Applied Mathematics & Information Sciences
*An International Journal*



# A New Steganographic Method For Grayscale Image Using Graph Coloring Problem

*Sidi Mohamed Douiri*[1], *Mohamed Boy Ould Medeni*[1], *Souad Elbernoussi*[1], *El Mamoun Souidi*[1]

Laboratory of Mathematic Informatics and Applications. University Mohammed V-Agdal, Faculty of Sciences, B.P. 1014, Rabat-Maroc.

Received: Jul. 10, 2012; Revised Dec. 1, 2012; Accepted Dec. 15, 2012
Published online: 1 Mar. 2013

**Abstract:** Statistical steganalysis schemes detect the existence of secret information embedded by steganography. The $\chi^2$ detection and Regular-Singular (RS)-attack methods are two well known statistical steganalysis schemes used against LSB (least significant bit) steganography. The embedded message length can be estimated accurately by these two steganalysis schemes. For secret communication, the resistance of steganography against steganalysis is very important for information security. To avoid the enemy's attempts, the statistical features between stego-images and cover images should be as similar as possible for better resistance to steganalysis. To ensure the security against the RS and $\chi^2$ analysis, we presents in this paper a new steganographic method based on graph coloring problem (GCP). Before embedding the secret message in LSB (least significant bit) of the cover image, we use a (GCP) algorithm to locate the optimal positions of the pixels in the cover image. Thus, the existence of the secret message is hard to be detected by the RS analysis. Meanwhile, better visual quality can be achieved by the proposed algorithm. The experimental results demonstrate the proposed algorithm's effectiveness in resistance to steganalysis with better visual quality.

**Keywords:** Steganography, graph coloring, heuristic DBG, LSB method.

## 1. Introduction

Nowadays internet is a popular communication channel. Transmitted data are easy to be copied or destroyed by unauthorized persons. Therefore, how to transmit data secretly by internet becomes an important topic. Encryption may provide a safe way, which transforms data into a ciphertext via cipher algorithms [17]. However, it makes the messages unreadable and suspicious enough to attract eavesdropper's attention. To overcome this problem, steganography offers different approaches to transmitting secret messages. Steganography is a technique that imperceptibly hides secret data into cover media by altering its most insignificant components for covert communication, such that an unauthorized user will not be aware of the existence of secret data. Many successful steganography methods have been proposed. Among all the methods, LSB (least significant bit) substitution, which embeds secret data by replacing $k$ LSBs of a pixel with $k$ secret bits directly [14]. The LSB embedding achieves good balance between the payload capacity and visual quality. However, the LSB replacing method flips one half of the least-significant bits. Thus the artifacts in the statistics of the image are easy to be detected. In a recent paper, different form of the LSB Steganographic schemes are proposed, in 2008, Wang et al. presented a steganographic method that utilizes the remainder of two consecutive pixels to record the information of secret data [15]. Yang et al. proposed an adaptive LSB steganographic method using the difference value of two consecutive pixels to distinguish between edge areas and smooth areas [18]. All pixels are embedded by the $k$-bit modified LSB substitution method, where $k$ is decided by the range which the difference value belongs to [18]. However, some of them seem not to consider the features of edge sufficiently [15,17,18]. The methods described in [19,20] have overcome the drawback, but unfortunately they result has propagated error and lower embedding capacity.

To resist to RS and $\chi^2$ steganalyses [22,23], the influence on the correlation of pixels needs to be compensated.

* Corresponding author: e-mail: douirisidimohamed@hotmail.fr

The compensation may be achieved by adjusting other bit planes. Nevertheless, the implementation may be computational infeasible. For example, if only two bit planes are modified in a $256 \times 256$ gray level image, there are $2^2$ possible bit selections for each pixel. For the entire image, there are $2^{524288}$ times of adjustments, it's not feasible in the practical application. For this reason, optimization algorithms have been employed in information hiding to find the optimal embedding positions. For example, genetic algorithm had been exploited in digital watermarking [21].

