

Weighted bipartite graph for locating optimal LSB substitution for secret embedding*

Cheng-Hsing Yang

*Department of Computer Science
National Pingtung University of Education
Pingtung
Taiwan 900
R.O.C.*

Shiuh-Jeng Wang[†]

*Department of Information Management
Central Police University
Taoyuan
Taiwan 333
R.O.C.*

Abstract

Simple least-significant-bit (LSB) substitution is an approach used to embed a secret image in the least significant bits of the pixels in a host image. In order to reduce the degradation of host image after embedding, an optimal LSB substitution scheme was proposed by [9]. Because the exhaustive search for an optimal solution was very time consuming, however, Wang et al. [9] proposed a genetic algorithm to search for approximate optimal solutions. Also, Chang et al. [10] proposed a dynamic programming strategy to efficiently obtain an optimal solution. In this paper, in virtue of efficient algorithm, we propose a new approach to find an optimal solution, which is inspired on the maximum matching of a weighted bipartite graph. Regardless the time required for constructing a weighted bipartite graph, in theory, the time complexity of our approach is $O(N^3)$ only, it is obviously better than the complexity analysis of $O(2^N)$ in [10], where N is equal to 2^k for k -LSB.

Keywords : Information hiding, LSP, bipartite graph, maximum matching.

*This work was supported in part by National Science Council in R.O.C. under Grant No. NSC-42186F and No. NSC 94-2213-E-015-001.

[†]E-mail: sjwang@mail.cpu.edu.tw

Journal of Discrete Mathematical Sciences & Cryptography

Vol. 9 (2006), No. 1, pp. 153–164

© Taru Publications

1. Introduction

With the development of internet technologies, enormous amounts of data are transmitted by networks, everyday. Because the Internet is an open system, this transmitted data can be stolen, altered, or destroyed. Therefore, ways of transmitting secret data, safely and correctly, has become an important topic. One of the main techniques of achieving this task is data encryption [1, 2], which transforms data into a ciphertext via cipher algorithms. Ciphertext is seemingly a random series of symbols. Only valid receivers are able to decrypt ciphertext, using a cryptography key. Because ciphertexts resemble a stream of meaningless codes, they easily attract grabbers, which may try to either recover them or simply destroy them. To circumvent this problem, data hiding [3,4] techniques offer different methods of transmitting data safely. These data hiding techniques embed secret data into multimedia, such as images or videos. In this paper, we have used images as carriers, referring to them as host images. After embedding secret data into a host image, it is referred to as a stego-image.

Different approaches to data hiding have been proposed for different goals, such as, invisibility, robustness and capacity [5-8,12]. In a data hiding procedure, the host image must not be degraded too much; otherwise, the quality of the stego-image would not be acceptable, with the embedded data being easily detected. Data hiding is easily achieved, using the least-significant-bit-based substitution technique, which replaces the least significant bits of the host image with secret data. In order to achieve a good quality of stego-image, one method uses a substitution matrix to transform secret data values, before they are embedded into the host image. For a k least significant bits substitution, it would take an extended period of time, and exhaustive search, to find an optimal substitution matrix. In order to overcome this long running time, Wang et al. [9] proposed a genetic algorithm to search for an approximate optimal solution, while Chang et al. [10] proposed a dynamic programming strategy to efficiently find an optimal solution. In this paper, we have proposed a different approach, based on the maximum matching of a weighted bipartite graph, to find an optimal substitution matrix. The time complexity of our approach is better than that of Chang et al.

The remainder of this paper is organized as follows: first, we introduce some prior works in Section 2. The approach taken is depicted

in Section 3. Analyses of this approach and comparisons are shown in Section 4. Finally, our conclusions are presented in Section 5.

2. Prior works

In this section, we describe some previous works that have attempted to find an optimal LSB substitution. First, the model of an optimal LSB substitution is depicted. Then, we introduce two prior approaches, the genetic algorithm and dynamic programming strategy, which have been used to solve this problem.

