IEICE TRANS. FUNDAMENTALS, VOL.E102–A, NO.2 FEBRUARY 2019

PAPER

A Universal Two-Dimensional Source Coding by Means of Subblock Enumeration*

Takahiro OTA[†a)], ***Member***, **Hiroyoshi MORITA**[††b)], ***Senior Member***, ***and*** **Akiko MANADA**[†††c)], ***Member***

SUMMARY The technique of lossless compression via substring enumeration (CSE) is a kind of enumerative code and uses a probabilistic model built from the circular string of an input source for encoding a one-dimensional (1D) source. CSE is applicable to two-dimensional (2D) sources, such as images, by dealing with a line of pixels of a 2D source as a symbol of an extended alphabet. At the initial step of CSE encoding process, we need to output the number of occurrences of all symbols of the extended alphabet, so that the time complexity increases exponentially when the size of source becomes large. To reduce computational time, we can rearrange pixels of a 2D source into a 1D source string along a space-filling curve like a Hilbert curve. However, information on adjacent cells in a 2D source may be lost in the conversion. To reduce the time complexity and compress a 2D source without converting to a 1D source, we propose a new CSE which can encode a 2D source in a block-by-block fashion instead of in a line-by-line fashion. The proposed algorithm uses the flat torus of an input 2D source as a probabilistic model instead of the circular string of the source. Moreover, we prove the asymptotic optimality of the proposed algorithm for 2D general sources.
key words: *compression via substring enumeration, enumerative code, universal source coding, two-dimensional, general source*

1. Introduction

Dubé and Beaudoin proposed an efficient off-line lossless data compression algorithm for a binary source known as *Compression via Substring Enumeration* (CSE) [1]. In [2], Yokoo proposed a universal CSE algorithm for an ergodic source with a binary alphabet, and various versions of CSE for a binary source have been proposed so far [3]–[5]. Reportedly, the performance of compression ratios of CSE [4] is better than that of an efficient off-line data compression algorithm using the Burrows-Wheeler transformation (BWT) [6]. It was proven that encoders of CSE and the antidictionary coding [7] are isomorphic for a binary source [8]. Moreover, an antidictionary coding algorithm [9] provided the first CSE for q-ary ($q > 2$) alphabet sources as a byproduct. It was also shown that encoders of the antidictionary coding and CSE are isomorphic for a q-ary source ($q > 2$) [9]. Iwata and Arimura modified CSE and evaluated the maximum redundancy rate of CSE for the k-th order Markov sources [10]. Furthermore, a universal CSE algorithm for an ergodic source with a finite alphabet source has been proposed [11].

CSE uses a probabilistic model built from the circular string which is obtained by concatenating the first symbol to the last symbol of an input source. The probabilistic model is also useful for the BWT and antidictionary coding [8], [9]. It was shown that the antidictionary built from the circular string is useful for genome comparison such as deoxyribonucleic acid (DNA) [12]. However, for a 2D source (*e.g.*, an image), the computational time of CSE is exponential with respect to the line length because CSE works in a line-by-line fashion. CSE deals with a line of a 2D source as a symbol of an extended alphabet. At the initial step of CSE encoding process, CSE needs to output frequencies including zero of all symbols of the extended alphabet. To reduce computational time, we can convert a 2D source to a 1D source by using space-filling curves as Hilbert curve, and the technique is used in image compression algorithms [13]. However, in converting, a 2D-ness has not been truly incorporated and information on adjacent cells in a 2D source may be lost.

To reduce the computational time and compress a 2D source without a space-filling curve, we propose a new CSE for a 2D source which uses the flat torus of an input 2D source as a probabilistic model instead of the circular string of the source. In the initial step, the total number of output blocks is constant because the new CSE works in a block-by-block fashion. Moreover, we prove the asymptotic optimality of the proposed algorithm for 2D general sources.

This paper is organized as follows. Section 2 gives the basic notations and definitions. Then, in Sect. 3, we review a conventional (1D) CSE. Section 4 proposed a 2D CSE algorithm. Section 5 proves that the proposed coding algorithm is asymptotically optimal for a 2D general source. Section 6 summarizes our results.

2. Basic Notations and Definitions

2.1 Alphabet and Block

Let $\mathcal{X}$ be a finite source alphabet $\{0, 1, \ldots, J-1\}$ and let $|\mathcal{X}|$ be the cardinality of $\mathcal{X}$, that is, $|\mathcal{X}| = J$. Let $\mathcal{X}^{[m,n]}$

Manuscript received March 30, 2018.
Manuscript revised September 10, 2018.
[†]The author is with Dept. of Computer & Systems Engineering, Nagano Prefectural Institute of Technology, Ueda-shi, 386-1211 Japan.
[††]The author is with Dept. of Computer and Network Engineering, Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu-shi, 182-8585 Japan.
[†††]The author is with Dept. of Information Science, Shonan Institute of Technology, Fujisawa-shi, 251-0046 Japan.
*This paper was presented in part at the IEEE International Symposium on Information Theory, Aachen, Germany, July 2017 [18].
a) E-mail: ota@pit-nagano.ac.jp
b) E-mail: morita@uec.ac.jp
c) E-mail: amanada@info.shonan-it.ac.jp
DOI: 10.1587/transfun.E102.A.440

1	0	0
1	1	0
0	0	1

Fig. 1 A 3×3 block $\boldsymbol{p}$.

be the set of all $m \times n$ finite blocks $\boldsymbol{p} = (\boldsymbol{p}_{(i,j)})_{1 \le i \le m, 1 \le j \le n}$ over $\mathcal{X}$, where $\boldsymbol{p}_{(i,j)} \in \mathcal{X}$ is the element of $\boldsymbol{p}$ at the (i, j)-coordinate. The (i, j)-coordinate in a block represents the location on the i-th row and the j-th column in the block where the numbers of the rows that increase downwards and the columns that does to the right and $i, j \ge 1$. For example, the $(1, 2)$-coordinate in a block represents the location on the top (first) row and the second column from the left in the block. Furthermore, let $\mathcal{X}^{[*,*]} := \cup_{m,n \ge 0} \mathcal{X}^{[m,n]}$, where $\mathcal{X}^{[m,n]}$ includes the *empty block* $\lambda^{[m,n]}$ when $m = 0$ or $n = 0$. For convenience, $\mathcal{X}^{[m,0]}$ and $\mathcal{X}^{[0,n]}$ are defined as $\{\lambda^{[m,0]}\}$ and $\{\lambda^{[0,n]}\}$, respectively. Let $|\boldsymbol{p}|_r$ and $|\boldsymbol{p}|_c$ be the numbers of rows (the *height*) and columns (the *width*), respectively. As an example, Fig. 1 illustrates a block $\boldsymbol{p} \in \mathcal{X}^{[3,3]}$.