In this paper, we propose a new LSB steganographic algorithm using the graph coloring problem (GCP). The (GCP) algorithm [12] is used to locate the optimum pixel. Using this positions, the artifacts caused by the steganography can be eliminated and the image quality will not be degraded. Embed secret bits into each pixel in the block independent set by modified LSB substitution method. Adjustment will be executed to extract secret data exactly by recipient and to minimize the perceptual distortion resulted from embedding. The experimental results show that our proposed method provides a large embedding capacity, and the quality of the stego-image is improved as well. The paper is organized as follows. In the next section, the LSB method is presented. In Section 3, we present the graph coloring problem. Section 4 describes our resolution approach. The experimental results will be in Section 5. Finally, the conclusion is given in Section 6.

## 2. Hiding Methods in Image Steganography

Image steganography has been widely studied by researchers [14]. There are a variety of methods using which information can be hidden in images that we recall:
Least Significant Bit Replacement Technique: In image steganography almost all data hiding techniques try to alter insignificant information in the cover image. Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image. For instance, a simple scheme proposed, is to place the embedding data at the least significant bit (LSB) of each pixel in the cover image [14,15,16]. The altered image is called stego-image. Altering LSB doesn't change the quality of image to human perception but this scheme is sensitive a variety of image processing attacks like compression, cropping etc. We will be emphasizing more on this technique for the various image formats.
Moderate Significant Bit Replacement Technique: The moderate significant bits of each pixel in the cover image can be used to embed the secret message. This method improves sensitivity to modification, but it degrades the quality of stego-image. Experiments have shown that the length of hidden messages embedded in the least significant bits of signal samples can be estimated with relatively high precision. A comprehensive survey of steganographic methods is presented in [14].

### *2.1. The LSB Method*

The least significant bit i.e. the eighth bit of each pixel inside an image is changed to a bit of the secret message. When using a 24-bit image, one can store 3 bits in each pixel by changing a bit of each of the red, green and blue color components, since each of them is represented by a byte. An $800 \times 600$ pixel image, can thus store a total amount of $1,440,000$ bits or $180,000$ bytes of embedded data For example a grid for 3 pixels of a 24-bit image can be as follows :

(01010101 01011100 11011000)
(10110110 11111100 00110100)
(11011110 10110010 01101011)

When the number 300, which binary representation is 101101100 is embedded into the least significant bits of this part of the image, the resulting grid is as follows :

(00101101 00011100 11011101)
(10100111 11000100 00001101)
(11010011 10110010 01100010)

Here the number 300 was embedded into the first 8 bytes of the grid, only the 5 bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. The human eye cannot perceive these changes thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the LSB without noticing the difference.

## 3. Graph coloring problem

The graph coloring problem (GCP) is a combinatorial optimization problems that is widely studied in computer science and mathematics. It's related to several traditional applications in various fields such as telecommunications, bioinformatics, and Internet. Among these applications we find, timetable problem [1], crew scheduling [2], supply chain and logistics optimization [3], register allocation [4], air traffic flow management [5] and frequency assignment problem [6]. Garey and Johnson in 1979 [7], demonstrated that the $k$-coloring problem is NP-complete and that the determination of the chromatic number $\chi(G)$ is NP-hard. Therefore several methods and heuristics have been proposed to solve this problem. The first used algorithms were constructive algorithms. Among the most employed include RFL [8] and DSATUR [9], both approaches use an order constructed dynamically on the vertices. Subsequently a large number of local search algorithms have been intended to solve the coloring problem. Among these methods, the tabu search that is in the first place. Several authors

have introduced this technique in their works [10,11]. Here after we recall some definitions and algorithms in graph coloring.

### *3.1. Definitions*

The graph coloring problem is to assign a color to each vertex so that two adjacent vertices do not have the same color. If the graph contains an edge $(x, y)$, then $x$ and $y$ will have different colors, see Figure 1.

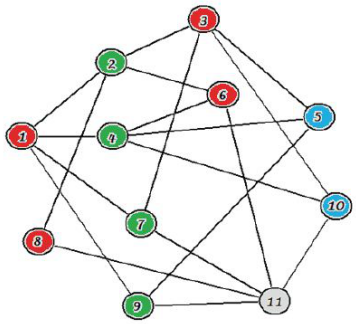



**Figure 1** Example of a graph coloring.

A valid $k$-coloring of vertices in a graph $G = (V, E)$ is an application $c : V \longrightarrow \{1, ..., k\}$ such as $c(x) \neq c(y)$, for all $(x, y) \in E$, the value $c(x)$ associated with vertex $x$ is called color of $x$. If $(x, y) \in E$ and $c(x) = c(y)$ we say that $x$ and $y$ are in conflict. The vertices of the same color define a color class noted $C$, such that there is no edge between two vertices of the same class. Since each color class induces an independent set of $G$, a coloring can also be seen as a partition of $V$ as independent sets.
The chromatic number of a graph $G$ denoted $\chi(G)$ is the smallest number of colors used to color all vertices of $G$ with a valid coloring.