2.1 The optimal LSB substitution problem

As shown in Figure 1, Wang et al. [9] proposed a model to describe the optimal LSB substitution problem. Suppose that the embedded data is an image called secret image C , while the host multimedia is an image called host image H . Both C and H are n -bit gray images. Using simple LSB substitution, the rightmost k least significant bits of H will be replaced by C . C' is defined by decomposing the bit streams of C into several k -bit units and treating each unit as a single pixel. Also, let R be the k -bit residual image, which is derived by extracting the rightmost k least significant bits from each pixel in the host image H . To increase security, the pixel location of C' is randomized by a bijection (i.e., one-to-one and onto) mapping function as follows. Suppose that pixel locations in C' are numbered sequentially from 0 to $m - 1$, where m is the image size. The pixel at location l of C' is transformed into a new location $f(l)$ as follows:

$$f(l) = (k_0 + k_1 \times l) \bmod m. \quad (1)$$

Here k_0 and k_1 are constants, and the greatest common divisor of k_1 and m , is 1. The above procedure will transform C' into a meaningless image C'' . In order to achieve a good embedding result, a substitution matrix $S = \{s_{ij}\}$ is defined by

$$s_{ij} = \begin{cases} 1 & \text{gray value } i \text{ is replaced by gray value } j, \\ 0 & \text{do nothing.} \end{cases} \quad (2)$$

The substitution matrix S is used to replace each pixel with gray value i in C'' , by a pixel with gray value j . Note that exactly one element in each row and one element in each column of S can have a value of 1. Thus, there

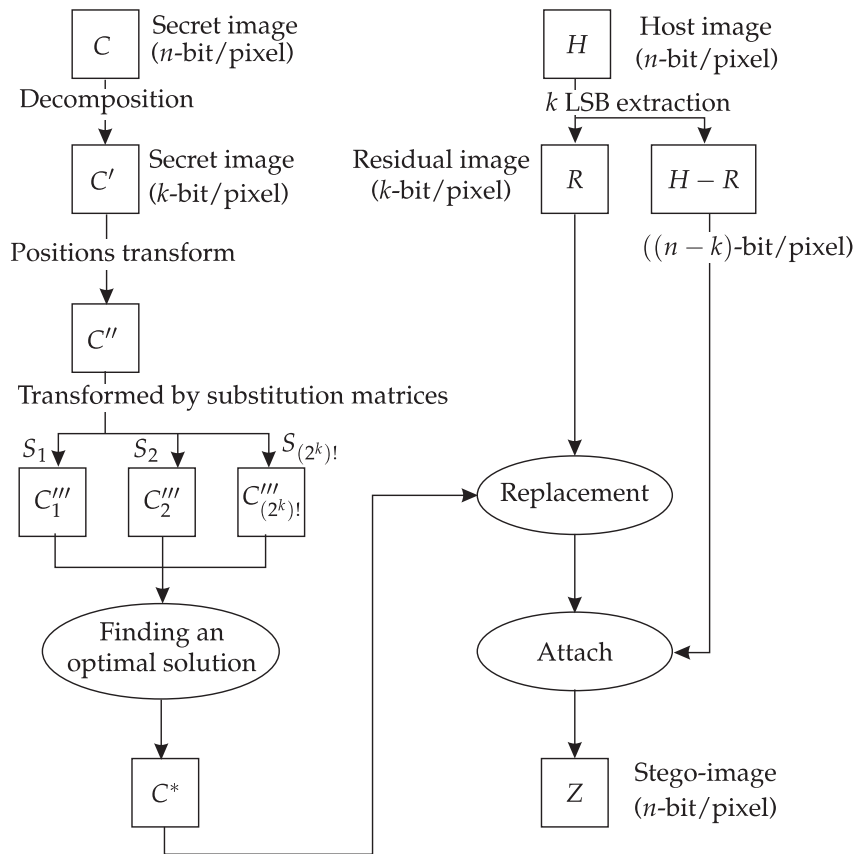


Figure 1
Embedding model via optimal LSB substitution

are $(2^k)!$ different substitution matrices, denoted as $S_1, S_2, \dots, S_{(2^k)!}$. A matrix S_i transforms C'' into C'''_i , and an embedding result can be obtained by replacing R by C'''_i . The resulting image is called the stego-image Z . In order to estimate the quality of the stego-image, a peak signal-to-noise ratio (*PSNR*) is proposed, in order to judge the similarity between the host image and the stego-image. The *PSNR* function is defined as follows:

$$PSNR = 10 \log \frac{(2^n - 1)^2}{MSE} (dB), \quad (3)$$

where n is the number of bits per pixel and MSE is the mean square error

defined as follows:

$$MSE = \frac{1}{m} \sum_{l=0}^{m-1} (z_l - h_l)^2 = \frac{1}{m} \sum_{l=0}^{m-1} (c_l''' - r_l)^2. \quad (4)$$

Here z_l , h_l , c_l''' , and r_l separately represent the pixel gray values of the stego-image Z , the host image H , and secret image C''' ; the residual image R at location l , and m is the image size of Z and H , respectively. Note that an optimal secret image C^* is the secret image C''' , which leads to an optimal embedding result (i.e. the result with the maximum $PSNR$). We also call the substitution matrix, which transforms C'' into C^* , the optimal substitution matrix S^* . We name the problem of finding an optimal substitution matrix S^* as the optimal LSB substitution problem.

