Sum Coloring: New upper bounds for the chromatic strength

Clément Lecat, Corinne Lucet, Chu Min Li

To cite this version:

Clément Lecat, Corinne Lucet, Chu Min Li. Sum Coloring: New upper bounds for the chromatic strength. 2016. hal-01321029

HAL Id: hal-01321029
https://hal.science/hal-01321029
Preprint submitted on 24 May 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License
Sum Coloring : New upper bounds for the chromatic strength

Clément Lecat, Corinne Lucet, Chu-Min Li
Laboratoire de Modélisation, Information et Système EA 4290, Université de Picardie Jules Verne, Amiens, France

Abstract
The Minimum Sum Coloring Problem (MSCP) is derived from the Graph Coloring Problem (GCP) by associating a weight to each color. The aim of MSCP is to find a coloring solution of a graph such that the sum of color weights is minimum. MSCP has important applications in fields such as scheduling and VLSI design. We propose in this paper new upper bounds of the chromatic strength, i.e. the minimum number of colors in an optimal solution of MSCP, based on an abstraction of all possible colorings of a graph called motif. Experimental results on standard benchmarks show that our new bounds are significantly tighter than the previous bounds in general, allowing to reduce substantially the search space when solving MSCP.

1. Introduction

The Graph Coloring Problem (GCP) is an important NP-hard combinatorial problem. A lot of effort has been devoted to study it. Two main types of algorithms (also called solvers) are developed for solving GCP: exact methods and approximate methods. Exact methods aim at finding an optimal solution of the problem, including the approaches based on branch-and-bound schema [23], on graph decomposition [21], and on SAT solving by encoding the problem into an equivalent propositional formula [30]. The minimum number of colors needed to color a graph $G$ is called the chromatic number of $G$ and is denoted by $\chi(G)$. The approximate methods aim at finding an upper bound or a lower bound of the optimal solution of the problem, including : greedy algorithms such as the famous DSATUR [6], and various heuristic or meta-heuristic algorithms [26, 11, 20]. In the literature, these methods are usually evaluated on standard benchmarks such as DIMACS and COLOR [7, 12].

The Minimum Sum Coloring Problem (MSCP) is derived from GCP and is introduced in 1989 by Kubicka et Schwenk [16], by associating a weight to each color. The aim of MSCP is to find a valid coloring solution that minimizes the sum of color weights. The minimum number of colors in an optimal solution of MSCP for a graph $G$ is called the chromatic strength of $G$ and is denoted by $s(G)$. Note that $s(G)$ can be bigger than $\chi(G)$. MSCP has important applications in fields such as scheduling, VLSI design and resource allocation [1, 22]. For example, to calculate the best quality of service in a distributed system with shared resource amounts to solve MSCP. The main results on MSCP include the theoretical bounds [13, 29, 22, 2] and the structural properties relative to the graph families for which efficient MSCP algorithms exist. Recently, heuristics and meta-heuristics for MSCP are proposed in [19] and [28, 15, 3, 25] respectively, giving bounds for DIMACS and COLOR graphs, and exact methods are proposed in [17].

When solving GCP and MSCP for a graph $G$, an algorithm generally has to explore the search space of GCP and MSCP that grows exponentially with the number of colors to be considered. In practice it is substantially harder to reduce the number of colors to be considered when solving MSCP than when solving GCP. In fact, for GCP when a valid coloring solution with $k$ colors is found, the sub-space with $k$ or more colors can be pruned. However, this is not the case for MSCP, because the optimal solution of MSCP can involve more than $k$ colors. So, establishing a tight upper bound of $s(G)$ is essential to solve MSCP.
Unfortunately, there are few works in the literature allowing to derive a tight upper bound of \( s(G) \). The only two existing upper bounds in our knowledge are proposed in [24] and in [9] respectively. In this paper, we propose two new upper bounds of \( s(G) \) by exploring an abstraction of the set of coloring solutions of \( G \) called motif. The notion of motif was already used by Bonomo and Valencia-Pabon in [4, 5] to solve MSCP for \( P_4 \)-sparse graphs. However, it is the first time in our knowledge that motifs are used for upper bounding \( s(G) \). By skillfully identifying and excluding those motifs that cannot correspond to an optimal solution of MSCP, we derive the two new upper bounds of \( s(G) \) from the remaining motifs.

The experimental results on standard benchmarks DIMACS and COLOR [7, 12] for coloring problems show that our bounds are substantially better than the existing bound proposed in [9] in general. In some instances, the gain is greater than 200 colors. The other existing bound proposed in [24] is not compared because it is trivial.

This paper is organized as follows. Section 2 presents the preliminaries necessary for our approach. Section 3 presents our approach for identifying the motifs that cannot correspond to an optimal solution of MSCP and for computing the new upper bounds of \( s(G) \). Section 4 compares our new upper bounds with the existing bound proposed in [9] on the DIMACS and COLOR graphs. Section 5 concludes.

2. Preliminaries

2.1. Basic definitions, GCP, MSCP, MaxClique and MaxStable

We consider an undirected graph \( G = (V, E) \), where \( V \) is a set of vertices \((|V| = n)\) and \( E \subseteq V^2 \) a set of edges. The set of adjacent (or neighbor) vertices of \( v \in V \), denoted by \( N \), is defined as : \( N(v) = \{u \mid (u, v) \in E\} \). The degree \( d(v) \) of a vertex \( v \) is the number of its adjacent vertices, i.e., \( d(v) = |N(v)| \). The degree of a graph, denoted by \( \Delta(G) \), is \( \max\{d(v) \mid v \in V\} \). A clique \( C \) is a subset of \( V \) such that \( \forall u, v \in C, (u, v) \in E \). A stable set \( S \) is a subset of \( V \) such that \( \forall u, v \in S, (u, v) \notin E \). The complement graph of \( G \) is defined as \( \overline{G} = (V, \overline{E}) \) where \( \overline{E} = V^2 \setminus E \). A clique in \( G \) is a stable set in \( \overline{G} \) and vice versa. A graph \( G' = (V', E') \) is a subgraph of \( G \) induced by \( V' \) if \( V' \subseteq V \) and \( E' = V'^2 \cap E \).