2.2 Subblock, Concatenation, and Dictionary

For $\boldsymbol{p} \in \mathcal{X}^{[m,n]}$, the *subblock* $\boldsymbol{p}_{(i,j)}^{(i+k-1,j+l-1)} \in \mathcal{X}^{[k,l]}$ is defined as

$$\boldsymbol{p}_{(i,j)}^{(i+k-1,j+l-1)} := \begin{cases} \lambda^{[0,l]} \ (k = 0 \text{ and } l \ge 0), \\ \lambda^{[k,0]} \ (k \ge 0 \text{ and } l = 0), \\ \begin{pmatrix} \boldsymbol{p}_{(i,j)} & \cdots & \boldsymbol{p}_{(i,j+l-1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{p}_{(i+k-1,j)} & \cdots & \boldsymbol{p}_{(i+k-1,j+l-1)} \end{pmatrix} \\ \quad (k > 0 \text{ and } l > 0) \end{cases}$$

where $1 \le i \le m$, $1 \le j \le n$, $0 \le k \le m - i + 1$, and $0 \le l \le n - j + 1$. Hereafter, without notice, we assume that the height and the width of $\boldsymbol{p}$ are respectively given by m (≥ 2) and n (≥ 2). For a given $\boldsymbol{q} \in \mathcal{X}^{[k,l]}$ $(k, l \ge 0)$, the $(k-1) \times l$ subblocks $\boldsymbol{q}_{(1,1)}^{(k-1,l)}$ (the subblock obtained by deleting the k-th row) and $\boldsymbol{q}_{(2,1)}^{(k,l)}$ (the subblock obtained by deleting the first row) are denoted by $\pi_r(\boldsymbol{q})$ and $\sigma_r(\boldsymbol{q})$, respectively, where both $\pi_r(\boldsymbol{q})$ and $\sigma_r(\boldsymbol{q})$ are $\lambda^{[0,l]}$ when $k = 0$ and 1. Similarly, the $k \times (l-1)$ subblocks $\boldsymbol{q}_{(1,1)}^{(k,l-1)}$ and $\boldsymbol{q}_{(1,2)}^{(k,l)}$ are denoted by $\pi_c(\boldsymbol{q})$ and $\sigma_c(\boldsymbol{q})$, respectively, where both $\pi_r(\boldsymbol{q})$ and $\sigma_r(\boldsymbol{q})$ are $\lambda^{[k,0]}$ when $l = 0$ and 1. Figure 2 shows $\pi_c(\boldsymbol{p}), \sigma_c(\boldsymbol{p}), \pi_r(\boldsymbol{p})$, and $\sigma_r(\boldsymbol{p})$ from left to right for $\boldsymbol{p}$ in Figure 1. The dictionary of $\boldsymbol{p}$ is defined as the set of all subblocks of $\boldsymbol{p}$; that is,

$$\mathcal{D}(\boldsymbol{p}) := \{\boldsymbol{p}_{(i,j)}^{(i+k-1,j+l-1)} \mid 1 \le i \le m, 1 \le j \le n, \\ 0 \le k \le m - i + 1, 0 \le l \le n - j + 1\}.$$

Now we define a column-wise concatenation of blocks. For two blocks $\boldsymbol{s}, \boldsymbol{t} \in \mathcal{X}^{[*,*]}$ such that $|\boldsymbol{s}|_r = |\boldsymbol{t}|_r$, define $\boldsymbol{s} : \boldsymbol{t} \in \mathcal{X}^{[|\boldsymbol{s}|_r, |\boldsymbol{s}|_c + |\boldsymbol{t}|_c]}$ to be the block obtained by concatenating $\boldsymbol{t}$ at the end of $\boldsymbol{s}$ in columns. Similarly, for two blocks $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{X}^{[*,*]}$ such that $|\boldsymbol{u}|_c = |\boldsymbol{v}|_c$, define $\boldsymbol{u}/\boldsymbol{v} \in \mathcal{X}^{[|\boldsymbol{u}|_r + |\boldsymbol{v}|_r, |\boldsymbol{u}|_c]}$ to be the block obtained by concatenating $\boldsymbol{v}$ at the end of $\boldsymbol{u}$ in rows.

1	0
1	1
0	0

,

0	0
1	0
0	1

,

1	0	0
1	1	0

,

1	1	0
0	0	1

.

Fig. 2 $\pi_c(\boldsymbol{p}), \sigma_c(\boldsymbol{p}), \pi_r(\boldsymbol{p})$, and $\sigma_r(\boldsymbol{p})$ of $\boldsymbol{p}$ in Fig. 1.

2.3 Flat Torus, Primitive, and Frequencies of Subblocks

The *flat torus* of $\boldsymbol{p}$, denoted by $\boldsymbol{p}^T$, is constructed by concatenating the leftmost column (*resp.* the top row) to the rightmost column (*resp.* the bottom row) of $\boldsymbol{p}$. The flat torus can be treated as an infinite pattern such that $\boldsymbol{p}_{(i,j)} = \boldsymbol{p}^T_{(i+km,j+ln)}$ for non-negative integers k and l.

For $\boldsymbol{q} \in \mathcal{X}^{[m,n]}$ and the $2m \times 2n$ subblock $\bar{\boldsymbol{p}} := (\boldsymbol{p} : \boldsymbol{p})/(\boldsymbol{p} : \boldsymbol{p})$ of $\boldsymbol{p}^T$, we say that $\boldsymbol{p}$ and $\boldsymbol{q}$ are equivalent, denoted by $\boldsymbol{p} \simeq \boldsymbol{q}$, if there exist positive integers i $(1 \le i \le m)$ and j $(1 \le j \le n)$ such that $\boldsymbol{q} = \bar{\boldsymbol{p}}_{(i,j)}^{(i+m-1,j+n-1)}$. In other words, $\boldsymbol{p} \simeq \boldsymbol{q}$ if and only if $\boldsymbol{q} \in \mathcal{D}(\bar{\boldsymbol{p}})$. Indeed, it satisfies the conditions to be an equivalence relation. Let $[\boldsymbol{p}]$ be the set of all blocks $\boldsymbol{q}$ such that $\boldsymbol{q} \simeq \boldsymbol{p}$; that is,

$$[\boldsymbol{p}] := \{\boldsymbol{q} \in \mathcal{X}^{[m,n]} \mid \boldsymbol{q} \in \mathcal{D}(\bar{\boldsymbol{p}})\}. \tag{1}$$

For a block $\boldsymbol{p} \in \mathcal{X}^{[m,n]}$, $\boldsymbol{p}^\dagger$ is defined as the smallest element in $[\boldsymbol{p}]$ in column-wise lexicographical order. From the definition, $\boldsymbol{p}^\dagger$ is equal to $\boldsymbol{q}^\dagger$ for any block $\boldsymbol{q} \in [\boldsymbol{p}]$. If $|[\boldsymbol{p}]| = mn$, $\boldsymbol{p}$ is called *primitive*. Hereafter, we always assume that $\boldsymbol{p}$ is primitive. For example, $\boldsymbol{p}$ shown in Figure 1 is primitive. For $\boldsymbol{p}$ and $\boldsymbol{u} \in \mathcal{X}^{[k,l]}$ $(0 \le k \le m$ and $0 \le l \le n)$, define

$$N(\boldsymbol{u} \mid \boldsymbol{p}) := |\{\boldsymbol{r} \mid \boldsymbol{u} = \boldsymbol{r}_{(1,1)}^{(k,l)}, \boldsymbol{r} \in [\boldsymbol{p}]\}|, \tag{2}$$

where $N(\lambda^{[k,l]}|\boldsymbol{p}) = mn$ $(k = 0$ or $l = 0)$. Thus, $N(\boldsymbol{u} \mid \boldsymbol{p})$ represents the frequency of $\boldsymbol{u}$ in $\boldsymbol{p}^T$. For convenience, we often adopt the notation $N(\boldsymbol{u})$ instead of $N(\boldsymbol{u}|\boldsymbol{p})$. For $0 \le k \le m$ and $0 \le l \le n$, observe that

$$\sum_{\boldsymbol{u} \in \mathcal{X}^{[k,l]}} N(\boldsymbol{u}) = mn. \tag{3}$$

