Sum Coloring : New upper bounds for the chromatic strength

Clément Lecat, Corinne Lucet, Chu-Min Li

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 ressource 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 $\mathcal{N}$, is defined as : $\mathcal{N}(v) = \{u \mid (u, v) \in E\}$. The degree $d(v)$ of a vertex v is the number of its adjacent vertices, i.e., $d(v) = |\mathcal{N}(v)|$. 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 $c : V \mapsto \{1, 2, \ldots, k\}$ 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 $w_i = i$. 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,* $X = \{\{a, e\}_1, \{b\}_2, \{c, d, f\}_3\}$ *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 $\Sigma(G)$:

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

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

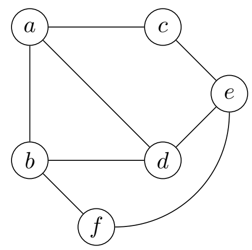

FIGURE 1: A simple graph

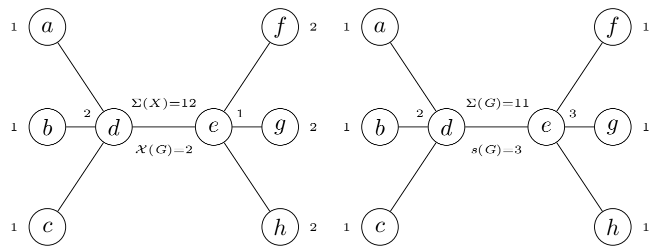

FIGURE 2: An optimal solution for GCP : X=$\{\{a,\ b,\ c,\ e\}_1,\ \{d,\ f,\ g,\ h\}_2\}$ with sum of color weights equal to 12, and an optimal solution for $MSCP$: X'=$\{\{a,\ b,\ c,\ f,\ g,\ h\}_1,\ \{d\}_2,\ \{e\}_3\}$ with sum of color weights equal to 11.

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 $\{\{a, e\}_1, \{b\}_2, \{c, d, f\}_3\}$ 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 : ($|X_1|$, $|X_2$, ..., $|X_k|$), 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 :

$$\Sigma(X) = \Sigma(p) = 1 \times p[1] + 2 \times p[2] + ... + k \times p[k] \tag{1}$$

Example 3. *The motif corresponding to $X = \{\{c, d, f\}_1, \{a, e\}_2, \{b\}_3\}$ 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 $\{c, d, f\}_1, \{b, e\}_2, \{a\}_3$ and $\{a, e, f\}_1, \{b, c\}_2, \{e\}_3$ 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 :

$$|\phi(n)| \sim \frac{1}{4n\sqrt{3}}\, e^{\pi\sqrt{2n/3}} \tag{2}$$

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

n	1	2	3	4	5	6	7	8	9	10	20	50	100	150
$\lvert\phi(n)\rvert$	1	2	3	5	7	11	15	22	30	42	697	204,226	190,569,292	40,853,235,313

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_4-sparse graphs, or an upper bound of the chromatic sum for general P_4-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 $p \succeq q$, 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 $\sum_{x=1}^{|p|} p[x] = \sum_{x=1}^{|q|} q[x] = n$, 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 $t = 2$: $\sum_{i=1}^{2} p[i] < \sum_{i=1}^{2} q[i]$ ($9+3 < 8+6$).
$p \not\succeq q$ and $q \not\succeq p$, i.e., p and q are not comparable.

So, the dominance relation $\succeq$ 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 $p \succeq q$ 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_k^i the i^{th} motif in $\phi(n, k)$.*

ϕ(n,k)	p$\in$ ϕ(n,k)	$\Sigma(p)$
ϕ(8,1)	(8)	8
ϕ(8,2)	(7,1)	9
	(6,2)	10
	(5,3)	11
	(4,4)	12
ϕ(8,3)	(6,1,1)	11
	(5,2,1)	12
	(4,3,1)	13
	(4,2,2)	14
	(3,3,2)	15
ϕ(8,4)	(5,1,1,1)	14
	(4,2,1,1)	15
	(3,3,1,1)	16
	(3,2,2,1)	17
	(2,2,2,2)	20
ϕ(8,5)	(4,1,1,1,1)	18
	(3,2,1,1,1)	19
	(2,2,2,1,1)	21
ϕ(8,6)	(3,1,1,1,1,1)	23
	(2,2,1,1,1,1)	24
ϕ(8,7)	(2,1,1,1,1,1,1)	29
ϕ(8,8)	(1,1,1,1,1,1,1,1)	36

TABLE 2: $\phi(8)$.

