Information Sciences 324 (2015) 257–269

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A further study of large payloads matrix embedding

Xiaolong Li[a], Siren Cai[a], Weiming Zhang[b], Bin Yang[a,*]

[a] Institute of Computer Science and Technology, Peking University, Beijing 100871, China
[b] School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China

ARTICLE INFO

Article history:
Received 16 June 2014
Revised 16 April 2015
Accepted 28 June 2015
Available online 3 July 2015

Keywords:
Steganography
Matrix embedding
Embedding efficiency
Matrix construction
Sub-optimal search

ABSTRACT

Matrix embedding (ME) is a well-known steganographic method that can improve the embedding efficiency. In ME, the sender and recipient agree in advance on a parity check matrix (PCM) of a binary linear code and utilize the PCM for data embedding. At the decoder side, the embedded message can be extracted by the recipient as the syndrome of the received stego data. The PCM can be taken as any full-ranked matrix and its selection is actually crucial to the embedding performance. In this paper, by extending some previous works, we propose a novel scheme to further improve the embedding efficiency of large payloads ME. First, we utilize a new and specifically designed matrix for ME. Then, instead of finding a coset leader as the modification to the cover which is time consuming, we turn to finding a vector in the coset with relatively small Hamming weight. By the proposed approach, effective and practically feasible large payloads ME can be realized, and extensive experiments show that, a significant increase of embedding efficiency is achieved compared with some state-of-the-art works.

1. Introduction

Steganography is the art and science of covert communication, whose aim is to hide not only a secret message but also the presence of communication [3,10,24]. Specifically, steganographic scheme embeds secret message into innocuous looking cover data (e.g., digital images) by slightly modifying the cover content in such a way that the intended recipient can precisely extract the embedded message. Unlike digital watermarking, steganography is a fragile data hiding technique, and the most important requirement for a steganographic scheme is its security, i.e., the perceptual and statistical undetectability of the hidden message [4,30,32,41]. Generally, there are mainly two ways to improve the stego-security. On one hand, less changes can be made to the cover for the same embedding capacity, which can be realized by the so-called matrix embedding (ME) [5,6,14,38]. By ME, for a given embedding capacity, the embedding distortion can be significantly reduced compared with the classical least-significant-bit (LSB) replacement steganography [2,37]. On the other hand, given an image-content adaptive distortion measurement, more appropriate changes can be made by a well-designed embedding method. For example, it is obvious that embedding modifications operated in rough regions of a natural image are less perceptible than that in flat regions. Moreover, the slight modifications to rough regions cannot be easily perceived by analyzing usual image statistics since the embedding noise is covered by the inherent noise. In this light, a common viewpoint is that the content adaptive approach for steganography has the potential to provide a higher level of security. Up to now, many content adaptive steganographic methods are already proposed [7,8,11,18–21,27–29,33,39,40,42].

[*] Corresponding author. Tel.: +86 10 82529693; fax: +86 10 82529207.
E-mail addresses: lixiaolong@pku.edu.cn (X. Li), caisiren@pku.edu.cn (S. Cai), zhangwm@ustc.edu.cn (W. Zhang), yang_bin@pku.edu.cn, yangbin.pku@gmail.com (B. Yang).

http://dx.doi.org/10.1016/j.ins.2015.06.046
0020-0255/© 2015 Elsevier Inc. All rights reserved.

ME is an effective steganographic scheme which can improve the embedding efficiency and enhance the stego-security [1,12,13,23,26,31]. The embedding efficiency is a commonly used stego-security measurement which is defined as the average number of data bits embedded per one embedding change [12]. In general, when the embedding rate (data bits embedded per pixel) is fixed, a steganographic scheme that has a higher embedding efficiency will be less detectable. In ME, it requires the sender and recipient to agree in advance on a parity check matrix (PCM) of a binary linear code, and the message will be embedded into the cover according to the PCM. At the decoder side, the embedded message can be extracted by the recipient as the syndrome (with respect to the PCM) of the received stego data. This technique was first proposed by Crandall [5], and made popular by the F5 algorithm of Westfeld [38]. Thereafter, the ME technique is systematically investigated by Fridrich et al. [1,12,13]. Particulary, Fridrich et al. proved that the theoretical upper bound of embedding efficiency of LSB-based steganography (i.e., only the LSB of cover pixel value can be changed in data embedding) can be achieved by using binary codes but with huge computational complexity. Moreover, ME with high embedding efficiency is also very valuable for designing content adaptive steganography [8,27].

To apply ME with feasible complexity, Fridrich and Soukal proposed using the PCM in systematic form by exploring random linear codes of small dimensions [13]. This approach can lead to high embedding efficiency for large payloads ME. Notice that, codes for large payloads embedding are important as efficient small payloads embedding can be generated from it [9,42]. Moreover, large payloads ME can also be exploited to design effective ±1 or ±2 embedding [43]. Therefore, in this paper, based on the previous works [13,17,34], we give a further study of large payloads ME by improving the embedding efficiency.

In ME, how to reduce the computational complexity while keeping high embedding efficiency is a critical problem. The methods proposed for this problem so far can be classified into two categories.

The essential idea of the first category is to construct specific matrix or code [13,15,16,22,25,34]. In [13], Fridrich and Soukal exploited the systematic matrix constituted by an identity matrix and a random matrix. In [15], Gao et al. constructed a specific matrix for ME such that the computational complexity is linear. In another work of Gao et al. [16], they studied the optimal matrix which can provide the highest embedding efficiency at a fixed matrix dimension. In [25], Li et al. proposed a method to reduce the embedding distortion via a tree structure of the cover. Li et al.'s method can be formulated as another specific matrix for ME, which is improved by Hou et al. with the majority-vote parity check (MPC) [22]. In a recent work [34], Wang et al. proposed a fast ME method by extending the systematic matrix with several referential columns. This method is experimentally verified better than [15] and [22].

The other category of methods adopt a sub-optimal solution instead of the optimal one, i.e., find a trade-off between computational complexity and embedding efficiency [17,35,36]. In [17], instead of finding a coset leader as the modification to the cover (which is the most time consuming step in ME), Gao et al. turned to finding a vector in the coset which has relatively small Hamming weight. Such a vector can be found at a reduced computational cost. Consequently, higher dimensional matrix can be used for practically implementable ME, and it leads to an improved embedding efficiency. Later on, a similar approach is proposed, in which a vector close to the coset leader is determined as the modification to the cover, and lower complexity is reached merely at the cost of a small deal of embedding efficiency [35].

In this paper, to get practically feasible large payloads ME, we combine the key thoughts of the above two categories of methods: matrix construction and sub-optimal search. The proposed method is an extension of the works of Fridrich and Soukal [13], Gao et al. [17] and Wang et al. [34]. In brief, inspired by [13] and [34], we propose a new and specific matrix construction for ME. The new matrix is constituted by a diagonal-like matrix and a random matrix in which the diagonal-like matrix is constructed by repeatedly placing a small dimensional sub-matrix along its main diagonal. Moreover, following the idea of our previous work [17], instead of the coset leader, we simply find a vector in the coset which has relatively small Hamming weight. By the proposed approach, practically efficient large payloads ME can be realized, and extensive experiments show that, a significant increase of embedding efficiency is achieved compared with the previous works [13,17,34].

The rest of this paper is organized as follows. The ME using systematic matrix [13] and the related works [17,34] are introduced in Section 2. The proposed method is presented in Section 3. The experimental results are shown in Section 4. Finally, our work is concluded in the last section.

2. Related works

2.1. Matrix embedding using systematic matrix

The presentation of ME in this subsection is based on the work of Fridrich and Soukal [13].

Let $\mathbf{H}$ be a PCM of a binary $[n,k]$ linear code $\mathcal{C}$. The matrix $\mathbf{H}$ can be actually taken as any full-ranked $(n-k)\times n$ binary matrix. Since different PCMs of a given linear code result in steganographic schemes with a same embedding efficiency, here we only consider the PCM in systematic form as follows:

$$\mathbf{H} = (\mathbf{I}_{n-k}, \mathbf{R}) \tag{1}$$

where $\mathbf{I}_{n-k}$ is an $(n-k)\times(n-k)$ identity matrix and $\mathbf{R}$ is a random matrix of dimension $(n-k)\times k$, i.e., each element of $\mathbf{R}$ is randomly chosen from $\{0,1\}$.