Moreover, for $\boldsymbol{v} \in \mathcal{X}^{[i,j]}$ $(0 \le i \le m, 0 \le j < n)$ and $\boldsymbol{v}' \in \mathcal{X}^{[k,l]}$ $(0 \le k < m, 0 \le l \le n)$, we have

$$N(\boldsymbol{v}) = \sum_{\boldsymbol{c} \in \mathcal{X}^{[i,1]}} N(\boldsymbol{c} : \boldsymbol{v}) = \sum_{\boldsymbol{c} \in \mathcal{X}^{[i,1]}} N(\boldsymbol{v} : \boldsymbol{c}), \tag{4}$$

$$N(\boldsymbol{v}') = \sum_{\boldsymbol{r} \in \mathcal{X}^{[1,l]}} N(\boldsymbol{r}/\boldsymbol{v}') = \sum_{\boldsymbol{r} \in \mathcal{X}^{[1,l]}} N(\boldsymbol{v}'/\boldsymbol{r}). \tag{5}$$

2.4 Classifications of Flat Tori and Core

For a block $\boldsymbol{p} \in \mathcal{X}^{[m,n]}$ and integers k $(0 \le k \le m)$ and

l $(0 \le l \le n)$, define

$$\mathcal{T}(\boldsymbol{p},k,l) := \{\boldsymbol{q}^{\dagger} \mid N(\boldsymbol{w}|\boldsymbol{q}) = N(\boldsymbol{w}|\boldsymbol{p}), {}^{\forall}\boldsymbol{w} \in \mathcal{X}^{[k,l]}, \boldsymbol{q} \in \mathcal{X}^{[m,n]}, \boldsymbol{q} \text{ is primitive}\}. \quad (6)$$

For convenience, we write $\mathcal{T}(k,l)$ instead of $\mathcal{T}(\boldsymbol{p},k,l)$.

For example, $\mathcal{T}(m,n) = \{\boldsymbol{p}^{\dagger}\}$, that is $|\mathcal{T}(m,n)| = 1$. The cardinality $|\mathcal{T}(k,l)|$ represents the number of the smallest (primitive) $m \times n$ blocks $\boldsymbol{q}^{\dagger}$ such that $N(\boldsymbol{w}|\boldsymbol{q}) = N(\boldsymbol{w}|\boldsymbol{p})$ for any $\boldsymbol{w} \in \mathcal{X}^{[k,l]}$. For $0 \le k < n$ and fixed $0 \le l \le n$, $\mathcal{T}(k,l)$ satisfies the subset relation in descending order of k; that is, $\mathcal{T}(k+1,l) \subseteq \mathcal{T}(k,l)$. Similarly, for fixed $0 \le k' \le n$ and $0 \le l' < n$, $\mathcal{T}(k', l'+1) \subseteq \mathcal{T}(k', l')$ holds.

We define $\mathcal{B}(\boldsymbol{p})$, which is used to encode $\boldsymbol{p}$, to be

$$\mathcal{B}(\boldsymbol{p}) := \{\boldsymbol{b} \in \mathcal{X}^{[k,l]} \mid \pi_r(\boldsymbol{b}) \in \mathcal{D}(\bar{\boldsymbol{p}}), \sigma_r(\boldsymbol{b}) \in \mathcal{D}(\bar{\boldsymbol{p}}), \pi_c(\boldsymbol{b}) \in \mathcal{D}(\bar{\boldsymbol{p}}), \sigma_c(\boldsymbol{b}) \in \mathcal{D}(\bar{\boldsymbol{p}}), 1 \le k \le m, 1 \le l \le n\} \cup \{\lambda^{[0,0]}\}. \quad (7)$$

We assume that the elements of $\mathcal{B}(\boldsymbol{p})$ are ordered in ascending order of their heights (if the heights of the elements are equal, then the elements are ordered with respect to their widths; if the widths of the elements are also equal, then the elements are ordered in column-wise lexicographical order), where $\boldsymbol{b}_i$ is the i-th element of $\mathcal{B}(\boldsymbol{p})$.

For an integer i $(1 \le i \le |\mathcal{B}(\boldsymbol{p})|)$, define

$$\mathcal{T}(\mathcal{B}(\boldsymbol{p}),\boldsymbol{p},i) := \{\boldsymbol{q}^{\dagger} \mid N(\boldsymbol{b}_j \mid \boldsymbol{q}) = N(\boldsymbol{b}_j \mid \boldsymbol{p}), 1 \le {}^{\forall}j \le i, \boldsymbol{q} \in \mathcal{X}^{[m,n]}, \boldsymbol{q} \text{ is primitive}\}. \quad (8)$$

For convenience, we write $\mathcal{T}(i)$ instead of $\mathcal{T}(\mathcal{B}(\boldsymbol{p}),\boldsymbol{p},i)$. For example, $\mathcal{T}(|\mathcal{B}(\boldsymbol{p})|) = \{\boldsymbol{p}^{\dagger}\}$, that is $|\mathcal{T}(|\mathcal{B}(\boldsymbol{p})|)| = 1$. The cardinality $|\mathcal{T}(i)|$ represents the number of representative (primitive) $m \times n$ blocks at encoding step i. For $1 \le i < |\mathcal{B}(\boldsymbol{p})|$, $\mathcal{T}(i)$ satisfies the subset relation in descending order of i; that is, $\mathcal{T}(i+1) \subseteq \mathcal{T}(i)$. For a block $\boldsymbol{u} \in \mathcal{B}(\boldsymbol{p})$, the block $\sigma_c(\pi_c(\boldsymbol{u}))$ is called *c-core* (column-core) if $\boldsymbol{a} : \boldsymbol{u}, \boldsymbol{b} : \boldsymbol{u}, \boldsymbol{u} : \boldsymbol{c}, \boldsymbol{u} : \boldsymbol{d} \in \mathcal{D}(\bar{\boldsymbol{p}})$, where $\boldsymbol{a}, \boldsymbol{b}(\neq \boldsymbol{a}), \boldsymbol{c}, \boldsymbol{d}(\neq \boldsymbol{c}) \in \mathcal{X}^{[|\boldsymbol{u}|_r,1]}$. Similarly, for a block $\boldsymbol{v} \in \mathcal{B}(\boldsymbol{p})$, the block $\sigma_r(\pi_r(\boldsymbol{v}))$ is called *r-core* (row-core) if $\boldsymbol{e}/\boldsymbol{v}, \boldsymbol{f}/\boldsymbol{v}, \boldsymbol{v}/\boldsymbol{g}, \boldsymbol{v}/\boldsymbol{h} \in \mathcal{D}(\bar{\boldsymbol{p}})$, where $\boldsymbol{e}, \boldsymbol{f}(\neq \boldsymbol{e}), \boldsymbol{g}, \boldsymbol{h}(\neq \boldsymbol{g}) \in \mathcal{X}^{[1,|\boldsymbol{v}|_c]}$. C-cores and r-cores are used to determine whether an element of $\mathcal{B}(\boldsymbol{p})$ is encoded. The details will be described in Sects. 3 and 4.

