Improved Lower Bounds for Sum Coloring via Clique Decomposition

Qinghua Wu and Jin-Kao Hao

LERIA, Université d’Angers
2 Boulevard Lavoisier, 49045 Angers Cedex 01, France

Abstract

Given an undirected graph \( G = (V, E) \) with a set \( V \) of vertices and a set \( E \) of edges, the minimum sum coloring problem (MSCP) is to find a legal vertex coloring of \( G \), using colors represented by natural numbers 1, 2, ... such that the total sum of the colors assigned to the vertices is minimized. This paper describes an approach based on the decomposition of the original graph into disjoint cliques for computing lower bounds for the MSCP. Basically, the proposed approach identifies and removes at each extraction iteration a maximum number of cliques of the same size (the largest possible) from the graph. Computational experiments show that this approach is able to improve on the current best lower bounds for 14 benchmark instances, and to prove optimality for the first time for 4 instances. We also report lower bounds for 24 more instances for which no such bounds are available in the literature. These new lower bounds are useful to estimate the quality of the upper bounds obtained with various heuristic approaches.

keywords: sum coloring; graph coloring; clique decomposition; bounds; heuristics

1 Introduction

Let \( G = (V, E) \) be an undirected graph with vertex set \( V \) and edge set \( E \). A clique is a subset of vertices \( C \subseteq V \) such that each pair of vertices in \( C \) are adjacent. An independent set is a subset of vertices, \( I \subseteq V \), such that no two vertices in \( I \) are adjacent. A legal \( k \)-coloring of \( G \) corresponds to a partition

* Corresponding author.

Email addresses: wu@info.univ-angers.fr (Qinghua Wu), hao@info.univ-angers.fr (Jin-Kao Hao).
of $V$ into $k$ independent sets (color classes) $I_1$, $I_2$, ..., $I_k$. The well-known graph coloring problem aims at finding the smallest $k$ for a given graph $G$ (its chromatic number $\chi(G)$) such that $G$ has a legal $k$-coloring. In this paper, we are interested in a related problem known as the minimum sum coloring problem (MSCP for short) \[17\].

The objective of MSCP is to find a legal coloring $c = \{I_1, I_2, ..., I_k\}$ of the graph $G$, such that the following total sum of the colors is minimized:

$$Sum(c) = \sum_{i=1}^{k} \sum_{v \in I_i} i$$  \quad (1)$$

The optimal value for the MSCP is called the chromatic sum of $G$ and is denoted by $\Sigma(G)$. For any optimal solution, the associated number of required colors is termed as the strength of the graph, and denoted by $s(G)$. Clearly, the chromatic number is a lower bound of chromatic sum, i.e., $s(G) \geq \chi(G)$.

Efficient approximation algorithms or polynomial algorithms exist for specific graph classes (e.g., tree, interval graphs, line graphs etc.) \[1, 2, 3, 9, 11, 12, 14, 18, 21, 23\]. However, for the general case, the MSCP, more precisely its decision version, is known to be NP-complete \[17\] and thus computationally hard. The MSCP also arises in a variety of real-world applications including those from VLSI design, scheduling and resource allocation \[18, 21\].

Given the importance of the MSCP, a number of heuristics and metaheuristics have been devised to find approximate solutions (upper bounds) of good quality with a reasonable computing time. This includes greedy algorithm based on the well-known RLF graph coloring heuristic \[19\], local search algorithms \[10\], tabu search \[4\], parallel genetic algorithm \[15\], hybrid algorithm \[5\] and heuristics based on independent set extraction \[1, 26\]. These approaches produce upper bounds to this minimization problem. Lower bounds are useful to assess the quality of these solutions.

Recently, several studies have been devoted to determining lower bounds for the MSCP. Many of them are based on a general approach that decomposes the original graph into specific graphs like trees \[14\], paths \[22\] or cliques \[6, 10, 22\]. Such a decomposition produces a partial graph to the original graph $G$, whose associated chromatic sum can be efficiently computed and leads naturally to a lower bound to the chromatic sum of the original graph. In \[22\], A. Moukrim et al. show that the clique decomposition provides better lower bounds than other graph decompositions like trees and paths. To obtain a clique decomposition of the original graph $G$, a simple method is to determine with a vertex coloring algorithm a legal coloring of the complementary graph $\overline{G}$ of the original graph $G$. Since an independent set (i.e., a color class) of $\overline{G}$
corresponds to a clique in \( G \), a coloring of \( \overline{G} \) defines a clique decomposition of \( G \).

This graph coloring based approach for determining MSCP lower bounds has been exploited in studies like [6, 10, 22]. These studies differ mainly by their way of coloring the complementary graph. In [22], the coloring is achieved with an adjusted greedy algorithm MRLF, while an ant colony optimization algorithm is used in [6] and a local search heuristic is employed in [10] to color \( \overline{G} \). Finally, some theoretical lower bounds for the MSCP are proposed in [15, 16].