ME can be defined using $\mathbf{H}$ as follows. Let $\mathbf{c}\in\mathbb{F}_2^n$ be a cover sequence composed of LSBs of cover pixel values and $\mathbf{m}\in\mathbb{F}_2^{n-k}$ be a to-be-embedded message. For convenience, we will use in the following context either row or column vectors depending on the choice. To embed $\mathbf{m}$ into $\mathbf{c}$, the key issue is to choose a vector denoted as $f(\mathbf{c},\mathbf{m})$ from the coset

$$\mathcal{C}_{\mathbf{H}}(\mathbf{u}) \triangleq \left\{\mathbf{v} \in \mathbb{F}_2^n : \mathbf{Hv} = \mathbf{u}\right\} \tag{2}$$

where $\mathbf{u} = \mathbf{m} - \mathbf{Hc}$, i.e., finding a solution to $\mathbf{Hv} = \mathbf{u}$. We will see later that the vector $f(\mathbf{c}, \mathbf{m})$ is actually the modification to the cover image. Therefore, to minimize the embedding distortion, $f(\mathbf{c}, \mathbf{m})$ will be taken as a coset leader, i.e., the vector in the coset which has a minimum Hamming weight

$$f(\mathbf{c}, \mathbf{m}) = \underset{\mathbf{v} \in \mathcal{C}_{\mathbf{H}}(\mathbf{u})}{\arg\min}\, w(\mathbf{v}) \tag{3}$$

where $w(\cdot)$ is the Hamming weight. To this end, we first precalculate and store in memory the 2^k vectors of the coset

$$\mathcal{C}_{\mathbf{H}}(\mathbf{0}_{n-k}) = \left\{\mathbf{v} \in \mathbb{F}_2^n : \mathbf{Hv} = \mathbf{0}_{n-k}\right\} \tag{4}$$

where $\mathbf{0}_t$ means the zero vector of length t. This preprocessing step is done before data embedding, and the required memory space is $n2^k$ bits. Then for data embedding, by utilizing the relation that

$$\mathcal{C}_{\mathbf{H}}(\mathbf{u}) = (\mathbf{u}, \mathbf{0}_k) + \mathcal{C}_{\mathbf{H}}(\mathbf{0}_{n-k}) \tag{5}$$

the Hamming weight of each vector in $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$ is computed based on the precalculated and stored vectors in $\mathcal{C}_{\mathbf{H}}(\mathbf{0}_{n-k})$, and the vector with the minimum Hamming weight is selected as $f(\mathbf{c}, \mathbf{m})$. Finally, take $\mathbf{s} = \mathbf{c} + f(\mathbf{c}, \mathbf{m})$ and replace LSBs of cover pixel values by $\mathbf{s}$ to get the stego image.

By this embedding procedure, $n-k$ bits are embedded into n cover pixels, and thus the embedding rate in bits per pixel (bpp) is $(n-k)/n$. And, the embedding distortion can be computed as the average distortion for each message $\mathbf{m} \in \mathbb{F}_2^{n-k}$, i.e.,

$$\frac{\sum_{\mathbf{m}\in\mathbb{F}_2^{n-k}} w(f(\mathbf{c}, \mathbf{m}))}{n2^{n-k}}. \tag{6}$$

Accordingly, the embedding efficiency is

$$\frac{\text{embedding rate}}{\text{embedding distortion}} = \frac{(n-k)2^{n-k}}{\sum_{\mathbf{m}\in\mathbb{F}_2^{n-k}} w(f(\mathbf{c}, \mathbf{m}))}. \tag{7}$$

The embedding distortion and efficiency are in fact independent to the cover sequence $\mathbf{c}$ since one can verify that $\sum_{\mathbf{m}\in\mathbb{F}_2^{n-k}} w(f(\mathbf{c}_1, \mathbf{m})) = \sum_{\mathbf{m}\in\mathbb{F}_2^{n-k}} w(f(\mathbf{c}_2, \mathbf{m}))$ holds for every $c_1, c_2 \in \mathbb{F}_2^n$.

The data extraction procedure of ME is simple. We only need to compute the syndrome $\mathbf{Hs}$ where $\mathbf{s}$ is the LSBs of stego pixel values. After data embedding, we have

$$\mathbf{Hs} = \mathbf{H}(\mathbf{c} + f(\mathbf{c}, \mathbf{m})) = \mathbf{Hc} + (\mathbf{m} - \mathbf{Hc}) = \mathbf{m}. \tag{8}$$

It means that the embedded message can be exactly extracted from the stego image.

We give an example to further explain the above notations and data embedding/extraction procedures. Take $(n, k) = (5, 2)$ and a 3×5 systematic matrix $\mathbf{H}$ as

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}. \tag{9}$$

Assume that $\mathbf{c} = (1, 0, 0, 1, 1)$ and $\mathbf{m} = (1, 0, 0)$ is a cover sequence and a to-be-embedded message, respectively. For data embedding, first compute $\mathbf{u} = \mathbf{m} - \mathbf{Hc} = (0, 1, 1)$. Notice that

$$\mathcal{C}_{\mathbf{H}}(\mathbf{0}_{n-k}) = \{(0, 0, 0, 0, 0), (1, 0, 1, 0, 1), (1, 1, 0, 1, 0), (0, 1, 1, 1, 1)\} \tag{10}$$

one can then get, according to (5),

$$\mathcal{C}_{\mathbf{H}}(\mathbf{u}) = \{(0, 1, 1, 0, 0), (1, 1, 0, 0, 1), (1, 0, 1, 1, 0), (0, 0, 0, 1, 1)\}. \tag{11}$$

In this case, there are two vectors (0, 1, 1, 0, 0) and (0, 0, 0, 1, 1) in $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$ which has the minimum Hamming weight. Finally, randomly take one between the two vectors as $f(\mathbf{c}, \mathbf{m})$, say, for example, (0, 1, 1, 0, 0), and determine the stego sequence as $\mathbf{s} = \mathbf{c} + f(\mathbf{c}, \mathbf{m}) = (1, 1, 1, 1, 1)$. Clearly, on the decoder side, one can get the embedded message as $\mathbf{Hs} = (1, 0, 0)$ in which it is correctly extracted. By using this specific matrix, the embedding rate is 0.6, and one can verify that the embedding distortion is

$$\frac{1 \cdot 0 + 5 \cdot 1 + 2 \cdot 2}{5 \cdot 2^3} = 0.225. \tag{12}$$

And, the embedding efficiency is $0.6/0.225 = 8/3$.

The above embedding procedure is implemented with $\mathcal{O}(n2^k)$ computations. The computational complexity per pixel (CCPP) of ME is thus $\mathcal{O}(2^k)$, and it only depends on the code dimension. In practice, k will be taken as a small number such as 15 to keep both low computational complexity and limited storage space, and n varies to provide different embedding rates. Large payloads ME can be then derived by taking sufficiently large n.

2.2. Gao et al.'s work

We now introduce the improved ME proposed by Gao et al. [17].

By the definitions of $\mathbf{H}$ and $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$ in (1) and (2), any vector $\mathbf{v}$ in the coset $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$ can be written in a form of $(\mathbf{u} - \sum_{i=1}^{k} v_i \mathbf{r}_i, v_1, \ldots, v_k)$, where $v_i \in \{0, 1\}$ and $\mathbf{r}_i$ is the i-th column vector of $\mathbf{R}$. Recall here that finding a solution to $\mathbf{Hv} = \mathbf{u}$ is equivalent to expressing $\mathbf{u}$ as a linear combination of column vectors of $\mathbf{H}$. Then, the sum $\sum_{i=1}^{k} v_i$ means the number of vectors $\{\mathbf{r}_1, \ldots, \mathbf{r}_k\}$ that participate in expressing $\mathbf{u}$.

For a coset leader $f(\mathbf{c}, \mathbf{m})$ of $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$ defined in (3), suppose that $f(\mathbf{c}, \mathbf{m}) = (\mathbf{u} - \sum_{i=1}^{k} v_i^* \mathbf{r}_i, v_1^*, \ldots, v_k^*)$. In [17], Gao et al. proposed to consider

$$\lambda_{\mathbf{u}} = v_1^* + \cdots + v_k^* \tag{13}$$

which is the number of vectors $\{\mathbf{r}_1, \ldots, \mathbf{r}_k\}$ used by the coset leader for expressing $\mathbf{u}$. Clearly, $\lambda_{\mathbf{u}} \in \{0, \ldots, k\}$. However, in [17], it is experimentally verified that $\lambda_{\mathbf{u}}$ is usually a relatively small number with respect to its maximum, k (see Fig. 1 of [17]). Based on this observation, instead of the coset $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$ composed of all solutions to $\mathbf{Hv} = \mathbf{u}$, Gao et al. proposed to consider a part of solutions defined by

$$\mathcal{C}_{\mathbf{H}}^{t}(\mathbf{u}) = \left\{ \left(\mathbf{u} - \sum_{i=1}^{k} v_i \mathbf{r}_i, v_1, \ldots, v_k \right) : v_i \in \{0, 1\}, \sum_{i=1}^{k} v_i \le t \right\} \tag{14}$$

where $t \le k$ is a given threshold.

