JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 38, 735-748 (2022)
DOI: 10.6688/JISE.202207_38(4).0002

Efficient Provably Algorithms for Boundary Bin-Pair Selection Problem*

CHUN-LIANG JHONG AND HSIN-LUNG WU+
Department of Computer Science and Information Engineering
National Taipei University
New Taipei City, 237 Taiwan
E-mail: liang851015@gmail.com; hsinlung@mail.ntpu.edu.tw

Given a non-negative vector, a boundary bin pair for this vector consists of two different vector entry indices and the boundary bin pair determines a set of bins whose indices are between the boundary bin indices. The sum of these bins is called the bounded area for this given non-negative vector. The boundary bin-pair selection problem (BBPS problem) is an optimization problem in which, n non-negative vectors (called histogram vectors) of length m and a capacity value T are given, the goal is to determine n bin pairs for these n histogram vectors such that the sum of corresponding n bounded areas is maximized under the constraint that the sum of $2n$ boundary bins is greater than or equal to T. The BBPS problem is shown to be equivalent to the modified multiple-histogram-modification problem which is a well-known research topic in reversible data hiding. A brute-force way to solve the BBPS problem requires exponential time. In this paper, we construct several efficient provably algorithms toward solving the BBPS problem. Our proposed algorithms are constructed in a dynamic-programming way. These proposed algorithms achieves the same time complexity $O(nm^2T)$. In order to compare their efficiency, we generate experimental data by constructing histogram vectors from the prediction errors of some test images. Experimental results give a clear performance comparison between our proposed algorithms for the BBPS problem.

Keywords: boundary bin-pair selection, histogram vectors, optimization, dynamic programming, multiple histogram modification

1. INTRODUCTION

Given a non-negative vector $\mathbf{V} = (V_1, \ldots, V_m) \in \mathbb{R}^m$ (called $\mathbf{V}$ a histogram vector) and two indices (ℓ, r) with $1 \leq \ell < r \leq m$, the two bins V_ℓ and V_r are called the boundary bins for $\mathbf{V}$. The boundary bin pair determines a set of bins whose indices are greater than ℓ and smaller than r and thus determines the sum $A_{\ell,r} = \sum_{k:\ell<k<r} V_k$ (called $A_{\ell,r}$ the bounded area). Given n histogram vectors $\mathbf{V}^1, \mathbf{V}^2, \cdots, \mathbf{V}^n \in \mathbb{R}^m$ and a natural number T, the goal of the boundary bin-pair selection problem (BBPS problem) is to find n boundary bin pairs $< (V^i_{\ell_i}, V^i_{r_i}) : 1 \leq i \leq n >$ such that the sum of bounded area $A^1_{\ell_1,r_1}, \ldots, A^n_{\ell_n,r_n}$ is maximized under the constraints that the sum of boundary bins is at least T.

Received July 7, 2020; accepted October 28, 2020.
Communicated by Shi-Chun Tsai.
+ Correponding author.
* This work was supported in part by the Ministry of Science and Technology of Taiwan under contract MOST-109-2221-E-305-005.

The BBPS problem has found applications in some research areas such as reversible data hiding [8]. One of invertible ways to embed secret message into an image is based on the method of multiple histograms modification (MHM) proposed by Li *et al*. [4] where the method for finding good pixel values to embed message is modeled as an optimization problem called the general MHM problem. Up to our knowledge, there is no efficient algorithm which can solve the general MHM problem. Instead of the general MHM problem, the modified MHM problem is suggested in [1, 7]. The BBPS problem captures the modified MHM problem. Thus, an optimal solution of the BBPS problem is an approximation of the optimal solution of the general MHM problem. Therefore, one can use it to obtain reversible data hiding with small distortion.

In [1], a dynamic-programming algorithm is proposed to solve the BBPS problem. However, the time complexity of this algorithm is $O(nT^2 + nm^2T)$. In [7], Qi *et al*. consider a special case of the modified MHM problem in which the selected boundary bin indices satisfy a specific constraint. The considered case can be seen as the special case of the BBPS problem. Qi *et al*. proposed a heuristic algorithm of time complexity $O(nmT)$ toward solving this special case. However, there is no correctness proof for their algorithm.

In this paper, we propose several provably efficient dynamic programming algorithms for solving the BBPS problem. The time complexity of the proposed algorithms is $O(nm^2T)$ which improves the time complexity of the algorithm proposed in [1]. Moreover, the correctness proof for these proposed algorithms is also given in order to guarantee the optimality. In addition, our algorithms also solves the BBPS problem with specific boundary bin-pair constraints within time $O(nmT)$. The correctness of the algorithms is also guaranteed. As an application, our proposed algorithm immediately solves the modified MHM problem and provides an improved MHM-based reversible data hiding scheme. Although our proposed algorithms have the same time complexity order in a theoretical view, we also use experimental data to give a performance comparison for these algorithms in a practical view. Experimental results show that two of our proposed algorithms called **Quick-BBPS** and **fQuick-BBPS** have much better performance than other proposed algorithms.

2. BOUNDARY BIN-PAIR SELECTION PROBLEM

In this section, we define the boundary bin-pair selection problem (BBPS problem) and show an efficient dynamic programming algorithm toward solving it. A vector $\mathbf{V} = (V_1, \cdots, V_m) \in \mathbb{R}^m$ is called a histogram vector if $V_i \geq 0$ for each $1 \leq i \leq m$. Let $\mathbb{R}^m_{\geq 0}$ denote the set of histogram vectors of length m.