In this paper, we present a heuristic approach (denoted by EXCLIQUE) based on a direct clique decomposition of \( G \) for computing the lower bounds for the general MSCP. The proposed approach tries to extract as many large cliques as possible from the graph. Since more colors and larger color numbers are needed to color the vertices of a large clique than to color the vertices of a small clique, a clique decomposition with more and large cliques tends to increase the chromatic sum of the resulting clique decomposition and therefore leads to a tighter lower bound to the original graph. To achieve this, we follow a similar approach which has been successfully used to establish upper bounds for the vertex coloring problem [25] and the MSCP [26]. At each iteration of the proposed approach, a maximum number of pairwise disjoint cliques of the largest possible size are identified and removed from the graph. This process is repeated until the graph becomes empty. The proposed approach is assessed on a set of 62 DIMACS and COLOR2 benchmark graphs in the literature. The computational outcomes show that this approach is able to improve on the current best lower bounds in 14 cases out of 38 instances with known lower bounds reported in the literature, and to prove optimality for the first time for 4 instances. Lower bounds are also reported for the first time for the remaining 24 instances for which no lower bounds are available in the literature.

The rest of the paper is organized as follows. In section 2, we explain the clique decomposition approach for computing lower bounds for the MSCP. In section 3, our proposed heuristic approach is presented. In section 4, we provide computational results and comparisons on a set of 62 benchmark graphs from the literature. In section 5, we show additional studies about the effect of the disjoint clique removal strategy, followed by conclusions in section 6.

2 Lower bounds for the MSCP based on clique decomposition

Given \( G = (V, E) \), a partial graph of \( G = (V, E) \) is a graph \( G' = (V, E') \) such that \( E' \) is a subset of \( E \). It is easy to observe that the chromatic sum of \( G' \) is a lower bound for the chromatic sum of \( G \). Indeed, any legal coloring of \( G \) is a
legal coloring of $G'$ while the reverse does not hold. Thus, to calculate a lower bound for the chromatic sum of the original graph $G$, one could try to find a partial graph of the original graph whose chromatic sum can be efficiently computed and maximized.

This can be achieved by decomposing the vertex set of $G$ into $k$ pairwise disjoint cliques $C_1, C_2,..., C_p$ such that $\forall i \neq j, C_i \cap C_j = \emptyset$ and $\bigcup_i C_i = V$ (see Fig. 1 for an illustrative example). Given a clique decomposition $F = \{C_1, C_2, ..., C_p\}$, each of its cliques $C_i$ ($i = 1..p$) needs exactly $|C_i|$ colors: $1..|C_i|$. Consequently, the chromatic sum of $F$ is given by $\sum_{i=1}^{p} \frac{|C_i|(|C_i|+1)}{2}$. Since a clique decomposition $F = \{C_1, C_2, ..., C_p\}$ is a partial graph of $G$ and the chromatic sum of $F$ is therefore a lower bound to the chromatic sum $\Sigma(G)$ of $G$.

The quality of this lower bound depends on the way to decompose the graph into cliques. For instance, consider the graph $G = (V, E)$ of Fig. 1(a), we decompose $G$ in two different ways (Fig. 1(b) and Fig. 1(c)), we obtain the following chromatic sum $\Sigma(G_1') = 12$ and $\Sigma(G_2') = 14$. Thus, in order to obtain a lower bound as tight (large) as possible, one could try to find a clique decomposition of $G$ whose chromatic sum is as large as possible. This can be considered as an optimization problem where we search for a decomposition of $G$ into cliques, such that the associated chromatic sum is maximized over all the possible clique decompositions:

$$LB^* = \max \{ \sum (F) \mid F \text{ is a clique decomposition of } G \} \quad (2)$$

It should be noted that $LB^*$ might be strictly smaller than $\Sigma(G)$ because, a clique decomposition, by ignoring some edges of the original graph $G$, is a less constrained problem for coloring.

![Fig. 1. Partial graphs of $G$ via clique decomposition](image)

This clique decomposition approach for computing lower bounds for the MSCP was originally proposed in [22] and were further exploited in [6,10]. In these
previous studies, clique decompositions are obtained by applying a vertex coloring algorithm to color the complementary graph $\overline{G}$ and to use the color classes to define a clique decomposition of $G$.

In this work, we present a heuristic approach (denoted by EXCLIQUE) based on a direct clique decomposition of the original graph $G$ to approximate $LB^*$. In order to obtain a clique decomposition which maximizes its chromatic sum, the proposed approach tries to extract as many large cliques as possible from the original graph by employing a recent maximum clique algorithm presented in [24].

3 EXCLIQUE: an algorithm for improved lower bounds of MSCP

3.1 Rationale and general procedure

Given a clique decomposition $F = \{C_1, C_2, ..., C_k\}$ of $G = (V, E)$, there is one unique coloring for $F$, i.e., for each clique $C_i \ (1 \leq i \leq k)$ of $F$, we need exactly $|C_i|$ colors to color its vertices. Moreover, the larger the clique $C_i$ is, the more the number of vertices in $C_i$ that need to be colored with large colors. Thus, a clique decomposition with more large cliques tends to have a larger chromatic sum and therefore gives a better lower bound to $G$. For instance, consider again the example of Fig. 1. Vertex A belongs respectively to a clique of size 2 (Fig.1(b)) and a clique of size 4 (Fig.1(c)). The later case increases the sum of colors by 2 because we need a large color (4) for the clique of size 4.