2.2 The genetic algorithm

Wang et al. [9] proposed a genetic algorithm to search for approximate optimal solutions. Each substitution matrix S is represented as a chromosome G ; $G = g_0g_1 \dots g_{N-1}$, where $N = 2^k$ and gene g_i means that the gray value i in C'' will be replaced by the gray value g_i . Note that there are $N!$ different chromosomes. In the generic algorithm, some chromosomes are specified as forming an initial population of the first generation. Then, the population of the next generation is created by the following operators, and sieved by a fitness function:

Reproduction: This randomly duplicates some chromosomes for the next generation.

Crossover: This randomly combines the left-hand side of one chromosome, with the right-hand side of another chromosome, to form a new chromosome. The new chromosome must be modified by replacing the repetitive genes with other genes, so that all genes are different, within each chromosome.

Mutation: This randomly chooses a chromosome and exchanges any two genes to form a new chromosome.

Fitness function: A chromosome's fitness function is the summation of all $(c_l''' - r_l)^2$, where c_l''' and r_l separately represent the pixel gray values of the secret image C''' and residual image R at location l , and C''' is the result transformed by this chromosome.

The above process repeats many times, until a predefined requirement is satisfied, or a constant number of iterations are exceeded. Finally,

chromosome G^* , with the minimum fitness value in the final generation, is the chosen substitution matrix.

2.3 Dynamic programming strategy

Chang et al. [10] proposed a dynamic programming strategy to efficiently find the optimal solution. Their main approach redefined the substitution matrix S as the matrix $M_{N \times N}$. The contents of $M_{N \times N}$ were not just 0 or 1, but the summation of the squares differences. If the pixels with gray value i in C''' were transformed into gray value j , the content of $m[i][j]$ in $M_{N \times N}$ represented the total square differences derived between the transformed pixels and the pixels of the corresponding locations in the residual image R . They also represented chromosome $G = g_0 g_1 \dots g_{N-1}$ as a substitution list $\langle g_0, g_1, \dots, g_{N-1} \rangle$. It is obvious that a list $\langle g_0, g_1, \dots, g_{N-1} \rangle$ with a minimum of $m[0][g_0] + m[1][g_1] + \dots + m[N-1][g_{N-1}]$ is the optimal solution for the least-significant substitution. A dynamic approach was proposed to find an optimal list as follows: Suppose that Set is a subset of $\{0, 1, \dots, N-1\}$. Let $m_Cost[r, Set]$ denote the minimum cost of the substitution list picked up from the sub-matrix M' of $M_{N \times N}$, where M' is constructed by rows from $N-1$ to r and columns listed in Set. The dynamic formula is expressed as follows:

$$m_Cost[r, Set] = \min_{j \in Set} \{m[r][j] + m_Cost[r+1, Set - \{j\}]\}. \quad (5)$$

The initial value of $m_Cost[N, \{\}]$ is zero in (5). Then, $m_Cost[0, \{0, 1, \dots, N-1\}]$ is the minimum cost of all $m[0][g_0] + m[1][g_1] + \dots + m[N-1][g_{N-1}]$; it is therefore easy to track back to derive the optimal substitution list $\langle g_0, g_1, \dots, g_{N-1} \rangle$.

3. Our approach

In this section, we present our new approach to finding an optimal solution. Our basic idea is to transform the optimal LSB substitution problem into a *maximum matching problem* of a weighted bipartite graph. Then, we can solve the maximum matching problem, using a well-known approach [11].

A bipartite graph consists of two sets of vertices in which there are weighted edges to connect the two disjoint sets. The *matching* of a bipartite graph is a set of edges with no endpoints in common. A *maximal matching* is a matching to which no edge in the graph can be added. A *maximum*

matching is a matching with maximum weight. Note that a maximum matching must be a maximal matching. Figure 2 shows a bipartite graph with two sets of vertices, $A = \{a_0, a_1, a_2, a_3\}$ and $B = \{b_0, b_1, b_2, b_3\}$ and a maximum matching $M = \{(a_0, b_0), (a_1, b_2), (a_2, b_1), (a_3, b_3)\}$. The weight of each edge in Figure 2 is the same.

