

# The Unconstrained Binary Quadratic Programming for the Sum Coloring Problem

Sidi Mohamed Douiri<sup>1</sup> & Souad Elbernoussi<sup>1</sup>

<sup>1</sup> Research laboratory Mathematics Computing and Applications, University Mohammed V, Rabat, Morocco

Correspondence: Sidi Mohamed Douiri, University Mohammed V, Faculty of Sciences-Agdal, 4 Avenue Ibn Battouta B. P. 1014 RP, Rabat, Morocco. E-mail: douirisidimohamed@hotmail.fr

Received: July 2, 2012 Accepted: August 8, 2012 Online Published: August 24, 2012

doi:10.5539/mas.v6n9p26 URL: <http://dx.doi.org/10.5539/mas.v6n9p26>

## Abstract

Several recent studies have shown the efficacy of unconstrained binary quadratic programming (UBQP) to model and solve many combinatorial problems. In this paper we are interested in the minimum sum coloring problem (MSCP), a new variant of the traditional graph coloring problem (GCP). We give a reformulation of the problem (MSCP) as an unconstrained binary quadratic binary programming, and we resolve it afterward by a genetic algorithms. The proposed algorithm is evaluated on the DIMACS challenge benchmarks and computational results show that the proposed UBQP model achieves highly competitive results, compared with 4 state-of-the-art algorithms.

**Keywords:** minimum sum coloring problem, unconstrained binary quadratic programming, genetic algorithms

## 1. Introduction

The graph coloring problem (GCP) is an NP-hard problem (Garey et al., 1979), it requires to assign a color to each vertex in such a way that colors on adjacent vertices are different and the number of used colors is minimized. For an undirected graph  $G = (V, E)$ , the chromatic number of a graph  $G$  is the smallest number of used colors for coloring all vertices of  $G$ , we denote it  $\chi(G)$ . The (GCP) problem has many applications such as the timetabling problem (de Werra, 1985), scheduling problem (Zufferey et al., 2008), and communication networks (Woo et al., 2002). In this paper we focus on the minimum sum coloring problem (MSCP), which is derived of the graph coloring one. Its principle is to minimize the total sum of colors assigned to all vertices of the graph. It was introduced first by Kubicka in 1987, and since, several theoretical works and algorithmics were realized among them we cite the work of (Kokosinski et al., 2007) using a parallel genetic algorithm based on GPX and CEX crossover with a number of iterations from 5000 to 10000. In 2010, a lower bound for MSCP has been presented by studying several approaches based on the extraction of partial graphs (Moukrim et al., 2010), for it they gave a coloring for the complementary graph using a greedy algorithm MRLF that has complexity  $O(n^3)$  (Li et al., 2009). Douiri et al. presented upper bounds by a genetic algorithm and a local heuristic (DBG) (Douiri et al., 2011a; 2011b), they also proposed a new ant colony optimization algorithm for the lower bound of sum coloring problem (Douiri et al., 2012). Wu et al. proposed a heuristic algorithm based on independent set extraction called EXSCOL (Wu et al., 2012).

Objective function of the unconstrained binary quadratic programming problem is to maximize (or minimize) the function:

$$f(x) = x^T Q x \quad (1)$$

where  $Q = (q_{ij})$  is an  $n \times n$  matrix of constants and  $x$  is an  $n$ -vector of binary variables, the great importance and the ability to model a wide of different problems on many areas as traffic management (Gallo et al., 1980), machine scheduling (Alidaee et al., 1994). UBQP gives evidence to its relevance and effectiveness in the face of known problems by their complexity such as the set packing problem (Alidaee et al., 2008), the vertex coloring problem (Kochenberger et al., 2005), and the linear ordering problem (Lewis et al., 2009). Given its NP-hard nature (Garey et al., 1979), various approaches have been proposed for solving this model using exact methods (Pardalos et al., 1990), and metaheuristic methods as memetic algorithms (Merz et al., 2004), scatter search (Amini et al., 1999), adaptive memory TS algorithms (Glover et al., 1998). In this paper we propose to transform

MSCP problem as an unconstrained binary quadratic programming (UBQP) and we solve it by a genetic algorithm.