For the purpose of obtaining a clique decomposition with more large cliques, one could try to identify as many large cliques as possible from the graph. To achieve this, we can iteratively identify and remove the maximum number of disjoint cliques of the maximum size from the graph until the graph becomes empty. Our proposed EXCLIQUE algorithm follows the above idea and can be summarized as follows:

1. Identify a maximum clique $C$ in $G$;
2. Collect in a set $M$ as many other cliques of size $|C|$ as possible;
3. Find from $M$ a maximum number of disjoint cliques $C_1, ..., C_t$;
4. Remove $C_1, ..., C_t$ from $G$;
5. Repeat the above steps until the graph becomes empty;

In step 1 of the above process, one needs to identify a maximum clique. Notice that finding a maximum clique in a graph is an NP-hard problem in the general case [7]. For this reason, we use the so-called adaptive tabu search heuristic (denoted by ATS) designed for the maximum clique problem to find
large cliques [24]. The same heuristic is also employed to build the pool $M$ composed of cliques of a given size (step 2) and identify the set of disjoint cliques (step 3).

In step 2, in order to build the pool $M$ of cliques of size $|C|$, we apply repeatedly ATS to generate as many cliques of size $|C|$ as possible and put them in set $M$. The search for a new clique of size $|C|$ stops when the number of cliques contained in $M$ reaches a desired threshold ($M_{\text{max}}$) or when no new clique of size $|C|$ is found after $p_{\text{max}}$ consecutive tries.

In step 3, it is required to find among the candidates of $M$ a maximum number of disjoint cliques. This task corresponds in fact to the maximum set packing problem, which is equivalent to the maximum clique problem [7]. To see this, it suffices to construct an instance of the maximum clique problem from $M = \{C_1, ..., C_n\}$ as follows. Define a new graph $G' = (V', E')$ where $V' = \{1, ..., n\}$ and $\{i, j\} \in E'$ (i, j $\in V'$) if and only if $C_i$ and $C_j$ share no common element, i.e., $C_i \cap C_j = \emptyset$. Now it is clear that there is a strict equivalence between a clique in $G' = (V', E')$ and a set of disjoint cliques in $M$. Consequently, one can apply again the ATS algorithm to approximate this problem.

**Fig. 2.** An illustration of the proposed EXCLIQUE algorithm

Fig. 2 illustrates how our EXCLIQUE approach works on a graph with 9 vertices. At the first step, we find a maximum clique of size 3 from the graph (e.g., $\{D, E, F\}$). Then we try to collect as many cliques of size 3 as possible in $M$, leading to a pool containing 6 cliques of size 3: $\{A, E, I\}$, $\{B, C, D\}$, $\{B, G, H\}$, $\{D, E, F\}$, $\{F, H, G\}$ and $\{G, F, H\}$. From these cliques, a set of 3 disjoint cliques $\{A, E, I\}$, $\{B, C, D\}$, $\{G, F, H\}$ is identified. Since the graph becomes empty after removing these 3 cliques, the procedure stops. We obtain a clique decomposition $F = \{\{A, E, I\}, \{B, C, D\}, \{G, F, H\}\}$ with a chromatic sum value equal to 18.
Note that if the first clique \( \{D, E, F\} \) would have been removed once it was identified, then all the resulting clique decompositions would have a chromatic sum inferior to 18, leading to a worse lower bound for graph \( G \).

4 Experimental results

To assess the efficiency of the proposed EXCLIQUE approach, we conduct experiments on two sets of benchmark instances from the literature. The first set is composed of 29 well-known DIMACS graphs\(^1\), which are very popular for testing graph coloring algorithms\(^2\). The second set of benchmarks is composed of 33 graphs from the COLOR02 competition\(^3\). In addition, we also assess the interest of the basic extraction method which extracts maximum cliques one by one and the pertinence of applying directly powerful graph vertex coloring algorithms on the complementary graph of the original graph.

Our algorithm is implemented in C++, and compiled with GNU gcc on an Intel Xeon E5440 processor with 2.83 GHz and 8GB RAM. To obtain our computational results, each instance is solved 20 times independently with different random seeds (the two very large instances Cxxxx.5 are solved 5 times). EXCLIQUE stops when the graph under consideration becomes empty.

The two main parameters for EXCLIQUE are \( M_{\text{max}} \) and \( p_{\text{max}} \). Obviously, larger values for \( M_{\text{max}} \) and \( p_{\text{max}} \) could include more cliques in \( M \), thus giving a higher chance of finding more pairwise disjoint clique sets of the largest possible size in \( M \). On the other hand, larger values for both parameters also imply longer computing time. According to our experiments, we have fixed \( p_{\text{max}} = 100 \) and \( M_{\text{max}} = 2000 \) for all our experiments. In addition to \( M_{\text{max}} \) and \( p_{\text{max}} \), ATS requires also several parameters. In our case, we adopt those used in the original paper\(^4\).