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

$$(n-k+1, \overbrace{1, 1, \ldots, 1}^{k-1\ times})$$

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

$$(\overbrace{\left\lceil \frac{n}{k} \right\rceil, \ldots, \left\lceil \frac{n}{k} \right\rceil}^{n\ mod\ k\ times}, \overbrace{\left\lfloor \frac{n}{k} \right\rfloor, \ldots, \left\lfloor \frac{n}{k} \right\rfloor}^{k\ -\ n\ mod\ k\ times})$$

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 *sup* that is $n-k+1$ if p is empty, $min(n-k+1, min(p))$ otherwise, and $\lceil \frac{n}{k} \rceil$, 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, $\emptyset$, $\emptyset$) (p=ϕ=$\emptyset$), the algorithm generates each first integer between 6 and 3. When the first integer is 6, the algorithm calls MOTIF(2, 2, {6}, $\emptyset$) 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.

Algorithme 1 : MOTIF(n, k, p, ϕ)

Input : two positive integers n and k such that $k \leq n$, motif p under construction, set of motifs ϕ under construction

Output : a set of motifs ϕ in decreasing lexicographic order

```
begin
    if k = 1 then
        φ ← φ ∪ {p ∪ {n}} ;
    else
        if p=∅ then
            sup ← n − k + 1 ;
        else
            sup ← min(n − k + 1, min(p)) ;
        inf ← ⌈n/k⌉ ;
        foreach x ← sup downto inf do
            Motif(n − x, k − 1, p ∪ {x}, φ) ;
```

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 :

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

Note that ψ is not empty, because it contains trivially the only motif $(1, 1, \ldots, 1)$ in $\phi(n, n)$ for $k_t = n$. Since ψ 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_s, 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 $1 \leq t \leq k$, $\sum_{x=t+1}^{k} p_k^1[x] \leq \sum_{x=t+1}^{k} q[x]$, 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^1=(n-k+1, 1, 1, ..., 1) dominates any motif in $\phi(n, k')$ such that $k \leq k'$. The sum coloring of p_k^1 is :

$$\Sigma(p_k^1) = (n - k + 1) + \sum_{x=2}^{k} x = \frac{1}{2}k^2 - \frac{1}{2}k + n \tag{4}$$

So UB_a is the smallest number max(k-1, $|X|$) of colors such that

$$\Sigma(p_k^1) \geq \Sigma(X) \tag{5}$$

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

$$\frac{1}{2}k^2 - \frac{1}{2}k + n - \Sigma(X) \geq 0 \tag{6}$$

Equation 6 is valid if and only if :

$$k \geq \frac{1 + \sqrt{1 + 8 \times (\Sigma(X) - n)}}{2}$$

Thus :

$$UB_a = max(\left\lceil \frac{1 + \sqrt{1 + 8 \times (\Sigma(X) - n)}}{2} \right\rceil - 1, |X|) \tag{7}$$

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_s, 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_s 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 λ denote the first integer of a motif. We have $\lceil \frac{n}{k} \rceil \leq \lambda \leq n-k+1$. Intuitively, a motif is major if it contains the maximum number (β) of integers equal to its first integer, λ. The remaining value of n (i.e. $n-\beta\times\lambda$) should be partitioned into $k-\beta$ positive integers. So β 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, …, 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 λ and β be two integers such that $\lceil \frac{n}{k} \rceil \leq \lambda \leq n-k+1$ and $\beta = \left\lfloor \frac{n-k}{\lambda-1} \right\rfloor$. A major motif in $\phi(n,k)$ is a motif p_k^i with the following properties :*

$$\begin{cases} p_k^i[x] = \lambda, \; if \;\; 1 \leq x \leq \beta; \\ p_k^i[x] = n-\beta\times\lambda-(k-\beta-1), \; if \;\; x=\beta+1; \\ p_k^i[x] = 1, \; if \;\; \beta+1 < x \leq k. \end{cases}$$

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 $1\leq t\leq\beta$, $\sum\limits_{x=1}^{t} p[x] \geq \sum\limits_{x=1}^{t} q[x]$.*

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, $\sum\limits_{x=1}^{t} p[x] = n-\sum\limits_{x=t+1}^{k} p[x] \geq n-\sum\limits_{x=t+1}^{k} q[x] = \sum\limits_{x=1}^{t} q[x]$.