In Part 2 we present some definitions and properties of (MSCP) in the third part we develop our model of (MSCP) in quadratic form. In the fourth section we describe our approach by a genetic algorithm, and we give numerical tests, and a conclusion.

### 2. Minimum Sum Coloring Problem

We consider an undirected graph  $G = (V, E)$  where  $V$  is the set of vertices and  $E$  denotes the set of edges. The sum coloring problem was introduced by (Kubicka et al., 1989) who demonstrated its NP-completeness. This problem aims to find a valid coloring of all vertices of a given graph so that the total sum of colors is minimal.

This minimum sum is called chromatic sum and denoted  $\Sigma(G) = \min_c \sum_{v \in V} c(v)$ , where  $c(v)$  is the color associated to the vertex  $v$ .

The smallest number of colors required in an optimal solution for (MSCP) is called the strength of the graph, and denoted  $s(G)$ . For any graph  $G = (V, E)$ , we recall some bounds of the chromatic sum:

$$card(V) + (s(G)(s(G) - 1)) / 2 \leq \Sigma(G) \tag{2}$$

$$card(V) + (\chi(G)(\chi(G) - 1)) / 2 \leq \Sigma(G) \tag{3}$$

$$\lceil \sqrt{8 \cdot card(E)} \rceil \leq \Sigma(G) \tag{4}$$

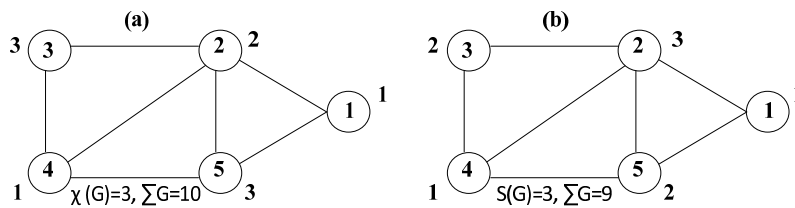


Figure 1. Two valid colorings with different sum coloring value

### 3. Quadratic Transformation

The minimum sum coloring problem (MSCP) can be expressed in a form of a linear problem with binary variables as follows:

$$(IP_{MSCP}) \left\{ \begin{array}{l} \min x_0 = \sum_{i=1}^n \sum_{p=1}^k p * x_{ip} \\ \sum_{p=1}^k x_{ip} = 1, i = 1, \dots, n \\ x_{ip} + x_{jp} \leq 1, \{i, j\} \in E, p = \{1, \dots, k\} \\ x_{ip} \text{ binary} \end{array} \right.$$

where

$$x_{ip} = \begin{cases} 1 & \text{if the vertex } i \text{ receives color } p \\ 0 & \text{otherwise} \end{cases}$$

and

$$x_i^2 = x_i \text{ since } x_i \text{ binary} \quad (5)$$

Each vertex must be colored we obtain:

$$\sum_{p=1}^k x_{ip} = 1, \quad i = 1, \dots, n$$

Two adjacent vertices have different colors are expressed as:

$$x_{ip} + x_{jp} \leq 1$$

The problem ( $IP_{MSCP}$ ) is constituted of a linear objective function with two types of constraints. The first one under shape equality  $A * x = b$ , where  $A$  is a matrix of dimension  $n \times (n \times k)$ , where  $n$  is the number of vertices and  $k$  the number of used colors. The second type of constraints is  $x_i + x_j \leq 1$ . The number of these inequality constraints is  $card(E) \times k$ . Our purpose is to reformulate the problem (IP) in a binary quadratic problem without constraints in the form:

$$(UBQP_{MSCP}) \begin{cases} \min x_0 = x^T Q x \\ x \text{ binary} \end{cases}$$

where  $Q$  is a square symmetric matrix of dimension  $(n \times k)$ . To obtain this formulation we apply two transformations mentioned above on the two types of constraints.