4.1 Improved lower bounds for the MSCP

Tables\(^5\) and\(^6\) respectively show the computational statistics of the EXCLIQUE algorithm on the DIMACS benchmarks and the COLOR02 instances. In both tables, columns 2–4 indicate the features of the tested graphs, including the number of vertices (\(|V|\)), the number of edges (\(|E|\)) and the density of the graph (\(\text{Den}\)). Columns 5 and 6 give respectively the current best upper and lower bounds reported in the literature 1\(^4\), 5\(^5\), 6\(^6\), 10\(^10\), 15\(^15\), 19\(^19\), 22\(^22\), 26\(^26\). The results of

\(^1\)ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/color/
\(^2\)http://mat.gsia.cmu.edu/COLOR02/
Table 1

Computational results of EXCLIQUE on 29 DIMACS challenge benchmarks. The symbol - means that the information is not available. Improved bounds or new bounds are indicated in bold.

| Instance      | |V|  | |E|  | Den | UB* | LB* | EXCLIQUE |
|---------------|----------------|------------|------|-----|------|------|-----|------|--------|
|               | LB             | Avg. (Std.) | T (sec) |
| DSJC125.1     | 125            | 736        | 0.09 | 326 | 238 | 246 | 244.10 (0.94) | 80 |
| DSJC125.5     | 125            | 3891       | 0.50 | 1015 | 504 | 536 | 522.40 (5.46) | 35 |
| DSJC125.9     | 125            | 6961       | 0.89 | 2511 | 1621 | 1664 | 1592.50 (26.61) | 47 |
| DSJC250.1     | 250            | 3218       | 0.10 | 977 | 537 | 567 | 561.95 (1.96) | 46 |
| DSJC250.5     | 250            | 15668      | 0.50 | 3246 | 1150 | 1270 | 1258.80 (5.44) | 37 |
| DSJC250.9     | 250            | 27897      | 0.90 | 8286 | 3972 | 4179 | 4082.40 (51.66) | 158 |
| DSJC500.1     | 500            | 12458      | 0.10 | 2850 | 1163 | 1250 | 1246.55 (1.37) | 1269 |
| DSJC500.5     | 500            | 62624      | 0.50 | 10910 | 2616 | 1270 | 1292.60 (5.44) | 60 |
| DSJC500.9     | 500            | 112437     | 0.90 | 29912 | 10074 | 1258 | 10734.50 (74.30) | 276 |
| DSJC1000.1    | 1000           | 49629      | 0.10 | 9003 | 2762 | 1250 | 1246.55 (1.37) | 1269 |
| DSJC1000.5    | 1000           | 249826     | 0.50 | 37598 | 5787 | 1270 | 2902.60 (11.94) | 60 |
| DSJC1000.9    | 1000           | 449449     | 0.90 | 103464 | 23863 | 1250 | 10734.50 (74.30) | 276 |
| flat300       | 20             | 0          | 300  | 21375 | -   | 1524 | 1505.65 (6.78) | 35 |
| flat300.26.0  | 300            | 21633      | 0.48 | 966  | -   | 1524 | 1511.40 (8.40) | 34 |
| flat300.28.0  | 300            | 21695      | 0.48 | 4822 | -   | 1532 | 1515.25 (7.81) | 43 |
| flat1000.50.0 | 1000           | 245000     | 0.49 | 25500 | -   | 1601 | 6571.80 (15.54) | 118 |
| flat1000.60.0 | 1000           | 245830     | 0.49 | 30100 | -   | 1601 | 6600.50 (18.01) | 414 |
| flat1000.76.0 | 1000           | 246708     | 0.49 | 37167 | -   | 1601 | 6583.15 (17.53) | 98 |
| le450.15a     | 450            | 8168       | 0.08 | 2642 | -   | 2329 | 2313.65 (15.32) | 252 |
| le450.15b     | 450            | 8169       | 0.08 | 2642 | -   | 2329 | 2315.65 (15.05) | 600 |
| le450.15c     | 450            | 16680      | 0.17 | 3866 | -   | 2329 | 2545.30 (24.67) | 187 |
| le450.15d     | 450            | 16750      | 0.17 | 3921 | -   | 2329 | 2572.40 (24.13) | 175 |
| le450.25a     | 450            | 8260       | 0.08 | 3153 | -   | 2329 | 2964.40 (28.08) | 967 |
| le450.25b     | 450            | 8263       | 0.08 | 3366 | -   | 2329 | 3304.10 (0.70) | 1550 |
| le450.25c     | 450            | 17343      | 0.17 | 4515 | -   | 2329 | 3597.10 (11.82) | 689 |
| le450.25d     | 450            | 17425      | 0.17 | 4544 | -   | 2329 | 3627.35 (45.33) | 850 |
| latin.sqr.10 | 900            | 307350     | 0.76 | 42223 | -   | 40950 | 40950 (0.00) | 15 |
| C2000.5       | 2000           | 999836     | 0.50 | 132515 | -   | 15091 | 15077.60 (11.74) | 3994 |
| C4000.5       | 4000           | 400268     | 0.50 | 473234 | -   | 15091 | 33018.80 (11.42) | 14413 |