Since $\lambda_{\mathbf{u}}$ is usually smaller than k, one can expect that the coset leader is contained in the set $\mathcal{C}_{\mathbf{H}}^{t}(\mathbf{u})$ for a small t. Then, instead of $\mathcal{C}_{\mathbf{H}}(\mathbf{0}_{n-k})$, Gao et al. proposed to precalculate and store in memory only the vectors in $\mathcal{C}_{\mathbf{H}}^{t}(\mathbf{0}_{n-k})$ and then take the vector with the minimum Hamming weight of $\mathcal{C}_{\mathbf{H}}^{t}(\mathbf{u}) = (\mathbf{u}, \mathbf{0}_k) + \mathcal{C}_{\mathbf{H}}^{t}(\mathbf{0}_{n-k})$ as the modification to the cover image. In this situation, instead of the coset leader, a sub-optimal solution is adopted. As a result, the modification to the cover image may be larger compared to the conventional ME presented in the previous subsection, however, the CCPP is reduced from $\mathcal{O}(2^k)$ to $\mathcal{O}(\theta(k, t))$, where

$$\theta(k, t) = \binom{k}{0} + \cdots + \binom{k}{t} \le 2^k. \tag{15}$$

Consequently, higher dimensional matrix can be utilized and it leads to an improved embedding efficiency. For example, for the same computational cost at an embedding rate of 0.5 bpp, compared with the conventional ME employing a 15 × 30 matrix, the embedding efficiency can be increased from 3.54 to 3.84 by Gao et al.'s method with a 30 × 60 matrix and $t = 4$.

2.3. Wang et al.'s work

To enhance the embedding speed and get practical efficient steganography, Wang et al. proposed a fast ME method by extending the systematic matrix via some referential columns [34]. Specifically, Wang et al. proposed to consider the matrix in the form of

$$\mathbf{H} = (\mathbf{I}_{n-k}, \mathbf{R}, \mathbf{J}) \tag{16}$$

where $\mathbf{R}$ is a random matrix of dimension $(n-k) \times k_1$, $\mathbf{J}$ is a $(n-k) \times k_2$ matrix whose i-th column is

$$\mathbf{j}_i = \left(\mathbf{0}_{t_1}, \ldots, \mathbf{0}_{t_{i-1}}, \mathbf{1}_{t_i}, \mathbf{0}_{t_{i+1}}, \ldots, \mathbf{0}_{t_{k_2}} \right) \tag{17}$$

where $k_1 + k_2 = k$ and t_i is defined by

$$t_i = \begin{cases} \lfloor \frac{n-k}{k_2} \rfloor & \text{if } i < k_2 \\ (n-k) - (k_2 - 1)\lfloor \frac{n-k}{k_2} \rfloor & \text{if } i = k_2 \end{cases}. \tag{18}$$

For example, the following 6 × 11 matrix is such an instance with $(n, k, k_1, k_2) = (11, 5, 2, 3)$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{19}$$

With the matrix defined in (16), every vector in the coset $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$ can be written as $\mathbf{v} = (\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2)$ where $\mathbf{v}_0, \mathbf{v}_1$ and $\mathbf{v}_2$ satisfy

$$\mathbf{v}_0 + \mathbf{R}\mathbf{v}_1 + \mathbf{J}\mathbf{v}_2 = \mathbf{u}. \tag{20}$$

Rewrite $\mathbf{u} - \mathbf{R}\mathbf{v}_1$ as $(\mathbf{w}_1, \ldots, \mathbf{w}_{k_2})$ where the length of $\mathbf{w}_i$ is t_i, and $\mathbf{v}_0$ as $(\mathbf{v}_{0,1}, \ldots, \mathbf{v}_{0,k_2})$ where the length of $\mathbf{v}_{0,\,i}$ is also t_i. Then, we have, according to (20)

$$\left(\mathbf{v}_{0,1}, \ldots, \mathbf{v}_{0,k_2} \right) + \mathbf{J}\mathbf{v}_2 = \left(\mathbf{w}_1, \ldots, \mathbf{w}_{k_2} \right). \tag{21}$$

Table 1
CCPP of the methods of Fridrich and Soukal [13], Gao et al. [17] and Wang et al. [34].

Fridrich and Soukal	Gao et al.	Wang et al.
$\mathcal{O}(2^k)$	$\mathcal{O}(\theta(k,t))$	$\mathcal{O}(2^{k_1})$

This yields that, for each $i \in \{1, \ldots, k_2\}$

$$\mathbf{v}_{0,i} + v_{2,i}\mathbf{1}_{t_i} = \mathbf{w}_i \tag{22}$$

where we take $\mathbf{v}_2 = (v_{2,1}, \ldots, v_{2,k_2})$. Thus

$$\mathbf{v}_{0,i} = \begin{cases} \mathbf{w}_i, & \text{if } v_{2,i} = 0 \\ \mathbf{1}_{t_i} - \mathbf{w}_i, & \text{if } v_{2,i} = 1 \end{cases} \tag{23}$$

and the Hamming weight of **v** is clearly

$$w(\mathbf{v}_1) + \sum_{i=1}^{k_2} (g_i + v_{2,i}) \tag{24}$$

where g_i is the Hamming weight of $\mathbf{v}_{0,\,i}$,

$$g_i = \begin{cases} w(\mathbf{w}_i), & \text{if } v_{2,i} = 0 \\ t_i - w(\mathbf{w}_i), & \text{if } v_{2,i} = 1 \end{cases}. \tag{25}$$

Consequently, the coset leader of $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$ can be determined by finding the vector $\mathbf{v}_1 \in \mathbb{F}_2^{k_1}$ that minimizing the following quantity:

$$w(\mathbf{v}_1) + \sum_{i=1}^{k_2} \min\{w(\mathbf{w}_i), t_i - w(\mathbf{w}_i) + 1\} \tag{26}$$

with $(\mathbf{w}_1, \ldots, \mathbf{w}_{k_2}) = \mathbf{u} - \mathbf{R}\mathbf{v}_1$.

By Wang et al.'s approach, compared with the conventional ME, the CCPP is reduced from $\mathcal{O}(2^k)$ to $\mathcal{O}(2^{k_1})$. As a result, at the same CCPP, higher dimensional matrix can be conducted and a better embedding efficiency is achieved. For example, by this method using a 20 × 40 matrix with $(k_1, k_2) = (15, 5)$, the embedding efficiency can be increased from 3.54 to 3.70 compared with the conventional ME.

For a summarization, the CCPP of the methods [13,17,34] presented in this section are listed in Table 1. For embedding rates 0.5 and 0.8 bpp, the comparison of embedding efficiency for these methods at the same CCPP is shown in Tables 2 and 3, respectively. According to the two tables, an observation is that, the embedding efficiency is better if employing a larger dimensional matrix.

Finally, we remark that Wang et al.'s method can be improved by incorporating it with Gao et al.'s sub-optimal search strategy. The idea is straightforward. By the matrix **H** defined in (16), for a coset leader $f(\mathbf{c}, \mathbf{m})$ of $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$ expressed as

$$\left(\mathbf{u} - \sum_{i=1}^{k_1} v_i^* \mathbf{r}_i - \sum_{i=1}^{k_2} v_{i+k_1}^* \mathbf{j}_i, v_1^*, \ldots, v_k^*\right) \tag{27}$$

where $\mathbf{r}_i$ and $\mathbf{j}_i$ are respectively column vectors of **R** and **J**, we define

Table 2
Comparison of embedding efficiency between the methods of Fridrich and Soukal [13], Gao et al. [17] and Wang et al. [34], for an embedding rate of 0.5 bpp. The dimension and parameters for the employed matrix of each method are also given.

	Fridrich and Soukal	Gao et al.	Wang et al.
Parameters	$(n,k) = (30, 15)$	$(n,k,t) = (60, 30, 4)$	$(n,k_1,k_2) = (40, 15, 5)$
Matrix dimension	15 × 30	30 × 60	20 × 40
Embedding efficiency	3.54	3.84	3.70

Table 3
Comparison of embedding efficiency between the methods of Fridrich and Soukal [13], Gao et al. [17] and Wang et al. [34], for an embedding rate of 0.8 bpp. The dimension and parameters for the employed matrix of each method are also given.

