$ be a random offset where Γ is sampled uniformly from $[0, 1)$
$j \leftarrow 0$
$k \leftarrow 0$ ▷ The next color to use
while $G \neq \emptyset$ **do**
  $k_j \leftarrow h \cdot c^j$
  $k'_j \leftarrow \min\{n, \lfloor k_j \rfloor\}$
  choose C randomly from $\mathcal{C}_{\gamma \cdot k'_j}$ with probability according to the LP values $z_{C',k'_j} \cdot \rho$ for $C' \in \mathcal{C}_{\gamma \cdot k'_j}$
  color C with $\lfloor \gamma \cdot k'_j \rfloor$ colors, call the color classes $C_1, C_2, \ldots, C_{\lfloor \gamma \cdot k'_j \rfloor}$
  randomly permute the color classes, let $C'_1, C'_2, \ldots, C'_{\lfloor \gamma \cdot k'_j \rfloor}$ be the reordering
  finally, assign nodes in C'_i color $k + i$ for each $1 \leq i \leq \lfloor \gamma \cdot k'_j \rfloor$ in the final solution
  $k \leftarrow k + \lfloor \gamma \cdot k'_j \rfloor$
  $G \leftarrow G - C$
  $j \leftarrow j + 1$
end while

Note that nodes colored during iteration j get assigned colors at most $\gamma \cdot (k_0 + k_1 + \ldots + k_j)$ and the expected color of such a node is at most $\gamma \cdot (k_0 +$

$k_1 + \ldots + k_{j-1} + (k_j + 1)/2$). The number of iterations is $O(\log n)$ because each vertex will appear in each $\gamma \cdot n$ coloring, as this is the largest color considered in $\mathbf{LP}^{(\rho,\gamma)}$.

We note that despite our approach following the main ideas of the algorithm and analysis for minimum latency given in [18], there are some key details that change. In [18], each iteration of the algorithm produces a tree, which is then doubled and shortcutted to produce a cycle with cost at most double the tree. While we randomly permute the colors in our coloring, they randomly choose which direction to walk along the cycle. For a tree with cost k, this gives an expected distance of k for each node. We save a factor of 2 because we do not have a doubling step, but our average color is $\frac{k+1}{2}$ as opposed to $\frac{k}{2}$. Some extra work is required in our analysis to account for the extra $\frac{1}{2}$ on each vertex.

Let $p_{v,j}$ be the probability that vertex v is not colored by the end of iteration j. For $j < 0$, we use $p_{v,j} = 1$ and $k_j = 0$. Finally, for $v \in V$, let $\phi(v)$ denote the color assigned to v in the algorithm.

The following is essentially Claim 5.2 in [18], with some changes based on the differences in our setting as outlined above.

Lemma 5. *For a vertex v,*

$$\mathbf{E}[\phi(v)|h] \le \frac{\gamma}{2} \cdot \frac{c+1}{c-1} \cdot \sum_{j \ge 0} p_{v,j-1} \cdot (k_j - k_{j-1}) + \gamma(\frac{1}{2} - \frac{h}{c-1})$$

Proof. There are at most $\gamma \cdot k_j$ colors introduced in iteration j. They are permuted randomly, so any vertex colored in iteration j has color, in expectation, at most $\gamma \cdot (k_j + 1)/2$ more than all colors used in previous iterations. That is, the expected color of v if colored in iteration j is at most

$$\gamma\left(k_0 + k_1 + \ldots + k_{j-1} + \frac{k_j + 1}{2}\right) \le \gamma\left(h \cdot \left(\frac{c^j - 1}{c-1} + \frac{c^j}{2}\right) + \frac{1}{2}\right)$$
$$= \gamma\left(\frac{k_j}{2} \cdot \frac{c+1}{c-1} + \frac{1}{2} - \frac{h}{c-1}\right),$$

where we have used $k_i = h \cdot c^i$ and summed a geometric sequence.

The probability v is colored in iteration j is $p_{v,j-1} - p_{v,j}$, so the expected color of v is bounded by

$$\frac{\gamma}{2} \cdot \frac{c+1}{c-1} \cdot \left(\sum_{j \ge 0} (p_{v,j-1} - p_{v,j}) \cdot k_j\right) + \gamma\left(\frac{1}{2} - \frac{h}{c-1}\right).$$

By rearranging, this is what we wanted to show.