Problem. *(Boundary Bin-Pair Selection Problem) Given n vectors* $\mathbf{V}^1, \mathbf{V}^2, \cdots, \mathbf{V}^n \in \mathbb{R}^m_{\geq 0}$ *and a natural number T, find n index pairs* $< (\ell_i, r_i) : 1 \leq i \leq n >$ *with* $1 \leq \ell_i < r_i \leq m$ *that maximizes the sum of bounded areas* $\sum_{i=1}^{n} \sum_{j=\ell_i+1}^{r_i-1} V^i_j$ *subject to* $\sum_{i=1}^{n} V^i_{\ell_i} + V^i_{r_i} \geq T$.

A brute-force way to solve the boundary bin-pair selection problem is to try all possible n pairs $< (\ell_i, r_i) : 1 \leq i \leq n >$ with $\sum_{i=1}^{n} V^i_{l_i} + V^i_{r_i} \geq T$ and find the maximum value of $\sum_{i=1}^{n} \sum_{j=\ell_i+1}^{r_i-1} V^i_j$. However, this requires $O(\binom{m}{2}^n)$ time which is extremely inefficient for large n.

Next, a special subclass of BBPS problem called the restricted boundary bin-pair selection problem is defined as follows.

Problem. *(Restricted Boundary Bin-Pair Selection Problem) Given n vectors $\mathbf{V}^1$, $\mathbf{V}^2, \cdots, \mathbf{V}^n \in \mathbb{R}^m_{\geq 0}$, a natural number T, and n constraint functions $f_1, \ldots, f_n$, find n index pairs $< (\ell_i, f_i(\ell_i)) : 1 \leq i \leq n >$ with $1 \leq \ell_i \leq m$ that maximizes the sum of bounded areas $\sum_{i=1}^n \sum_{j=\ell_i+1}^{f_i(\ell_i)-1} V_j^i$ subject to $\sum_{i=1}^n V_{\ell_i}^i + V_{f_i(\ell_i)}^i \geq T$.*

It is easy to see that the time complexity of the brute-force way to solve the restricted BBPS problem is $O(m^n)$ time. In the next section, we propose an efficient dynamic programming algorithm to solve the general BBPS problem.

3. A PRIMITIVE ALGORITHM FOR THE BBPS PROBLEM

The proposed algorithm is designed in a dynamic programming way. Given n vectors $\mathbf{V}^1, \mathbf{V}^2, \cdots, \mathbf{V}^n \in \mathbb{R}^m_{\geq 0}$ and an integer u, we define

$$B[s,t,u] = \max_{<(\ell_i, r_i): s \leq i \leq t>} \sum_{i=s}^{t} \sum_{j=\ell_i+1}^{r_i-1} V_j^i$$

subject to

$$\sum_{i=s}^{t} V_{\ell_i}^i + V_{r_i}^i \geq u.$$

If there is no $< (\ell_i, r_i) : s \leq i \leq t >$ satisfying $\sum_{i=s}^t V_{\ell_i}^i + V_{r_i}^i \geq u$, then we define $B[s,t,u] = -\infty$. In addition, we define $B[s,t,u] = B[s,t,0]$ for every integer $u < 0$.

Actually, $B[s,t,u]$ is the subproblem of the BBPS problem. In particular, $B[1,n,T]$ is exactly the BBPS problem.

Next, we show some properties of $B[s,t,u]$ and use them to construct an efficient algorithm for the BBPS problem. First of all, we give a recursive relation of $B[s,t,u]$ in the following theorem.

Theorem 1.

$$B[s,t,u] = \begin{cases} \max_{(\ell_s, r_s): V_{\ell_s}^s + V_{r_s}^s \geq u} \sum_{j=l_s+1}^{r_s-1} V_j^s & \textit{if } s = t, \\ \max_{0 \leq k \leq u} \left(B[s,t-1,k] + B[t,t,u-k]\right) & \textit{if } s < t. \end{cases}$$

Proof. We give a recursive relation for $B[s,t,u]$. Note that, for any integer s with $1 \leq s \leq n$ and any integer u with $0 \leq u \leq T$, $B[s,s,u]$ can be computed within time $O(m^2)$ by finding one of $\binom{m}{2}$ pairs (ℓ_s, r_s) with $V_{\ell_s}^s + V_{r_s}^s \geq u$ such that $\sum_{j=\ell_s+1}^{r_s-1} V_j^s$ is maximum. Thus, $B[s,s,u] = \max_{(\ell_s, r_s): V_{\ell_s}^s + V_{r_s}^s \geq u} \sum_{j=\ell_s+1}^{r_s-1} V_j^s$. We will give a more efficient algorithm to compute $B[s,s,u]$ later.

Next, we consider the case to compute $B[s,t,u]$ when $s<t$. By induction on $t-s$, we assume that it is efficient to find an optimal solution for $B[s',t',u']$ for $s \le s' \le t' \le t$ with $t'-s' < t-s$ and $0 \le u' \le T$. Let c be any integer with $s \le c < t$. Suppose that $<(\ell_i, r_i): s \le i \le t>$ is an optimal solution for $B[s,t,u]$. Let u_c be the value with $u_c = \sum_{i=s}^{c} V^i_{\ell_i} + V^i_{r_i}$. Note that $<(\ell_i, r_i): s \le i \le c>$ and $<(\ell_i, r_i): c+1 \le i \le t>$ must be optimal solutions that witness $B[s,c,u_c]$ and $B[c+1,t,u-u_c]$, respectively. Otherwise, we can find another solution $<(\ell'_i, r'_i): s \le i \le t>$ such that $\sum_{i=s}^{t}\sum_{j=\ell'_i+1}^{r'_i-1} V^i_j > \sum_{i=s}^{t}\sum_{j=\ell_i+1}^{r_i-1} V^i_j$ and $\sum_{i=s}^{t} V^i_{\ell'_i} + V^i_{r'_i} \ge u$. However, this contradicts the fact that $<(\ell_i, r_i): s \le i \le t>$ is an optimal solution for $B[s,t,u]$. Therefore, we conclude that $B[s,t,u] = B[s,c,u_c] + B[c+1,t,u-u_c]$. Since u_c is an integer between 0 and T, we obtain the following recurrence relation: $B[s,t,u] = \max_{0 \le k \le u}(B[s,c,k] + B[c+1,t,u-k])$. There are many ways to set the parameter c. In our algorithm, we set $c = t-1$. Thus, the $B[s,t,u]$ can be recursively defined by $B[s,t,u] = \max_{0 \le k \le u}(B[s,t-1,k] + B[t,t,u-k])$. □

Based on Theorem 1, we immediately obtain the following algorithm called **Primitive-BBPS** which computes an optimal solution that witnesses $B[1,n,T]$.