Concerning the 29 DIMACS instances, from Table 1 we observe that the results obtained by EXCLIQUE are very competitive when compared to the current best lower bounds reported in the literature. Indeed, for the 12 random DSJC graphs with a known lower bound, we managed to improve on the current best bounds in all the cases. For the remaining 17 DIMACS graphs,
we report the computational statistics for the lower bounds for the first time. Finally, Table 1 also discloses that for most of these DIMACS instances, the gaps between the best known upper bounds and our best lower bounds are still large.

Concerning the 33 COLOR02 instances, from Table 2 we observe that our
EXCLIQUE algorithm is able to improve the current best known lower bounds in the literature for 2 instances (fpsol2.i.1 and inithx.i.1) while equaling the best known lower bounds for 22 instances. Only for 2 instances (david and miles500), EXCLIQUE obtains a worse result. For the other 7 COLOR02 instances, we report for the first time lower bounds. Table 2 also indicates that we are able to prove optimality for 10 instances. Among these 10 instances, the optimality of 4 instances (qg.order30, qg.order40, fpsol2.i.1 and inithx.i.1) is proven for the first time.

4.2 Comparison with three state-of-the-art approaches

In this section, we compare EXCLIQUE with 3 recent state-of-the-art algorithms for computing lower bounds for the MSCP from the literature: MDS(5)+LS [10], RMDS(n) [22] and ANT [6]. All these reference algorithms are based on the clique decomposition method, which consists in transforming the original graph to its complement and then coloring the complementary graph with different coloring algorithms. First, we recall the basic experimental conditions (when they are available) used by these reference methods.

- MDS(5)+LS. The results of MDS(5)+LS were based on an Intel Core i7 processor 2.93Ghz with 8192MB cache L2 and 4 GB RAM. For each instance, MDS(5)+LS was run only 1 time with a time limit of 1 hour.
- RMDS(n). RMDS(n) was run on an Intel Core 2 Duo T5450 1.66-1.67 with 2GB Ram. For each instance, RMDS(n) repeats the MRLF algorithm [19] n times (n = |V|) starting from a different vertex for each replicate.
- ANT. ANT was run on an AMD Athlon(tm)X2 dual-core QL-65 (2cpus) 2.1GHz PC with 4GB RAM. For each instance, 20 runs of ANT were performed.

In Table 3 we show the best lower bounds obtained by EXCLIQUE in comparison with the other three approaches on 38 benchmark instances (results of the reference algorithms are not available for other instances). The results of MDS(5)+LS and RMDS(n) are directly extracted from [10]. From Table 3 we observe that our EXCLIQUE algorithm competes favorably with these 3 algorithms. Indeed, in comparison with each of these 3 algorithms, EXCLIQUE is able to achieve tighter lower bounds for at least 9 instances (indicated in bold) and only for at most 2 instances, EXCLIQUE’s result is worse than that of these reference algorithms (indicated in italics). These comparative results confirm the effectiveness of the proposed approach to deliver improved lower bounds for the MSCP.
Table 3
Comparison of lower bounds on the 38 instances reported in [10]. The symbol -
means that the related statistics are not available. The best $LB$ values are high-
lighted in bold.