For brevity, let $y_{v,j} = \sum_{k \le k_j} x_{v,k}$ denote the LP coverage for v up to color k_j. The next lemma is essentially Claim 5.3 from [18], but the dependence on ρ is better in our context[3].

[3] We note [18] does have a similar calculation in a single-vehicle setting of their problem whose dependence is more like that in Lemma 6. They just don't have a specific claim summarizing this calculation that we can reference.

Lemma 6. *For any $v \in V$ and $j \geq 0$, we have $p_{v,j} \leq (1-y_{v,j})\cdot\rho+(1-\rho)\cdot p_{v,j-1}$.*

Proof. If v is not covered by iteration j, then it is not covered in iteration j itself and it is not covered by iteration $j-1$, which happens with probability

$$\begin{aligned} p_{v,j-1}\cdot\left(1-\sum_{C\in\mathcal{C}_{\gamma\cdot k_j}:v\in C}\rho\cdot z_{C,k_j}\right) &\leq p_{v,j-1}\cdot(1-\rho\cdot y_{v,j}) \\ &= p_{v,j-1}\cdot\rho\cdot(1-y_{v,j})+p_{v,j-1}\cdot(1-\rho). \end{aligned}$$

Note that the first inequality follows from constraint (3) and the definition of $y_{v,j}$. The lemma then follows by using $p_{v,j-1}\leq 1$ and $y_{v,j}\leq 1$ to justify dropping $p_{v,j-1}$ from the first term.

From these lemmas, we can complete our analysis. Here, for $v\in V$, we let $col_v = \sum_{k=1}^{n} k\cdot x_{v,k}$ denote the fractional color of v, so the cost of (x,z) is $\sum_{v\in V} w_v\cdot col_v$. The following lemma is essentially Lemma 5.4 in [18] but with our specific calculations from the previous lemmas.

Lemma 7. *For any $v\in V$, we have $\mathbf{E}[\phi(v)] \leq \frac{\rho\cdot\gamma\cdot(c+1)}{2\cdot(1-(1-\rho)\cdot c)\cdot\ln c}\cdot col_v$.*

Proof. For brevity, let $\Delta_j = k_j - k_{j-1}$. We first consider a fixed offset h. Let $A=\sum_{j\geq 0} p_{v,j-1}\cdot\Delta_j$ and recall, by Lemma 5, that the expected color of v for a given h is at most $\frac{\gamma}{2}\cdot\frac{c+1}{c-1}\cdot A+\gamma(\frac{1}{2}-\frac{h}{c-1})$.

Note $\Delta_j = c\cdot\Delta_{j-1}$ for $j\geq 2$ and $\Delta_0+\Delta_1 = c\cdot\Delta_0$. So from Lemma 6,

$$\begin{aligned} A &\leq \sum_{j\geq 0}\rho\cdot(1-y_{v,j})\cdot\Delta_j+(1-\rho)\sum_{j\geq 0}p_{v,j-2}\cdot\Delta_j \\ &= \sum_{j\geq 0}\rho\cdot(1-y_{v,j})\cdot\Delta_j + c\cdot(1-\rho)\cdot A. \end{aligned}$$

Rearranging and using $c<1/(1-\rho)$, we have that

$$A \leq \frac{\rho}{1-c\cdot(1-\rho)}\cdot\sum_{j\geq 0}(1-y_{v,j})\cdot\Delta_j.$$

For $1\leq k\leq n$, let $\sigma(k)$ be k_j for the smallest integer j such that $k_j\geq k$. Simple manipulation and recalling $y_{v,j}=\sum_{k\leq k_j} x_{v,j}$ shows $\sum_{j\geq 0}(1-y_{v,j})\cdot\Delta_j = \sum_{k=1}^{n}\sigma(k)\cdot x_{v,k}$.