A coloring of a graph \( G \) with \( k \) colors is a function \( c : V \rightarrow \{1, 2, \ldots, k\} \) that assigns to each vertex \( v \in V \) a color \( c(v) \). A coloring is valid, if \( \forall (u, v) \in E, c(u) \neq c(v) \). We denote a coloring of \( G \) with \( k \) colors by \( X = \{X_1, X_2, \ldots, X_k\} \), where \( X_i = \{v \in V \mid c(v) = i\} \) is called a color class. The Graph Coloring Problem (GCP) consists in finding a valid coloring \( X \) of \( G \) with minimum \( k \). Such \( k \) is called the chromatic number of \( G \), and is denoted by \( \chi(G) \). GCP is NP-Hard [8].

The Minimum Sum Coloring Problem (MSCP) is derived from GCP by associating a weight \( w_i \) with each color \( i \). In this paper, we consider \( w_i = 1 \). We denote by \( \Sigma(X) \) the sum of color weights of a coloring:

\[
\Sigma(X) = 1 \times |X_1| + 2 \times |X_2| + ... + k \times |X_k|
\]

Example 1. Refer to the graph in Figure 1, \( X = \{\{a, e\}, \{b\}, \{c, d, f\}\} \) is a valid coloring. The vertices \( a \) and \( e \) are colored with the color 1, the vertex \( b \) with the color 2, and the vertices \( c, d \) and \( f \) with the color 3. The sum coloring is \( \Sigma(X) = 1 \times 2 + 2 \times 1 + 3 \times 3 = 13 \).

Given a graph \( G, MSCP \) consists in finding a valid coloring \( X \) of \( G \) with the minimum sum of color weights \( \Sigma(X) \). This minimum sum is called the chromatic sum of \( G \) and is denoted by \( \Sigma(G) \):

\[
\Sigma(G) = \min \{\Sigma(X) \mid X \text{ is a valid coloring of } G\}
\]

Kubicka and Schwenk proved that MSCP is NP-Hard [16]. An optimal solution of GCP does not necessarily correspond to an optimal solution for MSCP. For example, the optimal solution of GCP for the graph in Figure 2 uses 2 colors, for which the sum of color weights is 12, while an optimal solution of MSCP for this graph uses 3 colors. The chromatic sum of the graph is 11. The minimum number of
colors in an optimal solution of \( MSCP \) of a graph \( G \) is called chromatic strength (or simply strength) of \( G \), and is denoted by \( s(G) \).

A valid coloring of a graph \( G = (V, E) \) with \( k \) colors \( c : V \mapsto \{1, 2, \ldots, k\} \) is to be found in a set of colorings of cardinality \( k^{|V|} \), which forms the search space for both \( GCP \) and \( MSCP \). The number \( k \) must be large enough to achieve an optimal solution. While \( GCP \) and \( MSCP \) are both NP-hard, \( MSCP \) is much harder to solve than \( GCP \) in practice, because it is much more complex to prune search space when solving \( MSCP \). So, determining an upper bound of \( s(G) \) as tight as possible is essential for solving \( MSCP \).

Given a graph \( G \), the MaxClique (MaxStable) problem consists in finding a clique (stable set) with the maximum cardinality in \( G \). Note that finding a maximum stable set in \( G \) is equivalent to finding a maximum clique in \( \overline{G} \).

2.2. Major coloring

For a given coloring \( X \), each color class \( X_i \) is a stable set. We can exchange two colors \( i \) and \( j \) without impacting the validity of \( X \). The set of colorings that can be achieved by such exchanges from \( X \), are symmetric and form an equivalence class, denoted by \( \Theta(X) \). All colorings of \( \Theta(X) \) use the same number of colors, but they do not give the same sum of color weights. Refer to Figure 1, the sum of color weights of the coloring \( \{\{a, e\}_1, \{c, d, f\}_2, \{b\}_3\} \) is 13, while the sum of color weights of the coloring \( \{\{a, e\}_1, \{c, d, f\}_2, \{b\}_3\} \) is 11. Therefore, we define the notion of major coloring.

**Definition 1.** A major coloring, denoted by \( X^m \), is a coloring \( X^m = \{X_1, X_2, \ldots, X_k\} \) such that \( |X_1| \geq |X_2| \geq \ldots \geq |X_k| \).

**Example 2.** Refer to Figure 1, the coloring \( X = \{\{c, d, f\}_1, \{a, e\}_2, \{b\}_3\} \) is a major coloring of the graph. \( \Sigma(X^m) = 10 \).

The following property is a direct consequence of Definition 1.
Property 1. Let $\Theta(X)$ be the set of symmetric colorings of $X$ and $X^m$ a major coloring of $\Theta(X)$, then \( \forall X' \in \Theta(X), \Sigma(X^m) \leq \Sigma(X') \).

Consequently, only major colorings need to be considered when solving MSCP. In the sequel, all colorings we consider are major and are simply written as $X$.