### 3.1 Transformation

We introduce the constraints in the objective function in the following way:

The objective is transformed under shape  $x^T D x$  by (5), the constraint  $A * x = b$  is introduced by  $S(A * x - b)^T (A * x - b)$ :

$$\begin{aligned} x_0 &= x^T D x + S(A * x - b)^T (A * x - b) \\ x_0 &= x^T D x + S(x^T A^T A x) - 2S(b^T A x) + S b^T b \\ x_0 &= x^T Q_j x + m \end{aligned}$$

where  $m = S b^T b = n \times S$  is a constant, that we can remove:

$$x_0 = x^T Q_j x$$

The constraints  $x_i + x_j \leq 1$  are introduced by  $S x_i x_j$ , where  $S$  is a positive scalar. The problem ( $IP_{MSCP}$ ) is thus replaced by the quadratic problem without constraint after being rid of the constant:

$$(UBQP_{MSCP}) \begin{cases} \min x_0 = x^T Q x \\ x \text{ binary} \end{cases}$$

### 3.2 Example

We consider the example in Fig.1, illustrating two different valid coloring for the same graph with a chromatic sum  $\Sigma(G) = 9$  in Fig.(1.b). The example satisfies the following linear programming:

$$(IP_{MSCP}) \begin{cases} \min x_0 = \sum_{i=1}^5 \sum_{p=1}^3 p * x_{ip} \\ x_{i1} + x_{i2} + x_{i3} = 1, i = 1, \dots, 5 \\ x_{ip} + x_{jp} \leq 1, \{i, j\} \in E, p = \{1, 2, 3\} \\ x_{ip} \text{ binary} \end{cases}$$

This example has 15 variables and 26 constraints, we chose a scalar  $S = 6$ , and we apply both transformations then we get:

$$Q = \begin{pmatrix} -5 & 6 & 6 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 \\ 6 & -4 & 6 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 6 & 6 & -3 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 \\ 6 & 0 & 0 & -5 & 6 & 6 & 6 & 0 & 0 & 6 & 0 & 0 & 6 & 0 & 0 \\ 0 & 6 & 0 & 6 & -4 & 6 & 0 & 6 & 0 & 0 & 6 & 0 & 0 & 6 & 0 \\ 0 & 0 & 6 & 6 & 6 & -3 & 0 & 0 & 6 & 0 & 0 & 6 & 0 & 0 & 6 \\ 0 & 0 & 0 & 6 & 0 & 0 & -5 & 6 & 6 & 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 6 & -4 & 6 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 6 & 6 & -3 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 6 & 0 & 0 & -5 & 6 & 6 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 & 6 & 0 & 6 & -4 & 6 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & 6 & 6 & 6 & -3 & 0 & 0 & 6 \\ 6 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 & -5 & 6 & 6 \\ 0 & 6 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 6 & 0 & 6 & -4 & 6 \\ 0 & 0 & 6 & 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 6 & 6 & 6 & -3 \end{pmatrix}$$

where the constant  $m = 30$ .

On the Figure (1.a) the valid coloring  $x^T = (100010001100001)$  gives a solution for  $UBQP_{MSCP}$  equal to  $x_0 = -20$ , and for the case (1.b), the coloring  $x^T = (100001010100010)$  gives the optimal solution for  $UBQP_{MSCP}$ ,  $x_0 = -21$ .

#### 4. Solving $UBQP_{MSCP}$

After having transformed our problem (MSCP) in unconstrained binary quadratic programming form, we try to solve it using an adapted genetic algorithm by judicious choice of operators in (MSCP) problem.

##### 4.1 Genetic algorithms

Our approach of resolution of the problem  $UBQP_{MSCP}$  is based on genetic algorithms, their functioning is extremely simple, we start with an initial population, we evaluate the performance of each individual, we create a new population of potential solutions using evolutionary operators: selection, crossover and mutation. We re-do this cycle until it finds a satisfactory solution or the iteration number reaches.

The population will be generated in a random manner, the size of each individual is equal to  $n \times k$ , in the case of Figure 1, every individual is of size 15, and on each three genes we assign a single 1 and the two remaining genes to 0, see Figure 2.

2			1			1			3			2		
0	1	0	1	0	0	1	0	0	0	0	1	0	1	0

Figure 2. The coding of an individual using 3 colors

We consider the objective function  $x^T Q x$  as the evaluation function (fitness) of each individual of the population.