### *3.2. Maximal independent set*

An independent set in a graph $G = (V, E)$ is a subset $S \subseteq V$ of vertices pairwise non adjacent. The problems of maximal cardinality independent set and graph coloring are correlated. Thus, it is normal in coloring problems to search means to restructure the independent sets, that correspond to different given colors. The goal is to increase the size of the independent set to reduce the number of sets, and thus the numbers of colors.
We consider an undirected graph $G = (V, E)$, where $V = \{1, ..., n\}$ denotes the set of vertices of the graph and E denotes the set of edges. For each vertex $i \in V$ we have:

$$nodestar(i) = \{j : \{i, j\} \in E\}$$

$$d_i = card(nodestar(i))$$

$$d_0 = |E| = \text{number of edges}$$

The usual formulation in mathematical programming of the maximal independent set associates a binary variable $x_i$ to every vertex $i \in V$, where $x_i = 1$ if and only if the vertex $i$ is chosen like an element of the independent set. Therefore the problem can be expressed as the following integer programming problem:

$$(IP) \begin{cases} max\ x_0 = \sum(x_i : i \in V) \\ x_i + x_j \leq 1, \{i, j\} \in E \\ x_i\ binary, i \in V \end{cases}$$

To produce a quicker approached solution for (IP), we use the surrogate constraint heuristic, given by replacing the set of all constraints by only one constraint linear combination of the originals. For the problem (IP), we make a simple sum of all constraints like follows:

$$\sum(d_i x_i : i \in V) \leq d_0$$

The surrogate problem associated to (IP) is:

$$(SC) \begin{cases} max\ x_0 = \sum(x_i : i \in V) \\ \sum(d_i x_i : i \in V) \leq d_0 \\ x_i\ binary, i \in V \end{cases}$$

### *3.3. DBG Algorithm*

Let $G = (V, E)$ an undirected graph and $w = (w_1, ...., w_m)^t$ a surrogate constraint multiplier, where $m$ denotes the number of edges. Let (SC) the surrogate relaxation of the (IP) problem, we choose a variable $x_r = 1$. If there exists $j \in nodestar(r)$ and $1 \leq k \leq m$, so that $a_{k,j} = 1$ then $w_k = 0$, for all $1 \leq j \leq n$ and all $1 \leq k \leq m$, [12].

**Algorithm 1:** DBG

1. Let $w = (1, ...., 1)^t$ and $V' = \emptyset$.
2. Calculate the surrogate constraint $wA = \sum_{w_k \neq 0}(a_{k,.})$.
3. Give $i = index(\min(wA)_i : (wA)_i \neq 0)$, let $x_i = 1$ and $V' = V' \cup \{i\}$.
4. For all $j \in$ nodestar(i) if $a_{k,j} = 1$, then $w_k = 0$, if $\sum_{k=1}^{m} w_k = 0$ stop.
   Otherwise return to step 2.

### 3.4. Solving GCP

The decomposition of the graph $G$ as independent subsets $X_1, X_2, ..., X_k$ gives a valid $k$-coloring by assigning to each subset $X_i$ color $i$, $1 \leq i \leq k$. We propose a $k$-coloring to GCP problem using DBG heuristic to construct $k$ color classes of graph $G$, see Algorithm 2.

**Algorithm 2:** Graph coloring

1. While $V \neq \emptyset$.
2. Apply (DBG), give an approximation of the maximal independent set $V'$.
   a. Let $w = (1, ...., 1)^t$ and $V' = \emptyset$
   b. Calculate the surrogate constraint $wA = \sum_{w_k \neq 0}(a_{k,.})$
   c. Give $i = index(\min(wA)_i : (wA)_i \neq 0)$, let $x_i = 1$ and $V' = V' \cup \{i\}$
   d. For all $j \in$ nodestar(i) if $a_{k,j} = 1$, then $w_k = 0$, if $\sum_{k=1}^{m} w_k = 0$ stop. Otherwise return to step b
3. Make $V = V \backslash V'$.
4. If the vertices of $V$ are disjoint stop, otherwise go to 2.
5. End while.