3. Review of Conventional CSE

The conventional CSE is a lossless compression algorithm for a 1D source. We can regard $\boldsymbol{p} \in \mathcal{X}^{[m,n]}$ as a 1D source $\boldsymbol{x} \in \hat{\mathcal{X}}^{[1,n]}$ over an extended alphabet $\hat{\mathcal{X}}(= \mathcal{X}^{[m,1]})$, so that CSE can encode $\boldsymbol{p}$ as a 1D source $\boldsymbol{x}$. For $\boldsymbol{x}$ in $[\boldsymbol{x}]$, let $\text{rank}(\boldsymbol{x})$ be the number assigned for $\boldsymbol{x}$ when elements in $[\boldsymbol{x}]$ are arranged in lexicographical order, and $\epsilon(\text{rank}(\boldsymbol{x}))$ be the encoding of $\text{rank}(\boldsymbol{x})$ in binary (in $\lceil \log_2 n \rceil$ bits). CSE outputs

$$(E(n), e(\boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{|\mathcal{B}(\boldsymbol{x})|}), \epsilon(\text{rank}(\boldsymbol{x}))), \quad (9)$$

where $E(n)$ denotes the encoding of n by means of the Elias code for integers [14], and $e(\boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{|\mathcal{B}(\boldsymbol{x})|})$ represents the sequence of $N(\boldsymbol{b}_i)(= N(\boldsymbol{b}_i|\boldsymbol{x}))$, $2 \le i \le |\mathcal{B}(\boldsymbol{x})|$, which are encoded by an entropy coding. In encoding, an index i for $\boldsymbol{b}_i \in \mathcal{B}(\boldsymbol{x})$ is chosen between 2 and $|\mathcal{B}(\boldsymbol{x})|$ because when $i = 1$, $N(\boldsymbol{b}_1) = N(\lambda^{[0,0]}) = n$ and n is encoded as $E(n)$. Let $\boldsymbol{b}_{|\hat{\mathcal{X}}|+1}$ be the element of $\hat{\mathcal{X}}$ having the largest index in $\mathcal{B}(\boldsymbol{x})$. For $2 \le i \le |\mathcal{B}(\boldsymbol{x})|$, $N(\boldsymbol{b}_i)$ is encoded based on the following conditions.

(C-i) When $|\boldsymbol{b}_i|_c = 1$: Encode $N(\boldsymbol{b}_i)$ if $\boldsymbol{b}_i \neq \boldsymbol{b}_{|\hat{\mathcal{X}}|+1}$,
(C-ii) When $|\boldsymbol{b}_i|_c \ge 2$: Encode $N(\boldsymbol{b}_i)$ if (10) below holds and $\boldsymbol{a}, \boldsymbol{c} \in \hat{\mathcal{X}} \backslash \{\boldsymbol{b}_{|\hat{\mathcal{X}}|+1}\}$, where $\boldsymbol{b}_i = \boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c}$.

When we encode $N(\boldsymbol{b}_i)$ based on conditions (C-i) and (C-ii), we first assign the probability (as shown in (13), (14), or (15)) to $N(\boldsymbol{b}_i)$. We then encode the probability by using entropy coding.

Inequality (10) was first shown in [10]. Note that improved inequalities of (10) have been presented in [10] and [15]. They are omitted here to simplify discussions because they are complicated and only (10) is necessary for an asymptotic analysis. It is noteworthy that $N(\boldsymbol{b}_i)$ is encoded even though $N(\boldsymbol{b}_i) = 0$ in (C-i).

In (C-i), $N(\boldsymbol{b}_{|\hat{\mathcal{X}}|+1})$ can be inferred by using (3) and $\boldsymbol{b}_j (j < |\hat{\mathcal{X}}| + 1)$ which has been encoded. Similarly, in (C-ii), $N(\boldsymbol{b}_i)$ such that the first column of $\boldsymbol{b}_i$ is $\boldsymbol{b}_{|\hat{\mathcal{X}}|+1}$ or the last column of $\boldsymbol{b}_i$ is $\boldsymbol{b}_{|\hat{\mathcal{X}}|+1}$ can be inferred by using (4) and $\boldsymbol{b}_k$ $(k < i)$. Therefore, $N(\boldsymbol{b}_i)$ is not needed to be encoded.

$$\min(N(\boldsymbol{a} : \boldsymbol{w}), N(\boldsymbol{w} : \boldsymbol{c}), N(\boldsymbol{w}) - N(\boldsymbol{a} : \boldsymbol{w}), N(\boldsymbol{w}) - N(\boldsymbol{w} : \boldsymbol{c})) \ge 1. \quad (10)$$

As for $\boldsymbol{b}_i(= \boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c})$ in (C-ii), satisfying (10) is equivalent to satisfying the three conditions, $\boldsymbol{w}$ is a c-core, $\boldsymbol{a} : \boldsymbol{w} \in \mathcal{D}(\bar{\boldsymbol{x}})$, and $\boldsymbol{w} : \boldsymbol{c} \in \mathcal{D}(\bar{\boldsymbol{x}})$.

In (C-i), $N(\boldsymbol{b}_i)$ satisfies the following inequality

$$0 \le N(\boldsymbol{b}_i) \le n - 1. \quad (11)$$

In (C-ii), $N(\boldsymbol{b}_i)$ satisfies the following inequality [9]

$$\max\{0, N(\boldsymbol{a} : \boldsymbol{w}) - \sum_{\boldsymbol{d} \in \hat{\mathcal{X}} \backslash \{\boldsymbol{c}\}} N(\boldsymbol{w} : \boldsymbol{d}), N(\boldsymbol{w} : \boldsymbol{c}) - \sum_{\boldsymbol{b} \in \hat{\mathcal{X}} \backslash \{\boldsymbol{a}\}} N(\boldsymbol{b} : \boldsymbol{w})\} \le N(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c}) \le \min\{N(\boldsymbol{a} : \boldsymbol{w}), N(\boldsymbol{w} : \boldsymbol{c})\}. \quad (12)$$

The left-hand side of (10) is given by the difference between the upper bound and the lower bound of $N(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c})$ obtained in (12). Therefore, if (10) does not hold, then the upper bound and the lower bound turn out to be equal. In other words, $N(\boldsymbol{b}_i) = \min\{N(\boldsymbol{a} : \boldsymbol{w}), N(\boldsymbol{w} : \boldsymbol{c})\}$ holds, so that $N(\boldsymbol{b}_i)$ can be inferred. Hence, $N(\boldsymbol{b}_i)$ is not encoded if (10) does not hold. We define that $I(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c})$ for a given $\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c}$ is the left-hand side of (10) plus one; that is,

$$I(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c}) := \min(N(\boldsymbol{a} : \boldsymbol{w}), N(\boldsymbol{w} : \boldsymbol{c}), N(\boldsymbol{w}) - N(\boldsymbol{a} : \boldsymbol{w}), N(\boldsymbol{w}) - N(\boldsymbol{w} : \boldsymbol{c})) + 1.$$

For encoding $N(\boldsymbol{b}_i)$ by an entropy coding, the probability is assigned to $N(\boldsymbol{b}_i)$ [2], as

$$\frac{1}{n} \quad (|\boldsymbol{b}_i|_c = 1), \tag{13}$$

$$\frac{1}{I(\boldsymbol{b}_i)} \quad (2 \leq |\boldsymbol{b}_i|_c \leq \lfloor \log_2 \log_2 n \rfloor), \tag{14}$$

$$\frac{|\mathcal{T}(i)|}{|\mathcal{T}(i-1)|} \quad (|\boldsymbol{b}_i|_c \geq \lfloor \log_2 \log_2 n \rfloor + 1). \tag{15}$$

The assigned probabilities are encoded by an entropy coding such as an arithmetic coder or something like that [16].