	Fridrich and Soukal	Gao et al.	Wang et al.
Parameters	$(n,k) = (75, 15)$	$(n,k,t) = (90, 18, 6)$	$(n,k_1,k_2) = (100, 15, 5)$
Matrix dimension	60 × 75	72 × 90	80 × 100
Embedding efficiency	3.01	3.03	3.06

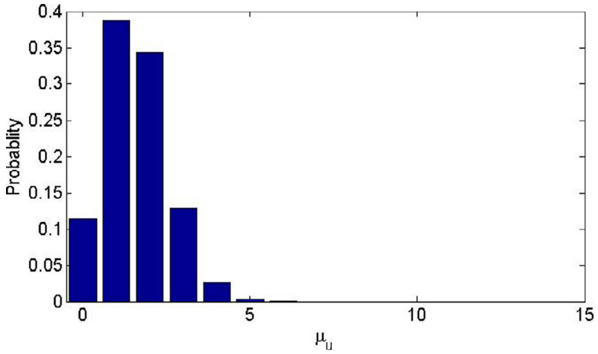

Fig. 1. Distribution of $\mu_{\mathbf{u}} \in \{0, \ldots, 15\}$ of Wang et al.'s method using a 20 × 40 matrix with $(k_1, k_2) = (15, 5)$.

$$\mu_{\mathbf{u}} = v_1^* + \cdots + v_{k_1}^*. \tag{28}$$

Similar to the case of $\lambda_{\mathbf{u}}$ defined in (13), one can also verify that $\mu_{\mathbf{u}}$ is usually a relatively small number with respect to its maximum, k_1. For example, we show in Fig. 1 the statistical distribution of $\mu_{\mathbf{u}} \in \{0, \ldots, 15\}$ of Wang et al.'s method using a 20 × 40 matrix with $(k_1, k_2) = (15, 5)$. This means, the coset leader $f(\mathbf{c}, \mathbf{m})$ probably lies in the set

$$\left\{ \left(\mathbf{u} - \sum_{i=1}^{k_1} v_i \mathbf{r}_i - \sum_{i=1}^{k_2} v_{i+k_1} \mathbf{j}_i, v_1, \ldots, v_k \right) : v_i \in \{0, 1\}, \sum_{i=1}^{k_1} v_i \leq t \right\} \tag{29}$$

with $t \ll k_1$. Based on this observation, instead of all $\mathbf{v}_1 \in \mathbb{F}_2^{k_1}$, one may consider only the vectors $\mathbf{v}_1 \in \mathbb{F}_2^{k_1}$ with $w(\mathbf{v}_1) \leq t$, to minimize (26). In this way, the optimality is lost and only a sub-optimal solution is selected instead of the coset leader, however, the CCPP can be reduced from $\mathcal{O}(2^{k_1})$ to $\mathcal{O}(\theta(k_1, t))$, where the function θ is defined in (15). For example, Wang et al.'s CCPP is $\mathcal{O}(2^{15})$ when taking a 20 × 40 matrix. At the same CCPP, by the improvement, one can utilize a 40 × 80 matrix with $(k_1, k_2) = (30, 10)$ and $t = 4$, and Wang et al.'s embedding efficiency can be increased from 3.70 to 3.94. Regarding Table 2, an even higher embedding efficiency is thus achieved.

3. Proposed method

The proposed method is motivated by the previous works [13,17,34]. We will construct a new matrix for ME and use sub-optimal search to reduce the computational complexity.

First of all, we point out that in Wang et al.'s work [34] when $k_2|(n-k)$ (i.e., k_2 is a factor of $n-k$), by column transformations (i.e., interchange two columns for several times), the matrix (16) can be transformed as a form of $(\mathbf{A}, \mathbf{R})$, where $\mathbf{R}$ is the $(n-k) \times k_1$ random matrix, and $\mathbf{A}$ is a $(n-k) \times (n-k+k_2)$ matrix constructed by repeatedly placing a $a \times (a+1)$ sub-matrix $\mathbf{B} = (\mathbf{I}_a, \mathbf{1}_a)$ along its main diagonal, i.e.,

$$\mathbf{A} = \begin{pmatrix} \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B} \end{pmatrix} \tag{30}$$

with $a = (n-k)/k_2$. For example, for the matrix (19), it can be transformed to $(\mathbf{A}, \mathbf{R})$, where $\mathbf{R}$ is composed of its 7-th and 8-th columns, and

$$\mathbf{A} = \begin{pmatrix} \mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B} \end{pmatrix} \tag{31}$$

with

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}. \tag{32}$$

On the other hand, notice that interchanging two columns of PCM will not change the embedding distortion. Then for Wang et al.'s method, the authors utilized actually aforementioned matrix $(\mathbf{A}, \mathbf{R})$ in which $\mathbf{A}$ is essential a diagonal matrix composed of $\mathbf{B} = (\mathbf{I}_a, \mathbf{1}_a)$ on its main diagonal, and the advantage is, the equation $\mathbf{B}\mathbf{v} = \mathbf{u}$ can be directly solved due to the specific structure of $\mathbf{B}$.

Based on the above discussion, our viewpoint is that the sub-matrix $\mathbf{B}$ in (30) can be taken as any full-ranked matrix. We then propose to consider the matrix $\mathbf{H}$ in a form of $(\mathbf{A}, \mathbf{R})$, where $\mathbf{A}$ is defined in (30) composed of m $a \times b$ full-ranked sub-matrix $\mathbf{B}$

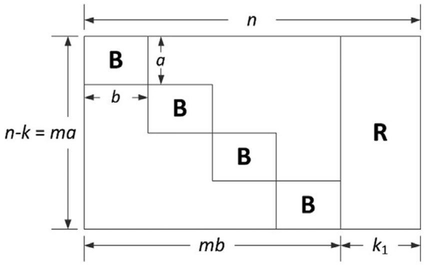

Fig. 2. Construction of the matrix $\mathbf{H} = (\mathbf{A}, \mathbf{R})$ used in our method, in which $\mathbf{A}$ is composed of m sub-matrix $\mathbf{B}$ on its main diagonal.

with $a \le b$, and $\mathbf{R}$ is a $(n-k) \times k_1$ random matrix. Notice that, considering the randomness of $\mathbf{R}$, the full-rank property is required for the sub-matrix $\mathbf{B}$ to guarantee that $\mathbf{H}$ is full-ranked whatever the matrix $\mathbf{R}$ is. In this case, we have

$$\begin{cases} n - k = ma \\ n = mb + k_1 \end{cases}. \tag{33}$$

An illustration for the construction of $\mathbf{H}$ is given in Fig. 2. Next, we consider solving the equation $\mathbf{Hv} = \mathbf{u}$. Rewrite $\mathbf{v}$ as $(\mathbf{v}_1, \mathbf{v}_2)$, where the lengths of $\mathbf{v}_1$ and $\mathbf{v}_2$ are mb and k_1, respectively. Then $\mathbf{Hv} = \mathbf{u}$ is equivalent to

$$\mathbf{A}\mathbf{v}_1 = \mathbf{u} - \mathbf{R}\mathbf{v}_2. \tag{34}$$

Denote respectively $\mathbf{v}_1$ and $\mathbf{u} - \mathbf{R}\mathbf{v}_2$ as $(\mathbf{v}_{1,1}, \ldots, \mathbf{v}_{1,m})$ and $(\mathbf{w}_1, \ldots, \mathbf{w}_m)$ where, for each $i \in \{1, \ldots, m\}$, the lengths of $\mathbf{v}_{1,i}$ and $\mathbf{w}_i$ are b and a, respectively. One can see that, to solve (34) for a fixed $\mathbf{v}_2$, one needs to find solutions to $\mathbf{B}\mathbf{v}_{1,i} = \mathbf{w}_i$ for each $i \in \{1, \ldots, m\}$. In this situation, as the Hamming weight of $\mathbf{v}$ is the sum $w(\mathbf{v}_2) + \sum_{i=1}^{m} w(\mathbf{v}_{1,i})$, one can see that each $\mathbf{v}_{1,i}$ is the coset leader of $\mathcal{C}_{\mathbf{B}}(\mathbf{w}_i)$ if $\mathbf{v}$ is the coset leader of $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$. So, one first needs to find the coset leader of each $\mathcal{C}_{\mathbf{B}}(\mathbf{w}_i)$ for fixed $\mathbf{v}_2$. Thus, finding the solution to $\mathbf{Hv} = \mathbf{u}$ having the minimum Hamming weight, i.e., finding the coset leader of $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$, contains two steps:

- Step 1: For a given $\mathbf{v}_2 \in \mathbb{F}_2^{k_1}$, find the following vector $\mathbf{v}^*_{1,i}$ which is the coset leader of $\mathcal{C}_{\mathbf{B}}(\mathbf{w}_i)$, for each $i \in \{1, \ldots, m\}$

$$\mathbf{v}^*_{1,i} = \arg\min_{\mathbf{B}\mathbf{v}_{1,i} = \mathbf{w}_i} w(\mathbf{v}_{1,i}) \tag{35}$$

  where $(\mathbf{w}_1, \ldots, \mathbf{w}_m) = \mathbf{u} - \mathbf{R}\mathbf{v}_2$.
- Step 2: Determine $\mathbf{v}_2 \in \mathbb{F}_2^{k_1}$ such that

$$w(\mathbf{v}_2) + \sum_{i=1}^{m} w(\mathbf{v}^*_{1,i}) \tag{36}$$

  is minimized. The corresponding vector $(\mathbf{v}^*_{1,1}, \ldots, \mathbf{v}^*_{1,m}, \mathbf{v}_2)$ is just the coset leader of $\mathcal{C}_{\mathbf{H}}(\mathbf{u})$.

We now describe how to speed up the above two steps for the proposed $(\mathbf{A}, \mathbf{R})$-based ME. First, for Step 1, as $\mathbf{B}$ is usually a small dimensional matrix, we then propose to precalculate and store in memory the coset leader of $\mathcal{C}_{\mathbf{B}}(\mathbf{w})$ for each $\mathbf{w} \in \mathbb{F}_2^a$. In this way, the vector $\mathbf{v}^*_{1,i}$ can be found immediately by a look-up table. For Step 2, motivated by the improvement of Wang et al.'s method presented at the end of previous section, we adopt Gao et al.'s sub-optimal search method. Specifically, only the vector $\mathbf{v}_2 \in \mathbb{F}_2^{k_1}$ with small Hamming weight will be processed to minimize (36), and a sub-optimal solution is adopted instead of the coset leader. We will see later that the sub-optimal search strategy is helpful for improving the embedding performance.

We give the detailed data embedding procedure of the proposed $(\mathbf{A}, \mathbf{R})$-based ME as follows.

First of all, we introduce the preprocessing step. Sort first in ascending order all vectors of $\mathbb{F}_2^{k_1}$ according to their Hamming weights, and select the first 2^c vectors to compose a set $\mathcal{S}$, where c is a predefined positive integer no more than k_1. Then, precalculate and store in memory the vector $\mathbf{Rv}$ for each $\mathbf{v} \in \mathcal{S}$. Finally, determine and store in memory a coset leader of the coset $\mathcal{C}_{\mathbf{B}}(\mathbf{w})$ for each $\mathbf{w} \in \mathbb{F}_2^a$, and this can be done by computing all linear combinations of column vectors of $\mathbf{B}$. The memory requirement for the preprocessing step is $(n-k)2^c + b2^a$ bits.

The data embedding procedure contains following steps. Consider here a cover sequence $\mathbf{c} \in \mathbb{F}_2^n$ and a message $\mathbf{m} \in \mathbb{F}_2^{n-k}$. Compute first $\mathbf{u} = \mathbf{m} - \mathbf{Hc}$. Then, for each $\mathbf{v}_2 \in \mathcal{S}$, utilizing the stored vector $\mathbf{R}\mathbf{v}_2$, determine $(\mathbf{w}_1, \ldots, \mathbf{w}_m) = \mathbf{u} - \mathbf{R}\mathbf{v}_2$ where the length of $\mathbf{w}_i$ is a, and take out the coset leader denoted $\mathbf{v}^*_{1,i}$ of the coset $\mathcal{C}_{\mathbf{B}}(\mathbf{w}_i)$ for each $i \in \{1, \ldots, m\}$. Next, determine the vector $\mathbf{v}_2 \in \mathcal{S}$ such that (36) is minimized, and take the corresponding vector $(\mathbf{v}^*_{1,1}, \ldots, \mathbf{v}^*_{1,m}, \mathbf{v}_2)$ as $\mathbf{v}$. Finally, take $\mathbf{s} = \mathbf{c} + \mathbf{v}$ as the stego sequence.

Clearly, the CCPP of the proposed method is $\mathcal{O}(2^c)$ which only depends on c. We call c the complexity parameter. Moreover, the extraction phase of the proposed method is just the same as the conventional ME. One needs only to compute $\mathbf{Hs}$ to extract the embedded message from the stego sequence $\mathbf{s}$. Besides, we remark that, the proposed method includes the conventional ME and Wang et al.'s method as special cases if taking the sub-matrix $\mathbf{B}$ as $\mathbf{I}_a$ and $(\mathbf{I}_a, \mathbf{1}_a)$, respectively.

Finally, we discuss the selection of the sub-matrix $\mathbf{B}$. Notice that the coset leader of the coset $\mathcal{C}_{\mathbf{B}}(\mathbf{w}_i)$, $\mathbf{v}^*_{1,i}$, is involved to generate the embedding distortion (36). Then, roughly speaking, the distortion is less if the expected value of $w(\mathbf{v}^*_{1,i})$ is smaller.

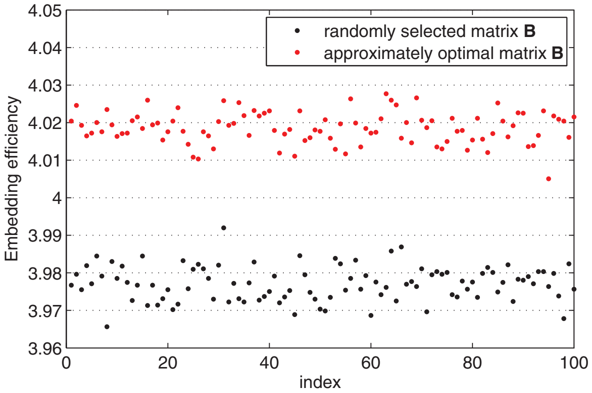

Fig. 3. Comparison of embedding efficiency of our method with different kinds of **B**.

On the other hand, this expected value is actually proportional to the embedding distortion of the conventional ME in which the PCM is taken as **B** [16]. Thus, we may expect that, for two matrices $\mathbf{B}_1$ and $\mathbf{B}_2$, if the embedding distortion of the conventional ME using $\mathbf{B}_1$ is smaller than that of $\mathbf{B}_2$, our method using $\mathbf{B}_1$ will achieve a reduced distortion compared with that of using $\mathbf{B}_2$. This thought is experimentally verified to be true as follows.

In our previous work [16], we studied the so-called optimal matrix defined as the matrix which can provide the highest embedding efficiency for the conventional ME at a fixed matrix dimension. As pointed out in [16], it is still unclear that how to determine an optimal matrix when the matrix dimension is large, then a greedy algorithm is proposed in [16] to determine the approximately optimal matrix (AOM) in systematic form $(\mathbf{I}_a, \mathbf{P})$ in which $\mathbf{P}$ is determined column by column. Specifically, by exhaustive search, we first select a vector $\mathbf{p}_1 \in \mathbb{F}_2^a$ such that the $a \times (a+1)$ matrix $(\mathbf{I}_a, \mathbf{p}_1)$ can achieve the highest embedding efficiency among all $a \times (a+1)$ matrices in systematic form. In other words, for each $\mathbf{p}_1 \in \mathbb{F}_2^a$, according to (7), we compute the embedding efficiency of ME using the $a \times (a+1)$ matrix $(\mathbf{I}_a, \mathbf{p}_1)$, and determine the vector $\mathbf{p}_1$ as the one such that the corresponding embedding efficiency is maximized. Then, also by exhaustive search, we select a vector $\mathbf{p}_2 \in \mathbb{F}_2^a$ such that the $a \times (a+2)$ matrix $(\mathbf{I}_a, \mathbf{p}_1, \mathbf{p}_2)$ can achieve the highest embedding efficiency among all $a \times (a+2)$ matrices in systematic form whose $(a+1)$-th column vector is fixed as $\mathbf{p}_1$. That is to say, for each $\mathbf{p}_2 \in \mathbb{F}_2^a$, according to (7), we compute the embedding efficiency of ME using the $a \times (a+2)$ matrix $(\mathbf{I}_a, \mathbf{p}_1, \mathbf{p}_2)$, and determine the vector $\mathbf{p}_2$ as the one with the maximized embedding efficiency. In the same way, the other column vectors of $\mathbf{P}$ can be determined iteratively. We adopt the AOM obtained in this way to verify the impact of **B** to the embedding performance.