**Algorithm 3:** Proposed steganographic scheme using graph coloring problem

1. Needed: A graph coloring algorithm (DBG), LSB substitution algorithm
2. Input: $I$ Grayscale image, $m$ a message to hide
3. Output: Stego-image $I^{'}$
4. Give $A$ matrix of pixels $(N_1 \times N_2)$ associated to the image
5. Divide $I$ in the form of $N$ blocks $H_l$ of same size
6. $l = 1$
7. While $l \leq N$ and $m \neq \emptyset$.
8. Calculate $s^l = \lceil (\sum |p_i^l - p_{i+1}^l|)/(card(H_l) - 1) \rceil$
9. Give the associated graph to each block $H_l$
10. Apply Algorithm 2 of coloring to $H_l$ to give the color classes $C_{H_l}$
11. Hide the message $m$ on color classes by LSB substitution $p^{'}_{C_{H_l}} = p_{C_{H_l}} + 2LSB(m)$
12. Adjusting block $H_l$:
13. For $i = 1$ to card($C_{H_l}$) do
14. $add = p_i^{'} - p_i$
15. $p_{nodestar(i)} \longleftarrow p_{nodestar(i)} + add$
16. End For
17. $m \longleftarrow m-$2bit-LSB
18. $l \leftarrow l + 1$
19. End While
20. Return $I^{'}$

## 4. Our approach

To escape steganalysis methods force's and prevent them to detect, there has been use of steganography, we use an approach which is to seek the suitable pixels in a given image, in which we can hide a secret message of a reasonable size, for this we must locate the positions of these pixels and render the probability of steganalysis very low. We consider matrix of pixels associated to an image, we propose to hide our message across a line of this matrix (pixel block). For this we use the graph coloring algorithm described previously to determine the optimal positions of pixels. Each row of associated matrix to the image corresponds to an undirected graph $G = (V, E)$ where $V$ denotes the set of vertices and $E$ denotes the set of edges such that two vertices $x$ and $y$ are adjacent if the difference between two pixels $p_i$ and $p_j$ is less than a threshold $s$ where $i \neq j$ and $i, j = 1, ..., card(V)$.

$$|p_i - p_j| < s \tag{1}$$

After giving a valid graph coloring, each color class $C$ corresponds to an independent set, and elements of each independent set indicate the optimal positions in which we can hide the message through LSB method by replacing the last two bits of each pixel noted $p^C$ of a class $C$ by two other bits of message to hide. Our approach based on graph coloring can process large size blocks. For examined images in this paper we have used 512 pixels on each line which makes the task of steganalysis very difficult.
The choice of threshold value $s$ is related to pixels values in each block of the associated matrix of image, and its determination is defined by the following rule.
For each block $H_l$ we have:

$$s^l = \lceil (\sum |p_i^l - p_{i+1}^l|)/(card(V) - 1) \rceil \tag{2}$$

### 4.1. Adjusting

After hiding the message, we carry an adjustment to pixels of block $H_l$ in order to keep the same graph as before hiding. For this we seek the vertices related to each vertex $x$ of a color class noted $nodestar(x)$. Afterwards we add the same value to their pixels as that added to associated pixel to vertex $x$ before hiding message. In this way we guarantee that the recipient will have the same graph as that before incorporation of message for each block $H_l$.

### 4.2. Extracting algorithm

In the extraction process, we can quickly extract secret data without the original image. Partition the stego-image into $N$ blocks $H_l$, which is identical with the embedding algorithm. For each block $H_l$, the following steps are executed to extract the secret data. The Extracting algorithm is explain in Algorithm 4.