The expected value of $\sigma(k)$ over the random choice of h, which is really over the random choice of $\Gamma\in[0,1)$, can be directly calculated as follows where j is the integer such that $k\in[c^j,c^{j+1})$.

$$\begin{aligned} \mathbf{E}_h[\sigma(k)] &= \int_0^{\log_c k-j} c^{\Gamma+j+1}d\Gamma + \int_{\log_c k-j}^{1} c^{\Gamma+j}d\Gamma \\ &= \frac{1}{\ln c}(c^{\log_c k+1}-c^{j+1}+c^{j+1}-c^{\log_c k}) = \frac{c-1}{\ln c}\cdot k. \end{aligned}$$

We have just shown $\mathbf{E}_h[\sum_{j\geq 0}(1-y_{v,j})\cdot \Delta_j] = \frac{c-1}{\ln c}\sum_{k\geq 0} k\cdot x_{v,k} = \frac{c-1}{\ln c}\cdot col_v$. So, we can now bound the unconditional color $\mathbf{E}_h[\phi(v)]$ using our previous lemmas.

$$\begin{aligned}
\mathbf{E}_h[\phi(v)] &= \frac{\gamma}{2}\cdot\frac{c+1}{c-1}\cdot\mathbf{E}_h[A] + \gamma(\frac{1}{2} - \mathbf{E}_h[h]/(c-1)) \\
&\leq \frac{\rho\cdot\gamma\cdot(c+1)}{2\cdot(1-(1-\rho)\cdot c)\cdot\ln c}\cdot col_v + \gamma\left(\frac{1}{2} - \mathbf{E}_h[h]/(c-1)\right) \\
&= \frac{\rho\cdot\gamma\cdot(c+1)}{2\cdot(1-(1-\rho)\cdot c)\cdot\ln c}\cdot col_v + \gamma\left(\frac{1}{2} - \frac{1}{\ln c}\right) \\
&\leq \frac{\rho\cdot\gamma\cdot(c+1)}{2\cdot(1-(1-\rho)\cdot c)\cdot\ln c}\cdot col_v
\end{aligned}$$

The first equality and inequality follow from linearity of expectation and known bounds on $\mathbf{E}[\phi(v)|h]$ and A. The second equality follows from the fact that $\mathbf{E}_h[h] = \int_0^1 c^{\Gamma} d\Gamma = \frac{c-1}{\ln c}$, and the last inequality is due to the fact that $c < e^2$ by assumption.

To finish the proof of Lemma 2, observe the expected vertex-weighted sum of colors of all nodes is then at most

$$\frac{\rho\cdot\gamma\cdot(c+1)}{2\cdot(1-(1-\rho)\cdot c)\cdot\ln c}\cdot\sum_{v\in V} w_v\cdot col_v \leq \frac{\rho\cdot\gamma\cdot(c+1)}{2\cdot(1-(1-\rho)\cdot c)\cdot\ln c}\cdot OPT.$$

Theorem 1 then follows by combing the $(1-\epsilon, 1)$ MkCS approximation (described in the next section) with this MSC approximation, choosing $c \approx 3.591$, and ensuring ϵ is small enough so $c < 1/\epsilon$.

We note Algorithm 1 can be efficiently derandomized. First, there are only polynomially-many offsets of h that need to be tried. That is, for each k_j, we can determine the values of h that would cause $\lfloor\gamma\cdot k_j\rfloor$ to change and try all such h over all j. Second, instead of randomly permuting the color classes in a $\gamma\cdot k_j$-coloring, we can order them greedily in non-increasing order of total vertex weight.

3 A PTAS for MkCS in Chordal Graphs

We first find a perfect elimination ordering of the vertices $v_1, v_2, \ldots, v_n$. This can be done in linear time, e.g., using lexicographical breadth-first search [9]. Let $N^{left}(v) \subseteq V$ be the set of neighbors of v that come before v in the ordering, so $N^{left}(v)\cup\{v\}$ is a clique. Recall that we are working with the following LP. The constraints we use exploit the fact that a chordal graph is k-colorable if and only if all left neighbourhoods of its nodes in a perfect elimination ordering have size at most $k-1$.

$$\begin{aligned} \text{maximize:} & \quad \sum_{v \in V} w_v \cdot x_v & & \textbf{(K-COLOR-LP)} \\ \text{subject to:} & \quad x_v + x(N^{left}(v)) \leq k & \forall\, v \in V & \quad (7) \\ & \quad x \in [0,1]^V & & \end{aligned}$$

Let OPT_{LP} denote the optimal LP value and OPT denote the optimal solution to the problem instance. Of course, $OPT_{LP} \geq OPT$ since the natural $\{0,1\}$ integer solution corresponding to a k-colorable subgraph of G is a feasible solution.

We can now give a rounding algorithm as follows.