The computational time results in a serious issue when encoding a 2D source $\boldsymbol{p}$ by the conventional CSE. In (C-i), the number of encoded $N(\boldsymbol{b}_i)$ $(2 \leq i \leq |\hat{\mathcal{X}}|)$ is exponential with respect to m because $|\hat{\mathcal{X}}| = |\mathcal{X}|^m$. In practice, m is greater than 1000 for an image $\boldsymbol{p} \in \mathcal{X}^{[m,n]}$, so that the number turns out to be greater than 2^{1000} even though $|\mathcal{X}| = 2$. It is noteworthy that the number of encoded $N(\boldsymbol{b}_i)$ is polynomial with respect to m and n in (C-ii), for the following reasons. Since $\boldsymbol{w}$ is a c-core, from (3) and (4), the total number of c-cores is polynomial with respect to m and n. Moreover, as $N(\boldsymbol{a} : \boldsymbol{w}) \geq 1$ and $N(\boldsymbol{w} : \boldsymbol{c}) \geq 1$ in (10), $\boldsymbol{a}, \boldsymbol{c} \in \mathcal{D}(\bar{\boldsymbol{x}}) \cap \hat{\mathcal{X}}$ also holds. From (3) and (4), $|\mathcal{D}(\bar{\boldsymbol{x}}) \cap \hat{\mathcal{X}}|$ never exceeds mn. Hence, the total number of candidates $\boldsymbol{b}_i (= \boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c})$ for encoding in (C-ii) is polynomial with respect to m and n. The set of all the candidates can be used to encode $\boldsymbol{x}$ instead of $\mathcal{B}(\boldsymbol{x})$ in (C-ii) for reducing the computational time. Furthermore, only the relation on columns is used as shown in (10) and a relation on rows is not used in encoding of the conventional CSE.

4. Proposed Algorithm

Assume that $m \leq n$, and let $K = \left\lfloor \sqrt{\log_{|\mathcal{X}|} \log_{|\mathcal{X}|} m} \right\rfloor$ and $L = \left\lfloor \sqrt{\log_{|\mathcal{X}|} \log_{|\mathcal{X}|} n} \right\rfloor$. We select K and L described above for the following reasons. We use the number of occurrences of $K \times L$ blocks $\boldsymbol{q} \in \mathcal{B}(\boldsymbol{p})$ (that is the empirical distribution of $\boldsymbol{q} \in \mathcal{X}^{[K,L]}$) to analyze remarkable properties of $\boldsymbol{p}$ that appear when the size of $\boldsymbol{p}$ goes to infinity. Therefore, it is desirable to choose K and L such that K and L grow with m and n, respectively. On the other hand, we need to store the $K \times L$ blocks for encoding the number of their occurrences. Since we want to make the cost of storing the $K \times L$ blocks smaller with respect to data compression, we select K and L such that the cost over the size of $\boldsymbol{p}$ converges to zero when the size of $\boldsymbol{p}$ goes to infinity. Hence, K and L are set to be small values compared with m and n, respectively. Observe that K and L above satisfy those two requirements.

We divide $\mathcal{B}(\boldsymbol{p})$ into four disjoint subsets, with respect to the size of elements, as

$$\mathcal{B}_0(\boldsymbol{p}) := \{\lambda^{[0,0]}\},$$
$$\mathcal{B}_1(\boldsymbol{p}) := \mathcal{X},$$
$$\mathcal{B}_2(\boldsymbol{p}) := \{\boldsymbol{b} \in \mathcal{B}(\boldsymbol{p}) \mid 1 \leq |\boldsymbol{b}|_r \leq K, 1 \leq |\boldsymbol{b}|_c \leq L, \boldsymbol{b} \notin \mathcal{X}\},$$
$$\mathcal{B}_3(\boldsymbol{p}) := \{\boldsymbol{b} \in \mathcal{B}(\boldsymbol{p}) \mid K < |\boldsymbol{b}|_r \text{ or } L < |\boldsymbol{b}|_c, \boldsymbol{b} \notin \mathcal{X}\}.$$

The elements of $\mathcal{B}_i(\boldsymbol{p})$ $(i = 0, 1, 2, 3)$ are arranged in ascending order of their heights (if the heights of the elements are equal, then the elements are arranged in ascending order of their widths; if the widths of the elements are also equal, then the elements are arranged in column-wise lexicographical order.) The elements of $\mathcal{B}(\boldsymbol{p})$ are further reordered based on the ascending order of indexes for $\mathcal{B}_i(\boldsymbol{p})$; that is, elements of $\mathcal{B}_i(\boldsymbol{p})$ are lined up before $\mathcal{B}_{i'}(\boldsymbol{p})$ when $i < i'$. For simplicity, we represent the fact as $(\mathcal{B}_0(\boldsymbol{p}), \mathcal{B}_1(\boldsymbol{p}), \mathcal{B}_2(\boldsymbol{p}), \mathcal{B}_3(\boldsymbol{p}))$.

For a given integer k, $\boldsymbol{x}(k, 1)$ (resp. $\boldsymbol{x}(1, k)$) is defined to be the last element (in ascending order) amongst all elements of $\mathcal{X}^{[k,1]} \cap \mathcal{B}(\boldsymbol{p})$ (resp. $\mathcal{X}^{[1,k]} \cap \mathcal{B}(\boldsymbol{p})$). For $2 \leq i \leq |\mathcal{B}(\boldsymbol{p})|$, $N(\boldsymbol{b}_i)$ is encoded based on the following conditions.

(P-i) If $\boldsymbol{b}_i \in \mathcal{B}_1(\boldsymbol{p})$: Encode $N(\boldsymbol{b}_i)$ if $\boldsymbol{b}_i \neq J - 1$,
(P-ii) If $\boldsymbol{b}_i \in \mathcal{B}_2(\boldsymbol{p}) \cup \mathcal{B}_3(\boldsymbol{p})$:
1) If $|\boldsymbol{b}_i|_c = 1$: Encode $N(\boldsymbol{b}_i)$ if (10) holds and $\boldsymbol{a}, \boldsymbol{c} \in \mathcal{X} \backslash \{J - 1\}$, where $\boldsymbol{b}_i = \boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c}$,
2) If $|\boldsymbol{b}_i|_r = 1$: Encode $N(\boldsymbol{b}_i)$ if (16) holds and $\boldsymbol{e}, \boldsymbol{g} \in \mathcal{X} \backslash \{J - 1\}$, where $\boldsymbol{b}_i = \boldsymbol{e}/\boldsymbol{v}/\boldsymbol{g}$,
3) If $|\boldsymbol{b}_i|_c \geq 2$ and $|\boldsymbol{b}_i|_r \geq 2$: Encode $N(\boldsymbol{b}_i)$ if both (10) and (16) hold, where $\boldsymbol{a}, \boldsymbol{c} \in \mathcal{X}^{[|\boldsymbol{b}_i|_r, 1]} \backslash \{\boldsymbol{x}(|\boldsymbol{b}_i|_r, 1)\}$ and $\boldsymbol{e}, \boldsymbol{g} \in \mathcal{X}^{[1, |\boldsymbol{b}_i|_c]} \backslash \{\boldsymbol{x}(1, |\boldsymbol{b}_i|_c)\}$.

When we encode $N(\boldsymbol{b}_i)$ based on conditions (P-i) and (P-ii), we first assign the probability (as shown in (18), (19), or (20)) to $N(\boldsymbol{b}_i)$. We then encode the probability by using entropy coding.

$$\min(N(\boldsymbol{e}/\boldsymbol{v}), N(\boldsymbol{v}/\boldsymbol{g}), N(\boldsymbol{v}) - N(\boldsymbol{e}/\boldsymbol{v}), N(\boldsymbol{v}) - N(\boldsymbol{v}/\boldsymbol{g})) \geq 1. \tag{16}$$

As for $\boldsymbol{b}_i (= \boldsymbol{e}/\boldsymbol{v}/\boldsymbol{g})$ in 2) and 3) of (P-ii), satisfying (16) is equivalent to satisfying the three conditions that $\boldsymbol{v}$ is an r-core, $\boldsymbol{e}/\boldsymbol{v} \in \mathcal{D}(\bar{\boldsymbol{p}})$, and $\boldsymbol{v}/\boldsymbol{g} \in \mathcal{D}(\bar{\boldsymbol{p}})$.