2.3. Motifs

Any major coloring $X$ with $k$ colors corresponds to a non-increasing sequence $p$ of integers : $(|X_1|, |X_2|, \ldots, |X_k|)$, called the motif of $X$. The $i^{th}$ integer is denoted by $p[i] = |X_i|$. The sum of color weights of $X$ can be computed as :

\[
\Sigma(X) = \Sigma(p) = 1 \times p[1] + 2 \times p[2] + \ldots + k \times p[k]
\]

Example 3. The motif corresponding to $X = \{\{c,d,f\}, \{a,e\}, \{b\}\}$ is $p = (3,2,1)$, with $p[1] = 3$, $p[2] = 2$ and $p[3] = 1$. $\Sigma(X) = \Sigma(p) = 10$.

The interest of the motif notion is that a motif provides an abstraction of several colorings, that is essential for MSCP. For example, refer to Figure 1, the two different colorings \(\{c,d,f\}, \{a,e\}\), \(\{b\}\) and \(\{a,e,f\}, \{b,c\}, \{e\}\) can be represented by the same motif $p = (3,2,1)$. Two different motifs represent necessarily different colorings. Nevertheless, a motif can represent an invalid coloring, because it does not include any structural property of a graph. For example, refer to Figure 1, the motifs $(5,1)$ and $(4,1,1)$ do not represent any valid coloring of the graph. As we will show in Section 3.4, some characteristics of a graph can be used to exclude a part of motifs representing invalid colorings, so that we can derive interesting properties for MSCP from the remaining motifs. We denote by $\phi(n)$ the set of all motifs for any graph with $n$ vertices, and $\phi(n,k)$ the set of motifs with $k$ colors for any graph with $n$ vertices.

\[
\phi(n,k) = \{p \in \phi(n) \mid |p| = k\}
\]

\[
\phi(n) = \bigcup_{k=1}^{n} \{\phi(n,k)\}
\]

The number of motifs in $\phi(n)$ is equal to the number of partitions of $n$ into non-increasing integers, which grows exponentially with $n$. Hardy and Ramanujan[10] give an approximate number of partitions of $n$ into non-increasing integers :

\[
|\phi(n)| \sim \frac{1}{4n\sqrt{3}} e^{\pi \sqrt{2n/3}}
\]

Table 1 illustrates the exponential growth of the cardinality of $\phi(n)$.

<table>
<thead>
<tr>
<th>$n$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>150</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\phi(n)</td>
<td>$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>11</td>
<td>15</td>
<td>22</td>
<td>30</td>
<td>42</td>
<td>697</td>
<td>204,226</td>
</tr>
</tbody>
</table>

Table 1: Cardinality of $\phi(n)$.

Our approach works in $\phi(n)$ to find a tight upper bound of the chromatic strength of any graph with $n$ vertices, using the dominance relation between the motifs defined in the next subsection.

2.4. Dominance relation

The dominance relation, denoted by $\succeq$, between the motifs of $\phi(n)$ was introduced by Bonomo and Valencia in [4, 5] to compute the chromatic sum of a subset of $P_3$-sparse graphs, or an upper bound of the chromatic sum for general $P_3$-sparse graphs. We adapt their definition for our approach to derive tight upper bounds of the chromatic strength.
Definition 2. Let \( p \) and \( q \) be two motifs in \( \phi(n) \). We say that \( p \) dominates \( q \), denoted by \( p \geq q \), if and only if \( \forall t \) such that \( 1 \leq t \leq \min\{|p|, |q|\} \), \( \sum_{x=1}^{t} p[x] \geq \sum_{x=1}^{t} q[x] \).

The main difference of Definition 2 with the original definition in [4] is that in this paper we explicitly require \( \sum_{x=1}^{|p|} p[x] = n \), because \( p \) and \( q \) are both in \( \phi(n) \).

Two motifs are not necessarily comparable, as shown in Example 4.

Example 4. Let \( p \) and \( q \) be two motifs of a graph \( G \) with 15 vertices, \( p = (9, 3, 3) \) and \( q = (8, 6, 1) \).

If \( t = 1 \) : \( \sum_{i=1}^{1} p[i] > \sum_{i=1}^{1} q[i] \) (9 > 8).

If \( t = 2 \) : \( \sum_{i=1}^{2} p[i] < \sum_{i=1}^{2} q[i] \) (9+3 < 8+6).

\( p \not\geq q \) and \( q \not\geq p \), i.e., \( p \) and \( q \) are not comparable.

So, the dominance relation \( \geq \) is a partial order. It is easy to prove the following property that makes the dominance relation useful for \( MSCP \) [4, 5].

Property 2. Let \( G \) be a graph, \( p \) and \( q \) two motifs corresponding to two valid colorings \( X \) and \( X' \) of \( G \), respectively. If \( p \geq q \) then \( \Sigma(X) \leq \Sigma(X') \).

3. Tight upper bounds for the chromatic strength

Much effort is spent to establish the relationship between the structural properties of a graph \( G \) and its chromatic sum [2, 27, 14, 5]. However, less attention is paid to the chromatic strength. Property 3 states a trivial upper bound of \( s(G) \) found in [24] :

Property 3.

\[ s(G) \leq \Delta(G) + 1 \]

A better upper bound based on a valid coloring of \( G \) is given in [9] and is stated in Property 4.

Property 4. Let \( G \) be a graph and \( X \) a valid coloring with \( k \) colors, then

\[ s(G) \leq \left\lceil \frac{\Delta(G) + \chi(G)}{2} \right\rceil \leq \left\lceil \frac{\Delta(G) + k}{2} \right\rceil \]

In this section, we propose two new upper bounds of \( s(G) \) based on a valid coloring of \( G \) by exploring \( \phi(n) \). For this purpose, we first present a total order in \( \phi(n) \) and an algorithm allowing to assign an index to each motif in \( \phi(n,k) \), which is needed for the understanding of the new upper bounds. Then we present the bounds after explaining their principle.

