Upper and Lower Bounds for the Minimum Sum Coloring Problem☆

Aziz Moukrim[a,*], Kaoutar Sghiouer[a], Corinne Lucet[b], Yu Li[b]

[a] *Université de Technologie de Compiègne, HeuDiaSyC UMR UTC-CNRS 7253, BP 20529, 60205 Compiègne, France*

[b] *Université de Picardie Jules Verne, M.I.S. EA 4290, 33 rue Saint Leu, 80000 Amiens, France*

Abstract

In this paper we study upper and lower bounds of the Minimum Sum Coloring Problem (MSCP) from a uniform point of view. MSCP is a graph coloring problem whose goal is to minimize the sum of colors, where colors are represented by natural numbers. We propose calculating upper and lower bounds for MSCP using a single coloring algorithm: a Memetic Algorithm (MA-MSCP) hybridizing a simple genetic algorithm with local search. Our lower bound is based on partitioning the original graph into cliques. In this context we define a new problem for obtaining this kind of general lower bound, namely Partition into Cliques for MSCP (PCMSCP), and prove its NP-completeness. We test our algorithm and compare it with other algorithms in the literature, and we also present the new results for some commonly used benchmark instances in order to provide a basis for future work. Experimental results show that our approach strictly improves or attains the lower bounds in the literature, and for the upper bounds the result of our algorithm is comparable. This work allows to show 27 instances optimally solved for MSCP and 36 instances for PCMSCP among 81 tested.

Keywords: Minimum Sum Coloring, Upper bounds, Lower bounds, Local search, Memetic Algorithm.

☆A preliminary version of this work was communicated in the International Symposium on Combinatorial Optimization (ISCO2010) [22].

This research was partially supported by *le Conseil Régional de Picardie*.

*Corresponding author

Email addresses: `aziz.moukrim@utc.fr` (Aziz Moukrim), `kaoutar.sghiouer@utc.fr` (Kaoutar Sghiouer), `corinne.lucet@u-picardie.fr` (Corinne Lucet), `yu.li@u-picardie.fr` (Yu Li)

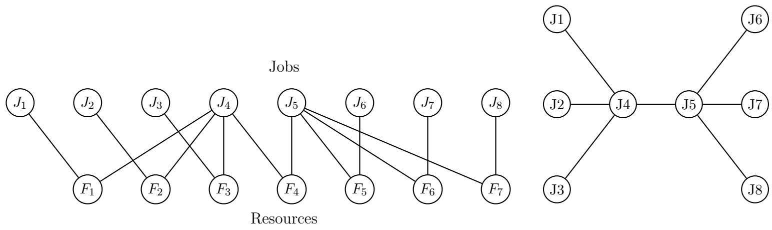
1. Introduction

The Minimum Sum Coloring Problem (MSCP) was first introduced by Kubicka and Schwenk, who also proved its NP-completeness [15]. MSCP is a graph coloring problem whose goal is to minimize the sum of colors, where the colors are represented by natural numbers. It is closely related to the basic Graph Coloring Problem (GCP), whose goal is to minimize the number of colors. MSCP has applications in VLSI design, scheduling and resource allocation [19].

GCP has been extensively studied in the literature, and many good results have been produced. These studies can be divided into two categories. The first category consists of exact methods such as constraint programming, column generation [9, 20] and decomposition [18], that give an optimal solution for small graphs or specific families of graphs. The second category involves basic heuristics such as greedy algorithms [4, 16] and meta-heuristics [8, 11]. Recently we proposed an adjusted greedy algorithm MDSAT [17] in $O(n^3)$, which improves the two well known existing greedy algorithms DSATUR and RLF for MSCP.

The literature features some theoretical results regarding upper and lower bounds of MSCP for specific graph classes [1, 14, 15, 24]. Algorithmic lower bound for MSCP was approached by different metaheuristics: an ant colony optimization algorithm (ANT) [6], a local search based on variable neighborhood (MDS(5)+LS) [10] and a decomposition into cliques (EX-CLIQUE) [27]. Also found in the literature, heuristic and metaheuristic algorithms providing an upper bound for the sum coloring: a genetic algorithm with a surrogate constraint heuristic (GA) proposed in [5], (MDS(5)+LS) [10] also used for upper bound, a Breakout Local Search (BLS) [2] that explores the search space by a joint use of local search and adaptive perturbation strategies for upper bounds, an efficient tabu search (EXSCOL) [26] based on the extraction of independent sets for upper bounds, and lately a memetic algorithm (MASC) proposed in [13].

The main purpose of this paper is to study upper and lower bounds of MSCP from a uniform point of view and then to propose an algorithm to calculate them. This work has its commencement in [17, 22]. An upper bound can be obtained by coloring the graph with a coloring algorithm. Obtaining a general lower bound involves extracting partial graphs for which the optimal solution can easily be computed, and in particular partial graphs generated by partitioning the graph into cliques. This has led us to introduce a new problem in relation to the lower bound, namely Partition into Cliques for MSCP (PCMSCP) and to prove its NP-completeness.



(a) Jobs and Resources.

(b) Graph of conflicts.

Figure 1: Scheduling Problem

We propose a Memetic Algorithm for MSCP (MA-MSCP), which is a simple hybrid genetic algorithm based on a crossover operator proposed in [8] and using local search techniques for mutation. Since PCMSCP can easily be transformed into the problem of coloring the complementary graph, we use the same coloring algorithm MA-MSCP to solve PCMSCP by considering the new objective function. We therefore compute both the upper and lower bounds, by obtaining valid solutions for MSCP and PCMSCP respectively.

We test our algorithm and compare it with other algorithms in the literature [2, 5, 6, 10, 13, 26, 27], and we also present the new results for some instances from the commonly used benchmark instances of DIMACS [12] and COLOR02 [7], in order to provide a basis for future work. Concerning MSCP, experimental results show that our approach strictly improves or attains the lower bounds in the literature, and for the upper bounds the result of our algorithm is comparable. The optimum has been reached for 27 instances out of the 81 tested ones. About PCMSCP, first results are presented where 36 of 81 instances are closed.

The paper is organized as follows. In the next section we give a formal definition for MSCP as well as GCP. In Section 3 we study some general lower bounds for MSCP and then introduce PCMSCP. In Section 4 we present MA-MSCP for obtaining upper and lower bounds for MSCP. In Section 5 we present and analyze experimental results. In Section 6, we conclude the paper.

2. Minimum Sum Coloring Problem

MSCP has recently begun to attract attention, because in a modern distributed system the average completion time is an important measure of quality of service.

As an illustration, consider the scheduling problem in distributed systems where a set of jobs will be executed by a cluster of processors in order to minimize the average time of executing a job. Let us denote the jobs to be executed as $J_1, J_2, \ldots, J_s$ and the resources used by these jobs as $F_1, F_2, \ldots, F_t$ (Fig. 1(a)). We suppose that each job runs in one unit of time and accesses exclusively to the resource. We also suppose that there is no limit on the number of processors. This scheduling problem can be modeled by a graph with vertices representing jobs, and edges representing access conflicts to resources (Fig. 1(b)). If we color the graph so that two adjacent vertices have different colors, a color can represent a period where the corresponding jobs can run together. The number of colors thus corresponds to the time taken by the cluster to execute all jobs. For this scheduling problem, this total time is equal to 2 (Fig. 2(a)). Furthermore, if we represent colors by the natural numbers $1, 2, \ldots, k$, the sum of colors corresponds to the total completion time of all jobs, and consequently to the average time for executing a job. For our scheduling problem, this average time is equal to 11/8 (Fig. 2(b)).

When coloring a graph, if the goal is to minimize the number of colors, we are dealing with the basic Graph Coloring Problem (GCP), whereas if the objective is to minimize the sum of colors, we are dealing with the Minimum Sum Coloring Problem (MSCP).

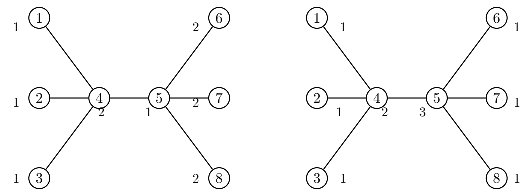
We consider an undirected graph $G = (V, E)$, where V is the set of $|V| = n$ vertices and E the set of $|E| = m$ edges.

A *coloring* of G is a function $c : v \mapsto c(v)$ that assigns to each vertex v a color $c(v)$. A coloring is said to be *feasible* if, for any pair of vertices $u, v \in V$ such that $[u, v] \in E$, we have $c(u) \neq c(v)$. If k colors are used in a feasible coloring, then this coloring of G is called a k-coloring. The minimum value of k among all the feasible colorings is called the *chromatic number* of the graph, denoted as $\chi(G)$. GCP involves finding this minimum number of colors.