The conventional CSE uses only condition (10) with respect to columns, while the proposed algorithm uses conditions (10) and (16) with respect to columns and rows, respectively. In 1) and 2) of (P-ii), $\boldsymbol{b}_i$ has one row and one column, so that (10) and (16) are used, respectively. In (P-i), $N(\boldsymbol{b}_i)$ satisfies $0 \leq N(\boldsymbol{b}_i) \leq mn - 1$. In (P-ii), $N(\boldsymbol{b}_i)$ such that $|\boldsymbol{b}_i|_c \geq 2$ satisfies a modified inequality (12) obtained by replacing $\hat{\mathcal{X}}$ by $\mathcal{X}^{[|\boldsymbol{a}|_r, 1]}$, and $N(\boldsymbol{b}_i)$ such that $|\boldsymbol{b}_i|_r \geq 2$ satisfies the following inequality

$$\max\{0, N(\boldsymbol{e}/\boldsymbol{v}) - \sum_{\boldsymbol{h} \in \mathcal{X}^{[1,|\boldsymbol{e}|_c]} \backslash \{\boldsymbol{g}\}} N(\boldsymbol{v}/\boldsymbol{h}), N(\boldsymbol{v}/\boldsymbol{g}) - \sum_{\boldsymbol{f} \in \mathcal{X}^{[1,|\boldsymbol{e}|_c]} \backslash \{\boldsymbol{e}\}} N(\boldsymbol{f}/\boldsymbol{v})\} \leq N(\boldsymbol{e}/\boldsymbol{v}/\boldsymbol{g}) \leq \min\{N(\boldsymbol{e}/\boldsymbol{v}), N(\boldsymbol{v}/\boldsymbol{g})\}. \tag{17}$$

Similarly, the left-hand side of (16) is given by the difference between the upper bound and the lower bound of $N(\boldsymbol{e}/\boldsymbol{v}/\boldsymbol{g})$ obtained in (17). Therefore, if (16) does not hold, then the upper bound and the lower bound turn to be equal. In other words, $N(\boldsymbol{b}_i) = \min\{N(\boldsymbol{e}/\boldsymbol{v}), N(\boldsymbol{v}/\boldsymbol{g})\}$ holds, so that $N(\boldsymbol{b}_i)$

can be inferred. Hence, $N(\boldsymbol{b}_i)$ is not encoded if (16) does not hold. Therefore, in 3), $N(\boldsymbol{b}_i)$ is encoded if both (10) and (16) hold. Moreover, $I'(\boldsymbol{e}/\boldsymbol{v}/\boldsymbol{g})$ for a given $\boldsymbol{e}/\boldsymbol{v}/\boldsymbol{g}$ is defined as the left-hand side of (16) plus one; that is,

$$I'(\boldsymbol{e}/\boldsymbol{v}/\boldsymbol{g}) := \min(N(\boldsymbol{e}/\boldsymbol{v}), N(\boldsymbol{v}/\boldsymbol{g}), N(\boldsymbol{v}) - N(\boldsymbol{e}/\boldsymbol{v}), N(\boldsymbol{v}) - N(\boldsymbol{v}/\boldsymbol{g})) + 1.$$

For encoding $N(\boldsymbol{b}_i)$ by an entropy coding, the probability is assigned to $N(\boldsymbol{b}_i)$, as

$$\frac{1}{mn} \quad (\boldsymbol{b}_i \in \mathcal{B}_1(\boldsymbol{p})), \tag{18}$$

$$\max\left(\frac{1}{I(\boldsymbol{b}_i)}, \frac{1}{I'(\boldsymbol{b}_i)}\right) \quad (\boldsymbol{b}_i \in \mathcal{B}_2(\boldsymbol{p})), \tag{19}$$

$$\frac{|\mathcal{T}(i)|}{|\mathcal{T}(i-1)|} \quad (\boldsymbol{b}_i \in \mathcal{B}_3(\boldsymbol{p})). \tag{20}$$

The assigned probabilities are encoded by an entropy coding such as an arithmetic coder or something like that. The proposed algorithm outputs the following quadruple

$$(E(m), E(n), e(\boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{|\mathcal{B}(\boldsymbol{p})|}), \epsilon(\mathrm{rank}(\boldsymbol{p}))). \tag{21}$$

In (21), $E(m)$ and $E(n)$ represent encoded m and n by means of the Elias code for integers, respectively. In addition, $\mathrm{rank}(\boldsymbol{p})$ represents an index for identifying $\boldsymbol{p}$ in $[\boldsymbol{p}]$ such as the rank of $\boldsymbol{p}$ in $[\boldsymbol{p}]$ with lexicographical order column-wisely. Then, $\epsilon(\mathrm{rank}(\boldsymbol{p}))$ is the encoding of $\mathrm{rank}(\boldsymbol{p})$ in binary (in $\lceil \log_2 mn \rceil$ bits), and $e(\boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{|\mathcal{B}(\boldsymbol{p})|})$ represents the sequence of $N(\boldsymbol{b}_i)$ $(2 \le i \le |\mathcal{B}(\boldsymbol{p})|)$ which is encoded by an entropy coding.

In the proposed algorithm, the number of encoded $N(\boldsymbol{b}_i)$ in (P-i) is $|\mathcal{X}| - 1$ (a constant), while that in (C-i) is $|\mathcal{X}|^m - 1$, which is exponential with respect to m. As for (P-ii), the number of candidates $N(\boldsymbol{b}_i)$ for encoding is polynomial with respect to m and n, for the following reasons. As for 1), it is the same as (C-ii). As for 2) and 3), since $\boldsymbol{v}$ is an r-core, from the discussions on a c-core described in Sect. 3, the total number of candidates $N(\boldsymbol{b}_i)$ for encoding is also polynomial with respect to m and n. The set of all the candidates can be used to encode $\boldsymbol{p}$ instead of $\mathcal{B}(\boldsymbol{p})$ in (P-ii) for reducing the computational time. Hence, for a 2D source $\boldsymbol{p}$, the total number of output blocks of the proposed algorithm is polynomial with respect to m and n, while that of the conventional CSE is exponential with respect to m.

5. Evaluation of the Proposed Algorithm

A general source $\mathbf{X}$ is defined as

$$\mathbf{X} := \{X^{[m,n]} = (X^{<m,n>}_{(1,1)}, X^{<m,n>}_{(1,2)}, \ldots, X^{<m,n>}_{(m,n)})\}_{m=1,n=1}^{\infty,\infty},$$

where a random variable $X^{[m,n]}$ takes a value in the $m \times n$ Cartesian product $\mathcal{X}^{[m,n]}$ of $\mathcal{X}$ [17]. The probability distribution of a random variable $X^{[m,n]}$ is denoted by $P_{X^{[m,n]}}$.