3.1. Establishing a total order in \( \phi(n) \)

The set of motifs \( \phi(n) \) can be sorted in the following order : \( \phi(n,1), \phi(n,2), \ldots, \phi(n,n) \), then each \( \phi(n,k) \) is sorted in the decreasing lexicographical order. As an example, Table 2 lists all motifs in \( \phi(8) \) in the above order.

Algorithm 1 generates the motifs of \( \phi(n,k) \) in the decreasing lexicographic order, so that we can assign an index \( i \) to each motif in \( \phi(n,k) \).

Definition 3. We denote by \( p_i^k \) the \( i \)th motif in \( \phi(n,k) \).
<table>
<thead>
<tr>
<th>$\phi(n,k)$</th>
<th>$p \in \phi(n,k)$</th>
<th>$\Sigma(p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi(8,1)$</td>
<td>(8)</td>
<td>8</td>
</tr>
<tr>
<td>$\phi(8,2)$</td>
<td>(7,1)</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>(6,2)</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>(5,3)</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>(4,4)</td>
<td>12</td>
</tr>
<tr>
<td>$\phi(8,3)$</td>
<td>(6,1,1)</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>(5,2,1)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>(4,3,1)</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>(4,2,2)</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>(3,3,2)</td>
<td>15</td>
</tr>
<tr>
<td>$\phi(8,4)$</td>
<td>(5,1,1,1)</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>(4,2,1,1)</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>(3,3,1,1)</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>(3,2,2,1)</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>(2,2,2,2)</td>
<td>20</td>
</tr>
<tr>
<td>$\phi(8,5)$</td>
<td>(4,1,1,1,1)</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>(3,2,1,1,1)</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>(2,2,2,1,1)</td>
<td>21</td>
</tr>
<tr>
<td>$\phi(8,6)$</td>
<td>(3,1,1,1,1,1)</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>(2,2,1,1,1,1)</td>
<td>24</td>
</tr>
<tr>
<td>$\phi(8,7)$</td>
<td>(2,1,1,1,1,1,1)</td>
<td>29</td>
</tr>
<tr>
<td>$\phi(8,8)$</td>
<td>(1,1,1,1,1,1,1,1)</td>
<td>36</td>
</tr>
</tbody>
</table>

Table 2: $\phi(8)$.

The first motif $p^1_k$ in $\phi(n,k)$ is

$$(n - k + 1, 1, 1, \ldots, 1)$$

$k - 1$ times

and the last motif in $\phi(n,k)$ is

$$\left(\left\lceil \frac{n}{k} \right\rceil, \ldots, \left\lceil \frac{n}{k} \right\rceil, \left\lfloor \frac{n}{k} \right\rfloor, \ldots, \left\lfloor \frac{n}{k} \right\rfloor \right)$$

Algorithm 1 works as follows. Given two positive integers $n$ and $k$ such that $k \leq n$, and a motif $p$ under construction that is initially empty, the algorithm generates all partitions of $n$ into $k$ integers in the decreasing lexicographic order. Each partition begins by the first integer that should be between $\text{sup}$ that is $n - k + 1$ if $p$ is empty, $\text{min}(n - k + 1, \text{min}(p))$ otherwise, and $\left\lceil \frac{n}{k} \right\rceil$, the other integers being generated by a recursive call of the algorithm. Note that the first integer should not be bigger than the minimum integer in $p$ if $p$ is not empty. The integers of each partition are appended to the end of $p$ to form a complete motif in decreasing order.

For example, in the call MOTIF(3, 8, [], []) ($p = \phi = \emptyset$), the algorithm generates each first integer between 6 and 3. When the first integer is 6, the algorithm calls MOTIF(2, 2, {6}, []) which generates only one partition of 2 into 2 integers (i.e. (1, 1)) appended into {6} to form a complete motif (6, 1, 1). When the first integer is 4, the algorithm calls MOTIF(4, 2, {4}, {(6, 1, 1), (5, 3, 2)}) that generates two partitions of 4 into 2 integers : (3, 1) and (2, 2), each partition being appended into {4} to form a complete motif. See Table 2 for all results.
Algorithm 1 : MOTIF\( (n,k,p,\phi) \)

**Input** : two positive integers \( n \) and \( k \) such that \( k \leq n \), motif \( p \) under construction, set of motifs \( \phi \) under construction

**Output** : a set of motifs \( \phi \) in decreasing lexicographic order

1 begin
2 if \( k = 1 \) then
3 \[ \phi \leftarrow \phi \cup \{p \cup \{n\}\}; \]
4 else
5 if \( p = \emptyset \) then
6 \[ \text{sup} \leftarrow n - k + 1; \]
7 else
8 \[ \text{sup} \leftarrow \min(n - k + 1, \min(p)); \]
9 \[ \text{inf} \leftarrow \left\lceil \frac{n}{k} \right\rceil; \]
10 foreach \( x \leftarrow \text{sup} \text{ downto } \text{inf} \) do
11 \[ \text{Motif}(n - x, k - 1, p \cup \{x\}, \phi); \]
12 end

3.2. Principle of the new upper bounds

Given a valid coloring \( X \) of a graph \( G \) and its associated motif \( p \), we will find the number \( k_t \) of colors such that all motifs with \( k_t \) or more colors are dominated by \( p \). Namely :

\[
\forall q \in \psi = \bigcup_{x=k_t}^{n} \phi(n,x), \quad p \succeq q \tag{3}
\]

Note that \( \psi \) is not empty, because it contains trivially the only motif \((1, 1, \ldots, 1)\) in \( \phi(n,n) \) for \( k_t = n \). Since \( \psi \) does not contain any valid coloring better than \( p \), \( k_t \) is an upper bound of \( s(G) \). The problem is to make \( k_t \) as small as possible.