Primitive-BBPS with input $\mathbf{V}^1, \cdots, \mathbf{V}^n \in \mathbb{R}^m_{\ge 0}$
$\arg\max_{<(\ell_i, r_i):1 \le i \le n} \sum_{i=1}^{n}\sum_{j=\ell_i+1}^{r_i-1} V^i_j$ subject to $\sum_{i=1}^{n} V^i_{\ell_i} + V^i_{r_i} \ge T$ is obtained as follows:

1. For $1 \le s \le n$ and for and integer u with $0 \le u \le T$, compute $B[s,s,u] = \max_{(\ell_s, r_s):V^s_{\ell_s}+V^s_{r_s} \ge u} \sum_{j=\ell_s+1}^{r_s-1} V^s_j$ and $P[s,s,u] = \arg\max_{(\ell_s, r_s):V^s_{\ell_s}+V^s_{r_s} \ge u} \sum_{j=\ell_s+1}^{r_s-1} V^s_j$.
2. For $s = 2$ to n and for $0 \le u \le T$, compute
$$B[1,s,u] = \max_{0 \le k \le u}(B[1,s-1,k] + B[s,s,u-k])$$
and
$$P[1,s,u] = \arg\max_{0 \le k \le u}(B[1,s-1,k] + B[s,s,u-k]).$$
3. Set $u_j = 0$ if $j = 0$, $u_j = P[1,j+1,u_{j+1}]$ if $0 < j < n$, and $u_j = T$ if $j = n$.
4. Output $\langle P[i,i,u_i - u_{i-1}] : 1 \le i \le n \rangle$.

Primitive-BBPS is proposed previously in [1]. It is an efficient algorithm since its running time is $O(nT^2 + nT\binom{m}{2})$ which is much lower than $O(\binom{m}{2}^n)$ for large n. However, for applications, the time complexity of **Primitive-BBPS** is still high in terms of T. We will give two improved versions of **Primitive-BBPS** in the next section.

4. IMPROVED VERSIONS OF PRIMITIVE-BBPS

In this section, we improve the time complexity of **Primitive-BBPS** algorithm to be a linear-time algorithm in terms of T. First of all, we show more properties of $B[s,t,u]$.

Lemma 1. *For any fixed s,t with $s \leq t$, $B[s,t,q] \leq B[s,t,p]$ for any integers p,q with $0 \leq p < q \leq T$.*

Proof. The proof is by induction on $d = t-s$. First, we prove the lemma for the base case that $d = t-s = 0$. By way of contradiction, suppose that there are two integers p,q with $p < q$ such that $B[t,t,p] < B[t,t,q]$. Let (ℓ_p, r_p) and (ℓ_q, r_q) denote the optimal selected bin pairs for $B[t,t,p]$ and $B[t,t,q]$, respectively. By definition, $\sum_{j=\ell_q+1}^{r_q-1} V_j^t$ is maximum subject to $V_{\ell_q}^t + V_{r_q}^t \geq q$. Since $p < q$ and $B[t,t,p] < B[t,t,q]$, we have

$$V_{\ell_q}^t + V_{r_q}^t \geq q > p$$

and

$$\sum_{j=\ell_q+1}^{r_q-1} V_j^t > \sum_{j=\ell_p+1}^{r_p-1} V_j^t = B[t,t,p].$$

Thus (ℓ_q, r_q) is a better feasible selection than (ℓ_p, r_p) for the sub-problem $B[t,t,p]$. We get a contradiction.

Now suppose that $B[s',t',u+1] \leq B[s',t',u]$ holds for any t' and s' with $t'-s' \leq d$. Let s,t be two integers such that $t-s = d+1$. We claim that, for any integer u with $0 \leq u < T$, $B[s,t,u+1] \leq B[s,t,u]$. Let k_0 and k_1 be two integers such that

$$\begin{aligned} B[s,t,u] &= \max_{0\leq k\leq u} (B[s,t-1,k] + B[t,t,u-k]) \\ &= B[s,t-1,k_0] + B[t,t,u-k_0] \end{aligned}$$

and

$$\begin{aligned} B[s,t,u+1] &= \max_{0\leq k\leq u} (B[s,t-1,k] + B[t,t,u+1-k]) \\ &= B[s,t-1,k_1] + B[t,t,u+1-k_1]. \end{aligned}$$

We consider the following two cases according to whether k_0 equals k_1. For the first case that $k_0 = k_1$, we have

$$B[t,t,u+1-k_1] \leq B[t,t,u-k_0]$$

since $B[t,t,u]$ is non-increasing with respect to u. Hence $B[s,t,u+1] \leq B[s,t,u]$ in this case. Next we consider the second case that $k_0 \neq k_1$. If $k_1 = u+1$, then

$$\begin{aligned} B[s,t,u+1] &= B[s,t-1,u+1] + B[t,t,0] \\ &\leq B[s,t-1,u] + B[t,t,0] \\ &\leq \max_{0\leq k\leq u} (B[s,t-1,k] + B[t,t,u-k]) \\ &= B[s,t,u] \end{aligned}$$

where the first inequality holds since, by induction, $B[s,t-1,u]$ is non-increasing with respect to u. If $k_1 \neq u+1$, then

$$\begin{aligned} B[s,t,u+1] &= B[s,t-1,k_1]+B[t,t,u+1-k_1] \\ &\leq B[s,t-1,k_1]+B[t,t,u-k_1] \\ &\leq \max_{0\leq k\leq u}(B[s,t-1,k]+B[t,t,u-k]) \\ &= B[s,t,u] \end{aligned}$$

where the first inequality holds since, by induction, $B[t,t,u]$ is non-increasing with respect to u. Therefore, for any fixed $s<t$, $B[s,t,u]$ is non-increasing with respect to u. □