The sup-entropy rate of $\mathbf{X}$ is defined as

$$\hat{H}(\mathbf{X}) := \limsup_{m,n\to\infty} \frac{1}{mn} H(X^{[m,n]}). \tag{22}$$

For $\boldsymbol{p}$, let $\ell(\boldsymbol{p})$ be the codeword length of the proposed algorithm. Let $\ell_0(\boldsymbol{p})$ be the total codeword lengths of $E(m)$, $E(n)$, and $\epsilon(\mathrm{rank}(\boldsymbol{p}))$ in (21). The codeword length of $e(\boldsymbol{b}_2, \boldsymbol{b}_3, \ldots, \boldsymbol{b}_{|\mathcal{B}(\boldsymbol{p})|})$ consists of three parts $\ell_1(\boldsymbol{p})$, $\ell_2(\boldsymbol{p})$, and $\ell_3(\boldsymbol{p})$, where $\ell_1(\boldsymbol{p})$, $\ell_2(\boldsymbol{p})$, and $\ell_3(\boldsymbol{p})$ are the total codeword lengths of $N(\boldsymbol{b}_i)$ for $\boldsymbol{b}_i \in \mathcal{B}_1(\boldsymbol{p})$, $\boldsymbol{b}_i \in \mathcal{B}_2(\boldsymbol{p})$, and $\boldsymbol{b}_i \in \mathcal{B}_3(\boldsymbol{p})$, respectively. Observe that $\ell(\boldsymbol{p}) = \ell_0(\boldsymbol{p}) + \ell_1(\boldsymbol{p}) + \ell_2(\boldsymbol{p}) + \ell_3(\boldsymbol{p})$.

Theorem 1 is one of our main results.

Theorem 1. *For a general source* $\mathbf{X}$,

$$\limsup_{m,n\to\infty} E\left[\frac{\ell(X^{[m,n]})}{mn}\right] = \hat{H}(\mathbf{X}).$$

To prove Theorem 1, we show four lemmas: Lemma 1, Lemma 2, Lemma 3, and Lemma A. Lemmas 1 and 2 are 2D versions of Lemma 6 in [11] and Lemma 3 in [2], respectively. Lemma A is stated without proof since it is equivalent to Corollary 2 in [11]. Note that we assign alphabetic letters to the lemma.

Lemma 1. *If* $\boldsymbol{b}_{i+1} \in \mathcal{B}(\boldsymbol{p})$ *such that* $|\boldsymbol{b}_{i+1}|_c \ge 2$ *does not satisfy (10), then* $\mathcal{T}(i+1) = \mathcal{T}(i)$. *Similarly, if* $\boldsymbol{b}_{i+1} \in \mathcal{B}(\boldsymbol{p})$ *such that* $|\boldsymbol{b}_{i+1}|_r \ge 2$ *does not satisfy (16), then* $\mathcal{T}(i+1) = \mathcal{T}(i)$.

Proof. $\mathcal{T}(i+1) \subseteq \mathcal{T}(i)$ holds from the monotonicity on the cardinalities, so we focus on showing $\mathcal{T}(i+1) \supseteq \mathcal{T}(i)$. Also, we only argue the first case since the latter case is obtained by swapping column and row.

From the assumption on $\boldsymbol{b}_{i+1}$, $\boldsymbol{b}_{i+1}$ can be written as $\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c}$ for $\boldsymbol{a}, \boldsymbol{c} \in \mathcal{X}^{[|\boldsymbol{b}_{i+1}|_r, 1]}$ and $\boldsymbol{w} \in \mathcal{X}^{[*,*]}$. Suppose that $\boldsymbol{b}_{i+1}$ does not satisfy (10). When (10) does not hold, a case among the four following ones has to hold

$$N(\boldsymbol{a} : \boldsymbol{w}) = 0 \text{ or}, \tag{23}$$

$$N(\boldsymbol{w} : \boldsymbol{c}) = 0 \text{ or}, \tag{24}$$

$$N(\boldsymbol{w}) - N(\boldsymbol{a} : \boldsymbol{w}) = 0 \text{ or}, \tag{25}$$

$$N(\boldsymbol{w}) - N(\boldsymbol{w} : \boldsymbol{c}) = 0. \tag{26}$$

If (23) holds, then $N(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{d}) = 0$ for any column $\boldsymbol{d} \in \mathcal{X}^{[|\boldsymbol{b}_{i+1}|_r, 1]}$. Therefore, if $\boldsymbol{y}$ is an element of

$$\mathcal{T}(i) = \{\boldsymbol{q}^\dagger \mid N(\boldsymbol{b}_j \mid \boldsymbol{q}) = N(\boldsymbol{b}_j \mid \boldsymbol{p}), 1 \le {}^\forall j \le i, \; N(\boldsymbol{a} : \boldsymbol{w} \mid \boldsymbol{q}) = N(\boldsymbol{a} : \boldsymbol{w} \mid \boldsymbol{p}) = 0, \; \boldsymbol{q} \in \mathcal{X}^{[m,n]}, \boldsymbol{q} \text{ is primitive}\}$$

then $\boldsymbol{y}$ is also an element of

$$\mathcal{T}(i+1) = \{\boldsymbol{s}^\dagger \mid N(\boldsymbol{b}_j \mid \boldsymbol{s}) = N(\boldsymbol{b}_j \mid \boldsymbol{p}), 1 \le {}^\forall j \le i, \; N(\boldsymbol{a} : \boldsymbol{w} \mid \boldsymbol{s}) = N(\boldsymbol{a} : \boldsymbol{w} \mid \boldsymbol{p}) = 0, \; N(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c} \mid \boldsymbol{s}) = N(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c} \mid \boldsymbol{p}) = 0,$$

$$s \in \mathcal{X}^{[m,n]}, s \text{ is primitive}\}.$$

Hence, $\mathcal{T}(i+1) \supseteq \mathcal{T}(i)$ holds. Similarly, if (24) holds, then $\mathcal{T}(i+1) \supseteq \mathcal{T}(i)$ holds.

If (25) holds, then $N(\boldsymbol{d} : \boldsymbol{w}) = 0$ for any column $\boldsymbol{d} \in \left(\mathcal{X}^{[|\boldsymbol{b}_{i+1}|_r, 1]} \backslash \{\boldsymbol{a}\}\right)$ because $N(\boldsymbol{w}) = N(\boldsymbol{a} : \boldsymbol{w})$. Hence, we have $N(\boldsymbol{w} : \boldsymbol{c}) = N(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c})$ because $N(\boldsymbol{d} : \boldsymbol{w}) = 0$ for any column $\boldsymbol{d} \in \left(\mathcal{X}^{[|\boldsymbol{b}_{i+1}|_r, 1]} \backslash \{\boldsymbol{a}\}\right)$. Let $N(\boldsymbol{w} : \boldsymbol{c}) = N(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c}) = C \geq 0$. If $\boldsymbol{y}$ is an element of

$$\begin{aligned}\mathcal{T}(i) = \{\boldsymbol{q}^\dagger \mid\ & N(\boldsymbol{b}_j \mid \boldsymbol{q}) = N(\boldsymbol{b}_j \mid \boldsymbol{p}), 1 \leq {}^\forall j \leq i, \\ & N(\boldsymbol{w} \mid \boldsymbol{q}) = N(\boldsymbol{w} \mid \boldsymbol{p}) = N(\boldsymbol{a} : \boldsymbol{w} \mid \boldsymbol{q}) = N(\boldsymbol{a} : \boldsymbol{w} \mid \boldsymbol{p}), \\ & N(\boldsymbol{w} : \boldsymbol{c} \mid \boldsymbol{q}) = N(\boldsymbol{w} : \boldsymbol{c} \mid \boldsymbol{p}) = C, \\ & \boldsymbol{q} \in \mathcal{X}^{[m,n]}, \boldsymbol{q} \text{ is primitive}\}\end{aligned}$$

then $\boldsymbol{y}$ is also an element of

$$\begin{aligned}\mathcal{T}(i+1) = \{\boldsymbol{s}^\dagger \mid\ & N(\boldsymbol{b}_j \mid \boldsymbol{s}) = N(\boldsymbol{b}_j \mid \boldsymbol{p}), 1 \leq {}^\forall j \leq i, \\ & N(\boldsymbol{w} \mid \boldsymbol{s}) = N(\boldsymbol{w} \mid \boldsymbol{p}) = N(\boldsymbol{a} : \boldsymbol{w} \mid \boldsymbol{s}) = N(\boldsymbol{a} : \boldsymbol{w} \mid \boldsymbol{p}), \\ & N(\boldsymbol{w} : \boldsymbol{c} \mid \boldsymbol{s}) = N(\boldsymbol{w} : \boldsymbol{c} \mid \boldsymbol{p}) = C, \\ & N(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c} \mid \boldsymbol{s}) = N(\boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c} \mid \boldsymbol{p}) = C, \\ & \boldsymbol{s} \in \mathcal{X}^{[m,n]}, \boldsymbol{s} \text{ is primitive}\}.\end{aligned}$$