Based on this above principle, we will establish two upper bounds for \( s(G) : UB_a \), an algebraic upper bound, and UB, an algorithmic upper bound based on a maximum stable set of \( G \).

3.3. UB\(_a\), an algebraic upper bound for \( s(G) \)

UB\(_a\) is based on the following two properties.

**Property 5.** Let \( G = (V,E) \) with \( |V|=n \), and \( k \leq n \) be an integer. The motif \( p_k^1 = (n-k+1,1,1,\ldots,1) \) dominates all other motifs in \( \phi(n,k) \).

In fact, let \( q \in \phi(n,k) \). Then \( \forall t \) such that \( 1 \leq t \leq k \), \[ \sum_{x=t+1}^{k} p_k^1[x] \leq \sum_{x=t+1}^{k} q[x], \text{because } p_k^1[x] = 1 \leq q[x] \]
when \( x > 1 \). So, \[ \sum_{x=1}^{t} p_k^1[x] = n - \sum_{x=t+1}^{k} p_k^1[x] \geq n - \sum_{x=t+1}^{k} q[x] = \sum_{x=1}^{t} q[x] \].

**Property 6.** Let \( G = (V,E) \) with \( |V|=n \), and \( k \) and \( k' \) be two integers such that \( 1 \leq k < k' \leq n \). The motif \( p_k^1 = (n-k+1,1,1,\ldots,1) \) dominates \( p_{k'}^1 = (n-k'+1,1,1,\ldots,1) \).

In fact, since \( n - k + 1 > n - k' + 1 \), \( \forall t \) such that \( 1 \leq t \leq k \), \[ \sum_{x=1}^{t} p_k^1[x] \geq \sum_{x=1}^{t} p_{k'}^1[x] \].
Property 5 and Property 6 mean that the motif \( p_k = (n-k+1, 1, 1, \ldots, 1) \) dominates any motif in \( \phi(n, k) \) such that \( k \leq k' \). The sum coloring of \( p_k \) is:

\[
\Sigma(p_k) = (n-k+1) + \sum_{x=2}^{k} x = \frac{1}{2}k^2 - \frac{1}{2}k + n \quad (4)
\]

So UB\(_a\) is the smallest number \( \max(k-1, |X|) \) of colors such that

\[
\Sigma(p_k) \geq \Sigma(X) \quad (5)
\]

Equation 5 means that a sum coloring with \( k \) or more colors cannot be better than the known valid coloring \( X \) according to Property 5 and Property 6. So, the optimal solution must be with at most \( k-1 \) colors if \( X \) uses fewer than \( k \) colors, \( k \) otherwise.

Using Equation 4, Equation 5 is transformed to

\[
\frac{1}{2}k^2 - \frac{1}{2}k + n - \Sigma(X) \geq 0 \quad (6)
\]

Equation 6 is valid if and only if:

\[
k \geq 1 + \sqrt{1 + 8 \times (\Sigma(X) - n)}
\]

Thus:

\[
UB_a = \max\left(\left\lceil 1 + \sqrt{1 + 8 \times (\Sigma(X) - n)}\right\rceil - 1, |X|\right) \quad (7)
\]

UB\(_a\) is a new algebraic upper bound of \( s(G) \). Note that apart from the valid coloring \( X \), UB\(_a\) does not consider any other structural information of \( G \). We will show in the next subsection that we can obtain a better upper bound by taking into account more structural information of \( G \).

3.4. UB\(_a\), an algorithmic upper bound of \( s(G) \)

Although MaxClique, GCP and MSCP are all NP-hard problems, MaxClique is relatively easier to solve than GCP and MSCP in practice. For example, the state-of-the-art exact algorithm IncMaxClique [18] finds a maximum clique of any random graph of 200 vertices in few seconds, but no exact algorithm in our knowledge is able to find the chromatic number of a random graph of 200 vertices and density 0.5 in reasonable time (a random graph of density \( d \) is generated by making each pair of vertices adjacent with probability \( d \)). So, we can find a maximum stable set of a graph \( G \), which is a maximum clique in the complement graph \( \overline{G} \), to compute an upper bound better than UB\(_a\), based on Property 7.

**Property 7.** Let \( \alpha(G) \) denote the cardinality of a maximum stable set of \( G \). A motif in which the first integer is bigger than \( \alpha(G) \) does not correspond to any valid coloring of \( G \).

We show in the sequel how to derive an upper bound of \( s(G) \) called UB\(_a\) by restricting ourselves to all motifs in which the first integer is smaller than or equal to \( \alpha(G) \). For this purpose, we define a notion called major motif which is motivated by the following observation. Given any valid major coloring \( X \), it is sometimes possible to find two color classes \( X_i \) and \( X_j \) \((i<j)\) such that a vertex of \( X_j \) can be moved into \( X_i \) to obtain another valid major coloring \( X' \). Clearly \( \Sigma(X') < \Sigma(X) \) and the motif \( p' \) associated with \( X' \) dominates the motif \( p \) associated with \( X \). The motif \( p' \) is obtained by decrementing the \( j \)th integer of \( p \) by 1 and incrementing the \( i \)th integer of \( p \) by 1. We call this transformation of \( p \) left-shifting operation.

Note that the left-shifting operation is not always possible, because one must keep the integers of the resulting motif in non-increasing order. Moreover, all integers in the resulting motif should be positive.
Given two integers $n$ and $k$, we are interested in those motifs in $\phi(n,k)$, called major motifs, that cannot be transformed into another motif in $\phi(n,k)$ by a left-shifting operation without incrementing the first integer.