Equivalently, a coloring can be seen as a partition of the set of vertices into k independent subsets, called *color classes:* $X_1, \ldots, X_k$, where the vertices in X_i are colored with color i. The number of vertices in X_i is denoted x_i and if $i < j$ then $x_i > x_j$. The associated sum of the colors can be written as follows:

$$\Sigma(G, c) = 1.x_1 + 2.x_2 + \cdots + k.x_k$$

MSCP consists in finding a feasible coloring such that the sum of the colors has the smallest possible value. This optimal value is called the



(a) Optimal solution for GCP. $\Sigma(G,c) = 12$, $\chi(G) = 2$

(b) Optimal solution for MSCP. $\Sigma(G) = 11$, $s(G) = 3$

Figure 2: Comparison of GCP and MSCP on trees.

chromatic sum of G and denoted $\Sigma(G)$. The smallest number of colors required by an optimal solution of MSCP is called the *chromatic strength* of the graph, and is denoted $s(G)$. The *chromatic strength* of G can be greater than its *chromatic number*.

As an example, the graph G in Figure 2(a) is a tree, so $\chi(G) = 2$. The best solution for MSCP using two colors has a sum coloring which is equal to 12. However, $s(G) = 3$ and $\Sigma(G) = 11$ (see Figure 2(b)).

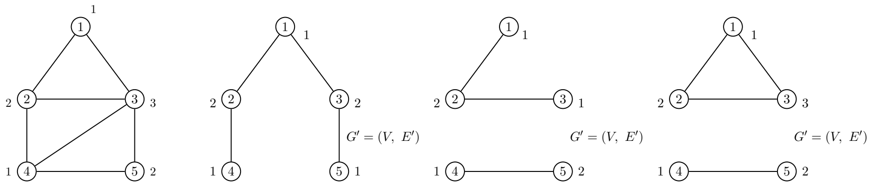
There are some theoretical results in relation to upper and lower bounds for MSCP, as well as properties for restricted graph families such as chain bipartite graphs, interval graphs and k-split-graphs [24]. However, to our knowledge, little work has been done regarding general bounds for the chromatic sum. In [25] the authors show that the chromatic sum $\Sigma(G)$ of any graph G (with n vertices, m edges, its chromatic number $\chi(G)$) is bounded as follows :

$$\Sigma(G) \leq n + m$$

$$\left\lceil \sqrt{8m} \right\rceil \leq \Sigma(G) \leq \left\lfloor \frac{3(m+1)}{2} \right\rfloor$$

$$n + \frac{\chi(G)(\chi(G)-1)}{2} \leq \Sigma(G) \leq \left\lfloor \frac{n(\chi(G)+1)}{2} \right\rfloor$$

Consequently, we note $LB_{th} = MAX\{\left\lceil \sqrt{8m} \right\rceil; n + \frac{\chi(G)(\chi(G)-1)}{2}\}$ the best theoretical lower bound and $UB_{th} = MIN\{n+m; \left\lfloor \frac{3(m+1)}{2} \right\rfloor; \left\lfloor \frac{n(\chi(G)+1)}{2} \right\rfloor\}$ the best theoretical upper bound. These theoretical bounds can be attained for specific graph classes, but they are still far away from the optimal solution



(a) $\Sigma(G) = 9$ (b) $\Sigma(G') = 7$ (c) $\Sigma(G') = 7$ (d) $\Sigma(G') = 9$

Figure 3: A graph G (a), some partial graphs of G and their associated chromatic sum (b)(c)(d).

in the general case. To overcome this, we propose an algorithmic solution for the upper and lower bounds of MSCP.

3. Lower Bounds for MSCP

A *partial graph* $G' = (V, E')$ of $G = (V, E)$ is a graph where E' is a subset of E. Any feasible coloring of G is a feasible coloring of G'. Therefore, the following result immediately holds.

Property 1. *If G' is a partial graph of G, then the chromatic sum of G' is a lower bound for the chromatic sum of G.*

To calculate a lower bound of MSCP we determine some partial graphs whose chromatic sums can be efficiently computed. In a previous work [22], we studied different kinds of partial graphs with such a characteristic, as bipartite graphs (trees and paths) and cliques. After we have tested on all instances of DIMACS and COLOR02, and put forward some properties, we concluded that bipartite graphs drop too many constraints (edges) to provide good lower bounds for the color problem benchmarks (see Fig. 3). As a consequence, in the next section we expose another family of partial graphs, where more constraints are preserved, to evaluate lower bounds for MSCP.

3.1. Partition into cliques

If we consider a clique of size k, there exists only one way of coloring that requires k colors. The associated sum of colors is $k(k+1)/2$.

Suppose that vertex set V is partitioned into $V_1, V_2, \ldots, V_l$ such that subgraph $G(V_i)$ induced by V_i is a clique. The graph $G(V_1) \cup G(V_2) \cup \ldots \cup G(V_l)$ is a partial graph of G, and its chromatic sum is $\sum_1^l \frac{|V_i|(|V_i|+1)}{2}$. This is a lower bound of MSCP for G.

For a good approximation of MSCP we therefore have to find a partition into cliques, such that the associated chromatic sum is as large as possible. This problem is an optimization problem that we term Partition into Cliques for MSCP (PCMSCP). Clearly, this problem is closely related to the well known Partition into Cliques Problem (PCP), but the two problems do not necessarily have the same optimal solution. In fact, if the minimum number of cliques is k for PCP, the optimal solution for PCMSCP may contain more than k cliques. In the example in Figure 4, the partition $\Lambda_1 = \{\{1,2,3\},\{4,5,6\},\{7,8,9\}\}$ is the optimal solution for PCP. Λ_1 is composed of three cliques and its chromatic sum is $3.\frac{3.4}{2} = 18$, while there exists a solution for PCMSCP $\Lambda_2 = \{\{1,3,4,6,7,9\},\{2\},\{5\},\{8\}\}$, having four cliques, and a chromatic sum equal to $\frac{6.7}{2} + 3.\frac{1.2}{2} = 24$.

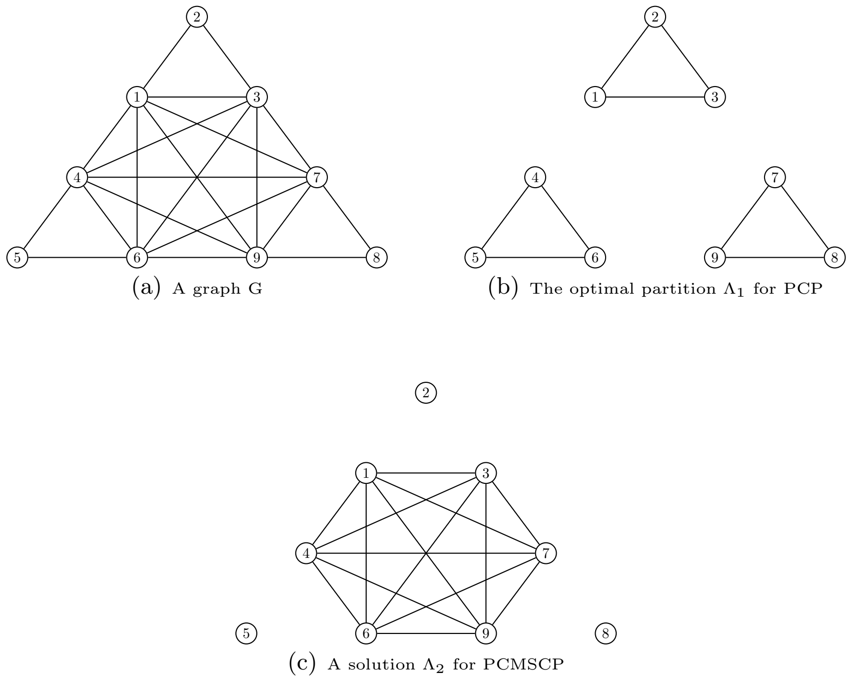


Figure 4: Partition into Cliques for MSCP (PCMSCP) and Partition into Cliques (PCP)

Definition 1. *Let $G = (V, E)$ be a graph. Partition into Cliques for MSCP (PCMSCP) consists in finding a partition of V into disjoint sets $V_1, V_2, \ldots, V_l$ such that for $1 \leq i \leq l$, the subgraph induced by V_i is a clique and $\sum_1^l \frac{|V_i|(|V_i|+1)}{2}$ is maximum. The optimal solution of PCMSCP is denoted $\Sigma_{clique}(G)$.*

Obviously, $\{\{1\}\,\{2\}\,,\ldots,\{\mid V\mid\}\}$ is a trivial partition of $G=(V,E)$, and in this case $\Sigma_{clique}(G)\geq n$. Graphs where $\omega(G)=2$ is a trivial case, for which PCMSCP amounts to solve the maximum matching problem, and can therefore be solved in polynomial time. Next, we establish a general upper bound of $\Sigma_{clique}(G)$ according to $\omega(G)$.