Lemma 2. *Let $\mathbf{V}^t$ be a histogram vector. Let p_t be a peak index such that $V^t_{p_t} = \max\{V^t_1, V^t_2,\ldots,V^t_m\}$. Then $B[t,t,u] = \max_{(\ell,r):\ell\leq p_t\leq r,\ell\neq r}\sum_{i=\ell+1}^{r-1} V^t_i$ subject to $V^t_\ell + V^t_r \geq u$.*

Proof. Without loss of generality, suppose that ℓ',r' are two different indices such that $p_t<\ell'<r'$, $V^t_{\ell'}+V^t_{r'}\geq u$, and $B[t,t,u]=\sum_{i=\ell'+1}^{r'-1}V^t_i$. In this case, we set $\ell''=p_t$. Then we have $\ell''\leq p_t<r'$, $V^t_{\ell''}+V^t_{r'}\geq V^t_{\ell'}+V^t_{r'}\geq u$, and $\sum_{i=\ell''+1}^{r'-1}V^t_i\geq\sum_{i=\ell'+1}^{r'-1}V^t_i=B[t,t,u]$. Therefore, the lemma holds. □

Based on Lemmas 1 and 2, we are able to design a more efficient algorithm called **ImBBPS** which improves the time complexity of **Primitive-BBPS**.

The improved algorithm consists of two parts. The first part is an algorithm which computes each $B[t,t,u]$. The second part is an algorithm for computing general term $B[s,t,u]$ for $s<t$. First, we define some notations. For a histogram vector $\mathbf{V}^t$ and a peak index p_t, we define

$$\Delta_t = \{(\ell,r) : \ell \leq p_t \leq r, \ell \neq r\}.$$

In addition, we define the multi-set

$$\Gamma_t = \{V^t_\ell + V^t_r : (\ell,r)\in\Delta_t\}\cup\{0\}.$$

Note that $|\Gamma_t|-1=|\Delta_t|\leq\binom{m}{2}/2$. Let $0=u_0\leq u_1\leq u_2\leq\ldots\leq u_{k_t}$ be the sorted elements of Γ_t where $k_t=|\Gamma_t|$. For each u_i, let $(\ell_i,r_i)\in\Delta_t$ be its corresponding index pair such that $V^t_{\ell_i}+V^t_{r_i}=u_i$. For each (ℓ_i,r_i), we define

$$\xi_i = \sum_{\ell_i<j<r_i} V^t_j \text{ and } \xi_0 = \sum_{j=1}^{m} V^t_j.$$

Lemma 2 states that, if $u_{i-1}<u\leq u_i$ for some i, then $B[t,t,u]=\max\{\xi_i,\xi_{i+1},\ldots,\xi_{k_t}\}$. By Lemma 1, this implies that $B[t,t,u]$ is a non-increasing step function. Now we introduce the way to compute $B[t,t,u]$ for $0\leq u\leq T$. First, we set

$$B[t,t,T] = \begin{cases} -\infty & \text{if } u_{k_t} < T \\ \max\{\xi_i,\xi_{i+1},\ldots,\xi_{k_t}\} & \text{if } u_{i-1}<T\leq u_i \text{ for some } i. \end{cases}$$

Moreover, we can compute $B[t,t,u]$ for $0 \le u < T$ recursively as follows,

$$B[t,t,u] = \begin{cases} \max\{B[t,t,u+1], \xi_i, \ldots, \xi_{i'}\} & \text{if } u_{i-1} < u \le u_i = u_{i'} < u+1 \le u_{i'+1}, \\ B[t,t,u+1] & \text{if } u_{i-1} < u < u+1 \le u_i, \\ \xi_0 & \text{if } u = u_0. \end{cases}$$

Remark 1. *The time complexity of the above procedure for computing $B[t,t,u]$ for any integer u with $0 \le u \le T$ is $O(|\Delta_t| + T)$ which improves the time complexity $O(\binom{m}{2}T)$ of the primitive algorithm. Thus, the time complexity of computing all terms $B[t,t,u]$ is $O(\sum_{t=1}^{n}(|\Delta_t| + T)) = O(nm^2 + nT)$.*

Before computing $B[s,t,u]$ for $s < t$. We show a useful lemma which helps us to construct the desired algorithm. Before stating it, we define some notations.

Definition 1. *Assume that $u_0 \le u_1 \le \ldots \le u_{k_t}$ be the sorted elements of Γ_t. For any z, let i_z be the minimum index such that $u_{i_z} \ge z$. Let Γ_t^z be the subset of Γ_t defined by $\Gamma_t^z = \{u_0, u_1, \ldots, u_{i_z}\}$.*

Lemma 3. *For any integer s,t with $s \ne t$,*

$$B[s,t,u] = \max_{q \in \Gamma_t^u}(B[s,t-1,u-q] + B[t,t,q]).$$

Proof. Based on Lemma 1, $B[s,t,u]$ is non-increasing for any s,t. Note that, there exists some q' such that

$$\begin{aligned} B[s,t,u] &= \max_{0 \le q \le u}(B[s,t-1,u-q] + B[t,t,q]) \\ &= B[s,t-1,u-q'] + B[t,t,q']. \end{aligned}$$