Algorithm 2 MCSRound(G, k)

let $0 \leq f(k) \leq 1$ be a value we will optimize later
find a perfect elimination ordering $v_1, v_2, \ldots, v_n$ for G
let x be an optimal feasible solution to (**K-COLOR-LP**)
form S' by adding each $v \in V$ to S' independently with probability $(1 - f(k)) \cdot x_v$.
$S \leftarrow \emptyset$
for $v \in \{v_1, v_2, \ldots, v_n\}$ **do**
  if $v \in S'$ and $S \cup \{v\}$ is k-colorable, add v to S
end for
return S

3.1 Analysis

Observe that when we consider adding some $v \in S'$ to S, $S \cup \{v\}$ is k-colorable if and only if $|S \cap N^{left}(v)| \leq k - 1$. This is easy to prove by noting that the restriction of a perfect elimination ordering of G to a subset S yields a perfect elimination ordering of $G[S]$. Because we consider the nodes v according to a perfect elimination ordering of G, by adding v the only possible left-neighbourhood of a node that could have size $\geq k$ is $N^{left}(v)$ itself.

We bound the probability that we select at least k vertices from $N^{left}(v)$. The second moment method is used so that derandomization is easy. Let Y_u indicate the event that $u \in S'$. Then $\mathbf{E}[Y_u^2] = \mathbf{E}[Y_u] = (1 - f(k)) \cdot x_u$. Fix some vertex v. Let $Y = \sum_{u \in N^{left}(v)} Y_u$. By constraint (7), we have

$$\mathbf{E}[Y] = \sum_{u \in N^{left}(v)} (1 - f(k)) \cdot x_u \leq (1 - f(k)) \cdot k.$$

And since each Y_u is independent, we have again by constraint (7) that

$$\mathbf{Var}[Y] = \sum_{u \in N^{left}(v)} \mathbf{Var}[Y_u] = \sum_{u \in N^{left}(v)} \left(\mathbf{E}[Y_u^2] - \mathbf{E}[Y_u]^2\right) \leq \sum_{u \in N^{left}(v)} \mathbf{E}[Y_u^2] \leq k.$$

We are interested in

$$\mathbf{Pr}[Y \geq k] \leq \mathbf{Pr}[|Y - E[Y]| \geq f(k) \cdot k].$$

By Chebyshev's inequality,

$$\mathbf{Pr}[|Y - E[Y]| \geq f(k) \cdot k] \leq \frac{\mathbf{Var}[Y]}{f(k)^2 \cdot k^2} \leq \frac{k}{f(k)^2 \cdot k^2} = \frac{1}{f(k)^2 \cdot k}.$$

From this, we find that the probability we actually select vertex v is at least

$$\mathbf{Pr}[Y_v \wedge (Y \leq k-1)] = \mathbf{Pr}[Y_v] \cdot \mathbf{Pr}[Y \leq k-1] \geq (1 - f(k)) \cdot x_v \cdot \left(1 - \frac{1}{f(k)^2 \cdot k}\right).$$

The first equality is justified because Y only depends on Y_u for $u \neq v$, so these two events are independent.

Choosing $f(k) = k^{-1/3}$ results in $v \in S$ with probability at least $x_v \cdot (1 - 2 \cdot k^{-1/3})$. By linearity of expectation, the expected value of S is at least $(1 - 2 \cdot k^{-1/3}) \cdot \sum_{v \in V} w_v \cdot x_v$.

The PTAS for MkCS in chordal graphs is now immediate. For any constant $\epsilon > 0$, if $k \geq 8/\epsilon^3$, then we run our LP rounding algorithm to get a k-colorable subgraph with weight at least $(1 - \epsilon) \cdot OPT_{LP}$. Otherwise, we run the exact algorithm in [19], which runs in polynomial time since k is bounded by a constant.

It is desirable to derandomize this algorithm so it always finds a solution with the stated guarantee. This is because we use it numerous times in the approximate separation oracle for (**DUAL-MSC**). Knowing it works all the time does not burden us with providing concentration around the probability we successfully approximately solve $\mathbf{LP}^{(\rho,\gamma)}$ as in Lemma 4. We can derandomize Algorithm 2 using standard techniques, since it only requires that the variables $Y_u, u \in V$ be pairwise-independent (in order to bound $\mathbf{Var}[Y]$).