Therefore, $p \succeq q$. □

Property 9 compares two majors 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 $1\leq t\leq\beta_p$, $\sum\limits_{x=1}^{t} p[x] > \sum\limits_{x=1}^{t} q[x]$.*

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 $\sum\limits_{x=1}^{t} p[x] = n-\sum\limits_{x=t+1}^{k} p[x] \geq n-\sum\limits_{x=t+1}^{k} q[x] = \sum\limits_{x=1}^{t} q[x]$. 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 $p_k^i \succeq p_k^j$.*

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 $\forall p_k^i \in \phi(n,k)$, $p_k^1 \succeq p_k^i$, because p_k^1 is major.

The following property compares the major motif p_k^1 to $p_{k'}^i$ 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 $\forall q \in \psi$, $p_k^1 \succeq q$.*

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

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 $\forall q \in \psi$ such that $q[1] \leq p_k^i[1]$, $p_k^i \succeq q$.*

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 $\sum_{x=1}^{t} p_k^i[x] = n - \sum_{x=t+1}^{k} p_k^i[x] \geq n - \sum_{x=t+1}^{k} q[x] = \sum_{x=1}^{t} q[x]$. Therefore, $p_k^i \succeq q$. □

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 $\mathrm{UB}_s = k-1$. Algorithm 2 implements UB_s. 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 ($\chi(G)$), Σ^* the best-known upper bound for the chromatic sum ($\Sigma(G)$), $\alpha(G)$ the cardinality of a maximum stable set of G, Time_s 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 $\Sigma(G)^*$.

Algorithme 2 : $\text{CUB}_s(n, \alpha(G), X)$

Input : the number of vertices n, the cardinality of a maximum stable set $\alpha(G)$, 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	$\lvert V\rvert$	$\lvert E\rvert$	k^*	Σ^*	$\Delta(G)$	$\alpha(G)$	Time_s	UB_{hmt}	UB_a	UB_s
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	41603	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	30313	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	20	3150	160	15	0	90	76	20
flat300-26-0	300	21633	26	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	65	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
games120	120	638	9	443	13	22	0	11	25	15
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
le450-15a	450	8168	15	2740	99	75	45	57	68	53

Continued on next page...

Graph	$\|V\|$	$\|E\|$	k^*	Σ^*	$\Delta(G)$	$\alpha(G)$	Time$_s$	UB_{hmt}	UB$_a$	UB$_s$
le450-15b	450	8169	15	2733	94	78	7	55	68	54
le450-15c	450	16680	15	3829	139	41	1602	77	82	57
le450-15d	450	16750	15	3751	138	41	2266	77	81	56
le450-25a	450	8260	25	3291	128	91	0	77	75	66
le450-25b	450	8263	25	3492	111	78	0	68	78	68
le450-25c	450	17343	25	4906	179	47	24	102	94	79
le450-25d	450	17425	25	4953	157	43	82	91	95	78
le450-5a	450	5714	5	1350	42	N/A	N/A	24	42	N/A
le450-5b	450	5734	5	1363	42	N/A	N/A	24	43	N/A
le450-5c	450	9803	5	1356	66	90	2	36	43	8
le450-5d	450	9757	5	1350	68	90	2	37	42	5
miles1000	128	3216	42	1690	86	8	0	64	56	48
miles1500	128	5198	73	3354	106	5	0	90	80	77
miles250	128	387	8	325	16	44	0	12	20	14
miles500	128	1170	20	712	38	18	0	29	34	26
miles750	128	2113	31	1179	64	12	0	48	46	38
mulsol.2	188	3885	31	1191	156	90	0	94	45	44
mulsol.i.1	197	3925	49	1957	121	100	0	85	59	59
mulsol.i.3	184	3916	31	1187	157	86	0	94	45	44
mulsol.i.4	185	3946	31	1189	158	86	0	95	45	44
mulsol.i.5	186	3973	31	1160	159	88	0	95	44	43
myciel3	11	20	4	21	5	5	0	5	5	4
myciel4	23	71	5	45	11	11	0	8	7	6
myciel5	47	236	6	93	23	23	0	15	10	8
myciel6	95	755	7	189	47	47	0	27	14	11
myciel7	191	2360	8	381	95	95	0	52	20	15
queen10-10	100	1470	10	553	35	10	0	23	30	12
queen11-11	121	1980	11	730	40	11	0	26	35	13
queen12-12	144	2596	12	940	43	12	0	28	40	14
queen13-13	169	3328	13	1190	48	13	0	31	45	16
queen14-14	196	4186	14	1478	51	14	0	33	51	17
queen15-15	225	5180	15	1811	56	15	0	36	56	19
queen16-16	256	6320	16	2190	59	16	0	38	62	20
queen5-5	25	160	5	75	16	5	0	11	10	5
queen6-6	36	290	6	138	19	6	0	13	14	10
queen7-7	49	476	7	196	24	7	0	16	17	7
queen8-12	96	1368	12	624	32	8	0	22	33	12
queen8-8	64	728	9	291	27	8	0	18	21	10
queen9-9	81	1056	10	409	32	9	0	21	26	11
school1	385	19095	14	2674	282	41	2	148	68	45
school1-nsh	352	14612	14	2392	232	39	0	123	64	43
zeroin.i.1	211	4100	49	1822	111	120	0	80	57	56
zeroin.i.2	211	3541	30	1004	140	127	0	85	40	39
zeroin.i.3	206	3540	30	998	140	123	0	85	40	39