**Algorithm 4:** Extracting algorithm

1. Needed: a graph coloring algorithm
2. Input: $I'$ stego-image Gray-scale
3. Output: $m$ a secret message
4. Divide $I'$ in the form of $N$ blocks $H_l$ of same size
5. $m =$' '
6. $l = 1$
7. While $l \leq N$
8. Calculate $s^l = \lceil (\sum |p_i^l - p_{i+1}^l|)/(card(H_l) - 1) \rceil$
9. Give the associated graph to each block $H_l$
10. Apply Algorithm 2 to $H_l$ to give the color classes $C_{H_l}$
11. Extract 2bit-LSB secret data from each pixels in $C_{H_l}$
12. $m \leftarrow m$+2bit-LSB
13. $l \leftarrow l + 1$
14. End While
15. Return $m$

### *4.3. Example*

We consider block of pixels below. We apply the Algorithm 3 to hide a message $m = (1\ 0\ 0\ 1)$ in optimal positions. The threshold corresponding to this block $s = 2$, and therefore the graph $G = (V, E)$ associated to this block contains six vertices, such that $(i, j) \in E$ if $|p_i - p_j| < 2$, see Figure 2. Algorithm 3 provides two optimal positions for the first color class such as $C = (1, 4)$. After hiding the message $m$ by LSB method on those pixels and adjust the other values of pixels to maintain the same original graph, we obtain a new block of pixels as:

The associated pixels to class $(1, 4)$: 103 and 105 after

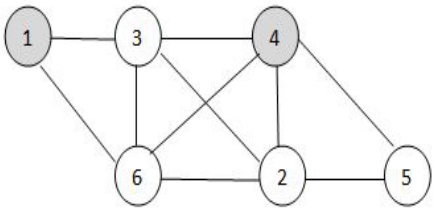



**Figure 2** Associated graph to block A.

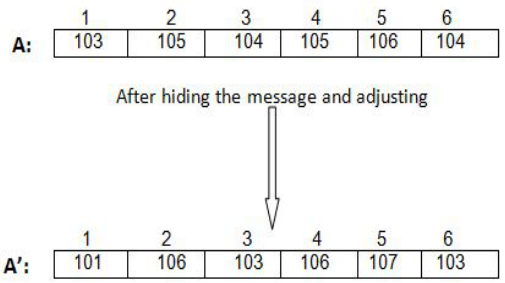
hiding becoming 101 and 106.

The adjusting of the block is done by changing the pixel values associated to the vertices connected to the class $(1, 4)$:

The pixel value associated to vertex 1 has been decremented by 2 then the associated pixel values to $nodestar(1) = (3, 6)$ are also decremented by 2.

The associated pixel value to vertex 4 was incremented by 1, so the associated pixel values to $nodestar(4) = (2, 3, 5, 6)$ are also incremented by 1.

The recipient receives block $A^{'}$. Then he computes the threshold with $s = 4$, he constructs the associated graph to




$A^{'}$ that will be identical to block $A$ (Figure 2). Through, the adjusting done after incorporation of message, he applies the coloring algorithm and then obtain the same class $(1, 4)$ associated to pixels 101 and 106. The recipient has only taking the last two bits of each pixel to construct the message $m$.

## 5. Experimental results

Several experiments were carried out to evaluate our proposed method. Thirty gray-scale images with size equal to $128 \times 128$, $256 \times 256$ and $512 \times 512$ are used as cover images, and four of them are presented in Figures 3, 4, 5, 6. A series of pseudo-random number as secrets bit stream are embedded in the cover images.

The peak signal to noise ratio (PSNR) is used to evaluate the quality of the stego-image, it's expressed in decibels (db). The human visual system is unable to distinguish gray-scale images with a value of PSNR over 36 dB [13]. For an $M \times N$ gray-scale image, the PSNR value is defined as follows:

$$PSNR = 10 \times log_{10} \frac{255 \times 255 \times M \times N}{\sum_{i=1}^{M} \sum_{j=1}^{N} (P_{ij} - Q_{ij})^2} (dB)$$

where $P_{ij}$ and $Q_{ij}$ denote the pixel values in row $i$ and column $j$ of the cover image and the stego-image, respectively, $M$ and $N$ are the image sizes.