| Instance      | $LB^*$ | Lower bounds             |
|---------------|--------|--------------------------|
|               |        | EXCLIQUE | MDS(5)+LS | RMDS(n)  | ANT       |
|               |        |        |        |        |           |
|               |        |        |        |        |           |
| DSJC125.1     | 238    | 246     | 238    | 238    | -         |
| DSJC125.5     | 504    | 536     | 493    | 504    | -         |
| DSJC125.9     | 1621   | 1664    | 1621   | 1600   | -         |
| DSJC250.1     | 537    | 567     | 521    | 537    | -         |
| DSJC250.5     | 1150   | 1270    | 1128   | 1150   | -         |
| DSJC250.9     | 3972   | 4179    | 3779   | 3972   | -         |
| DSJC500.1     | 1163   | 1250    | 1143   | 1163   | -         |
| DSJC500.5     | 2616   | 2921    | 2565   | 2616   | -         |
| DSJC500.9     | 10074  | 10881   | 9731   | 10074  | -         |
| DSJC1000.1    | 2499   | 2762    | 2456   | 2499   | -         |
| DSJC1000.5    | 5787   | 6708    | 5660   | 5787   | -         |
| DSJC1000.9    | 23863  | 26557   | 23208  | 23863  | -         |
| myciel3       | 16     | 16      | 16     | 16     | 16        |
| myciel4       | 34     | 34      | 34     | 34     | 34        |
| myciel5       | 70     | 70      | 70     | 70     | 70        |
| myciel6       | 142    | 142     | 142    | 142    | 142       |
| myciel7       | 286    | 286     | 286    | 286    | 286       |
| anna          | 273    | 273     | 273    | 272    | 272       |
| david         | 234    | 229     | 234    | 234    | 234       |
| luck          | 243    | 243     | 243    | 243    | 243       |
| jean          | 216    | 216     | 216    | 216    | 216       |
| queen5.5      | 75     | 75      | 75     | 75     | 75        |
| queen6.6      | 126    | 126     | 126    | 126    | 126       |
| queen7.7      | 196    | 196     | 196    | 196    | 196       |
| queen8.8      | 288    | 288     | 288    | 288    | 288       |
| games120      | 442    | 442     | 442    | 442    | 442       |
| miles250      | 318    | 318     | 318    | 318    | 318       |
| miles500      | 686    | 677     | 686    | 677    | 677       |
| 2-Insertions  | 55     | 55      | 55     | 55     | 55        |
| 3-Insertions  | 84     | 84      | 84     | 84     | 84        |
| fpsol2.i.1    | 3402   | 3403    | 3151   | 3402   | 2590      |
| initlx.i.1    | 3581   | 3676    | 3486   | 3581   | 2801      |
| mug88_1       | 188    | 188     | 188    | 186    | 187       |
| mug88_25      | 186    | 186     | 186    | 183    | 185       |
| mug100_1      | 164    | 164     | 164    | 163    | 163       |
| mug100_25     | 162    | 162     | 162    | 161    | 162       |
| zeroin.i.2    | 1004   | 1004    | 1004   | 1004   | 1003      |
| zeroin.i.3    | 998    | 998     | 998    | 998    | 997       |
### Table 4
Comparison of EXCLIQUE with the graph coloring approach

| Instance    | LB\(^*\) | EXCLIQUE | MACOL |
|-------------|----------|----------|-------|
| DSJC125.1   | 238      | 246      | 245   |
| DSJC125.5   | 504      | 536      | 534   |
| DSJC125.9   | 1621     | 1664     | 1637  |
| DSJC250.1   | 537      | 567      | 567   |
| DSJC250.5   | 1150     | 1270     | 1266  |
| DSJC250.9   | 3972     | 4179     | 4062  |
| DSJC500.1   | 1163     | 1250     | 1257  |
| DSJC500.5   | 2616     | 2921     | 2916  |
| DSJC500.9   | 10074    | 10881    | 10830 |
| DSJC1000.1  | 2499     | 2762     | 2775  |
| DSJC1000.5  | 5787     | 6708     | 6545  |
| DSJC1000.9  | 23863    | 26557    | 25879 |

### 4.3 Comparison with the graph coloring approach

Using graph coloring methods to compute the lower bound of the MSCP is exploited in [6,10,22]. With this approach, a graph coloring algorithm is applied to the complement of the original graph \(G\) and the color classes of the coloring induces a clique decomposition of \(G\). In this section, we revisit this approach by employing one of the most recent and effective coloring algorithm (i.e., Memetic Coloring Algorithm–MACOL [20]) and show a comparison with the EXCLIQUE approach.

For this purpose, we consider the 12 DSJC random graphs and run MACOL 20 times to solve each instance. The results of EXCLIQUE are extracted from Table 1. The comparative results between MACOL and EXCLIQUE are summarized in Table 4.

From Table 4, we observe that EXCLIQUE has a globally better performance than MACOL. Indeed, for 9 out of the 12 instances, EXCLIQUE is able to reach a larger best lower bound than MACOL while the reverse is true only for two cases. When it comes to the the average results, the differences between the two approaches are less pronounced. As to the the computing times, it is clear that EXCLIQUE dominates largely MACOL, requiring at most one third of the time required by MACOL. Finally, it is interesting to observe that even if MACOL is inferior to EXCLIQUE, its bounds are still much better than the previous bounds that were obtained with other (less powerful) coloring algorithms.
5 Analysis of the effect of the disjoint clique removal

Different methods can be applied to extract cliques from the graph, one simple and conventional method (denoted by OBOCLIQUE) is to extract at each time exactly one maximum clique until the graph becomes empty. Compared to this basic method, our EXCLIQUE algorithm, which uses a heuristic method to extract at each iteration as many disjoint cliques as possible of the largest possible size, is able to identify more large cliques contained in the final decomposition, thus leading to a larger chromatic sum.

We present in this section computational evidences to show the advantage of our disjoint clique removal approach (EXCLIQUE) over the one-by-one clique removal approach (OBOCLIQUE). For OBOCLIQUE, we apply repetitively the ATS clique algorithm [24] to extract one largest possible clique until the graph becomes empty. We consider again the 12 DIMACS random DSJC graphs and report in Table 5 a detailed comparison between these two methods. For each instance, we run OBOCLIQUE 20 times and the following statistics are provided: the best lower bounds, the averaged lower bound and the average CPU time in seconds. The results of EXCLIQUE are extracted from Table 1.