Table 3: Comparison of our new algebraic bound (UB$_a$) and algorithmic bound (UB$_s$) with the bound of Hajiabolhassan et al. [9] (UB$_{hmt}$).

The results in Table 3 shows that our algebraic bound (UB$_a$) is already better than the results of Hajiabolhassan et al. given by Property 4 (UB$_{hmt}$), except for some graphs with low degree (*le*450, *queen*). UB$_a$ 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$_s$) is significantly better than UB$_{hmt}$ in general. UB$_s$ gives a better lower bound for 66 instances among 74. For example, while UB_{hmt} for the three graphs *inithx* is 278, 286 and 287 respectively, UB_s is 72, 48 and 48 respectively (UB_a is 75, 53 and 52 respectively). Another example is the graph *le450-5d* for which $\text{UB}_{hmt} = 37$, while $\text{UB}_s = 5$. 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 UB_a and UB_s respectively. UB_a and 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. UB_a is obtained by identifying the number of colors from which a coloring better than X cannot be obtained.

Apart from X, UB_a does not exploit any other structure property of the graph. UB_s is a more established upper bound. In order to determine 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 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. UB_s is derived from the remaining colorings thanks to our algorithm CUB_s.

We evaluated UB_a and UB_s on DIMACS and COLOR graphs. The experimental results show that UB_a is already better than the previous bounds except for some graphs with low degree. The algorithmic upper bound 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 UB_s, and to develop efficient algorithms to solve $MSCP$ based on UB_s.

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

6. References

[1] Amotz Bar-Noy, Mihir Bellare, Magnús M Halldórsson, Hadas Shachnai, and Tami Tamir. On chromatic sums and distributed resource allocation. *Information and Computation*, 140(2) :183–202, 1998.

[2] Amotz Bar-Noy and Guy Kortsarz. Minimum color sum of bipartite graphs. *Journal of Algorithms*, 28(2) :339–365, 1998.

[3] Una Benlic and Jin-Kao Hao. A study of breakout local search for the minimum sum coloring problem. In *Simulated Evolution and Learning*, pages 128–137. Springer, 2012.

[4] Flavia Bonomo and Mario Valencia-Pabon. Minimum sum coloring of p 4-sparse graphs. *Electronic Notes in Discrete Mathematics*, 35 :293–298, 2009.

[5] Flavia Bonomo and Mario Valencia-Pabon. On the minimum sum coloring of p 4-sparse graphs. *Graphs and Combinatorics*, 30(2) :303–314, 2014.

[6] Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4) :251–256, 1979.

[7] ftp ://dimacs.rutgers.edu/pub/challenge/graph/ benchmarks/color/.

[8] Michael R Garey and David S Johnson. A guide to the theory of np-completeness. *WH Freemann, New York*, 1979.