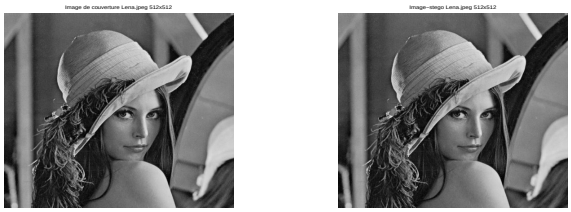
We computed the PSNR for evaluating image quality, obtained after that the steganography hides a secret message in the cover image.

The proposed steganography algorithm is applied to gray-scale cover images "Lena", "Baboon", "Lake" and "Couple" with a size $512 \times 512$ to 8-bit, all experiments were executed on a PC Windows 7, AMD Athlon (tm) X2 dual-core QL-65 (2cpus) 2.1GHz with 4 GB of RAM. The messages are randomly generated.

For Lena image we hid a message $m$ of size 355648 bits, randomly generated on blocks, which each contain 512 pixels, see Figure 3. Similarly we hid a message of size 564544 bits on Baboon image, message of size 676832 bits on Lake image, and another message of size 689120 bits on Couple image, see figures 4, 5 and 6. The provided

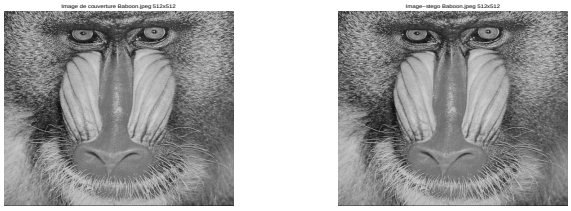
**Table 1** Comparison of PSNR for different size images.

| Cover image | Capacity(bit) | PSNR(db) |
|---|---|---|
| Baboon 512x512 | 564544 | 44.39 |
| Baboon 256x256 | 233472 | 38.86 |
| Baboon 128x128 | 155648 | 35.52 |
| Lake 512x512 | 676832 | 38.11 |
| Lake 256x256 | 311296 | 37.95 |
| Lake 128x128 | 233472 | 35.27 |
| Couple 512x512 | 689120 | 42.76 |
| Couple 256x256 | 307200 | 40.09 |
| Couple 128x128 | 182624 | 37.54 |



(a) (b)

**Figure 3** Hiding a message on Lena image. (a) cover images with size $512 \times 512$, (b) is stego-image.



(c) (d)

**Figure 4** Hiding a message on Baboon image. (c) cover images with size $512 \times 512$, (d) is stego-image.

results in these figures show that there is no difference between the original image and the stego-image for human visual.

## 6. Conclusion

In this paper, we have introduced a new steganographic method to hide information in a gray-scale image. Our approach is based on graph coloring problem in order to locate the optimal positions to hide our message, and be able to increase capacity and imperceptibility of the image after embedding, the use of an appropriate adjusting facilitates the extraction of the hidden message by recipient and

(e) (f)

**Figure 5** Hiding a message on Lake image. (e) cover images with size $512 \times 512$, (f) is stego-image.

(g) (h)

**Figure 6** Hiding a message on Couple image. (g) cover images with size $512 \times 512$, (h) is stego-image.

complicates at the same time the task of steganalysis methods since we change during the adjustment the pixels that we have not incorporated the message. The experimental results has found, and showed that the proposed method gave good values for the parameter PSNR for different image sizes. Which means that there is no difference between the original image and the stego-image, and permet to hide a large capacity of the message.
In future work, we plan to do the following modofications of our given method:

–Investigating the proposed method on color image
–Modifying the proposed approach to embed image inside another image
–Test other optimization algorithm in steganography

## References

[1] D. de Werra, An introduction to timetabling. European Journal of Operational Research, **19** (2) :151-162, (1985).
[2] M. Gamache, A. Hertz and J. O. Ouellet, A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding, Computers Operations Research, **34** (8): 2384-2395, (2007).
[3] A. Lim and F. Wang, Robust graph coloring for uncertain supply chain management. In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)-Track 3-Volume 03. IEEE Computer Society, **11** (8): 263-277, (2005).