Table 5 discloses that EXCLIQUE dominates OBOCLIQUE in terms of solution quality. Indeed, for each of these 12 instances, EXCLIQUE is able to find a larger lower bound than OBOCLIQUE. To get some insights about the performance difference, we show in Table 6 the number of cliques of different sizes extracted by both methods on the instance DSJC1000.5. We see clearly that compared to the OBOCLIQUE method, our EXCLIQUE method is able to extract more larger cliques (leading to a larger chromatic sum, thus better lower bound).

It is also interesting to observe that even if OBOCLIQUE can not compete with EXCLIQUE, the results of OBOCLIQUE remain very competitive compared to the current best lower bounds reported in the literature. This observation highlights the interest of the general clique extraction approach for computing the lower bounds for the MSCP when an effective maximum clique like ATS is employed.

Finally, due to the fact that EXCLIQUE may need to call the ATS clique algorithm many times at each extraction step (only one call per extraction for OBOCLIQUE), EXCLIQUE requires naturally more computing times than OBOCLIQUE.
Table 5
Comparison of clique extraction methods: disjoint clique removal (EXCLIQUE) v.s. one-by-one removal (OBOCLIQUE)

| Instance | LB* | EXCLIQUE | OBOCLIQUE |
|----------|-----|----------|-----------|
|          | LBbest | Avg | T[second] | LBbest | Avg | T[second] |
| DSJC125.1| 238 | 246 | 244.10 | 80 | 239 | 232.55 | 20 |
| DSJC125.5| 504 | 536 | 522.40 | 35 | 524 | 509.6 | 19 |
| DSJC125.9| 1621 | 1664 | 1592.50 | 47 | 1646 | 1586.20 | 32 |
| DSJC250.1| 537 | 567 | 561.95 | 46 | 544 | 533.20 | 34 |
| DSJC250.5| 1150 | 1270 | 1258.80 | 37 | 1234 | 1223.45 | 34 |
| DSJC250.9| 3972 | 4179 | 4082.40 | 158 | 4116 | 4054.05 | 85 |
| DSJC500.1| 1163 | 1250 | 1246.55 | 1269 | 1184 | 1174.85 | 391 |
| DSJC500.5| 2616 | 2921 | 2902.60 | 60 | 2868 | 2840.25 | 51 |
| DSJC500.9| 10074 | 10881 | 10734.50 | 276 | 10827 | 10687.60 | 160 |
| DSJC1000.1| 2499 | 2762 | 2758.55 | 5193 | 2635 | 2616.10 | 922 |
| DSJC1000.5| 5787 | 6708 | 6665.90 | 155 | 6549 | 6526.65 | 119 |
| DSJC1000.9| 23863 | 26557 | 26300.25 | 2741 | 26455 | 26168.85 | 320 |

Table 6
Detailed comparison on DSJC1000.5 of disjoint clique removal (EXCLIQUE) and one-by-one removal (OBOCLIQUE)

| | EXCLIQUE | | OBOCLIQUE |
|-------|-------|-------|-------|
| | [C] | No. of cliques of size [C] | | [C] | No. of cliques of size [C] |
| 15 | 7 | | 15 | 7 |
| 14 | 26 | | 14 | 19 |
| 13 | 12 | | 13 | 15 |
| 12 | 11 | | 12 | 10 |
| 11 | 6 | | 11 | 10 |
| 10 | 5 | | 10 | 6 |
| 9 | 5 | | 9 | 5 |
| 8 | 2 | | 8 | 5 |
| 7 | 4 | | 7 | 1 |
| 6 | 3 | | 6 | 3 |
| 5 | 1 | | 5 | 4 |
| 4 | 2 | | 4 | 0 |
| 3 | 0 | | 3 | 2 |
| 2 | 3 | | 2 | 3 |
| 1 | 1 | | 1 | 2 |

6 Conclusion

In this paper, we have presented an effective heuristic algorithm based on clique decomposition for computing lower bounds for the minimum sum coloring problem. Basically, the proposed EXCLIQUE algorithm identifies and removes at each extraction iteration as many disjoint cliques as possible of the same size that is as large as possible. Compared to the conventional one-by-
one extraction strategy, EXCLIQUE is able to extract more large cliques from
the graph such that a higher number of vertices need to be colored with large
color numbers, thus leading to a clique decomposition with a large chromatic
sum which corresponds to a tighter lower bound for the MSCP.

We have assessed the performance of EXCLIQUE on a set of 62 benchmark
graphs (29 from DIMACS and 33 from COLOR02). For the 38 instances with
reported lower bounds, our EXCLIQUE algorithm has improved on the cur-
rent best lower bounds in 14 cases, proved optimality for the first time for 4
instances and attained the previously best bounds for 22 other instances. Only
in two cases, our bounds are slightly worse than the best known bounds. We
also reported for the first time lower bounds for the remaining 24 instances
that were missing in the literature.