Proposition 1. *Let ω be the size of a maximum clique of G and $r=n-\omega\left\lfloor\frac{n}{\omega}\right\rfloor$. $\Sigma_{clique}(G)$ is upper bounded by $MAX_{LBclique}=\frac{\omega(\omega+1)}{2}\left\lfloor\frac{n}{\omega}\right\rfloor+\frac{r(r+1)}{2}$.*

Proof. If G has a maximum clique of size ω, any partition Λ has at most $\left\lfloor\frac{n}{\omega}\right\rfloor$ cliques of size ω. Each of these has an associated sum equal to $\frac{\omega(\omega+1)}{2}$. The r remaining vertices in the best case form a clique with an associated sum equal to $\frac{r(r+1)}{2}$.
Now let $\Lambda_{opt}=V_1,V_2,\ldots,V_l$ be an optimal solution of PCMSCP. Without loss of generality, we can assume that $|V_1|\geq|V_2|\geq\ldots\geq|V_l|$. We construct a new partition of V into subsets $X_1,X_2,\ldots,X_p$ $(p\leq l)$, each subset X_i being initialized with V_i. So, initially, the vertices in X_i are colored from 1 to $|V_i|$. Consider the first subset $X_j(1\leq j<p)$ such that $|X_j|<\omega$. Remove from X_p vertex u of color $|V_p|$ and add u to X_j. The new color of u will be $|V_j|+1>|V_p|$. Note that the subsets X_i are not necessarily cliques in G and $\sum_1^l\frac{|V_i|(|V_i|+1)}{2}<\sum_1^p\frac{|X_i|(|X_i|+1)}{2}$. By continuing in the same manner, we obtain a series of partitions, until we end up with the final partition, each of whose subsets has ω vertices, with the exception of the last subset, which has r vertices. The result consequently holds.

Corollary 1. *Let $G=(V,E)$ be a graph such that $|V|$ is a multiple of $\omega(G)$. If G is partitioned into k cliques $V_1,\ldots,V_k$ such as $\sum_1^k\frac{|V_i|(|V_i|+1)}{2}\geq\frac{\omega(\omega+1)}{2}\frac{|V|}{\omega}$, then $|V_i|=\omega\ \forall i$ and $\sum_1^k\frac{|V_i|(|V_i|+1)}{2}=\frac{\omega(\omega+1)}{2}\frac{|V|}{\omega}$.*

Theorem 1. *PCMSCP is an NP-hard problem.*

Proof. We transform Exact Cover by 3-sets to PCMSCP. First we recall the two decision problems:

Exact Cover by 3-Sets (X3C). Instance: a finite set X with $|X|=3k$ and a collection U of 3-element subsets of X.
Question: Does U contain an exact cover of X, that is, a sub-collection $U'\subseteq U$ such that every element of X occurs in exactly one member of U'?

Partition into cliques for MSCP (PCMSCP). Instance: let $G = (V, E)$ be a graph and B an integer.
Question: Is there a partition of V into k disjoint subsets $V_1, V_2, \ldots, V_k$ of V such that $\forall i$ $1 \leq i \leq k$, the subgraph induced by V_i is a clique and $\sum_1^k \frac{|V_i|(|V_i|+1)}{2} \geq B$?

It is easy to see that PCMSCP is in NP. We shall show that the Exact Cover by 3-Sets problem can be transformed into Partition into cliques for MSCP.

Let set X with $|X| = 3k$ and the collection $U = \{U_1, \ldots, U_s\}$ be an arbitrary instance of X3C.

We shall construct an instance of PCMSCP as follows: for each i in $\{1, \ldots, s\}$ and $U_i = \{x_i, y_i, z_i\}$ in U, we introduce 9 vertices $a_i(j), (1 \leq j \leq 9)$ and 18 edges as in Figure 5. These edges will be denoted by E_i. We introduce a graph $G = (V, E)$ where $V = X \cup \bigcup_1^s \{a_i(j), 1 \leq j \leq 9\}$, $E = \bigcup_1^s E_i$ and $B = 2|V|$. We have $|V| = |X| + 9.|U|$ so that $|V| = 3.(k + 3.s)$. Note that graph G and the number B can easily be obtained from X3C in polynomial time.

Let $\{U_{i_1}, \ldots, U_{i_q}\}$ be subsets in any exact cover for X. If $i \in \{1, \ldots, s\}$, there are two possible cases:

Case 1: $U_i \in \{U_{i_1}, \ldots, U_{i_q}\}$. We can therefore construct 4 cliques of G: $\{a_i(1), a_i(2), x_i\}$, $\{a_i(4), a_i(5), y_i\}$, $\{a_i(7), a_i(8), z_i\}$, $\{a_i(3), a_i(6), a_i(9)\}$.

Case 2: $U_i \notin \{U_{i_1}, \ldots, U_{i_q}\}$. We can therefore build 3 cliques of G: $\{a_i(1), a_i(2), a_i(3)\}$, $\{a_i(4), a_i(5), a_i(6)\}$, $\{a_i(7), a_i(8), a_i(9)\}$.

We obtain a partition $Z = \left\{V_1, \ldots, V_{\frac{|V|}{3}}\right\}$ of G into 3-cliques and $\sum_1^{\frac{|V|}{3}} \frac{|V_i|(|V_i|+1)}{2} = \sum_1^{\frac{|V|}{3}} \frac{3.4}{2} = 2|V|$.

Conversely, assume that $T = \{V_1, \ldots, V_k\}$ is a partition of G into k cliques such that $\sum_1^k \frac{|V_i|(|V_i|+1)}{2} \geq 2|V|$.

Note that $\omega(G) = 3$, using Corollary 1, we have: $\sum_1^k \frac{|V_i|(|V_i|+1)}{2} = 2|V|$ and $|V_i| = 3$ $\forall i \in \{1, \ldots, k\}$. Consequently, $\forall i \in \{1, \ldots, s\}$, $\{a_i(3), a_i(6), a_i(9)\} \in T$. Therefore, the subsets $\{x_i, y_i, z_i\}$ constitute an exact cover for X.

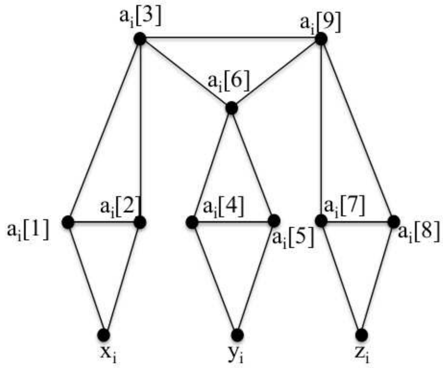


Figure 5: Transformation X3C to PCMSCP

4. Memetic algorithm

4.1. General outline

As discussed previously, an upper bound can be obtained by coloring the original graph G while a lower bound is obtained by optimally coloring a partial graph of G. Such a partial graph can be generated by partitioning G into cliques. Moreover, the problem of finding such a partition can easily be reduced to that of coloring the complementary graph of G, given the relation between a clique and an independent set.

Let $\overline{G} = (V, E')$ be the complementary graph of G such that $\forall(u, v) \in V \times V$, $[u, v] \in E'$ if and only if $[u, v] \notin E$. So, by definition, any independent set of $\overline{G}$ is a clique of G. Therefore, any partition of $\overline{G}$ into independent sets $X_1, X_2, \ldots, X_k$ can be seen as a partition of G into k cliques. Now, finding a partition of $\overline{G}$ into independent sets is simply equivalent to coloring $\overline{G}$.