**Example 5.** Consider $\phi(8,4)$ (refer to Table 2). The major motifs are $(5,1,1,1)$, $(4,2,1,1)$, $(3,3,1,1)$ and $(2,2,2,2)$. In fact, unless the first integer is incremented, these motifs cannot be transformed into any motif in $\phi(8,4)$ by a left-shifting operation.

Let $\lambda$ denote the first integer of a motif. We have $\left\lceil \frac{n}{k} \right\rceil \leq \lambda \leq n - k + 1$. Intuitively, a motif is major if it contains the maximum number ($\beta$) of integers equal to its first integer, $\lambda$. The remaining value of $n$ (i.e. $n - \beta \times \lambda$) should be partitioned into $k - \beta$ positive integers. So $\beta$ is the maximum integer satisfying $n - \beta \times \lambda \geq k - \beta$, or $\beta \leq \frac{n - k}{\lambda - 1}$ after excluding the trivial motif $(1, 1, \ldots, 1)$ and assuming $\lambda > 1$. So, $\beta = \left\lfloor \frac{n - k}{\lambda - 1} \right\rfloor$.

A major motif is formally defined in Definition 4.

**Definition 4.** Let $\lambda$ and $\beta$ be two integers such that $\left\lceil \frac{n}{k} \right\rceil \leq \lambda \leq n - k + 1$ and $\beta = \left\lfloor \frac{n - k}{\lambda - 1} \right\rfloor$. A major motif in $\phi(n,k)$ is a motif $p^k_\lambda$ with the following properties:

\[
\begin{align*}
p^k_\lambda[x] &= \lambda, \text{ if } 1 \leq x \leq \beta; \\
p^k_\lambda[x] &= n - \beta \times \lambda - (k - \beta - 1), \text{ if } x = \beta + 1; \\
p^k_\lambda[x] &= 1, \text{ if } \beta + 1 < x \leq k.
\end{align*}
\]

The interest of the major motif notion lies in the following two properties. The first one, Property 8, says that a major motif in $\phi(n,k)$ dominates all motifs $\phi(n,k)$ with the same first integer and represents the best sum coloring among these motifs.

**Property 8.** Let $p$ and $q$ be two motifs in $\phi(n,k)$ such that $p$ is major and $p[1] = q[1]$, then $p \succeq q$.

**Proof 1.** Let $\beta = \left\lfloor \frac{n-k}{p[1]-1} \right\rfloor$. Since $p$ is major, we have $p[1] = p[2] = \ldots = p[\beta] = q[1] \geq q[2] \geq \ldots \geq q[\beta]$.

So, $\forall t$ such that $1 \leq t \leq \beta$, $\sum_{x=1}^{t} p[x] \geq \sum_{x=1}^{t} q[x]$.

Moreover, $\forall t$ such that $\beta + 1 \leq t \leq k$, we have $p[t+1] = p[t+2] = \ldots = p[k] = 1 \leq q[k] \leq q[k-1] \leq \ldots \leq q[t+1]$. So, $\sum_{x=1}^{t} p[x] = n - \sum_{x=t+1}^{k} p[x] \geq n - \sum_{x=t+1}^{k} q[x] = \sum_{x=1}^{t} q[x]$.

Therefore, $p \succeq q$. □

Property 9 compares two major motifs in $\phi(n,k)$.

**Property 9.** Let $p$ and $q$ be two major motifs in $\phi(n,k)$. If $p[1] > q[1]$ then $p \succeq q$.

**Proof 2.** Let $\beta_p = \left\lfloor \frac{n-k}{p[1]-1} \right\rfloor$ and $\beta_q = \left\lfloor \frac{n-k}{q[1]-1} \right\rfloor$. Clearly, $\beta_p \leq \beta_q$. We have $p[1] = p[2] = \ldots = p[\beta_p] > q[1] = q[2] = \ldots = q[\beta_p]$. So, $\forall t$ such that $1 \leq t \leq \beta_p$, $\sum_{x=1}^{t} p[x] > \sum_{x=1}^{t} q[x]$.

Moreover, $\forall t$ such that $\beta_p + 1 \leq t \leq k$, we have $p[t+1] = p[t+2] = \ldots = p[k] = 1 \leq q[k] \leq q[k-1] \ldots \leq q[t+1]$, implying $\sum_{x=1}^{t} p[x] = n - \sum_{x=t+1}^{k} p[x] \geq n - \sum_{x=t+1}^{k} q[x] = \sum_{x=1}^{t} q[x]$. Therefore, $p \succeq q$. □

Since motifs in $\phi(n,k)$ are in decreasing lexicographical order, Property 8 and Property 9 imply that a major motif in $\phi(n,k)$ dominates all subsequent motifs in $\phi(n,k)$, as stated in Property 10.
Property 10. Let \( p_k^i \in \phi(n, k) \) be a major motif and \( p_k^j \in \phi(n, k) \) such that \( j > i \). Then \( p_k^j \geq p_k^i \).

Observe that while a major coloring is the best sum coloring in a set of symmetric colorings, a major motif is the best motif in a set of motifs in the sense of Property 10. A direct consequence is that \( \forall p_k^i \in \phi(n, k), p_k^j \geq p_k^i \), because \( p_k^i \) is major.

The following property compares the major motif \( p_k^1 \) to \( p_k^{k'} \), such that \( k < k' \).

Property 11. Let \( k \) and \( k' \) be two integers such that \( k < k' \), and \( \psi = \bigcup_{x=k'}^n \{ \phi(n, x) \} \), then \( \forall \psi \in \psi, p_k^1 \geq q \).

Proof 3. According to Property 6, \( p_k^1 \) dominates \( p_k^x \) \( \forall x \) such that \( k' \leq x \leq n \). Since \( p_k^1 \) dominates all motifs in \( \phi(n, x) \) according to Property 10, \( p_k^1 \) dominates all motifs in \( \psi \). \( \square \)

We now prove the most important property of this paper.