Referring to Fig. 3, it shows the comparison of embedding efficiency of our method with different kinds of **B**. The parameters are fixed as $(a, b, m, k_1, c) = (15, 26, 5, 20, 10)$, and the dimension of **H** is 75 × 150 where the embedding rate is 0.5 bpp. The dimension of **B** is 15 × 26 and it is taken as the AOM or randomly selected for 100 times. Recall that in each case, the matrix **R** is randomly selected when **B** is given. The corresponding embedding efficiencies of our method using the AOM and randomly selected **B** are marked as red and black dots, respectively. According to this figure, one can see that a better embedding efficiency is achieved using the AOM. Thus, to get the best embedding performance for our method, we will take the sub-matrix **B** as the AOM determined using the greedy algorithm. For reference, the 20 × 19 matrix **P** of the 20 × 39 AOM $(\mathbf{I}_{20}, \mathbf{P})$ is given below

$$\begin{pmatrix}
1&0&0&0&0&1&1&1&0&0&1&1&0&1&0&0&1&0&1\\
1&0&0&1&0&1&0&0&1&0&1&1&1&1&0&0&0&0&0\\
1&0&1&0&1&0&0&1&0&1&1&0&1&0&0&0&0&0&1\\
1&0&0&1&1&0&0&1&0&0&0&1&1&0&0&1&1&0&1\\
1&0&1&0&1&1&1&0&1&0&1&1&1&0&0&1&1&1&0\\
1&0&1&0&0&1&0&1&1&1&1&0&0&0&0&1&1&1&0\\
1&0&0&1&1&1&1&1&1&1&0&0&0&0&1&1&0&0&1\\
1&0&0&0&1&0&1&0&1&1&0&0&1&1&0&0&1&1&1\\
1&1&1&1&1&0&0&1&1&1&1&1&0&0&0&1&0&0&0\\
1&1&1&0&0&0&0&0&0&0&1&1&0&1&0&1&1&1&0\\
1&1&1&1&0&1&0&1&0&0&1&0&0&1&0&0&0&0&0\\
1&0&1&1&0&0&1&0&1&1&1&0&1&0&1&1&1&0&1\\
1&1&1&0&0&1&1&0&1&1&0&0&0&0&1&0&0&0&0\\
1&1&0&0&1&1&0&0&1&1&1&1&0&1&1&0&1&0&1\\
1&1&0&1&0&1&1&0&0&0&0&0&1&1&0&1&1&1&1\\
1&1&1&0&1&0&1&1&1&0&0&1&0&1&0&1&1&0&1\\
1&1&0&1&1&0&1&0&0&1&1&0&0&0&1&0&1&1&0\\
1&1&1&1&1&1&1&1&0&1&0&1&1&1&0&1&0&1&1\\
1&1&0&1&0&0&0&1&1&0&0&0&1&0&1&1&0&1&0\\
1&1&0&0&0&0&1&1&0&1&0&0&1&0&0&1&1&0&0
\end{pmatrix}. \tag{37}$$

According to the greedy algorithm, we know that for each $l \in \{1, \ldots, 18\}$, the $20 \times (20 + l)$ AOM $(\mathbf{I}_{20}, \mathbf{P}')$ can be constructed by taking $\mathbf{P}'$ as the first l columns of $\mathbf{P}$.

Finally, before closing this section, we discuss the practical issue of the proposed method that how to communicate the PCM $\mathbf{H} = (\mathbf{A}, \mathbf{R})$. Actually, the matrix can be delivered along with the stego data. Notice that, $\mathbf{R}$ is a random binary matrix and it can be generated in a pseudo-random manner, e.g., using a pseudo random number generator (PRNG). On the other hand, as $\mathbf{A}$ is composed of the sub-matrix $\mathbf{B}$ which is taken as the AOM of a fixed dimension, only the matrix dimension of $\mathbf{B}$ needs to be delivered since the AOM can be predetermined and stored in memory. In this case, to communicate the PCM, only the following information should be delivered: the seed of PRNG and the parameters (a, b, m, k_1, c). The above information can be embedded into some reserved cover pixels using simple LSB replacement.

4. Experimental results

We first illustrate the effectiveness of the sub-optimal search strategy in our method. The experiment is conducted as follows. We consider the embedding rate q/p with conditions

$$p, q \in \{1, \ldots, 10\}, \quad q/p \in [0.5, 1) \quad \text{and} \quad (p, q) = 1, \tag{38}$$

i.e., 16 different embedding rates {1/2, 2/3, 3/4, 3/5, 4/5, 5/6, 4/7, 5/7, 6/7, 5/8, 7/8, 5/9, 7/9, 8/9, 7/10, 9/10} are taken into account in the experiment. The complexity parameter c is fixed as 10. For the paraments a, b and m, they are empirically selected as

$$10 \leq a \leq 20, \quad a + 1 \leq b < 2a, \quad 2 \leq m \leq 20. \tag{39}$$

Recall that $a \times b$ is the dimension of $\mathbf{B}$, m is the numbers of $\mathbf{B}$ used to form $\mathbf{A}$, and the AOM of dimension $a \times b$ is required for each (a, b) satisfying (39). In addition, we remark that, to get high embedding rates, the parameter b is thus restricted by the condition $b < 2a$. Then, for each (p, q) satisfying (38) and (a, b, m) satisfying (39), the parameter k_1 (the width of matrix $\mathbf{R}$) is determined such that the embedding rate is q/p, i.e.,

$$\frac{q}{p} = \frac{n-k}{n} = \frac{ma}{mb + k_1} \tag{40}$$

which is equivalent to

$$k_1 = \frac{p}{q}ma - mb. \tag{41}$$

If k_1 is an integer no less than c, the proposed method is then repeatedly implemented for 100 times using the matrix $\mathbf{H} = (\mathbf{A}, \mathbf{R})$ determined by the parameters (a, b, m, k_1), i.e., the matrix $\mathbf{R}$ is randomly chosen for 100 times, and the results are averaged as the final embedding efficiency denoted $E^c_{a,b,m,k_1}(p, q)$. Notice that, for a given (p, q), there are many parameters (a, b, m, k_1) where the corresponding embedding rate is q/p, we then select the best parameters that can yield the least distortion, i.e., we take

$$E^c(p, q) = \max_{(a,b,m,k_1)} E^c_{a,b,m,k_1}(p, q) \tag{42}$$

as the embedding efficiency of our method at the embedding rate q/p. For example, for $q/p = 1/2$, the best parameters are $(a, b, m, k_1) = (20, 37, 6, 18)$ where the dimensions of $\mathbf{H}$ and $\mathbf{B}$ are 120×240 and 20×37, respectively. On the other hand, reviewing (41), if $k_1 = c$, it means that all the vectors in $\mathbb{F}_2^{k_1}$ are tested for minimizing (36) and a coset leader is thus selected without any loss of optimality. We then define

$$\widetilde{E}^c(p, q) = \max_{(a,b,m),\, k_1 = c} E^c_{a,b,m,k_1}(p, q) \tag{43}$$

in which k_1 is fixed as c and the sub-optimal search is disabled. The comparison between $E^c(p, q)$ and $\widetilde{E}^c(p, q)$ is shown in Fig. 4. One can see that our method with sub-optimal search can provide a larger embedding efficiency especially for moderate embedding rate. It confirms the effectiveness of the sub-optimal search.

The proposed method is then evaluated by comparing it with the previous works [13,17,34]. Referring to Table 1, for a given complexity parameter c, the parameters for [13,17,34] and our method are determined as follows to get the same CCPP:

- For the method [13] described in Section 2.1, k is fixed as c and n varies to generate different embedding rate.
- For the method [17] described in Section 2.2, every (n, k, t) is tested if $\theta(k, t) \leq 2^c$. For a given embedding rate, the optimal (n, k, t) that yields maximum embedding efficiency is selected, and the maximum embedding efficiency is taken as the result.
- For the method [34] described in Section 2.3, (k_1, k_2) is taken as $(c, c/3)$ and n varies to generate different embedding rate.
- For our method, only the embedding rate q/p satisfying (38) is tested and the parameters (a, b, m, k_1) are determined according to (39) and (41). The best parameters are selected for a given (p, q) and $E^c(p, q)$ defined in (42) is taken as our embedding efficiency at the embedding rate q/p.