Let $u_1 \le u_2 \le \ldots \le u_{k_t}$ be the sorted elements of Γ_t. We consider two cases as follows. In the first case, we assume that $u_{i-1} < q' \le u_i \le u$. In this case, we have

$$\begin{aligned} B[s,t-1,u-u_i] + B[t,t,u_i] &\le B[s,t-1,u-q'] + B[t,t,q'] \\ &= B[s,t-1,u-q'] + B[t,t,u_i] \\ &\le B[s,t-1,u-u_i] + B[t,t,u_i] \end{aligned}$$

where the first inequality holds since q' is the optimal selection, the first equality holds since $B[t,t,u]$ is a step function, and the second inequality holds since $B[s,t-1,u]$ is non-increasing. In this case, we obtain that

$$B[s,t,u] = B[s,t-1,u-u_i] + B[t,t,u_i] = \max_{q \in \Gamma_t^u}(B[s,t-1,u-q] + B[t,t,q]).$$

In the second case, we assume that $u_{i-1} < q' \le u < u_i$. In this case, we have

$$\begin{aligned} B[s,t-1,0] + B[t,t,u_i] &= B[s,t-1,0] + B[t,t,u] \\ &\le B[s,t-1,u-q'] + B[t,t,q'] \\ &= B[s,t-1,u-q'] + B[t,t,u_i] \\ &\le B[s,t-1,0] + B[t,t,u_i] \end{aligned}$$

where the first and second equalities hold since $B[t,t,u]$ is a step function, the first inequality holds since q' is the optimal selection, and the second inequality holds since $B[s,t-1,u]$ is non-increasing. So we have $B[s,t,u] = B[s,t-1,0] + B[t,t,u_i]$. Since $u - u_i < 0$, we have $B[s,t,u-u_i] = B[s,t,0]$. Therefore we obtain that

$$B[s,t,u] = B[s,t-1,u-u_i] + B[t,t,u_i] = \max_{q \in \Gamma_t^u} (B[s,t-1,u-q) + B[t,t,q]). \qquad \square$$

By Lemma 3, we can efficiently compute $B[s,t,u]$ by searching those elements $q \in \Gamma_t^u$ instead of the whole elements from 0 to u. Now based on the above discussion, the improved version of **Primitive BBPS** called **ImBBPS** is presented as follows.

ImBBPS with input $\mathbf{V}^1, \cdots, \mathbf{V}^n \in \mathbb{R}^m_{\geq 0}$

$\arg\max_{<(\ell_i,r_i):1\leq i\leq n} \sum_{i=1}^n \sum_{j=\ell_i+1}^{r_i-1} V_j^i$ subject to $\sum_{i=1}^n V_{\ell_i}^i + V_{r_i}^i \geq T$ is obtained as follows:

1. For each histogram vector $\mathbf{V}^t$, compute a peak index p_t for $\mathbf{V}^t$ and generate two sets Δ_t and Γ_t.

2. For $1 \leq t \leq n$, do the following:

   (a) List the sorted elements $u_0, u_1, \ldots, u_{k_t}$ in Γ_t and compute their corresponding values $\xi_0, \xi_1, \xi_2, \ldots, \xi_{k_t}$.

   (b) Compute

   $$B[t,t,T] = \begin{cases} -\infty & \text{if } u_{k_t} < T \\ \max\{\xi_i, \xi_{i+1}, \ldots, \xi_{k_t}\} & \text{if } u_{i-1} < T \leq u_i \text{ for some } i. \end{cases}$$

   (c) For any integer u with $0 \leq u < T$, compute

   $$B[t,t,u] = \begin{cases} \max\{B[t,t,u+1], \xi_i, \ldots, \xi_{i'}\} & \text{if } u_{i-1} < u \leq u_i = u_{i'} < u+1 \leq u_{i'+1} \\ B[t,t,u+1] & \text{if } u_{i-1} < u < u+1 \leq u_i. \\ \xi_0 & \text{if } u = u_0. \end{cases}$$

   and

   $$P[t,t,u] = \arg \max_{(\ell_t,r_t)\in\Delta_t : V_{\ell_t}^t + V_{r_t}^t \geq u} \sum_{j=\ell_t+1}^{r_t-1} V_j^t.$$

3. For $t = 2$ to n and for any integer u with $0 \leq u \leq T$, compute

   $$B[1,t,u] = \max_{k \in \Gamma_t^u} (B[1,t-1,u-k] + B[t,t,k])$$

   and

   $$P[1,t,u] = \arg\max_{k \in \Gamma_t^u} (B[1,t-1,u-k] + B[t,t,k]).$$

4. Set $u_j = 0$ if $j = 0$, $u_j = u_{j+1} - P[1,j+1,u_{j+1}]$ if $0 < j < n$, and $u_j = T$ if $j = n$.

5. Output $\langle P[i,i,u_i-u_{i-1}]:1\le i\le n\rangle$.

Remark 2. *The time complexity of* **ImBBPS** *for computing* $B[1,n,u]$ *for any integer* u *with* $0\le u\le T$ *is* $O((\sum_{t=1}^{n}(|\Delta_t|+T))+T\sum_{t=2}^{n}(|\Delta_t|))=O(nm^2+nT+nm^2T)$.

Remark 3. *For the restricted boundary bin-pair selection problem, one can also apply* **ImBBPS** *on this problem. Since the sizes of* Δ_t *and* Γ_t *is* $O(m)$*, the time complexity of* **ImBBPS** *for computing restricted BBPS problem is* $O((\sum_{t=1}^{n}(|\Delta_t|+T))+T\sum_{t=2}^{n}(|\Delta_t|))=O(nm+nT+nmT)$.