Our comparison between EXCLIQUE and the basic one-by-one clique extrac-
tion method has showed a clear dominance of EXCLIQUE in terms of solution
quality even though the one-by-one extraction method is faster. We have also
revisited the clique decomposition approach based on coloring the comple-
mentary graph by using a powerful graph coloring algorithm. This experiment
demonstrated that it is globally less effective than the proposed EXCLIQUE
even though in some cases the graph coloring approach can attain better re-
results.

Acknowledgment

This work was partially supported by the Region of “Pays de la Loire” (France)
within the Radapop (2009-2012) and LigeRO Projects (2009-2013).

References

[1] A. Bar-Noy, M. Bellareb, M. M. Halldórsson, H. Shachnai, T. Tamir.
On chromatic sums and distributed resource allocation. Information and
Computation 140(2): 183–202, 1998.

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

[3] F. Bonomo, G. Durán, J. Marenco, M. Valencia-Pabon. Minimum sum set
coloring of trees and line graphs of trees. Discrete Applied Mathematics 159(5):
288–294, 2011.

[4] H. Bouziri and M. Jouini. A tabu search approach for the sum coloring problem.
Electronic Notes in Discrete Mathematics 36(1): 915–922, 2010.
[5] S.M. Douiri and S. Elberoussi. New algorithm for the sum coloring problem. International Journal of Contemporary Mathematical Sciences 6(10): 453–463, 2011.

[6] S.M. Douiri and S. Elberoussi. A New Ant Colony Optimization Algorithm for the Lower Bound of Sum Coloring Problem. Journal of Mathematical Modelling and Algorithms 11(2), 181–192, 2012.

[7] M.R. Garey and D.S. Johnson. Computers and intractability: A guide to the theory of NP-completeness. W.H. Freeman and Company, San Francisco, 1979.

[8] R. Gandhi, M.M. Halldórsson, G. Kortsarz and H. Shachnai, Improved bounds for sum multicoloring and scheduling dependent jobs with minsum criteria, In: Proc. of the Second Workshop on Approximation and Online Algorithms, 68-82, 2005.

[9] H. Hajiabolhassan, M.L. Mehrabadi, R. Tusserkani. Minimal coloring and strength of graphs. Discrete Mathematics 215(1–3): 265–270, 2000.

[10] A. Helmar and M. Chiarandini. A local search heuristic for chromatic sum. In L. D. Gaspero, A. Schaerf, T. Stützle (Eds.), Proceedings of the 9th Metaheuristics International Conference, MIC, pp 161–170, 2011.

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

[12] T. Jiang and D. West. Coloring of trees with minimum sum of colors. Journal of Graph Theory 32(4): 354–358, 1999.

[13] D.S. Johnson and M.A.Trick (Eds). Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. vol 26, AMS, Providence, RI, 1996.

[14] L. G. Kroon, A. Sen, H. Deng, A. Roy. The optimal cost chromatic partition problem for trees and interval graphs. Lecture Notes in Computer Science 1197: 279–292, 1996.

[15] Z. Kokosiński and K. Kawarciany. On sum coloring of graphs with parallel genetic algorithms. Lecture Notes In Computer Science 4431: 211–219, 2007.

[16] E. Kubicka. The chromatic sum of graphs; history and recent developments. The International Journal of Mathematical Sciences, 30: 1563-1573, 2004.

[17] E. Kubicka and A. J. Schwenk. An introduction to chromatic sums. Proceedings of the 17th Annual ACM Computer Science Conference, pp 39–45, 1989.

[18] E. Kubicka, G. Kubicki, D. Kountanis. Approximation algorithms for the chromatic sum. Proceedings of the First Great Lakes Computer Science Conference. Lecture Notes in Computer Science 507: 15–21, 1991.

[19] Y. Li, C. Lucet, A. Moukrim, K. Sghiouer. Greedy algorithms for minimum sum coloring algorithm. Proceedings LT2009 Conference, Tunisia, March 2009.
[20] Z. Lü and J.K. Hao. A memetic algorithm for graph coloring. European Journal of Operational Research 200(1): 235–244, 2010.

[21] M. Malafiejski. Sum coloring of graphs. Graph Colorings, Contemporary Mathematics 352, AMS, 55–65, 2004.

[22] A. Moukrim, K. Sghiouer, C. Lucet, Y. Li. Lower bounds for the minimal sum coloring problem. Electronic Notes in Discrete Mathematics 36: 663–670, 2010.

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

[24] Q. Wu and J.K. Hao. Adaptive multistart tabu search for the maximum clique problem. Accepted to Journal of Combinatorial Optimization, Nov. 2011. DOI: 10.1007/s10878-011-9437-8

[25] Q. Wu and J.K. Hao. Coloring large graphs based on independent set extraction. Computers and Operations Research, 39(2): 283–290, 2012.

[26] Q. Wu and J.K. Hao. An effective heuristic algorithm for sum coloring of graphs. Computers and Operations Research 39(7): 1593-1600, 2012.