For both these methods, the matrix $\mathbf{R}$ is randomly chosen for 100 times when the parameters are given, and the corresponding 100 embedding efficiencies are averaged as the final result.

The embedding efficiency comparison is shown in Figs. 5, 6, and 7, for $c = 9$, 12, and 15, respectively. Notice that in these figures, the blue dots in our method are obtained using embedding rate interpolation. That is, for two embedding methods with

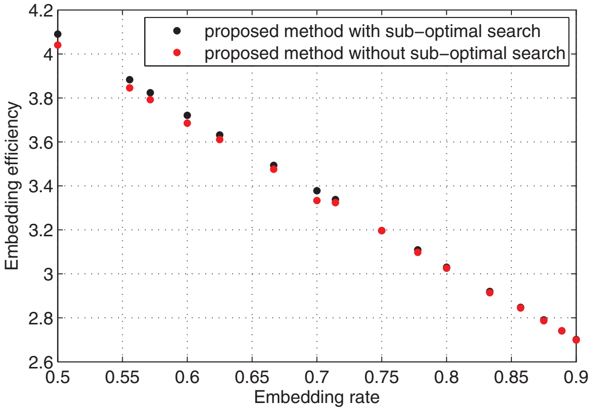

Fig. 4. Comparison of embedding efficiency of our method with and without the sub-optimal search, where the complexity parameter c is fixed as 10.

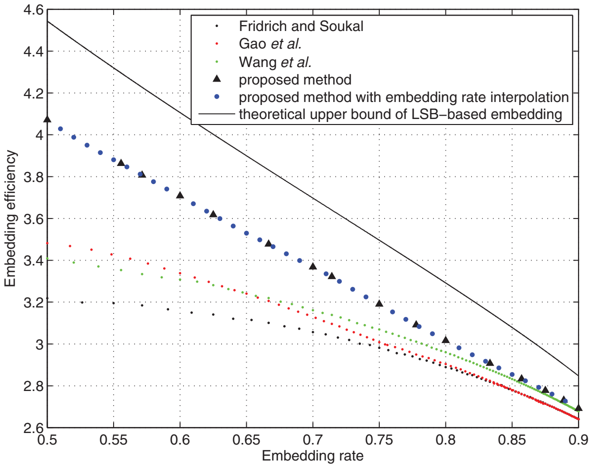

Fig. 5. Comparison of embedding efficiency between our method and the methods of Fridrich et al. [13], Gao et al. [17] and Wang et al. [34], for the same CCPP with the complexity parameter $c = 9$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

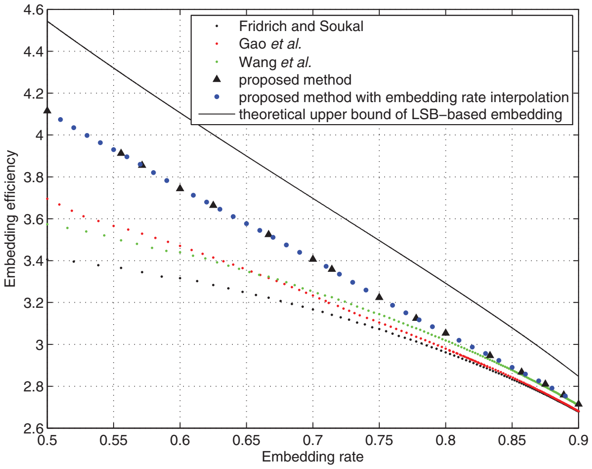

Fig. 6. Comparison of embedding efficiency between our method and the methods of Fridrich et al. [13], Gao et al. [17] and Wang et al. [34], for the same CCPP with the complexity parameter $c = 12$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

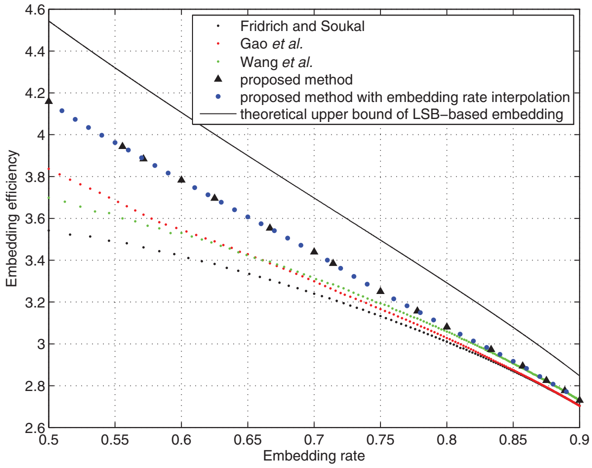

Fig. 7. Comparison of embedding efficiency between our method and the methods of Fridrich et al. [13], Gao et al. [17] and Wang et al. [34], for the same CCPP with the complexity parameter $c = 15$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

Table 4
Best parameters (a, b, m, k_1) that yield the least distortion in the proposed method, the corresponding matrix dimension and embedding efficiency, and the running time for embedding a gray-scale cover image of 1024 × 1024 pixels.

Embedding rate (bpp)		0.5	0.6	0.7	0.8	0.9
Best (a, b, m, k_1)	$c = 9$	(20,37,5,15)	(20,31,6,14)	(20,27,7,11)	(20,23,5,10)	(18,19,9,9)
	$c = 12$	(20,37,6,18)	(20,30,6,20)	(20,26,7,18)	(20,23,7,14)	(18,19,13,13)
	$c = 15$	(20,35,6,30)	(20,30,6,20)	(20,26,7,18)	(12,13,9,18)	(18,19,16,16)
Dimension of **H**	$c = 9$	100 × 200	120 × 200	140 × 200	100 × 125	162 × 180
	$c = 12$	120 × 240	120 × 200	140 × 200	140 × 175	234 × 260
	$c = 15$	120 × 240	120 × 200	140 × 200	108 × 135	288 × 320
Embedding efficiency	$c = 9$	4.07	3.71	3.37	3.02	2.69
	$c = 12$	4.12	3.74	3.41	3.05	2.72
	$c = 15$	4.16	3.78	3.44	3.08	2.73
Running time (s)	$c = 9$	0.09	0.10	0.10	0.21	0.15
	$c = 12$	0.45	0.52	0.54	0.92	0.90
	$c = 15$	4.17	4.86	5.52	6.55	6.34

the pairs of embedding rate and distortion, (r_1, d_1) and (r_2, d_2), one can get an embedding method with embedding rate r and distortion d, where $r = \alpha r_1 + (1 - \alpha)r_2$ and $d = \alpha d_1 + (1 - \alpha)d_2$. Here, $\alpha \in [0, 1]$ is an arbitrary real number. This can be done by a mixed embedding using the two methods and the obvious detail is omitted here. Moreover, for a better illustration, the theoretical upper bound of embedding efficiency of LSB-based embedding is also plotted as the black line. According to these figures, one can see that the proposed method always achieves the best performance among all the tested methods, and our superiority is significant for moderate embedding rates. For example, for an embedding rate of 0.5 bpp, comparing our method with Gao et al.'s which performs the best among [13,17,34], the embedding efficiency is significantly improved from 3.48 to 4.07, 3.70 to 4.12, and 3.84 to 4.16, when c is 9, 12, and 15, respectively. It can be concluded that the proposed method outperforms these state-of-the-art works [13,17,34].

Finally, for different complexity parameter $c \in \{9, 12, 15\}$ and embedding rates of 0.5, 0.6, 0.7, 0.8 and 0.9 bpp, we show in Table 4 the following results:

- The best parameters (a, b, m, k_1) that yield the least distortion in our method, i.e., the parameters (a, b, m, k_1) such that $E^c_{a,b,m,k_1}(p, q)$ is maximized for a given (p, q) (see (42)).
- The dimension of the matrix **H** determined by the best parameters.
- The embedding efficiency of the proposed method using the matrix **H** determined by the best parameters.
- The running time of our method for embedding a gray-scale cover image of 1024 × 1024 pixels. The test is run on a personal computer utilizing a single CPU core of Intel Core i7-4770 running at 3.4 GHz with 8GB RAM. The proposed method is implemented in C++ and compiled under Windows 7 with Visual C++ 10.

According to this table, one can see that a very promising embedding efficiency is obtained by our method with the complexity parameter $c = 12$ in which $1024 \times 1024 \approx 10^6$ cover pixels can be processed within 1 s.