4 Conclusion

It is natural to wonder if MSC admits a better approximation in perfect graphs. Unfortunately, our techniques do not extend immediately. In [1], Addario-Berry et al. showed MkCS is NP-HARD in a different subclass of perfect graphs than chordal graphs. Their proof reduces from the maximum independent set problem and it is easy to see it shows MkCS is APX-HARD in the same graph class if one reduces from bounded-degree instances of maximum independent set.

However, our approach, or a refinement of it, may succeed if one has good constant approximations for MkCS in perfect graphs. Note that MkCS can be approximated within $1 - 1/e$ in perfect graphs simply by using the *maximum coverage* approach. That is, for k iterations, we greedily compute a maximum independent set of nodes that are not yet covered. This is not sufficient to get an improved MSC approximation in perfect graphs using Lemma 2. Lemma 2 can be used if we get a sufficiently-good (≈ 0.704) approximation for MkCS. As a starting point, we ask if there is a ρ-approximation for MkCS in perfect graphs for some constant $\rho > 1 - 1/e$.

References

1. Addario-Berry, L., Kennedy, W., King, A., Li, Z., Reed, B.: Finding a maximum-weight induced k-partite subgraph of an i-triangulated graph. Discrete Applied Mathematics **158**(7), 765–770 (2010)
2. Bar-Noy, A., Bellare, M., Halldórsson, M.M., Shachnai, H., Tamir, T.: On chromatic sums and distributed resource allocation. Information and Computation **140**(2), 183–202 (1998)
3. Bar-Noy, A., Kortsarz, G.: The minimum color-sum of bipartite graphs. Journal of Algorithms **28**(2), 339–365 (1998)
4. Chakaravarthy, V., Roy, S.: Approximating maximum weight k-colorable subgraphs in chordal graphs. Information Processing Letters **109**(7), 365–368 (2009)
5. Chakrabarty, D., Swamy, C.: Facility location with client latencies: Linear programming based techniques for minimum latency problems. In: Integer Programming and Combinatoral Optimization. pp. 92–103 (2011)
6. Feige, U.: Approximating maximum clique by removing subgraphs. SIAM Journal on Discrete Mathematics **18**(2), 219–225 (2004)
7. Feige, U., Kilian, J.: Zero knowledge and the chromatic number. Journal of Computer and System Sciences **57**(2), 187–199 (1998)
8. Gandhi, R., Halldórsson, M.M., Kortsarz, G., Shachnai, H.: Improved bounds for sum multicoloring and scheduling dependent jobs with minsum criteria. In: Approximation and Online Algorithms. pp. 68–82 (2005)
9. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press (1980)
10. Gonen, M.: Coloring Problems on Interval Graphs and Trees. Master's thesis, School of Computer Science, The Open Univ., Tel-Aviv (2001)
11. Halldórsson, M.M., Kortsarz, G., Shachnai, H.: Sum coloring interval and k-claw free graphs with application to scheduling dependent jobs. Algorithmica **37**, 187–209 (2003)
12. Halldórsson, M., Kortsarz, G.: Tools for multicoloring with applications to planar graphs and partial k-trees. Journal of Algorithms **42**(2), 334–366 (2002)
13. Halldórsson, M., Kortsarz, G., Sviridenko, M.: Sum edge coloring of multigraphs via configuration lp. ACM Trans. Algorithms **7**(2), 1–21 (2011)
14. Kubicka, E.: The Chromatic Sum and Efficient Tree Algorithms. Ph.D. thesis, Western Michigan University (1989)
15. Malafiejski, M., Giaro, K., Janczewski, R., Kubale, M.: Sum coloring of bipartite graphs with bounded degree. Algorithmica **40**, 235–244 (2004)
16. Marx, D.: Complexity results for minimum sum edge coloring. Discrete Applied Mathematics **157**(5), 1034–1045 (2009)
17. Pereira, F., Palsberg, J.: Register allocation via coloring of chordal graphs. In: Programming Languages and Systems. pp. 315–329 (2005)
18. Post, I., Swamy, C.: Linear programming-based approximation algorithms for multi-vehicle minimum latency problems. In: Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 512–531 (2014)
19. Yannakakis, M., Gavril, F.: The maximum k-colorable subgraph problem for chordal graphs. Information Processing Letters **24**(2), 133–137 (1987)