4.1 Another Improved Algorithm for BBPS Problem

In this subsection, we provide a slightly better algorithm which improves the performance of **ImBBPS**. The main bottleneck of **ImBBPS** is the procedure for computing $B[s,t,u]$. Recall that $B[s,t,u]=\max_{q\in\Gamma_t^u}(B[s,t-1,u-q]+B[t,t,q])$. **ImBBPS** needs to check those $q\in\Gamma_t^u$ for computing the maximum value. However, this procedure takes $\Omega(|\Gamma_t|)$ steps for most u. We design a new approach to improve this drawback. To obtain this goal, our key observation is that $B[s,t,u]$ is a non-increasing step function as stated in the following lemma. First of all, we define some notations.

Definition 2. *Assume that* $\tilde{u}^T$ *is the minimum discontinuous input of the function* $B[s,t,\cdot]$ *which is larger than or equal to* T*. Let* $\Gamma_{s,t}^T$ *be the subset of discontinuous inputs of* $B[s,t,\cdot]$ *defined by* $\Gamma_{s,t}^T=\{u:u\le T$ *and* u *is a discontinuous input of* $B[s,t,\cdot]\}\cup\{\tilde{u}^T\}$.

Definition 3. *Given two sets* $A,B\subset\mathbb{R}$*, let* $A+B$ *be the sumset of* A *and* B *defined by* $A+B\doteq\{u'+u'':u'\in A$ *and* $u''\in B\}$.

The following lemma states that $\Gamma_{s,t}^T$ is a finite set and $\Gamma_{s,t}^T$ is computable inductively.

Lemma 4. *For any integers* s,t *with* $s\neq t$*,* $B[s,t,u]$ *is a non-increasing step function with respect to* u*. In particular,* $\Gamma_{s,t}^T\subseteq\Gamma_{s,t-1}^T+\Gamma_{t,t}^T$.

Proof. The non-increasing property has been shown in Lemma 1. In addition, we have shown that $B[t,t,u]$ is a non-increasing step function with respect to u previously. Now we prove that $B[s,t,u]$ is a step function by induction on $d=|t-s|$. Note that, for each function $B[t,t,u]$, $\Gamma_{t,t}^T$ is contained in the support set of Γ_t and is computable within time $O(|\Delta_t|)$. Next, suppose that $\Gamma_{s,t-1}^T$ and $\Gamma_{t,t}^T$ are found. We claim that

$$\Gamma_{s,t}^T\subseteq\Gamma_{s,t-1}^T+\Gamma_{t,t}^T\doteq\{u'+u'':u'\in\Gamma_{s,t-1}^T\text{ and }u''\in\Gamma_{t,t}^T\}.$$

Let $0=\tilde{u}_0<\tilde{u}_1<\ldots<\tilde{u}_k$ be the sorted elements of $\Gamma_{s,t-1}^T+\Gamma_{t,t}^T$. If $u=0$, then $B[s,t,0]=B[s,t-1,0]+B[t,t,0]$. Suppose that $w_0<w_1<\ldots,<w_k$ and $z_0<z_1<\ldots<z_h$ are sorted elements of $\Gamma_{s,t-1}^T$ and $\Gamma_{t,t}^T$, respectively. Now, suppose that $\tilde{u}_i<u\le\tilde{u}_{i+1}$ for some i. By Lemma 3 and the non-increasing property of $B[s,t,u]$, there exist some $z_p\in\Gamma_{t,t}^T$ and some

$w_j, w_{j+1} \in \Gamma^T_{s,t-1}$ such that $w_j < u - z_p \leq w_{j+1}$ and

$$\begin{aligned}
& B[s,t,\tilde{u}_{i+1}] \\
\leq\ & B[s,t,u] \\
=\ & \max_{q \in \Gamma^u_t} \left(B[s,t-1,u-q] + B[t,t,q]\right) \text{ by Lemma 3} \\
=\ & B[s,t-1,u-z_p] + B[t,t,z_p] \text{ by Lemma 3} \\
=\ & B[s,t-1,w_{j+1}] + B[t,t,z_p] \text{ since } w_{j+1} \text{ is the minimum discontinuous input} \\
& \text{larger than } u - z_p \leq \max_{0 \leq q \leq w_{j+1}+z_p} \left(B[s,t-1,w_{j+1}+z_p-q] + B[t,t,q]\right) \\
=\ & B[s,t,w_{j+1}+z_p] \text{ where we define } w_{j+1}+z_p = \tilde{u}_{i'} \in \Gamma^T_{s,t-1} + \Gamma^T_{t,t} \\
=\ & B[s,t,\tilde{u}_{i'}].
\end{aligned}$$

Since $\tilde{u}_i < u \leq \tilde{u}_{i'}$, we have $\tilde{u}_{i+1} \leq \tilde{u}_{i'}$. Since $B[s,t,\cdot]$ is non-increasing, $B[s,t,\tilde{u}_{i'}] \leq B[s,t,\tilde{u}_{i+1}]$. Therefore, $B[s,t,u] = B[s,t,\tilde{u}_{i+1}]$ for any $\tilde{u}_i < u \leq \tilde{u}_{i+1}$. Thus, the subset $\Gamma^T_{s,t}$ is a subset of the sumset $\Gamma^T_{s,t-1} + \Gamma^T_{t,t}$, that is $\Gamma^T_{s,t} \subseteq \Gamma^T_{s,t-1} + \Gamma^T_{t,t}$. □

Based on Lemma 4, we have the following algorithm called **Quick-ImBBPS** for computing $B[1,n,T]$.