Property 12. Let \( k \) and \( k' \) be two integers such that \( k < k' \). Let \( p_k^i \) be a major motif and \( \psi = \bigcup_{y=k'}^n \{ \phi(n, y) \} \). Then \( \forall \psi \in \psi \) such that \( q \) \( [1] \leq p_k^i \), \( p_k^i \geq q \).

Proof 4. Let \( \beta = \left\lfloor \frac{n-k}{p_k^i [1]-1} \right\rfloor \). Since \( p_k^i \) is major, we have \( p_k^i [1] = p_k^i [2] = \ldots = p_k^i [\beta] \geq q [1] \geq q [2] \geq \ldots \geq q \{\beta\} \). So, \( \forall t \) such that \( 1 \leq t \leq \beta \), \( \sum_{x=1}^t p_k^i [x] \geq \sum_{x=1}^t q [x] \).

Moreover, \( \forall t \) such that \( \beta + 1 \leq t \leq k \), we have \( p_k^i [t+1] = p_k^i [t+2] = \ldots = p_k^i [k] = 1 \leq q [k] \leq q [k-1] \leq \ldots \leq q [t+1] \) (\( q \) is a sequence of non-increasing positive integers), implying \( \sum_{x=t+1}^k p_k^i [x] \geq n - \sum_{x=t+1}^k q [x] \). Therefore, \( p_k^i \geq q \). \( \square \)

Given a valid coloring \( X \) of a graph \( G \) with \( n \) vertices and the cardinality \( \alpha(G) \) of a maximum stable set of \( G \), we can search for the smallest \( k \) and the major motif \( p_k^i \) with \( p_k^i [1] = \alpha(G) \) such that \( \Sigma(X) < \Sigma(p_k^i) \). Property 12 says that a valid coloring of \( G \) with \( k \) or more colors better than \( X \) cannot be found. So \( UB_a = k - 1 \). Algorithm 2 implements UB_a. The function constructMajorMotif(\( \lambda, k \)) constructs a major motif \( p \in \phi(n, k) \) such that \( p [1] = \lambda \), as defined in Definition 4. The function computeSumColoring(\( p \)) computes the sum of color weights of \( p \), as shown in Equation 1.

The complexity of the constructMajorMotif(\( \lambda, k \)) function and the complexity of the computeSumColoring(\( p \)) function are both \( O(k) \). Since \( k \leq n \), the complexity of Algorithm 2 is \( O(n^2) \).

4. Empirical evaluation

In this section, we compare our algebraic and algorithmic bounds (UB_a and UB_s) for the chromatic strength of a graph \( G \) with that proposed by Hajiabolhassan et al. [9] (Property 4). To find a maximum stable set of \( G \), we run the state-of-the-art exact MaxClique algorithm IncMaxCLQ [18] to find a maximum clique in the complement graph \( \overline{G} \). We evaluate our approach using the COLOR [12] and DIMACS [7] graphs.

Table 3 gives the experimental results. For each graph \( G \), we denote by \( k^* \) the best-known upper bound for the chromatic number (\( \chi(G) \)), \( \Sigma^* \) the best-known upper bound for the chromatic sum (\( \Sigma(G) \)), \( \alpha(G) \) the cardinality of a maximum stable set of \( G \), Time_a the runtime in seconds to find a maximum stable set in \( G \), UB_hmt the results of Hajiabolhassan et al. given by Property 4 using \( \Delta(G) \) and \( k^* \), UB_a the results of our algebraic bound and UB_s the results of our algorithmic bound based on a maximum stable of \( G \). Both UB_a and UB_s are computed using \( \Sigma(G)^* \).
Algorithm 2: CUB₃(n, α(G), X)

Input: the number of vertices n, the cardinality of a maximum stable set α(G), and a valid coloring X
Output: an upper bound of s(G)

begin
SUM ← 0;
k ← |X|; /*initialize k*/
λ ← α(G);
while SUM ≤ Σ(X) do
    k ← k + 1;
    if n − k + 1 < λ then
        λ ← n − k + 1;
        p ← constructMajorMotif(λ, k);
        SUM ← computeSumColoring(p);
return k − 1;