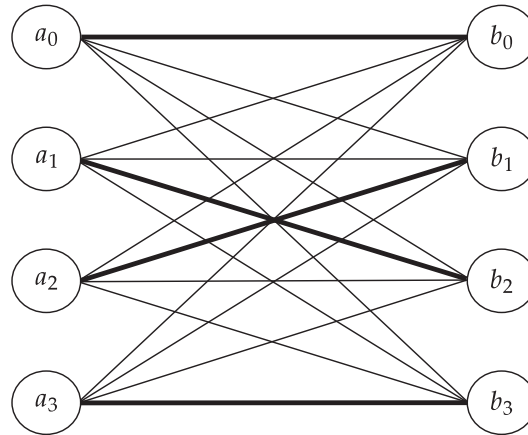


Figure 2
An example of a bipartite graph G and a matching
 $M = \{(a_0, b_0), (a_1, b_2), (a_2, b_1), (a_3, b_3)\}$

Given an optimal LSB substitution problem with a k -bit secret image C'' , it is easy to see that we can represent a substitution matrix by the maximal matching of a bipartite graph. For a k -bit secret image C''' , each pixel has gray value from 0 to $N - 1$, where $N = 2^k$. We construct a bipartite graph G , with two sets of vertices, $A = \{a_0, a_1, \dots, a_{N-1}\}$ and $B = \{b_0, b_1, \dots, b_{N-1}\}$, where both a_i and b_i represent gray value i , for $i = 0, 1, \dots, N - 1$. An edge exists between a_i and b_j , for all $0 \leq i, j \leq N - 1$. Let M be a maximal matching of G . Note that the cardinality of M is equal to N . Then, M is a corresponding substitution S , where any edge $(a_i, b_j) \in M$ is corresponding to s_{ij} with value 1 in S .

Below, we give an example to depict the above idea. Assume that the value of N is 4, the representations of image C'' and substitution matrix S are listed, respectively, as follows:

$$C'' = \begin{bmatrix} 11 & 01 \\ 10 & 10 \end{bmatrix}_2 = \begin{bmatrix} 3 & 1 \\ 2 & 2 \end{bmatrix}_{10},$$

and

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_2.$$

The matching M of G , corresponding to S , is shown in Figure 2. Let C''' be the image transformed from C'' by the substitution matrix S . Then,

$$C''' = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}_{10} = \begin{bmatrix} 11 & 10 \\ 01 & 01 \end{bmatrix}_2.$$

In order to find an optimal substitution matrix S^* , we assign a weight to each edge (a_i, b_j) of the graph G , to find a maximum matching of the graph G . This matching will correspond to the optimal substitution matrix S^* . From the definition of *PSNR*, we note that lower *MSE* will result in higher *PSNR*, and therefore, will obtain a higher quality stego-image. It should be noted that the *MSE* between images Z and H , is identical to the *MSE* between images C''' and R . Therefore, our goal is to find an image C''' , such that, the total square error between images R and C''' is as small as possible. Suppose C''' is transformed from C'' by a matching M . If $(a_i, b_j) \in M$, it means that we can transform each pixel of gray value i in C'' into a pixel of gray value j . We define $\text{cost}(a_i, b_j)$ to be the total square error of all pixels in C''' with gray value j as follows:

$$\text{cost}(a_i, b_j) = \sum_l (c_l''' - r_l)^2 = \sum_l (j - r_l)^2, \quad (6)$$

where the location l must satisfy $c_l'' = i$. Here c_l'' , c_l''' and r_l are the pixel gray values of C'' , C''' and R , respectively, at location l . If we assign each edge (a_i, b_j) with a value $\text{cost}(a_i, b_j)$ in (6), then, for a matching with cardinality N , the summation of all $\text{cost}(a_i, b_j)$ is equal to the total square error between C''' and R . Continuing with the above example, we further suppose that the form of the residual image R is as follows:

$$R = \begin{bmatrix} 2 & 0 \\ 1 & 0 \end{bmatrix}_{10}.$$

Then, the cost values of all edges in the matching M are as follows:

$$\begin{aligned} \text{cost}(a_0, b_0) &= 0, \\ \text{cost}(a_1, b_2) &= (2 - 0)^2 = 4, \end{aligned}$$

$$\begin{aligned}\text{cost}(a_2, b_1) &= (1 - 1)^2 + (1 - 0)^2 = 1, \\ \text{cost}(a_3, b_3) &= (3 - 2)^2 = 1.\end{aligned}$$

The summation of above values is equal to the total square error between images C''' and R , which is $(3 - 2)^2 + (2 - 0)^2 + (1 - 1)^2 + (1 - 0)^2 = 6$.