The quality of the upper and lower bounds depends on the methods used for coloring G and $\overline{G}$ respectively. We have chosen to use a single coloring heuristic algorithm, MA-MSCP, to compute a lower bound denoted $LB_{MA} = MA\text{-}MSCP(\overline{G}, Max\ f_{PCMSCP})$, with the objective function to maximize $f_{PCMSCP}(S) = \sum_{i=1}^{k} \frac{x_i.(x_i+1)}{2}$ for PCMSCP, and an upper bound $UB_{MA} = MA\text{-}MSCP(G, Min\ f_{MSCP})$, by minimizing $f_{MSCP}(S) = \sum_{i=1}^{k} i.x_i$ for MSCP. MA-MSCP is a Memetic Algorithm dedicated to MSCP.
To summarize :

$$LB_{MA} \leq \Sigma_{clique}(G) \leq \Sigma(G) \leq UB_{MA}$$

Memetic algorithms are recent metaheuristics [21] that belong to the family of Evolutionary Algorithms. A memetic algorithm consists of two phases: a genetic evolution phase and a phase of local search. In the phase of genetic evolution a memetic algorithm causes a population of individuals to evolve according to the genetic principle, while in the local search phase individuals evolve according to the principle of local improvement. It includes a number of basic components: a representation of the solution, an initial population, the selection of individuals, the combination of individuals (crossover), local improvement and update of the population. We shall now describe in detail how each of these components contributes to our approach for solving the problem. Concerning PCMSCP, it is sufficient to consider graph $\overline{G}$ and the objective function $f_{PCMSCP}(S) = \sum_{i=1}^{k} \frac{x_i.(x_i+1)}{2}$ to maximize in different components of MA-MSCP.

4.2. Representation and evaluation of the solution

Each individual S is a coloring encoded as a partition $X_1, \dots, X_k$ of the vertex set of the graph, such that $x_1 \geq x_2 \geq \dots \geq x_k$. The quality of S is estimated by $f_{MSCP}(S) = \sum_{i=1}^{k} i.x_i$ for MSCP.

4.3. Initial population

The population POP is a list of P individuals $(S_1, \dots, S_P)$, sorted in ascending order of f_{MSCP}. A good initial population is composed of individuals of good quality and diversity. In this context, we initialize our population with $\frac{1}{4}P$ individuals generated by polynomial greedy algorithms proposed in [17] and $\frac{3}{4}P$ individuals randomly generated. To maintain the diversity of this population, we do not allow two different individuals to have the same evaluation value, i.e. for S_i and S_j $(i \neq j)$, $f_{MSCP}(S_i) \neq f_{MSCP}(S_j)$. In order to take into account the computation times consumed in the phase of local search, the population size in memetic algorithms is small. In our experimentation we use 20 individuals.

4.4. Selection of parents

To select the parents we use a binary tournament selection, which is experimentaly showed better than a random selection or roulette-wheel selection. First, we randomly select four different individuals out of the population. From the first two we choose the better individual according to f_{MSCP} to be parent P_1, and from the remaining the better individual to be the second parent P_2.

4.5. Crossover

The crossover operator used in our MA-MSCP to produce a new individual S_{new}, is an adaptive GPX crossover operator proposed by Galinier and Hao [8]. From two parents P_1 with k_1 colors and P_2 with k_2 colors, our crossover builds the k ($k = min\{k_1, k_2\}$) color classes of S_{new}. At step l ($1 \leq l \leq k$) we build color class X_l of S_{new} as follows: we consider parent P_1 if l is odd, otherwise we consider parent P_2. In the considered parent, we choose the largest color class to become class X_l in S_{new}, and remove all its vertices from P_1 and P_2. At the end of these steps, some vertices may remain unassigned. These vertices are then randomly assigned to an available color class. If no available color class exists, the vertices are assigned to a new color class.

Thanks to the data structures we used, the complexity of this $Crossover$ operator is $O(n^2)$. Indeed, there are at most n color classes ($k_1, k_2 \leq n$) and the update of the vertices in the color classes is bounded by $O(n)$.

4.6. Local improvement

The purpose of the local search in MA-MSCP is to improve a new individual produced by the above crossover, before inserting it into the population. Different neighborhoods, inspired of [3], are tested in an exploration of the solution space. Among all the neighborhoods tested, the two presented here give the best solutions for MSCP as well as PCMSCP.

Hill climbing: a vertex v colored with $c(v)$ is randomly chosen in the individual S, extracted from its color class $X_{c(v)}$ and then inserted, if possible, into a color class X_p such that $x_{c(v)} \leq x_p$ and p is an available color for v. We then obtain a new individual S', and obviously $f_{MSCP}(S') \leq f_{MSCP}(S)$. This process is iterated n times.

The time complexity of such a move is $O(n)$ for determining color class X_p and updating the available colors of the vertices if an insertion is performed.

Destroy and repair: the general principle is to remove a small part of the individual (some vertices) and improve it by reconstruction [23]. We apply this procedure to the individual obtained using the hill climbing method outlined above. Let S be an individual composed of k color classes and a random value d between 1 and n/k. First, randomly remove d vertices $v_1, v_2, \ldots, v_d$ from S. Next, the solution is reconstructed such that each removed vertex v_i is inserted into the largest available color class X_p ($1 \leq p \leq k$). If such a color class does not exist, the vertex v_i is assigned to a new color class, and the number of colors is incremented.

Since Hill climbing is used d times, the time complexity of this second move is $O(d.n)$.

The two procedures are alternately applied until there is no further improvement. This schema defines our MA-MSCP mutation operator which is systematically applied to each new individual obtained by the crossover operator.

4.7. Update population

To build the new population POP (of size P) from the previous one, we proceed as follows. After the crossover and the local improvement, we obtain a new individual S_{new} characterized by $f_{MSCP}(S_{new})$. If there is an individual S in the population such that $f_{MSCP}(S_{new}) = f_{MSCP}(S)$ then S is replaced by S_{new} to form the next population. Otherwise, S_{new} is inserted according to its rank, and the worst individual S_P of the population is rejected. This update is in $O(\log(P))$.

The stopping criterion of our MA-MSCP algorithm is time. This algorithm will be executed for a maximum time T, and its result is the best individual in the last population. Our memetic algorithm is summarized in Algorithm 1.

5. Experimental Results

In this section, we present the experimental results for the lower and upper bounds of MSCP obtained by our algorithm as well as other algorithms in the literature [2, 5, 6, 10, 13, 26, 27]. We will not report our previous results presented in [22] which are based on greedy algorithms (see Li et al. [17]) since they are dominated by MA-MSCP.

A total of 81 instances from the DIMACS and COLOR02 Libraries are used for these tests. These 81 instances were not all tested by all the methods of the literature. We present in Tables 1 and 3 the entire experiment results for the lower and upper bounds of MSCP and summarize these comparisons in Tables 2 and 4. Among them, for lower bound (resp. upper bound), 49 (resp. 65) instances have been used in the literature while the 32 (resp. 16) other instances are first reported. We include the families of graphs, as $fpsol$, $inithx.i$, $mulsol.i$, $zeroin.i$, and large graphs as $dsjr500.i$, $school1$ and $school1$-nsh. We also complete the results on $queen$, $le450$, and $miles$ families. In Table 5 we show experimental results for the specific problem PCMSCP defined in Section 3.1. Our algorithm MA-MSCP was

Require: POP a list of P individuals $(S_1, \ldots, S_P)$
Ensure: S_1 (*best solution*)

Sort individuals of POP in ascending order of f_{MSCP}
while (T is not over) **do**
  Choose two parents P_1 and P_2 by binary tournament
  $S_{new} \leftarrow Crossover(P_1,\ P_2)$
  $S_{new} \leftarrow Local\ Search(S_{new})$
  if $(\exists\ S \in POP,\ f_{MSCP}(S_{new}) = f_{MSCP}(S))$ **then**
    Replace S by S_{new}
  else
    if $(f_{MSCP}(S_{new}) \leq f_{MSCP}(S_P))$ **then**
      Insert S_{new} into POP
      Reject S_P from POP
    end if
  end if
end while
return S_1

Algorithm 1: Memetic Algorithm for MSCP

programmed in C and run on an Intel Core 2 Duo T5450- 1.66-1.67 with 2GB Ram running under Windows Vista Home Premium. The execution times are given for guidance but are difficult to compare. We set a limit for the total CPU time which is equal to 2 hours (following Benlic and Hao [2]). In other methods in literature (Jin et al. [13], Wu and Hao [26]), much more computational effort is allowed for huge instances as $DSJC1000.1$, $DSJC1000.9$, $qg.order60$.

5.1. Lower bounds results

Table 1 sums up the lower bounds of MSCP on 81 instances obtained by our algorithm MA-MSCP. We also compare MA-MSCP with MDS(5)+LS on 32 instances in [10], with ANT on 20 instances in [6] and with EXCLIQUE on 47 instances in [27]. The schema used to calculate lower bounds are all based on the decomposition into cliques on the complementary graph of G, which is presented in [22] and further developed in Section 3 in this paper. Therefore, all these results could be exploited as upper bounds for PCMSCP.