Graph | V | E | k* | Σ* | Δ(G) | α(G) | Timeₙ | UBₙ | UBₙ | UBₙ
anna | 138 | 493 | 11 | 276 | 71 | 80 | 0 | 41 | 17 | 14
David | 87 | 406 | 11 | 237 | 82 | 36 | 0 | 47 | 17 | 15
DSJC1000.1 | 1000 | 49629 | 21 | 9931 | 127 | N/A | N/A | 74 | 134 | N/A
DSJC1000.5 | 1000 | 249826 | 87 | 46003 | 551 | 15 | 408 | 319 | 285 | 187
DSJC1000.9 | 1000 | 449449 | 224 | 106452 | 924 | 6 | 223 | 574 | 459 | 360
DSJC125.1 | 125 | 736 | 5 | 326 | 23 | 34 | 0 | 14 | 20 | 11
DSJC125.5 | 125 | 3891 | 17 | 1012 | 75 | 10 | 0 | 46 | 42 | 29
DSJC125.9 | 125 | 6961 | 44 | 2503 | 120 | 4 | 0 | 82 | 69 | 58
DSJC250.1 | 250 | 3218 | 8 | 996 | 38 | 44 | 117 | 23 | 39 | 23
DSJC250.5 | 250 | 15668 | 28 | 3306 | 147 | 12 | 0 | 88 | 78 | 53
DSJC250.9 | 250 | 27897 | 72 | 8288 | 234 | 5 | 24 | 153 | 127 | 105
DSJC500.1 | 500 | 12458 | 12 | 2997 | 68 | N/A | N/A | 40 | 71 | N/A
DSJC500.5 | 500 | 62624 | 48 | 11759 | 286 | 13 | 16 | 167 | 150 | 97
DSJC500.9 | 500 | 112437 | 126 | 30113 | 471 | 5 | 84 | 299 | 244 | 190
flat1000-50-0 | 1000 | 245000 | 50 | 39315 | 520 | 20 | 263 | 285 | 277 | 212
flat1000-60-0 | 1000 | 245830 | 60 | 40648 | 524 | 17 | 296 | 292 | 282 | 200
flat1000-76-0 | 1000 | 246708 | 76 | 41199 | 532 | 15 | 466 | 304 | 284 | 183
flat300-20-0 | 300 | 21375 | 21 | 3150 | 160 | 15 | 0 | 90 | 76 | 20
flat300-26-0 | 300 | 21633 | 25 | 3966 | 158 | 12 | 0 | 92 | 86 | 36
flat300-28-0 | 300 | 21695 | 28 | 4330 | 162 | 12 | 0 | 95 | 90 | 53
fpsol2.i.1 | 496 | 11654 | 54 | 3403 | 252 | 307 | 0 | 159 | 76 | 75
fpsol2.i.2 | 451 | 8691 | 30 | 1668 | 346 | 261 | 0 | 188 | 49 | 46
fpsol2.i.3 | 425 | 8688 | 30 | 1636 | 346 | 238 | 0 | 188 | 49 | 46
huck | 74 | 301 | 11 | 243 | 53 | 27 | 0 | 32 | 18 | 16
inithx.i.1 | 864 | 18707 | 54 | 3676 | 502 | 566 | 0 | 278 | 75 | 72
inithx.i.2 | 645 | 13979 | 31 | 2050 | 541 | 365 | 0 | 286 | 53 | 48
inithx.i.3 | 621 | 13969 | 31 | 1986 | 542 | 360 | 0 | 287 | 52 | 48
jean | 80 | 254 | 10 | 217 | 36 | 38 | 0 | 23 | 17 | 15
lc450-15a | 450 | 8168 | 15 | 2740 | 99 | 75 | 45 | 57 | 68 | 53

Continued on next page...
Table 3: Comparison of our new algebraic bound (UB\textsubscript{a}) and algorithmic bound (UB\textsubscript{s}) with the bound of Hajiabolhassan et al. [9] (UB\textsubscript{hmt}).

The results in Table 3 shows that our algebraic bound (UB\textsubscript{a}) is already better than the results of Hajiabolhassan et al. given by Property 4 (UB\textsubscript{hmt}), except for some graphs with low degree (le\textsubscript{450}, queen). UB\textsubscript{a} gives a better lower bound for 42 instances among 74. However, when a maximum stable set can be found in reasonable time using IncMaxCLQ, our algorithmic bound (UB\textsubscript{s}) is significantly better than UB\textsubscript{hmt} in general. UB\textsubscript{s} gives a better
lower bound for 66 instances among 74. For example, while \( \text{UB}_{hmt} \) for the three graphs \( \text{inithx} \) is 278, 286 and 287 respectively, \( \text{UB}_a \) is 72, 48 and 48 respectively (\( \text{UB}_a \) is 75, 53 and 52 respectively). Another example is the graph \( \text{le450-5d} \) for which \( \text{UB}_{hmt} = 37 \), while \( \text{UB}_a = 5 \). These results show the performance of our approach for reducing the number of colors to be considered when solving MSCP.

5. Conclusion

In this paper we focused on one component of the Minimum Sum Coloring Problem (MSCP), the chromatic strength \( s(G) \) of the graph \( G \). We have proposed two new upper bounds of \( s(G) \), called \( \text{UB}_a \) and \( \text{UB}_s \) respectively. \( \text{UB}_a \) and \( \text{UB}_s \) both use a known valid coloring \( X \) of the graph and explore a set of motifs representing an abstraction of all possible colorings of the graph. \( \text{UB}_a \) is obtained by identifying the number of colors from which a coloring better than \( X \) cannot be obtained.

Apart from \( X \), \( \text{UB}_a \) does not exploit any other structure property of the graph. \( \text{UB}_s \) is a more established upper bound. In order to determine \( \text{UB}_s \), we introduced a notion called major motif that exploits the dominance relation on the set of motifs. Indeed, such a motif represents the best sum coloring among all motifs with the same or more number of colors and the same or smaller first integer.

Computing \( \text{UB}_s \) consists in identifying a major motif whose the first integer is the cardinality of a maximum stable set of the graph, and whose the sum coloring is greater than the sum of \( X \). The maximum stable set is computed using the exact MaxClique algorithm IncMaxCLQ. Thus, we exclude the colorings that cannot be valid for the graph and the colorings that cannot be better than \( X \). \( \text{UB}_s \) is derived from the remaining colorings thanks to our algorithm \( \text{CUB}_s \).

We evaluated \( \text{UB}_a \) and \( \text{UB}_s \) on DIMACS and COLOR graphs. The experimental results show that \( \text{UB}_a \) is already better than the previous bounds except for some graphs with low degree. The algorithmic upper bound \( \text{UB}_s \), based on the major motif notion and a maximum stable set, outperforms generally all others bounds allowing to reduce substantially the search space when solving MSCP.

In the future, we plan to integrate more structural properties of a graph to further improve \( \text{UB}_s \), and to develop efficient algorithms to solve MSCP based on \( \text{UB}_s \).

Acknowledgements. This work is supported by the Ministry of Higher Education and Research, of the French state.

6. References