For the selection, first, we randomly choose two individuals, and then we apply the tournament selection operator in order to keep the best individual. The comparison between two individuals is carried out according to their fitness.

$$C = \begin{cases} C_i & \text{if } C_i^T Q C_i < C_j^T Q C_j \\ C_j & \text{otherwise} \end{cases}$$

We opted for a simple crossing in a single point, we choose in a random way a point inter-gene  $b$  in each parents

$P_1$  and  $P_2$  such as  $b = k \times r$ , where  $k$  is the number of colors and  $r = 1, \dots, (n - 1)$ . The crossing is made in the following way:

$$C_1(i) = \begin{cases} P_1(i) & \text{if } i \in [1, b[ \\ P_2(i) & \text{if } i \in [b, n] \end{cases}$$

$$C_2(i) = \begin{cases} P_2(i) & \text{if } i \in [1, b[ \\ P_1(i) & \text{if } i \in [b, n] \end{cases}$$

The individuals  $C_1$  and  $C_2$  are not necessarily generate according to the rule above, but sometimes, we bring some modifications to obtain individuals verifying the conditions of our problem. In the event where a color  $l$  is lacking in the individual  $C$  after the crossover operation, we follow the rule:

If the individual  $C$  resulting after crossing has no color  $1 \leq l \leq k$  then for  $1 \leq j < b$  and  $b \leq i \leq n$  if  $C(i) = C(j)$  then  $C(j) = l$ , see Figure 3.

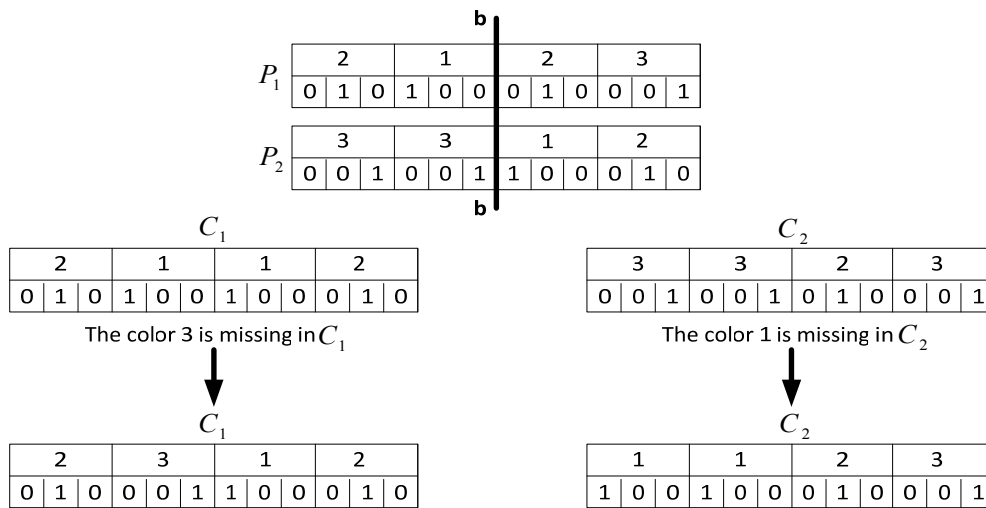


Figure 3. Crossing example

The mutation is used with the aim to further explore the search space and reaching solutions that the crossing cannot touch them, for this we choose randomly two separate blocks (different colors) of  $k$  genes and we permute them, we replace the solution  $p$  by  $p'$  if  $f(p') \leq f(p)$ , see in Figure 4.

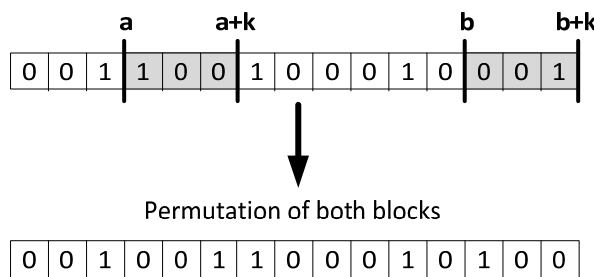


Figure 4. Mutation of an individual with five vertices

#### 4.2 Remark

We fix  $k$  at some large value and it decreased by 1 in each subsequent problem until it is no longer possible to find a feasible coloring. For each value of  $k$  we obtain the associated  $UBQP_{MSCP}$  problem. The number of variables of  $IP_{MSCP}$  change and therefore the size of the symmetric matrix  $Q$  changes.  $Q$  depends of the number of used colors  $k$ .