For each instance in Table 1, we denote n as the number of vertices, m as the number of edges, w as the best lower bound for the size of the maximum clique, Ki as the best upper bound for the chromatic number. w and Ki are marked in bold when optimal. LB_{th} is computed as discussed in Section 3, by using the chromatic number when it is known, and by using w otherwise.

Concerning our algorithm MA-MSCP, it is run 10 times and the lower bound averages are reported in column LB_{MA}^{Av}, the corresponding standard deviations in column LB_{MA}^{St}, the best lower bound in LB_{MA} and the average CPU time in minutes required to reach the best result in column T_{MA}. For the decomposition into cliques EXCLIQUE, best lower bounds of the 47 instances are in column LB_{EXCL} and the corresponding CPU time in minutes in column T_{EXCL}. For the algorithm MDS(5)+LS, the best lower bounds are reported in column LB_{MDS}. MDS(5)+LS is run during one hour for each of 32 instances. For the algorithm ANT, the best lower bounds on 20 instances and the CPU times in minutes are reported in columns LB_{ANT} and T_{ANT}. When they are comparisons between algorithms, the best solutions are in bold.

In Table 2 we report the scores of the comparison of MA-MSCP with the others algorithms. The first line $LB_{MA} = LB_*$ presents the number of equivalent lower bounds. As an illustration, MA-MSCP gives 21 lower bounds equal to those provided by EXCLIQUE, among 47 common tested instances. The second line $LB_{MA} > LB_*$ gives the number of times that MA-MSCP is strictly better. MA-MSCP computes 16 lower bounds better than those provided by EXCLIQUE, among 47 common tested instances. And finally, line $LB_{MA} < LB_*$ gives the number of times that MA-MSCP is strictly worse.

We observe that the lower bounds obtained by MA-MSCP are systematically better than or equal to those given by MDS(5)+LS and ANT and of equivalent quality to EXCLIQUE.

We note also that all lower bounds computed by these algorithms are better than the theoretical lower bounds, except for the instance myciel3, where the theoretical lower bound is 17, but the lower bounds obtained by algorithms are all equal to 16. Using Proposition 1 and since the maximum clique for myciel3 is 2, we have $MAX_{LBclique} = 16$. This exception suggests the limit of the schema based on the decomposition into disjoint cliques to compute lower bounds of MSCP for some particular instances.

$G(V,E)$	n	m	w	Ki	LB_{th}	LB_{MA}^{Av}	LB_{MA}^{St}	LB_{MA}	T_{MA}	LB_{EXCL}	T_{EXCL}	LB_{MDS}	LB_{ANT}	T_{ANT}
dsjc125.1	125	736	4	5	135	244.6	2.42	**247**	34	246	1	238		
dsjc125.5	125	3891	10	17	177	541	7.14	**549**	8	536	1	493		
dsjc125.9	125	6961	34	44	686	1677.7	9.24	**1689**	19	1664	1	1621		
dsjc250.1	250	3218	4	8	256	558.4	7.89	**569**	33	567	1	521		
dsjc250.5	250	15668	12	28	355	1249.4	25.63	**1280**	39	1270	1	1128		
dsjc250.9	250	27897	43	72	1153	4160.9	73.11	**4279**	15	4179	3	3779		
dsjc500.1	500	12458	5	12	510	1214.9	22.84	1241	22	**1250**	21	1143		
dsjc500.5	500	62624	13	48	708	2797.7	52.29	2868	50	**2921**	1	2565		
dsjc500.9	500	112437	56	126	2040	10443.8	263.38	10759	16	**10881**	5	9731		
dsjc1000.1	1000	49629	6	21	1015	2651.2	31.64	2707	98	**2762**	87	2456		
dsjc1000.5	1000	249826	15	87	1414	6182.5	183.53	6534	34	**6708**	3	5660		
dsjc1000.9	1000	449449	67	224	3211	24572	1128.36	26157	73	**26557**	46	23208		
dsjr500.1	500	3555	12	12	566	2052.9	4.61	2061	21					
dsjr500.1c	500	121275	83	84	3986	14443.9	504.65	15025	11					
dsjr500.5	500	58862	122	122	7881	22075	579.8	22728	30					
flat300-20-0	300	21375	11	20	490	1479.3	26.78	1515	58	**1524**	1			
flat300-26-0	300	21633	11	26	625	1501.6	25.74	**1536**	41	1525	1			
flat300-28-0	300	21695	12	28	678	1503.9	30	**1541**	26	1532	1			
flat1000-50-0	1000	245000	14	50	2225	6121.5	187.09	6433	63	**6601**	2			
flat1000-60-0	1000	245830	15	60	2770	6047.7	173.26	6402	42	**6640**	7			
flat1000-76-0	1000	246708	15	76	3850	6074.6	138.88	6330	66	**6632**	2			
fpsol2.i.1	496	11654	65	65	2576	3403	0	**3403**	63			3151	2590	8
fpsol2.i.2	451	8691	30	30	886	1668	0	1668	28					
fpsol2.i.3	425	8688	30	30	860	1636	0	1636	18					
le450-5a	450	5714	5	5	460	1171.5	15.25	1190	29					
le450-5b	450	5734	5	5	460	1166.5	13.98	1186	15					
le450-5c	450	9803	5	5	460	1242.3	20.58	1272	41					
le450-5d	450	9757	5	5	460	1245.2	21.02	1269	45					
le450-15a	450	8168	15	15	555	2324.3	4.8	**2329**	3	**2329**	4			
le450-15b	450	8169	15	15	555	2335	13.15	**2348**	2	2343	10			
le450-15c	450	16680	15	15	555	2569.1	22.42	**2593**	6	2591	3			
le450-15d	450	16750	15	15	555	2587.2	23.03	**2622**	24	2610	3			
le450-25a	450	8260	25	25	750	3000.4	3.07	**3003**	5	2997	16			
le450-25b	450	8263	25	25	750	3304.1	1.04	**3305**	2	**3305**	26			
le450-25c	450	17343	25	25	750	3617	16.72	**3638**	31	3619	11			
le450-25d	450	17425	25	25	750	3683.2	11.08	**3697**	20	3684	14			
mulsol.i.1	197	3925	49	49	1373	1957	0	1957	4					
mulsol.i.2	188	3885	31	31	653	1191	0	1191	5					
mulsol.i.3	184	3916	31	31	649	1187	0	1187	6					
mulsol.i.4	185	3946	31	31	650	1189	0	1189	4					
mulsol.i.5	186	3973	31	31	651	1160	0	1160	2					
inithx.i.1	864	18707	54	54	2295	3616	96.93	**3676**	107			3486	2801	13
inithx.i.2	645	13979	31	31	1110	1989.2	70.98	2050	16					
inithx.i.3	621	13969	31	31	1086	1961.8	45.8	1986	89					
zeroin.i.1	211	4100	49	49	1387	1822	0	1822	9					
zeroin.i.2	211	3541	30	30	646	1002.1	5.7	**1004**	13	**1004**	8	**1004**	1003	3
zeroin.i.3	206	3540	30	30	641	998	0	**998**	36	**998**	7	**998**	997	2
queen5-5	25	160	5	5	36	75	0	**75**	0	**75**	0	**75**	**75**	0
queen6-6	36	290	6	7	57	126	0	**126**	0	**126**	0	**126**	**126**	0

Continued on next page...