**Table 2** Experimental results with different parameters.

| | Lena512x512 | | Baboon512x512 | | Lake512x512 | | Couple512x512 | | Barbara512x512 | | Peppers512x512 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| k-LSB | Cap(bit) | PSNR | Cap(bit) | PSNR | Cap(bit) | PSNR | Cap(bit) | PSNR | Cap(bit) | PSNR | Cap(bit) | PSNR |
| 2-LSB | 355648 | 40.69 | 564544 | 44.39 | 676832 | 38.11 | 689120 | 42.76 | 539692 | 40.51 | 564346 | 41.37 |
| 3-LSB | 390624 | 39.17 | 662538 | 41.72 | 842538 | 36.45 | 749538 | 39.05 | 628752 | 35.26 | 697920 | 38.92 |
| 4-LSB | 523388 | 33.20 | 683384 | 39.08 | 899538 | 33.26 | 879384 | 36.92 | 728692 | 33.82 | 721676 | 34.58 |


[4] M. Allen, G. Kumaran and T. Liu, A combined algorithm for graph-coloring in register allocation. In D. S. Johnson, A. Mehrotra, M. Trick (eds.), Proceedings of the Computational Symposium on Graph Coloring and its Generalizations, pages 100-111, Ithaca, New York, USA, (2002).

[5] N. Barnier and P. Brisset, Graph coloring for air traffic flow management. In CPAIOR'02 : Fourth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems, pages 133-147. Le Croisic, France, (2002).

[6] A. Gamst, Some lower bounds for a class of frequency assignment problems. IEEE Transactions of Vehicular Technology, **35**: 8-14, (1986).

[7] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness. W.H. Freeman and Company, New York, Nantes, France, (1979).

[8] F. T. Leighton, A graph coloring algorithm for large scheduling problems. Journal of Research of the National Bureau of Standards, **84** (6): 489-506, (1979).

[9] D. Brlaz, New methods to color the vertices of a graph. Communications of the ACM, **22** (4): 251-256, (1979).

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

[11] R. Dorne and J.K. Hao, In: Voss, s., martello, s., osman, i.h., roucairol, c.(eds.), tabu search for graph coloring, t-colorings and set t-colorings metaheuristics. Advances and Trends in Local Search Paradigms for Optimization. Kluwer, **117**: 77-92, (1998).

[12] S. M. Douiri and S. Elbernoussi, New Heuristic for the Sum Coloring Problem, Applied Mathematical Sciences, Vol. **5**, no. 63, 3121-3129, 2011.

[13] R.O. El Safy, H. H. Zayed and A. El Dessouki, An adaptive steganography technique based on integer wavelet transform, ICNM International Conference on Networking and Media Convergence, pp 111-117, (2009).

[14] D.W. Bender, N.M. Gruhl and A. Lu, Techniques for data hiding, IBM Syst. J. **35**, 313-316, (1996).

[15] C.M. Wang, N.I. Wu, C.S. Tsai and M.S. Hwang, A high quality steganography method with pixel-value differencing and modulus function, J. Syst. Softw. **81**, 150-158, (2008).

[16] J. Fridrich and P. Lisonek , Grid coloring in steganography", IEEE Transactions on Information Theory, **53** (4): 1547-1549, (2007).

[17] H.J. Highland, Data encryption : a non-mathematical approach, Comput. Secur. **16**, 369-386, (1997).

[18] C.H. Yang, C.Y. Weng, S.J. Wang and H.M. Sun, Adaptive data hiding in edge areas of images with spatial LSB domain systems, IEEE Trans. Inf. Forensics Secur. **3** (3) 488-497, (2008).

[19] C.C. Chang and H.W. Tseng, A steganographic method for digital images using side match, Pattern Recognit. Lett. **25** (12), 1431-1437, (2004).

[20] Y.R. Park, H.H. Kang, S.U. Shin and K.R. Kwon, A Steganographic Scheme in Digital Images Using Information of Neighboring Pixels, Vol. **3612**, Springer-Verlag, Berlin, Germany, pp. 962-967, (2005).

[21] C. Lu, S. Huang, C. Sze, and H. Y. M. Liao. Cocktail Watermarking for Digital Image Protection. In IEEE Transactions on Multimedia **2**, pp.209-224, (2000).

[22] J. Fridrich, M. Goljan and R. Du, Detecting lsb steganography in color, and gray-scale images, IEEE MultiMedia, pp. 22-28, (2001).

[23] A. Westfeld and A. Pfitzmann, Attacks on steganographic systems, Proc. of Information Hiding-Third International Workshop, (1999).