In order to apply the maximum matching approach of the weighted bipartite graph, we extended the $\text{cost}(a_i, b_j)$ to define the weight of each edge (a_i, b_j) of the bipartite graph G as follows:

$$\text{weight}(a_i, b_j) = \text{MAX_COST} - \text{cost}(a_i, b_j), \quad (7)$$

where $\text{MAX_COST} = \max\{\text{cost}(a_i, b_j) \mid i = 0, 1, \dots, N - 1 \text{ and } j = 0, 1, \dots, N - 1\}$. Up to this point, we have transformed the optimal LSB substitution problem into a maximum matching problem of the weighted bipartite graph.

4. Analyses and comparisons

Before our discussions, we would like to emphasize that the k -LSB approaches are almost limited in the short bit-length of k . While, the complexity when compared with other schemes of k -LSB models will be analyzed in algorithms. In this section, we analyze the time complexity of our matching approach. Then, we compare our scheme to the algorithm of dynamic approach proposed by Chang et al. [10]. Consider a secret image C'' and a residual image R , both of which have m pixels of k bits. The weighted bipartite graph is constructed as follows: Two vertex sets $A = \{a_0, a_1, \dots, a_{N-1}\}$, $B = \{b_0, b_1, \dots, b_{N-1}\}$ and each edge (a_i, b_j) connecting A and B are created first. In order to calculate the cost value of each edge, we scan the pixels of C'' from location 0 to location $m - 1$. Whenever a pixel at location l is scanned, the value $(j - r_l)^2$ is added to $\text{cost}(a_{c_l''}, b_j)$, for all $j = 0, \dots, N - 1$, where c_l'' and r_l are the pixel gray values of C'' and R at location l , respectively. Figure 3 shows the condition of a pixel at location l with grade value i being scanned. After each pixel of C'' has been scanned, all $\text{cost}(a_i, b_j)$ defined in (6) are computed. Then, the value of $\text{weight}(a_i, b_j)$ in (7) can be computed from $\text{cost}(a_i, b_j)$. Obviously, a time complexity of $O(N^2 + Nm)$ is required to construct a weighted bipartite graph.

Finally, a well-know approach [11] is applied to find a maximum matching of the weighted bipartite graph, which takes $O(N^3)$ time. The time complexity of our matching approach is, therefore, $O(N^3 + Nm)$.

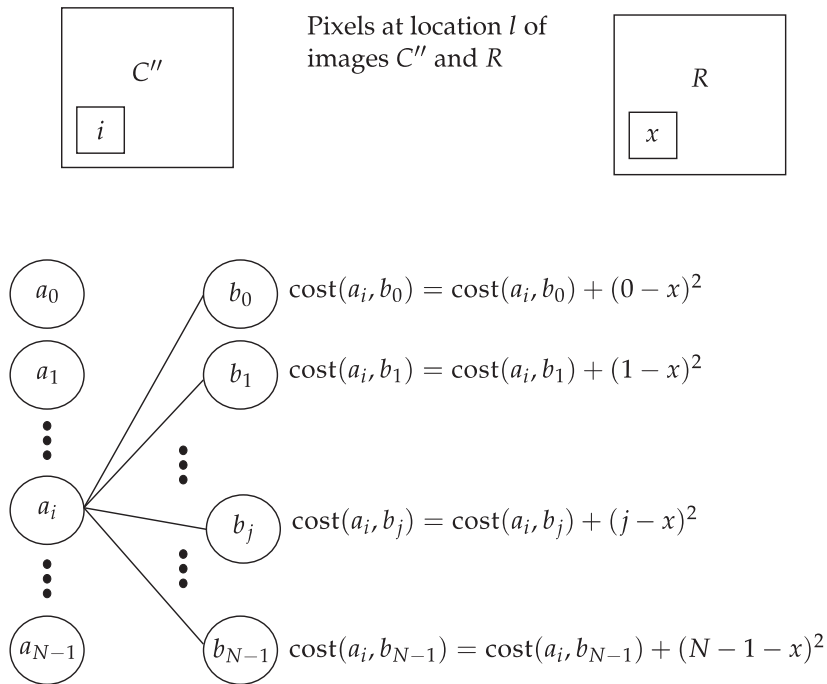


Figure 3

Looking location l , let the gray values of pixels be i and x for images C'' and R , respectively. Once the pixel has been scanned, the changing cost values are shown in the figure