## 5. Results and Discussion

In this section, we report intensive experimental results of our approach on the well-known DIMACS and COLOR02 coloring benchmarks (<ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/color/2>; <http://mat.gsia.cmu.edu/COLOR02/>). We compare the results with 4 other state of the art sum coloring algorithms from the literature. For each instance, we indicate the number of vertices  $n$ , the chromatic number  $\chi(G)$ . Column 5 presents the lower bound (LB) given in (Douiri et al., 2012), column 6 indicates the current best known sum values ( $UB_{best}$ ) from the literature along with the references reporting these values in brackets,  $S_{UBQP}$  is the strength of G given by our approach, the success rate (#hit) we run our algorithm on each graph 10 times, last column shows the average computation time in seconds to reach the given  $T_{max}$ . The symbol '\*' means that the related statistics are not available. Our algorithm has been implemented on a PC windows 7, AMD Athlon(tm) X2 dual-core QL-65 (2cpus) 2.1 GHz with 4 GB RAM.

The set of benchmark graphs from the literature are considered in the experiments. The Table 1 is composed of 38 well-known graphs. Among these graphs, sum coloring results have been reported in the literature (Kokosinski et al., 2007; Douiri et al., 2011a; Li et al., 2009; Wu et al., 2012) denoted respectively by ([1], [2], [3], [4]). The parameters of the GA for the  $UBQP_{MSCP}$  problem are: population size= 80, crossover rate  $p_c= 0.75$ , mutation rate  $p_m= 0.2$ , and the maximum iterations number  $T_{max}$  is taken between 120 and 800 iterations depending on the size of graph treated. Table 1 shows that our algorithm improves a large sets of benchmark graphs treated. Our results disclose that UBQP approach competes favorably with (Kokosinski et al., 2007; Li et al., 2009; Douiri et al., 2011a; Wu et al., 2012) on the set of 35 instances. Indeed, UBQP method improves the best known upper bounds of sum coloring for 8 graphs, and finds identical results for 19 graphs. The algorithm fails to achieve the best known results on 8 graphs (queen8.8, miles500, dsjc125.1, dsjc125.9, dsjc250.5, dsjc250.9, dsjc500.5, dsjc500.9). Concerning the CPU time, for the majority of graphs tested (25 instances) the CPU time to reach the smallest sum coloring is less than one minute, but for few hard graphs, the computation time is long as the 12 DIMACS random dsjc graphs.

Table 1. Computational results

Graph	$n$	$\chi(G)$	LB	UBbest	[1]	[2]	[3]	[4]	UB <sub>MSCP</sub>	S <sub>UBQP</sub>	#hit	Time(s)
anna	138	11	272	277[3]	281	*	<b>277</b>	283	<b>277</b>	11	7/10	13.65
david	87	11	234	237[4]	243	*	241	<b>237</b>	<b>237</b>	11	9/10	19.70
huck	74	11	<b>243</b>	243[1,2,4]	<b>243</b>	<b>243</b>	244	<b>243</b>	<b>243</b>	11	10/10	1.23
jean	80	10	216	217[3,4]	218	*	<b>217</b>	<b>217</b>	<b>217</b>	11	8/10	1.07
queen5.5	25	5	<b>75</b>	75[5,7,21]	<b>75</b>	*	<b>75</b>	<b>75</b>	<b>75</b>	5	10/10	0.19
queen6.6	36	7	126	138[1,3,2]	<b>138</b>	<b>138</b>	<b>138</b>	150	<b>138</b>	8	10/10	0.27
queen7.7	49	7	<b>196</b>	196[1,3,4]	<b>196</b>	*	<b>196</b>	<b>196</b>	<b>196</b>	7	10/10	0.42
queen8.8	64	9	288	291[4]	302	*	303	<b>291</b>	296	10	9/10	0.76
miles250	128	8	316	328[4]	347	343	334	<b>328</b>	<b>328</b>	9	10/10	3.4
miles500	128	20	677	709[4]	762	755	715	<b>709</b>	715	22	8/10	5.61
games120	120	9	442	443[4]	460	446	446	<b>443</b>	<b>443</b>	9	9/10	0.92
myciel3	11	4	16	21[1,2,3,4]	<b>21</b>	<b>21</b>	<b>21</b>	<b>21</b>	<b>21</b>	4	10/10	0.06
myciel4	23	5	34	45[1,2,3,4]	<b>45</b>	<b>45</b>	<b>45</b>	<b>45</b>	<b>45</b>	5	10/10	0.11
myciel5	47	6	70	93[1,2,3,4]	<b>93</b>	<b>93</b>	<b>93</b>	<b>93</b>	<b>93</b>	6	10/10	0.21
myciel6	95	7	142	189[1,2,3,4]	<b>189</b>	<b>189</b>	<b>189</b>	<b>189</b>	<b>189</b>	7	10/10	3.74
myciel7	191	8	286	381[2,3,4]	382	<b>381</b>	<b>381</b>	<b>381</b>	<b>381</b>	8	10/10	8.96
fpsol2.i.1	496	65	2590	3405[2]	*	3405	*	*	<b>3402</b>	65	7/10	131.3
inithx.i.1	864	54	2801	3679[2]	*	3679	*	*	<b>3676</b>	54	6/10	478.11
mug88-1	88	4	163	190[2]	*	<b>190</b>	*	*	<b>190</b>	4	10/10	2.35