[9] Hossein Hajiabolhassan, Mojtaba L Mehrabadi, and Ruzbeh Tusserkani. Minimal coloring and strength of graphs. *Discrete Mathematics*, 215(1) :265–270, 2000.

[10] Godfrey H Hardy and Srinivasa Ramanujan. Asymptotic formulaæ in combinatory analysis. *Proceedings of the London Mathematical Society*, 2(1) :75–115, 1918.

[11] Alain Hertz, Matthieu Plumettaz, and Nicolas Zufferey. Variable space search for graph coloring. *Discrete Applied Mathematics*, 156(13) :2551–2560, 2008.

[12] http ://mat.gsia.cmu.edu/COLOR/instances.html.

[13] Klaus Jansen. The optimum cost chromatic partition problem. In *Algorithms and complexity*, pages 25–36. Springer, 1997.

[14] Klaus Jansen. Approximation results for the optimum cost chromatic partition problem. *Journal of Algorithms*, 34(1) :54–89, 2000.

[15] Yan Jin, Jin-Kao Hao, and Jean-Philippe Hamiez. A memetic algorithm for the minimum sum coloring problem. *Computers & Operations Research*, 43 :318–327, 2014.

[16] Ewa Kubicka and Allen J Schwenk. An introduction to chromatic sums. In *Proceedings of the 17th conference on ACM Annual Computer Science Conference*, pages 39–45. ACM, 1989.

[17] Clément Lecat, Chu-Min Li, Corinne Lucet, and Yu Li. Exact methods for the minimum sum coloring problem. In *Doctoral Program of Constraint Programming (CD-DP'15)*, pages 61–69, 2015.

[18] Chu-Min Li, Zhiwen Fang, and Ke Xu. Combining maxsat reasoning and incremental upper bound for the maximum clique problem. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 939–946. IEEE, 2013.

[19] Yu Li, Corinne Lucet, Aziz Moukrim, and Kaoutar Sghiouer. Greedy algorithms for the minimum sum coloring problem. In *Logistique et transports*, pages LT–027, 2009.

[20] Zhipeng Lü and Jin-Kao Hao. A memetic algorithm for graph coloring. *European Journal of Operational Research*, 203(1) :241–250, 2010.

[21] Corinne Lucet, Florence Mendes, and Aziz Moukrim. An exact method for graph coloring. *Computers & operations research*, 33(8) :2189–2207, 2006.

[22] Michal Malafiejski. *Sum coloring of graphs*, chapter 4, pages 55–66. Graph Coloring. 2004.

[23] Isabel Méndez-Díaz and Paula Zabala. A branch-and-cut algorithm for graph coloring. *Discrete Applied Mathematics*, 154(5) :826–847, 2006.

[24] John Mitchem and Patrick Morriss. On the cost-chromatic number of graphs. *Discrete Mathematics*, 171(1) :201–211, 1997.

[25] Aziz Moukrim, Kaoutar Sghiouer, Corinne Lucet, and Yu Li. Lower bounds for the minimal sum coloring problem. *Electronic Notes in Discrete Mathematics*, 36 :663–670, 2010.

[26] Daniel Cosmin Porumbel, Jin-Kao Hao, and Pascale Kuntz. A search space âĂIJcartographyâĂİ for guiding graph coloring heuristics. *Computers & Operations Research*, 37(4) :769–778, 2010.

[27] Mohammad R Salavatipour. On sum coloring of graphs. *Discrete Applied Mathematics*, 127(3) :477–488, 2003.

[28] Kaoutar Sghiouer, Li Yu, Corinne Lucet, and Aziz Moukrim. A memetic algorithm for the minimum sum coloring problem. In *Conférence Internationale en Recherche Opérationnelle (CIRO'2010)*, 2010.

[29] Carsten Thomassen, Paul Erdös, Yousef Alavi, Paresh J Malde, and Allen J Schwenk. Tight bounds on the chromatic sum of a connected graph. *Journal of Graph Theory*, 13(3) :353–357, 1989.

[30] Zhaoyang Zhou, Chu-Min Li, Chong Huang, and Ruchu Xu. An exact algorithm with learning for the graph coloring problem. *Computers & Operations Research*, 51 :282–301, 2014.