5. Conclusion

In this paper, a study of large payloads ME is proposed by improving the embedding efficiency while keeping low computational complexity. The key issues of the proposed method are the utilization of a new designed matrix and the adoption of the sub-optimal search strategy. This work is an extension of previous methods [13,17,34], and our superiority over these state-of-the-art works is experimentally verified. By the proposed method, effective and practically feasible large payloads ME can be realized. A future work is to try/design different matrix **H** to further enhance the embedding efficiency of ME.

Acknowledgments

This work was supported by National High-tech Technology R&D Program (863 Program) of China under Grant 2013AA013504 and National Natural Science Foundation of China under contract No. 61471009. The work of Weiming Zhang was supported by the Natural Science Foundation of China under Grant 61170234 and Grant 60803155, and by the Strategic Priority Research Program through the Chinese Academy of Sciences under Grant XDA06030601.

References

[1] J. Bierbrauer, J. Fridrich, Constructing good covering codes for applications in steganography, Transactions on Data Hiding and Multimedia Security III, Springer, 2008, pp. 1–22.
[2] C.C. Chang, J.Y. Hsiao, C.S. Chan, Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy, Pattern Recogn. 36 (7) (2003) 1583–1595.
[3] A. Cheddad, J. Condell, K. Curran, P.M. Kevitt, Digital image steganography: survey and analysis of current methods, Signal Process. 90 (3) (2010) 727–752.
[4] R. Cogranne, F. Retraint, Application of hypothesis testing theory for optimal detection of LSB matching data hiding, Signal Process. 93 (7) (2013) 1724–1737.
[5] R. Crandall, Some notes on steganography, 1998[online], http://www.di.unisa.it/˜ads/corso-security/www/CORSO-0203/steganografia/LINKSLOCALI/matrix-encoding.pdf.
[6] M. van Dijk, F. Willems, Embedding information in grayscale images, in: Proceedings of the 22nd Symposium on Information Theory in the Benelux, 2001, pp. 147–154.
[7] N.N. El-Emam, R.A.S. AL-Zubidy, New steganography algorithm to conceal a large amount of secret message using hybrid adaptive neural networks with modified adaptive genetic algorithm, J. Syst. Softw. 86 (6) (2013) 1465–1481.
[8] T. Filler, J. Judas, J. Fridrich, Minimizing additive distortion in steganography using syndrome-trellis codes, IEEE Trans. Inf. Forensic Secur. 6 (2011) 920–935.
[9] J. Fridrich, Asymptotic behavior of the ZZW embedding construction, IEEE Trans. Inf. Forensic Secur. 4 (1) (2009) 151–154.
[10] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications, Cambridge University Press, Cambridge, UK, 2010.
[11] J. Fridrich, P. Lisoněk, D. Soukal, Writing on wet paper, IEEE Trans. Signal Process. 53 (10) (2005) 3923–3935.
[12] J. Fridrich, P. Lisoněk, D. Soukal, On steganographic embedding efficiency, in: Proceedings of the 8th Int. Workshop on Information Hiding, 2006, pp. 282–296.
[13] J. Fridrich, D. Soukal, Matrix embedding for large payloads, IEEE Trans. Inf. Forensic Secur. 1 (3) (2006) 390–395.
[14] F. Galand, G. Kabatiansky, Information hiding by coverings, in: Proceedings of IEEE Information Theory Workshop, ITW, 2003, pp. 151–154.
[15] Y. Gao, X. Li, B. Yang, Constructing specific matrix for efficient matrix embedding, in: Proceedings of IEEE International Conference on Multimedia and Expo, ICME, 2009, pp. 1006–1009.
[16] Y. Gao, X. Li, B. Yang, Employing optimal matrix for efficient matrix embedding, in: Proceedings of Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP, 2009, pp. 161–165.
[17] Y. Gao, X. Li, T. Zeng, B. Yang, Improving embedding efficiency via matrix embedding: a case study, in: Proceedings of IEEE International Conference on Image Processing, ICIP, 2009, pp. 109–112.
[18] T. Han, W. Zhang, C. Wang, N. Yu, Y.F. Zhu, Adaptive ±1 steganography in extended noisy region, Comput. J. 57 (4) (2014) 557–566.
[19] V. Holub, J. Fridrich, Designing steganographic distortion using directional filters, in: Proceedings of IEEE International Workshop on Information Forensics and Security, WIFS, 2012, pp. 234–239.
[20] V. Holub, J. Fridrich, Digital image steganography using universal distortion, in: Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security, 2013, pp. 59–68.
[21] W. Hong, Adaptive image data hiding in edges using patched reference table and pair-wise embedding technique, Inf. Sci. 221 (2013) 473–489.
[22] C. Hou, C. Lu, S. Tsai, W. Tzeng, An optimal data hiding scheme with tree-based parity check, IEEE Trans. Image Process. 20 (3) (2011) 880–886.
[23] M. Khatirinejad, P. Lisoněk, Linear codes for high payload steganography, Discret. Appl. Math. 157 (5) (2009) 971–981.
[24] B. Li, J. He, J. Huang, Y.Q. Shi, A survey on image steganography and steganalysis, J. Inf Hiding Multimed. Signal Process. 2 (2) (2011) 142–172.
[25] R.Y.M. Li, O.C. Au, K.K. Lai, C.K. Yuk, S.Y. Lam, Data hiding with tree based parity check, in: Proceeedings of IEEE International Conference on Multimedia and Expo, ICME, 2007, pp. 635–638.
[26] R.D. Lin, C.T. Chang, A compact covering method to exploit embedding capacity for matrix encoding, Inf. Sci. 188 (2012) 170–181.
[27] G. Liu, W. Liu, Y. Dai, S. Lian, Adaptive steganography based on block complexity and matrix embedding, Multimed. Syst. 20 (2) (2014) 227–238.
[28] D.C. Lou, N.I. Wu, C.M. Wang, Z.H. Lin, C.S. Tsai, A novel adaptive steganography based on local complexity and human vision sensitivity, J. Syst. Softw. 83 (7) (2010) 1236–1248.
[29] W. Luo, F. Huang, J. Huang, Edge adaptive image steganography based on LSB matching revisited, IEEE Trans. Inf. Forensic Secur. 5 (2) (2010) 201–214.
[30] X. Luo, D. Wang, P. Wang, F. Liu, A review on blind detection for image steganography, Signal Process. 88 (9) (2008) 2138–2157.
[31] Q. Mao, A fast algorithm for matrix embedding steganography, Digit. Signal Process. 25 (2014) 248–254.
[32] A. Nissar, A. Mir, Classification of steganalysis techniques: a study, Digit. Signal Process. 20 (6) (2010) 1758–1770.
[33] T. Pevny, T. Filler, P. Bas, Using high-dimensional image models to perform highly undetectable steganography, in: Proceedings of the 12th International Workshop on Information Hiding, 2010, pp. 161–177.
[34] C. Wang, W. Zhang, J. Liu, N. Yu, Fast matrix embedding by matrix extending, IEEE Trans. Inf. Forensic Secur. 7 (1) (2012) 346–350.
[35] J.J. Wang, H. Chen, A suboptimal embedding algorithm with low complexity for binary data hiding, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2012, pp. 1789–1792.
[36] J.J. Wang, C.Y. Lin, H. Chen, T.Y. Yang, A suboptimal embedding algorithm for binary matrix embedding, in: Proceedings of International Symposium on Computer, Consumer and Control, IS3C, 2012, pp. 165–168.
[37] Z.H. Wang, C.C. Chang, M.C. Li, Optimizing least-significant-bit substitution using cat swarm optimization strategy, Inf. Sci. 192 (2012) 98–108.
[38] A. Westfeld, High capacity despite better steganalysis: F5 – a steganographic algorithm, in: Proceedings of the 4th International Workshop on Information Hiding, 2001, pp. 289–302.
[39] D.C. Wu, W.H. Tsai, A steganographic method for images by pixel-value differencing, Pattern Recogn. Lett. 24 (9–10) (2003) 1613–1626.

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

[41] T. Zhang, W. Li, Y. Zhang, E. Zheng, X. Ping, Steganalysis of LSB matching based on statistical modeling of pixel difference distributions, Inf. Sci. 180 (23) (2010) 4685–4694.

[42] W. Zhang, X. Zhang, S. Wang, Maximizing steganographic embedding efficiency by combining Hamming codes and wet paper codes, in: Proceedings of the 10th Int. Workshop on Information Hiding, 2008, pp. 60–71.

[43] X. Zhang, Efficient data hiding with plus-minus one or two, IEEE Signal Process. Lett. 17 (7) (2010) 635–638.