mug88-25	88	4	162	187[2]	*	<b>187</b>	*	*	<b>187</b>	4	10/10	2.08
mug100-1	100	4	187	211[2]	*	211	*	*	<b>202</b>	4	10/10	2.32
mug100-25	100	4	185	214[2]	*	214	*	*	<b>204</b>	4	10/10	3.14
2-Inser-3	37	4	55	62[2]	*	<b>62</b>	*	*	<b>62</b>	4	10/10	0.69
3-Inser-3	56	4	84	92[2]	*	<b>92</b>	*	*	<b>92</b>	4	10/10	0.91
zeroin.i.2	211	30	1003	1013[2]	*	1013	*	*	<b>1009</b>	30	6/10	94.02
zeroin.i.3	206	30	997	1007[2]	*	1007	*	*	<b>1002</b>	30	4/10	77.58
dsjc125.1	125	*	*	326[4]	*	*	352	<b>326</b>	332	8	5/10	42.13
dsjc125.5	125	*	*	1017[4]	*	*	1141	1017	<b>1015</b>	19	8/10	39.27
dsjc125.9	125	*	*	2512[4]	*	*	2653	<b>2512</b>	2519	45	4/10	41.30
dsjc250.1	250	*	*	985[4]	*	*	1068	985	<b>978</b>	9	6/10	71.1
dsjc250.5	250	*	*	3246[4]	*	*	3658	<b>3246</b>	3248	31	7/10	89.82
dsjc250.9	250	*	*	8286[4]	*	*	8942	<b>8286</b>	8302	74	4/10	225.91
dsjc500.1	500	*	*	2850[4]	*	*	3229	<b>2850</b>	<b>2850</b>	13	6/10	185.26
dsjc500.5	500	*	*	10910[4]	*	*	12717	<b>10910</b>	11016	52	6/10	830.07
dsjc500.9	500	*	*	29912[4]	*	*	32713	<b>29912</b>	29916	130	3/10	862.14

## 6. Conclusion

In this paper we proposed a resolution approach for the minimum sum coloring problem (MSCP) which is an NP-complete problem in combinatorial optimization. We transformed the linear model into an unconstrained binary quadratic programming model UBQP. We have subsequently solved it with an adapted genetic algorithm. We have shown that this approach obtains highly competitive results on a large number of DIMACS challenge benchmark graphs. The obtained results show the efficiency of our method and its competitiveness with the other cited approaches.