Consider the dynamic programming approach proposed by [10]. It takes $O(Nm)$ to construct a modified substitution matrix. In order to calculate the value of $m_Cost[0, \{0, 1, \dots, N - 1\}]$, all subsets of the set, $\text{Set} = \{0, 1, \dots, N - 1\}$ must be created. Given a size i , there exists $\binom{N}{i}$ different subsets of Set . Therefore, $\binom{N}{0} + \binom{N}{1} + \dots + \binom{N}{i} = 2^N$ different subsets must be created, the time complexity of their approach being $O(2^N + Nm)$. Comparing the time complexity of ours, $O(N^3 + Nm)$, to that of [10], with $O(2^N + Nm)$, although m is usually so large that it would be the main factor affecting the running time, our approach still provides a better solution, reducing the time complexity of $O(2^N)$ to $O(N^3)$. In practice, if the bit-length of k is small, the great improvement of ours in time complexity is not significantly involved in the k -LSB

approaches. However, the model of weighted bipartite graph in our scheme is only an example to optimize the k -LSB substitution in data hiding. It can be systematically transformed to a general model in polynomial time so that the k -LSB substitution problem is bounded in more efficient algorithms, i.e. lower time complexity is required only in the viewpoint of algorithms. It is the key feature highlighted in our contribution.

5. Concluding remarks

In this paper, we have proposed a new approach to solve the optimal LSB substitution problem. Our idea transforms the optimal LSB substitution problem into a maximum matching problem, using a weighted bipartite graph. After constructing the weighted bipartite graph, it only takes $O(N^3)$ time to find an optimal substitution matrix. As a matter of fact, the time evaluation of $N = 2^k$ is only for k -LSB, it is not much large N in LSB approaches. Nevertheless, we propose an efficient model on the basis of time complexity for the solution of LSB substitution. In theoretical view, this complexity is much better than the previous dynamic programming approach, which requires $O(2^N)$ time to solve [10] in order to find an optimal substitution matrix.

References

- [1] Y. H. Chu and S. Chang, Dynamical cryptography based on synchronized chaotic systems, *Electron. Lett.*, Vol. 35 (12) (1999), pp. 974–975.
- [2] H. J. Highland, Data encryption: a non-mathematical approach, *Comput. Security*, Vol. 16 (1997), pp. 369–386.
- [3] D. W. Bender, N. M. Gruhl and A. Lu, Techniques for data hiding, *IBM System J.*, Vol. 35 (1996), pp. 313–336.
- [4] W. D. Chun and T. E. Hsiang, Data hiding in images via multiple-based number conversion and lossy compression, *IEEE Trans. Consumer Electron.*, Vol. 44 (4) (1998), pp. 1406–1412.
- [5] F. A. P. Petitcolas, R. J. Anderson and M.G. Kuhn, Information hiding – a survey, in *Proceeding of IEEE*, Vol. 87 (7) (July 1999), pp. 1062–1078.
- [6] S. Katzenbeisser and F. A. P. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, London, 2000.
- [7] S. D. Lin and C. F. Chen, A robust DCT-based watermarking for copyright protection, *IEEE Trans. Consumer Electron.*, Vol. 46 (3) (2000), pp. 415–421.

- [8] Y. K. Lee and L. H. Chen, High capacity image steganography, in *IEE Proceedings on Vision Image and Signal Processing*, Vol. 147 (3) (2000), pp. 288–294.
- [9] R. Z. Wang, C. F. Lin and J. C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, *Pattern Recognition*, Vol. 34 (2000), pp. 671–683.
- [10] C. C. Chang, J. Y. Hsiao and C. S. Chan, Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy, *Pattern Recognition*, Vol. 36 (2003), pp. 1583–1595.
- [11] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Inc., Upper Saddle River, NJ., 1982.
- [12] S. J. Wang, Steganography of capacity required using modulo operator for embedding secret image, *Applied Mathematics and Computation*, Vol. 164 (1) (2005), pp. 99–116.

Received April, 2005