$G(V,E)$	n	m	w	Ki	LB_{th}	LB_{MA}^{Av}	LB_{MA}^{St}	LB_{MA}	T_{MA}	LB_{EXCL}	T_{EXCL}	LB_{MDS}	LB_{ANT}	T_{ANT}
queen7-7	49	476	7	7	70	196	0	**196**	0	**196**	1	**196**	**196**	0
queen8-8	64	728	8	9	100	288	0	**288**	0	**288**	1	**288**	**288**	0
queen8-12	96	1368	12	12	162	624	0	624	0					
queen9-9	81	1056	9	10	126	405	0	405	0					
queen10-10	100	1470	10	11	155	550	0	550	0					
queen11-11	121	1980	11	11	176	726	0	726	0					
queen12-12	144	2596	12	12	210	936	0	936	0					
queen13-13	169	3328	13	13	247	1183	0	1183	0					
queen14-14	196	4186	14	14	287	1470	0	1470	0					
queen15-15	225	5180	15	15	330	1800	0	1800	1					
queen16-16	256	6320	16	16	376	2176	0	2176	1					
school1.col	385	19095	14	14	476	2283.3	34.24	2345	5					
school1-nsh	352	14612	14	14	443	2064.6	24.36	2106	41					
anna	138	493	11	11	193	273	0	**273**	1	**273**	3	**273**	272	1
david	87	406	11	11	142	234	0	**234**	0	229	1	**234**	**234**	1
homer	561	1628	13	13	639	1129		1129	51		0			
huck	74	301	11	11	129	243	0	**243**	0	**243**	1	**243**	**243**	1
jean	80	254	10	10	125	216	0	**216**	0	**216**	1	**216**	**216**	1
games120	120	638	9	9	156	442	0	**442**	0	**442**	2	**442**	**442**	1
miles250	128	387	8	8	156	318	0	**318**	0	**318**	2	**318**	316	1
miles500	128	1170	20	20	318	686	0	**686**	0	677	2	**686**	677	1
miles750	128	2113	31	31	593	1145	0	1145	0					
miles1000	128	3216	42	42	989	1623	0	1623	12					
miles1500	128	5198	73	73	2756	3239	0	3239	0					
myciel3	11	20	2	4	**17**	16	0	16	0	16	0	16	16	0
myciel4	23	71	2	5	33	34	0	**34**	0	**34**	1	**34**	**34**	0
myciel5	47	236	2	6	62	70	0	**70**	0	**70**	1	**70**	**70**	0
myciel6	95	755	2	7	116	139.5	2.5	**142**	0	**142**	2	**142**	**142**	0
myciel7	191	2360	2	8	219	277.5	8.51	**286**	0	**286**	2	**286**	**286**	2
qg.order30	900	26100	30	30	1335	13950	0	**13950**	0	**13950**	8			
qg.order40	1600	62400	40	40	2380	32800	0	**32800**	1	**32800**	23			
qg.order60	3600	212400	60	60	5370	109800	0	**109800**	11	**109800**	125			

Table 1: Experimental results for Lower Bounds (LB)

	$EXCLIQUE$	MDS	ANT
$LB_{MA} = LB_*$	21/47	18/32	13/20
$LB_{MA} > LB_*$	16/47	14/32	7/20
$LB_{MA} < LB_*$	10/47	0/32	0/20

Table 2: Lower Bound: MA versus $EXCLIQUE$, MDS, ANT

5.2. Upper bounds results

Table 3 presents the results obtained by our algorithm MA-MSCP for the upper bounds of MSCP on the same instances as in Table 1. We also compare MA-MSCP with five algorithms in the literature [2, 5, 10, 13, 26]. The comparison is mainly based on the quality criterion, i.e. the best upper bound, due to the fact that these tests were conduced in different experimental conditions.

MA-MSCP is run 10 times and the upper bound averages are reported in column UB_{MA}^{Av}, the standard deviation in column UB_{MA}^{St}, the best upper bound in column UB_{MA}, and the average CPU time in minutes required to reach the best result in column T_{MA}. The columns UB_{BLS} and T_{BLS} are the best upper bounds and average CPU times required by BLS to reach its best result, it is a local search heuristic in [2] (27 tested instances) whereas UB_{Exscol} and T_{Exscol} are those of the tabu search algorithm EXSCOL in [26] (45 tested instances). UB_{MASC} and T_{MASC} are related to the memetic algorithm MASC in [13] (66 common tested instances). The column UB_{MDS} relates the results of the MDS(5)+LS algorithm that is run one hour for each instance [10] (32 tested instances). Finally, UB_{GA} are the upper bounds of a hybridized genetic algorithm that we denoted GA in [5] (20 tested instances and no CPU times are available). When they are comparisons between algorithms, the best upper bounds are in bold. Besides, we provide also the best lower bounds from Table 1 in column LB_{best}, followed by an asterisk when the LB_{best} reaches the best upper bound. The theoretical upper bound computed as discussed in Section 3, by using Ki, is reported in column UB_{th}.

In Table 4, for each algorithm (BLS, EXSCOL, MASC, MDS, GA), we compare the number of times MA-MSCP is equivalent (in line $UB_{MA} = UB_*$), better (in line $UB_{MA} < UB_*$) and worse (in line $UB_{MA} > UB_*$). As an example, MA-MSCP and MASC reach the same upper bound 44 times among 66 common instances, MA-MSCP is better on 6 instances and worse on 16. MASC and MA-MSCP have tested large instances.

The conjunction of the results obtained by the lower bounds and the upper bounds allow us to find the optimal value of 27 instances for MSCP. The following families are specially concerned : fpsol2.i, mulsol.i, inithx.i, zeroin.i, queen (9 out of 13 are closed) and qg.order. We note that 12 instances (some queen and qg.order) are closed thanks to the theoretical upper bound UB_{th}.

$G(V,E)$	LB_{best}	UB_{th}	UB^{Av}_{MA}	UB^{St}_{MA}	UB_{MA}	T_{MA}	UB_{BLS}	T_{BLS}	UB_{Exscol}	T_{Exscol}	UB_{MASC}	T_{MASC}	UB_{MDS}	UB_{GA}
dsjc125.1	247	375	327.3	1.49	**326**	10	**326**	18	**326**	1	**326**	4	**326**	
dsjc125.5	549	1125	1018.5	5.59	1013	5	**1012**	51	1017	1	**1012**	3	1015	
dsjc125.9	1689	2812	2519	20.03	**2503**	2	**2503**	1	2512	1	**2503**	2	2511	
dsjc250.1	569	1125	995.8	9.45	983	42	**973**	111	985	4	974	17	977	
dsjc250.5	1280	3625	3285.5	55.32	**3214**	26	3219	104	3246	6	3230	23	3281	
dsjc250.9	4279	9125	8348.8	66.64	**8277**	33	8290	41	8286	7	8280	5	8412	
dsjc500.1	1250	3250	2990.5	90.16	2897	4	2882	112	2850	9	**2841**	29	2951	
dsjc500.5	2921	12250	11398.3	314.09	11082	42	11187	118	10910	11	**10897**	73	11717	
dsjc500.9	10881	31750	30361.9	360.21	29995	51	30097	37	29912	15	**28896**	59	30818	
dsjc1000.1	2762	11000	9667.1	509.47	9188	31	9520	112	9003	28	**8995**	70	10123	
dsjc1000.5	6708	44000	40260.9	1582.08	38421	23	40661	113	37598	24	**37594**	200	43067	
dsjc1000.9	26557	112500	107349	1329.70	105234	61			**103464**	27	**103464**	125	112593	
dsjr500.1	2061	3250	2253.1	75.01	2173	4				0				
dsjr500.1c	15025	21250	16408.5	76.34	16311	34				0				
dsjr500.5	22728	30750	26978	525.57	25630	44				0				
flat300-20-0	1524	**3150**	3150	0.00	**3150**	0			**3150**	3	**3150**	0		
flat300-26-0	1536	4050	3966	0.00	**3966**	0			**3966**	3	**3966**	0		
flat300-28-0	1541	4350	4389.4	89.31	4261	28			4282	3	**4238**	22		
flat1000-50-0	6601	**25500**	25500	0.00	**25500**	28			**25500**	9	**25500**	0		
flat1000-60-0	6640	30500	30100	0.00	**30100**	16			**30100**	11	**30100**	114		
flat1000-76-0	6632	38500	39722.7	1594.82	38213	8			**37167**	19	**37167**	1		
fpsol2.i.1	3403*	12150	3403	0.00	**3403**	4				0	**3403**	8	**3403**	3405
fpsol2.i.2	1668*	6990	1668	0.00	**1668**	3				0	**1668**	5		
fpsol2.i.3	1636*	6587	1636	0.00	**1636**	2				0	**1636**	7		
le450-5a	1190	**1350**	1350	0.00	**1350**	1				0	**1350**	0		
le450-5b	1186	**1350**	1350	0.00	**1350**	2				0	**1350**	0		
le450-5c	1272	**1350**	1350	0.00	**1350**	3				0	**1350**	0		
le450-5d	1269	**1350**	1350	0.00	**1350**	2				0	**1350**	0		
le450-15a	2329	3600	2733.1	41.18	2681	19			**2632**	5	2706	41		
le450-15b	2348	3600	2730.6	35.40	2690	19			**2642**	7	2724	40		
le450-15c	2593	3600	4048.4	54.15	3943	6			3866	6	**3491**	45		
le450-15d	2622	3600	4032.4	79.41	3926	3			3921	5	**3506**	59		
le450-25a	3003	5850	3204.3	22.00	3178	5			**3153**	7	3166	39		
le450-25b	3305	5850	3416.2	27.95	3379	7			**3366**	6	**3366**	40		
le450-25c	3638	5850	4700.7	51.38	4648	16			**4515**	8	4700	25		
le450-25d	3697	5850	4740.3	38.46	4696	3			**4544**	7	4722	27		
mulsol.i.1	1957*	4122	1957	0.00	**1957**	2				0	**1957**	0		
mulsol.i.2	1191*	3008	1191	0.00	**1191**	2				0	**1191**	0		
mulsol.i.3	1187*	2944	1187	0.00	**1187**	1				0	**1187**	0		
mulsol.i.4	1189*	2960	1189	0.00	**1189**	1				0	**1189**	0		
mulsol.i.5	1160*	2976	1160	0.00	**1160**	1				0	**1160**	0		
inithx.i.1	3676*	19571	3679.6	5.62	**3676**	5				0	**3676**	7	**3676**	3679
inithx.i.2	2050*	10320	2053.7	3.61	**2050**	10				0	**2050**	4		
inithx.i.3	1986*	9936	1986	0.00	**1986**	2				0	**1986**	1		
zeroin.i.1	1822*	4311	1822	0.00	**1822**	2				0	**1822**	0		
zeroin.i.2	1004*	3270	1004	0.00	**1004**	1				0	**1004**	0	**1004**	1013
zeroin.i.3	998*	3193	998	0.00	**998**	1				0	**998**	0	**998**	1007
queen5-5	75*	**75**	75	0.00	**75**	0	**75**	0	**75**	1	**75**	0	**75**	**75**
queen6-6	126	144	138	0.00	**138**	0	**138**	0	150	1	**138**	1	**138**	**138**