## References

- Alidaee, B., Kochenberger, G. A., & Ahmadian, A. (1994). 0-1 Quadratic programming approach for the optimal solution of two scheduling problems. *International Journal of Systems Science*, 25, 401-408. <http://dx.doi.org/10.1080/00207729408928968>
- Alidaee, B., Kochenberger, G. A., Lewis, K., Lewis, M., & Wang, H. (2008). A new approach for modelling and solving set packing problems. *European Journal of Operational Research*, 86(2), 504-512. <http://dx.doi.org/10.1016/j.ejor.2006.12.068>
- Amini, M., Alidaee, B., & Kochenberger, G. A. (1999). A Scatter Search Approach to Unconstrained Quadratic Binary Programs. *New Methods in Optimization*, McGraw-Hill, New York, NY, 317-330.
- de Werra, D. (1985). An introduction to timetabling. *European Journal of Operational Research*, 19, 151-162. [http://dx.doi.org/10.1016/0377-2217\(85\)90167-5](http://dx.doi.org/10.1016/0377-2217(85)90167-5)
- Douiri, S. M., & Elberoussi, S. (2011a). New algorithm for the sum coloring problem. *Int. J. Contemp. Math. Sciences*, 6(10), 453-463.
- Douiri, S. M., & Elberoussi, S. (2011b). A New Heuristic for the Sum Coloring Problem. *Applied Mathematical Sciences*, 5(63), 3121-3129.
- Douiri, S. M., & Elberoussi, S. (2012). A new ant colony optimization algorithm for the lower bound of sum coloring problem. *Journal of Mathematical Modelling and Algorithms*, 11(2), 181-192. <http://dx.doi.org/10.1007/s10852-012-9172-x>
- Gallo, G., Hammer, P., & Simeone, B. (1980). Quadratic knapsack problems. *Mathematical Programming*, 12, 132-149. <http://dx.doi.org/10.1007/BFb0120892>
- Garey, M. R., & Johnson, D. S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. *W.H. Freeman and Company, New York*.

- Glover, F., Kochenberger, G. A., & Alidaee, B. (1998). Adaptive memory tabu search for binary quadratic programs. *Management Science*, 44, 336-345. <http://dx.doi.org/10.1287/mnsc.44.3.336>
- Kokosinski, Z., & Kwarciany, K. (2007). On sum coloring of graphs with parallel genetic algorithms. In *ICANNGA'07, part I, LNCS 4431*, pp. 211-219.
- Kochenberger, G. A., Glover, F., Alidaee, B., & Rego, C. (2005). An unconstrained quadratic binary programming approach to the vertex coloring problem. *Annals of Operations Research*, 139, 229-241. <http://dx.doi.org/10.1007/s10479-005-3449-7>
- Kubicka, E., & Schwenk, A. J. (1989). An introduction to chromatic sums, *Proceedings of the ACM Computer Science Conference*, 39-45.
- Lewis, M., Alidaee, B., Glover, F., & Kochenberger, G. A. (2009). A note on xQx as a modelling and solution framework for the Linear Ordering Problem. *International Journal of Operational Research*, 5(2), 152-162. <http://dx.doi.org/10.1504/IJOR.2009.025005>
- Li, Y., Lucet, C., Moukrim, A., & Sghiouer, K. (2009). Greedy Algorithms for the Minimum Sum Coloring Problem. *International Workshop: Logistics and transport*.
- Merz, P., & Katayama, K. (2004). Memetic algorithms for the unconstrained binary quadratic programming problem. *BioSystems*, 78, 99-118. <http://dx.doi.org/10.1016/j.biosystems.2004.08.002>
- Moukrim, A., Sghiouer, K., Lucet, C., & Li, Y. (2010). Lower Bounds for the Minimal Sum Coloring Problem. *Electronic Notes in Discrete Mathematics*, 36, 663-670. <http://dx.doi.org/10.1016/j.endm.2010.05.084>
- Pardalos, P., & Rodgers, G. P. (1990). Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45, 131-144. <http://dx.doi.org/10.1007/BF02247879>
- Wu, Q., & Hao, J. K. (2012). An effective heuristic algorithm for sum coloring of graphs. *Computers & Operations Research*, 39(7), 1593-1600. <http://dx.doi.org/10.1016/j.cor.2011.09.010>
- Woo, T. K., Su, S. Y. W., & Wolfe, R. N. (2002). Resource allocation in a dynamically partitionable bus network using a graph coloring algorithm. *IEEE Transactions on Communications*, 39, 1794-1801. <http://dx.doi.org/10.1109/26.120165>. <http://dx.doi.org/10.1109/26.120165>
- Zufferey, N., Amstutz, P., & Giaccari, P. (2008). Graph colouring approaches for a satellite range scheduling problem. *Journal of Scheduling* 11, 263-277. <http://dx.doi.org/10.1007/s10951-008-0066-8>