Quick-ImBBPS with input $\mathbf{V}^1, \cdots, \mathbf{V}^n \in \mathbb{R}^m_{\geq 0}$
$\arg\max_{<(\ell_i,r_i):1 \leq i \leq n} \sum_{i=1}^n \sum_{j=\ell_i+1}^{r_i-1} V^i_j$ subject to $\sum_{i=1}^n V^i_{\ell_i} + V^i_{r_i} \geq T$ is obtained as follows:

1. For $1 \leq t \leq n$, do the following:
   (a) Compute the set $\Gamma^T_{t,t}$ by examining the discontinuity of $B[t,t,u]$ for $u \in \Gamma_t$.
   (b) Keep $B[t,t,u]$ and $P[t,t,u]$ for $u \in \Gamma^T_{t,t}$.
2. For $t = 2$ to n do the following:
   (a) Compute the sumset $\Gamma^T_{1,t-1} + \Gamma^T_{t,t}$.
   (b) Find the subset $\Gamma^T_{1,t}$ of discontinuous inputs by examining function values of inputs in the sumset $\Gamma^T_{1,t-1} + \Gamma^T_{t,t}$.
   (c) Keep $B[1,t,u]$ and $P[1,t,u]$ for $u \in \Gamma^T_{1,t}$.
3. Set
$$u_j = \begin{cases} \min\{q \in \Gamma^T_{1,n} : q \geq T\} & \text{if } j = n, \\ P[1,j+1,u_{j+1}] & \text{if } 0 < j < n, \\ 0 & \text{if } j = 0. \end{cases}$$
4. Output $\langle P[i,i,u_i - u_{i-1}] : 1 \leq i \leq n \rangle$.

Remark 4. *The time complexity of* **Quick-ImBBPS** *for computing* $B[1,n,T]$ *is*

$$O\left(\sum_{t=1}^{n} |\Gamma_t| + \sum_{t=1}^{n-1} (|\Gamma_{1,t}^{T}| \times |\Gamma_{t+1,t+1}^{T}|)\right) = O\left(nm^2 + nm^2 T\right).$$

Note that the order of the time complexity of **Quick-ImBBPS** *is the same as the one of* **ImBBPS**. *However, in practice, the running time of* **Quick-ImBBPS** *is better than* **ImBBPS**. *We will show some experimental results to support it in the next section.*

The main procedure of **Quick-ImBBPS** is to find the set $\Gamma_{1,t}^{T}$ of discontinuous inputs from the sumset $\Gamma_{1,t-1}^{T} + \Gamma_{t,t}^{T}$. However, it is not necessary to check all values in the sumset $\Gamma_{1,t-1}^{T} + \Gamma_{t,t}^{T}$. Given $u' \in \Gamma_{1,t-1}^{T}$, let us define $q_{T,u'} \doteq \min_{q \in \Gamma_{t,t}^{T}} \{q : q \geq T - u'\}$. Instead of searching values in the whole sumset $\Gamma_{1,t-1}^{T} + \Gamma_{t,t}^{T}$, we only need to check its subset

$$\mathrm{Sub}\left(\Gamma_{1,t-1}^{T} + \Gamma_{t,t}^{T}\right) \doteq \{u' + u'' : u' \in \Gamma_{1,t-1}^{T}, u'' \in \Gamma_{t,t}^{T}, \text{ and } u'' \leq q_{T,u'}\}.$$

For convenience, we call the updated algorithm **fQuick-BBPS** that means faster **Quick-ImBBPS**. In the next section, we demonstrate the efficiency of **Quick-ImBBPS** and **fQuick-ImBBPS** by using some experimental test data.

5. EXPERIMENTAL RESULTS

In this section, we generate some sets of histogram vectors as our experimental data in order to show the efficiency of the algorithm **Quick-BBPS**. In [1], it is shown that the BBPS problem is equivalent to the so-called modified multiple-histogram-modification problem (the modified MHM problem) where it is raised as an application of reversible data hiding on images. We refer readers to the appendix of this paper where the background knowledge of the modified MHM problem is introduced. Based on the above reason, the experimental data is generated from Six standard 512×512 gray-scale images including Lena, Baboon, Airplane, Boat, Lake and Barbara selected from [6]. For each image, we generate two sets of histogram vectors by using the techniques proposed in [4]. Each set consists of 16 histogram vectors of length 127, that is $n = 16$ and $m = 127$ in our setting. The first set of histogram vectors is constructed from the prediction error of the shadow pixels of the images. The second set of histogram vectors is constructed from the prediction errors of the blank pixels of the watermarked image. In Fig. 1, the set of histogram vectors constructed from the prediction errors of the shadow pixels of the image Lena is presented as an illustration. Each test set is assigned a capacity value T.

In Table 1, we show the total number of times for updating the table $B[s,t,u]$ in order to compute the final value $B[1,n,T]$ for sets of histogram vectors constructed from prediction errors for shadow and blank pixels, respectively.

6. CONCLUDING

In this paper, we study the so-called boundary-bin-pair selection problem (BBPS problem) which is arose from the multiple-histogram-modification problem in reversible