Continued on next page...

$G(V,E)$	LB_{best}	UB_{th}	UB_{MA}^{Av}	UB_{MA}^{St}	UB_{MA}	T_{MA}	UB_{BLS}	T_{BLS}	UB_{Exscol}	T_{Exscol}	UB_{MASC}	T_{MASC}	UB_{MDS}	UB_{GA}
queen7-7	196*	**196**	196	0.00	**196**	0	**196**	0	**196**	1	**196**	0	**196**	**196**
queen8-8	288	320	291	0.00	**291**	0	**291**	0	**291**	1	**291**	12	**291**	302
queen8-12	624*	**624**	624	0.00	**624**	0				0	**624**	0		
queen9-9	405	445	411.9	4.61	**409**	2				0	**409**	1		
queen10-10	550	600	555.2	2.23	553	2				0				
queen11-11	726*	**726**	735.4	1.43	733	6				0				
queen12-12	936*	**936**	948.7	3.55	944	14				0				
queen13-13	1183*	**1183**	1197	3.97	1192	6				0				
queen14-14	1470*	**1470**	1490.8	5.78	1482	24				0				
queen15-15	1800*	**1800**	1823	5.20	1814	21				0				
queen16-16	2176*	**2176**	2205.9	5.94	2197	32				0				
school1.col	2345	2887	2766.8	100.85	2674	25				0				
school1-nsh	2106	2640	2477.1	71.31	2392	8				0				
anna	273	631	276	0.00	**276**	1	**276**	14	283	2	**276**	0	**276**	279
david	234	493	237	0.00	**237**	1	**237**	2	**237**	1	**237**	0	**237**	241
homer	1129	2189	1481.9	391.82	1157	32				0				
huck	243*	375	243	0.00	**243**	0	**243**	0	**243**	1	**243**	0	**243**	**243**
jean	216	334	217	0.00	**217**	0	**217**	0	**217**	1	**217**	0	**217**	**217**
games120	442	600	443	0.00	**443**	0	**443**	0	**443**	2	**443**	0	**443**	446
miles250	318	515	325.4	0.66	**325**	8	327	31	328	2	**325**	0	**325**	343
miles500	686	1298	711.2	2.23	708	12	710	86	709	2	**705**	1	712	755
miles750	1145	2048	1183.9	7.67	1173	11				0				
miles1000	1623	2752	1697.3	13.27	1679	5				0				
miles1500	3239	4736	3357.2	3.92	3354	1				0				
myciel3	17	27	21	0.00	**21**	0	**21**	0	**21**	1	**21**	0	**21**	**21**
myciel4	34	69	45	0.00	**45**	0	**45**	0	**45**	1	**45**	0	**45**	**45**
myciel5	70	164	93	0.00	**93**	0	**93**	1	**93**	1	**93**	0	**93**	**93**
myciel6	142	380	189	0.00	**189**	0	**189**	20	**189**	2	**189**	0	**189**	**189**
myciel7	286	859	381	0.00	**381**	0	**381**	38	**381**	2	**381**	1	**381**	**381**
qg.order30	13950*	**13950**	13950	0.00	**13950**	1			**13950**	15		4		
qg.order40	32800*	**32800**	32800	0.00	**32800**	1			**32800**	35	**32800**	11		
qg.order60	109800*	**109800**	109800	0.00	**109800**	7			110925	87	**109800**	290		

Table 3: Experimental results for the Upper Bounds (UB)

	BLS	$EXSCOL$	$MASC$	MDS	GA
$UB_{MA} = UB_*$	16/27	19/45	44/66	20/32	10/20
$UB_{MA} < UB_*$	8/27	11/45	6/66	11/32	10/20
$UB_{MA} > UB_*$	3/27	15/45	16/66	1/32	0/20

Table 4: Upper Bound: MA versus BLS, $EXSCOL$, $MASC$, MDS, GA

5.3. *PCMSCP results*

Table 5 is devoted to the study of the new problem PCMSCP defined in Section 3.1, and presents experimental results for the 81 instances previously mentioned. It is clear that any partition into cliques of G leads to a lower bound of $\Sigma_{clique}(G)$, the optimal value of PCMSCP. So, we report in the column LB_{PCMSCP} the best of such lower bounds from Table 1. In the same way, we can ensure that any upper bound for $\Sigma(G)$ is an upper bound for $\Sigma_{clique}(G)$ too (see Section 4.1). Then, we report the best upper bounds from Table 3 in column UB_{best}. Furthermore, $MAX_{LBclique}$ defined in Proposition 1 is also an upper bound for $\Sigma_{clique}(G)$. To compute $MAX_{LBclique}(G)$, we use the size of the maximum clique when it is known, and Ki otherwise. We define the best upper bound for $\Sigma_{clique}(G)$ as follows: $UB_{PCMSCP} = MIN\{UB_{best}, MAX_{LBclique}\}$.

All of the 27 instances closed for MSCP, are also closed for PCMSCP, and our study allowed to show 9 more closed instances for this new problem. Moreover, 9 of these 36 are closed thanks to $MAX_{LBclique}$, and more especially the myciel family. As mentioned in Section 3.1, when $\omega(G) = 2$, the instance is a trivial case. Also, our method allows many instances with regular structure (queen family) to be closed.

G (V,E)	**w**	**Ki**	$MAX_{LBclique}$	UB_{best}	UB_{PCMSCP}	LB_{PCMSCP}
dsjc125.1	4	**5**	375	326	326	247
dsjc125.5	10	17	1092	1012	1012	549
dsjc125.9	34	44	2683	2503	2503	1689
dsjc250.1	4	8	1119	973	973	569
dsjc250.5	12	28	3599	3214	3214	1280
dsjc250.9	43	72	8479	8277	8277	4279
dsjc500.1	5	12	3234	2841	2841	1250
dsjc500.5	13	48	11970	10897	10897	2921
dsjc500.9	56	126	31506	28896	28896	10881
dsjc1000.1	6	21	10948	8995	8985	2762
dsjc1000.5	15	87	43054	37594	37594	6708
dsjc1000.9	67	224	106260	103464	103464	26557
dsjr500.1	**12**	**12**	3234	2173	2173	2061
dsjr500.1c	83	**84**	21090	16311	16311	15025
dsjr500.5	**122**	**122**	30090	25630	25630	22728
flat300-20-0	11	**20**	3150	3150	3150	1524
flat300-26-0	11	**26**	3966	3966	3966	1536
flat300-28-0	12	**28**	4270	4238	4238	1541
flat1000-50-0	14	**50**	25500	25500	25500	6601
flat1000-60-0	15	**60**	30100	30100	30100	6640
flat1000-76-0	15	**76**	38116	37167	37167	6632
fpsol2.i.1	**65**	**65**	15876	3403	**3403**	**3403**

Continued on next page...