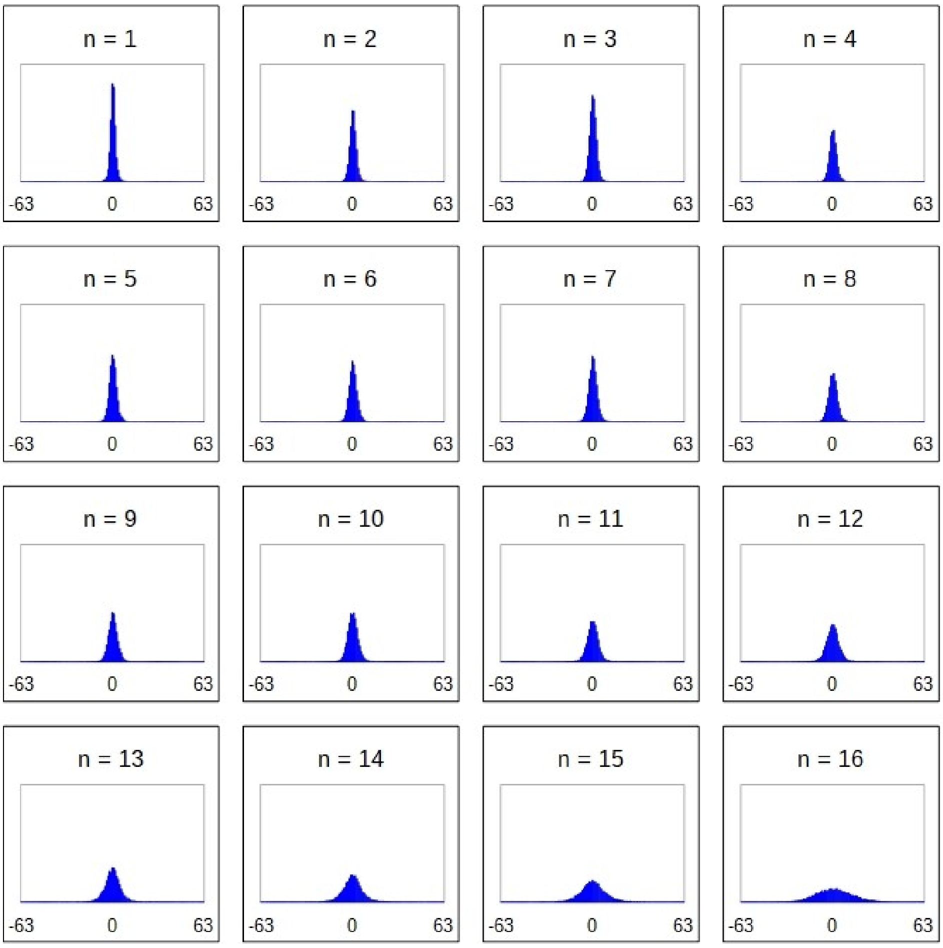

Fig. 1. The histogram vectors constructed from the prediction errors of the shadow pixels of the image Lena.

data hiding. We propose several efficient provably algorithms toward solving the BBPS problem with correctness proof as a guarantee. Since these algorithms achieve the same order of time complexity in a theoretical view, we provide some experimental results to demonstrate the efficiency of some proposed algorithms.

REFERENCES

1. Y.-H. Chou, "Index-pair selection problem with application to efficient reversible data hiding," Master Thesis, Department of Computer Science and Information Engineering, National Taipei University, 2016.
2. G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, and C. Roux, "Reversible watermarking based on invariant image classification and dynamic histogram shifting,"

Table 1. Number of times for updating $B[s,t,u]$ in order to compute $B[1,n,T]$ under the parameter setting $n = 16$ and $m = 127$ where I-B, Q-B, and fQ-B denote ImBBPS, Quick-ImBBPS, and fQuick-ImBBPS, respectively. The test histogram vectors are constructed from prediction errors for shadow and blank pixels, respectively.

Shadow	T and Methods				T and Methods			
Image	T	I-B	Q-B	fQ-B	T	I-B	Q-B	fQ-B
Lena	10430	14471512	9166169	8974278	20430	28911512	13348683	13264687
Baboon	5462	8547846	5572915	5476696	10462	16672846	7596116	7581772
Boat	10526	16142931	7993701	7906424	15526	24037931	9390629	9350768
Lake	10462	17035058	10447128	10335931	15462	25415058	12020357	11976741
Airplane	10494	12245234	8942641	8619310	20494	24605234	14009367	13883031
Barbara	10462	14590887	9350178	9204226	20462	28990887	13976290	13933439
Blank	T and Methods				T and Methods			
Image	T	I-B	Q-B	fQ-B	T	I-B	Q-B	fQ-B
Lena	10430	14588325	9217060	9036074	20430	29700518	12779628	12717908
Baboon	5462	8350788	5234792	5140250	10462	16032348	6760340	6748992
Boat	10540	16358669	7129767	7056644	15540	24615007	8757302	8721816
Lake	10462	16810400	9472096	9383048	15462	24591716	10002061	9982122
Airplane	10494	13488862	9682458	9400697	20494	28405813	12623123	12544636
Barbara	10462	13724603	8236741	8120050	20462	28509238	12128899	12105610

IEEE Transactions on Information Forensics and Security, Vol. 8, 2013, pp. 111-120.
3. H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam, and H. G. Choo, "A novel difference expansion transform for reversible data embedding," *IEEE Transactions on Information Forensics and Security*, Vol. 3, 2008, pp. 456-465.
4. X. Li, W. Zhang, X. Gui, and B. Yang, "Efficient reversible data hiding based on multiple histograms modification," *IEEE Transactions on Information Forensics and Security*, Vol. 10, 2015, pp. 2016-2027.
5. B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, "Pairwise prediction-error expansion for efficient reversible data hiding," *IEEE Transactions on Image Process*, Vol. 22, 2013, pp. 5010-5021.
6. Miscelaneous Gray Level Images, http://decsai.ugr.es/cvg/dbimagenes/g512.php.
7. W. Qi, X. Li, T. Zhang, and Z. Guo, "Optimal reversible data hiding scheme based on multiple histograms modification," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 30, 2020, pp. 2300-2312.
8. Y. Shi, X. Li, X. Zhang, H. Wu, and B. Ma, "Reversible data hiding: Advances in the past two decades," *IEEE Access*, Vol. 4, 2016, pp. 3210-3237.

Chun-Liang Jhong received his BS degree in 2019 from National Taipei University, New Taipei City, Taiwan. Currently, he is a master student with the Department of Computer Science and Information Engineering at National Taipei University. His research interests include reversible data hiding, algorithm design, and machine learning.

Hsin-Lung Wu received his Ph.D. degree in Computer Science and Information Engineering from National Chiao Tung University, Taiwan in 2008. He is currently an Associate Professor in the Department of Computer Science and Information Engineering at National Taipei University, Taiwan. His main research interests include design and analysis of algorithms, computational complexity, theory of machine learning, deep learning, and reversible data hiding.