G (V,E)	w	Ki	$MAX_{LBclique}$	UB_{best}	UB_{PCMSCP}	LB_{PCMSCP}
fpsol2.i.2	**30**	**30**	6976	1668	**1668**	**1668**
fpsol2.i.3	**30**	**30**	6525	1636	**1636**	**1636**
le450-5a	**5**	**5**	1350	1350	1350	1190
le450-5b	**5**	**5**	1350	1350	1350	1186
le450-5c	**5**	**5**	1350	1350	1350	1272
le450-5d	**5**	**5**	1350	1350	1350	1269
le450-15a	**15**	**15**	3600	2632	2632	2329
le450-15b	**15**	**15**	3600	2642	2642	2348
le450-15c	**15**	**15**	3600	3491	3491	2593
le450-15d	**15**	**15**	3600	3506	3506	2622
le450-25a	**25**	**25**	5850	3153	3153	3003
le450-25b	**25**	**25**	5850	3366	3366	3305
le450-25c	**25**	**25**	5850	4515	4515	3638
le450-25d	**25**	**25**	5850	4544	4544	3697
mulsol.i.1	**49**	**49**	4901	1957	**1957**	**1957**
mulsol.i.2	**31**	**31**	2979	1191	**1191**	**1191**
mulsol.i.3	**31**	**31**	2915	1187	**1187**	**1187**
mulsol.i.4	**31**	**31**	2945	1189	**1189**	**1189**
mulsol.i.5	**31**	**31**	2976	1160	**1160**	**1160**
inithx.i.1	**54**	**54**	23760	3676	**3676**	**3676**
inithx.i.2	**31**	**31**	10245	2050	**2050**	**2050**
inithx.i.3	**31**	**31**	9921	1986	**1986**	**1986**
zeroin.i.1	**49**	**49**	5020	1822	**1822**	**1822**
zeroin.i.2	**30**	**30**	3256	1004	**1004**	**1004**
zeroin.i.3	**30**	**30**	3141	998	**998**	**998**
queen5-5	**5**	**5**	75	75	**75**	**75**
queen6-6	**6**	**7**	126	138	**126**	**126**
queen7-7	**7**	**7**	196	196	**196**	**196**
queen8-8	**8**	**9**	288	291	**288**	**288**
queen8-12	**12**	**12**	624	624	**624**	**624**
queen9-9	**9**	**10**	405	409	**405**	**405**
queen10-10	**10**	**11**	550	553	**550**	**550**
queen11-11	**11**	**11**	726	726	**726**	**726**
queen12-12	**12**	**12**	936	936	**936**	**936**
queen13-13	**13**	**13**	1183	1183	**1183**	**1183**
queen14-14	**14**	**14**	1470	1470	**1470**	**1470**
queen15-15	**15**	**15**	1800	1800	**1800**	**1800**
queen16-16	**16**	**16**	2176	2176	**2176**	**2176**
school1.col	**14**	**14**	2863	2674	2674	2345
school1-nsh.col	**14**	**14**	2628	2392	2392	2106
anna	**11**	**11**	813	276	276	273
david	**11**	**11**	517	237	237	234
Homer	**13**	**13**	3916	1157	1157	1129
huck	**11**	**11**	432	243	**243**	**243**
jean	**10**	**10**	440	217	217	216
games120	**9**	**9**	591	443	443	442
miles250	**8**	**8**	576	325	325	318
miles500	**20**	**20**	1296	705	705	686
miles750	**31**	**31**	1994	1173	1173	1145
miles1000	**42**	**42**	2712	1679	1679	1623
miles1500	**73**	**73**	4241	3354	3354	3239
myciel3	**2**	**4**	16	21	**16**	**16**
myciel4	**2**	**5**	34	45	**34**	**34**
myciel5	**2**	**6**	70	93	**70**	**70**
myciel6	**2**	**7**	142	189	**142**	**142**
myciel7	**2**	**8**	286	381	**286**	**286**
qg.order30.col	**30**	**30**	13950	13950	**13950**	**13950**
qg.order40.col	**40**	**40**	32800	32800	**32800**	**32800**
qg.order60.col	**60**	**60**	109800	109800	**109800**	**109800**

Table 5: Experimental results for PCMSCP

6. Conclusion

In this paper we investigated lower and upper bounds for MSCP. Because the chromatic sum of any partial graph is a lower bound for MSCP, we considered relevant partial graphs, partitions into cliques, for which the chromatic sum is easy to compute. In this context we proposed a new combinatorial problem, Partition into Cliques for MSCP (PCMSCP), such that any solution of PCMSCP is a lower bound for MSCP. We proved the NP-Completeness of PCMSCP and showed that building a solution for PCMSCP is simply a matter of coloring the complementary graph in order to work out a partition of the graph into cliques. We then presented a single coloring algorithm MA-MSCP, with different objective functions, for computing the lower and upper bounds. MA-MSCP is a memetic algorithm, based on an adaptive GPX crossover operator and on a local search dedicated to MSCP. Experimental results show that our approach improves or attains the lower bounds for a large panel of instances. We tested 81 instances of the DIMACS and COLOR02 libraries. This work helped to close 27 among 81 instances for the MSCP and 36 among 81 instances for the PCMSCP, thanks also to the theoretical bounds proposed here. It is also the first results shown for PCMSCP.

References

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

[2] U. Benlic and J. Hao. A study of breakout local search for the minimum sum coloring problem. *SEAL 2012, Lecture Notes in Computer Science*, 7673:128–137, 2012.

[3] H. Bouly, D.-C. Dang, and A. Moukrim. A memetic algorithm for the team orienteering problem. *4OR*, 8(1):49–70, 2010.

[4] D. Brelaz. New methods to color the vertices of a graph. *Commun. ACM*, 22(4):251–256, 1979.

[5] S. M. Douiri and S. Elbernoussi. New algorithm for the sum coloring problem. *Int. J. Contemp. Math. Sciences*, 6(10):453–463, 2011.

[6] S. M. Douiri and S. Elbernoussi. A local search heuristic for chromatic sum. *J Math Model Algorithm*, 11(1):181–192, 2012.

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

[8] P. Galinier and J. Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization*, 3(4):379–397, 1999.

[9] S. Gualandi and F. Malucelli. Exact solution of graph coloring problems via constraint programming and column generation. *INFORMS Journal on Computing*, 24:81–100, 2012.

[10] A. Helmar and M. Chiarandini. A local search heuristic for chromatic sum. *MIC 2011: The IX Metaheuristics International Conference*, pages 1–10, 2011.

[11] A. Hertz and D. de Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, 1987.

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

[13] Y. Jin, J. Hao, and J. Hamiez. A memetic algorithm for the minimum sum coloring problem. *Submitted*, 2013.

[14] L. Kroon, A. Sen, H. Deng, and A. Roy. The optimal cost chromatic partition problem for trees and interval graphs. In *WG'96*, pages 279–292, 1997.

[15] E. Kubicka and A. Schwenk. An introduction to chromatic sums. In *CSC'89: Proceedings of the 17th conference on ACM Annual Computer Science Conference*, pages 39–45, 1989.

[16] F. Leighton. A graph coloring algorithm for large scheduling problems. *Journal of research of the national institute of standards and technology*, 84(6):251–256, 1979.

[17] Y. Li, C. Lucet, A. Moukrim, and K. Sghiouer. Greedy algorithms for the minimum sum coloring problem. In *International Workshop: Logistics and transport*, 2009.

[18] C. Lucet, F. Mendes, and A. Moukrim. An exact method for graph coloring. *Computers & Operations Research*, 33(8):2189–2207, 2006.

[19] M. Malafiejski. *Sum coloring of graphs, in Graph Colorings (M.Kubale, Ed.)*. 2004.

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

[21] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts - towards memetic algorithms, 1989.

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

[23] R. Ruiz and T. Sttzle. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177:2033–2049, 2006.

[24] M. Salavatipour. On sum coloring of graphs. *Discrete Appl. Math.*, 127 (3):477–488, 2003.

[25] C. Thomassen, P. Erds, Y. Alavi, P. Malde, and A. Schwenk. Tight bounds on the chromatic sum of a connected graph. *J. Graph Theory*, 13:353–357, 1989.

[26] Q. Wu and J. Hao. An effective heuristic algorithm for sum coloring of graphs. *Comput. Oper. Res.*, 39(7):1593–1600, 2012.

[27] Q. Wu and J. Hao. Improved lower bounds for sum coloring via clique decomposition. *preprint : http://arxiv.org/abs/1303.6761*, 2013.