Therefore, $\mathcal{T}(i+1) \supseteq \mathcal{T}(i)$ holds. Similarly, if (26) holds, then $\mathcal{T}(i+1) \supseteq \mathcal{T}(i)$ holds.

Hence, in any case, we have $\mathcal{T}(i+1) \supseteq \mathcal{T}(i)$, which completes the proof. □

Remark 1. *Lemma 1 holds when the rearranged order of elements of $\mathcal{B}(\boldsymbol{p})$ is used, that is $(\mathcal{B}_0(\boldsymbol{p}), \mathcal{B}_1(\boldsymbol{p}), \mathcal{B}_2(\boldsymbol{p}), \mathcal{B}_3(\boldsymbol{p}))$. When $\boldsymbol{b}_{i+1}$ $(= \boldsymbol{a} : \boldsymbol{w} : \boldsymbol{c})$ such that $|\boldsymbol{b}_{i+1}|_c \geq 2$ (resp. $\boldsymbol{b}_{i+1}$ $(= \boldsymbol{e}/\boldsymbol{v}/\boldsymbol{g})$ such that $|\boldsymbol{b}_{i+1}|_r \geq 2$) is encoded based on the rearranged order, $N(\boldsymbol{a} : \boldsymbol{w}), N(\boldsymbol{w} : \boldsymbol{c})$, and $N(\boldsymbol{w})$ (resp. $N(\boldsymbol{e}/\boldsymbol{v}), N(\boldsymbol{v}/\boldsymbol{g})$, and $N(\boldsymbol{v})$) have been already encoded or can be inferred.*

We give the reasons why Remark 1 holds. Since $|\boldsymbol{b}_{i+1}|_c \geq 2$ (resp. $|\boldsymbol{b}_{i+1}|_c \geq 2$) from the assumption, $\boldsymbol{b}_{i+1} \in (\mathcal{B}_2(\boldsymbol{p}) \cup \mathcal{B}_3(\boldsymbol{p}))$. For $\boldsymbol{b}_{i+1} \in (\mathcal{B}_2(\boldsymbol{p}) \cup \mathcal{B}_3(\boldsymbol{p}))$, any block $\boldsymbol{x} \in \{\boldsymbol{a} : \boldsymbol{w}, \boldsymbol{w} : \boldsymbol{c}, \boldsymbol{w}\}$ (resp. $\{\boldsymbol{e}/\boldsymbol{v}, \boldsymbol{v}/\boldsymbol{g}, \boldsymbol{v}\}$) is the empty block or an element of $\mathcal{B}(\boldsymbol{p})$ because $\boldsymbol{x}$ is a proper subblock of $\boldsymbol{b}_{i+1}$ and any proper subblock of $\boldsymbol{b}_{i+1}$ is in $\mathcal{D}(\bar{\boldsymbol{p}})$ from the definition of $\mathcal{B}(\boldsymbol{p})$. When $\boldsymbol{x}$ is the empty block, $N(\boldsymbol{x}) = mn$ from the definition, so $N(\boldsymbol{x})$ can be inferred.

In case that $\boldsymbol{b}_{i+1} \in \mathcal{B}_2(\boldsymbol{p})$, any block $\boldsymbol{x} \in \{\boldsymbol{a} : \boldsymbol{w}, \boldsymbol{w} : \boldsymbol{c}, \boldsymbol{w}\}$ (resp. $\{\boldsymbol{e}/\boldsymbol{v}, \boldsymbol{v}/\boldsymbol{g}, \boldsymbol{v}\}$) is the empty block or an element of $(\mathcal{B}_1(\boldsymbol{p}) \cup \mathcal{B}_2(\boldsymbol{p}))$. For $\boldsymbol{x} \in (\mathcal{B}_1(\boldsymbol{p}) \cup \mathcal{B}_2(\boldsymbol{p}))$, $\boldsymbol{x}$ comes before $\boldsymbol{b}_{i+1}$ in the rearranged order because $|\boldsymbol{x}|_r < |\boldsymbol{b}_{i+1}|_r$ or $|\boldsymbol{x}|_r = |\boldsymbol{b}_{i+1}|_r$ and $|\boldsymbol{x}|_c < |\boldsymbol{b}_{i+1}|_r$.

In case that $\boldsymbol{b}_{i+1} \in \mathcal{B}_3(\boldsymbol{p})$, any block $\boldsymbol{x} \in \{\boldsymbol{a} : \boldsymbol{w}, \boldsymbol{w} : \boldsymbol{c}, \boldsymbol{w}\}$ (resp. $\{\boldsymbol{e}/\boldsymbol{v}, \boldsymbol{v}/\boldsymbol{g}, \boldsymbol{v}\}$) is the empty block or an element of $(\mathcal{B}_1(\boldsymbol{p}) \cup \mathcal{B}_2(\boldsymbol{p}) \cup \mathcal{B}_3(\boldsymbol{p}))$. For $\boldsymbol{x} \in (\mathcal{B}_1(\boldsymbol{p}) \cup \mathcal{B}_2(\boldsymbol{p}) \cup \mathcal{B}_3(\boldsymbol{p}))$, $\boldsymbol{x}$ comes before $\boldsymbol{b}_{i+1}$ in the rearranged order because $|\boldsymbol{x}|_r < |\boldsymbol{b}_{i+1}|_r$ or $|\boldsymbol{x}|_r = |\boldsymbol{b}_{i+1}|_r$ and $|\boldsymbol{x}|_c < |\boldsymbol{b}_{i+1}|_r$.

Therefore, $N(\boldsymbol{x})$ has been already encoded or can be inferred when $N(\boldsymbol{b}_{i+1})$ is encoded based on the rearranged order, where $\boldsymbol{x} \in \{\boldsymbol{a} : \boldsymbol{w}, \boldsymbol{w} : \boldsymbol{c}, \boldsymbol{w}\}$ (resp. $\{\boldsymbol{e}/\boldsymbol{v}, \boldsymbol{v}/\boldsymbol{g}, \boldsymbol{v}\}$).

Lemma A (Corollary 2 [11])**.** *For a positive integer n such that $n = a_1 + a_2 + \cdots + a_d$ and non-negative integers $a_1, a_2, \ldots, a_d$,*

$$\frac{n!}{\Pi_{i=1}^d a_i!} \leq \frac{n^n}{\Pi_{i=1}^d a_i^{a_i}},$$

where $0! = 1$ and $0^0 = 1$.

Lemma 2. *For $1 \leq k \leq m$ and $1 \leq l \leq n$, we have*

$$\log_2 |\mathcal{T}(k,l)| \leq -\frac{mn}{kl} \sum_{\boldsymbol{w} \in \mathcal{X}^{[k,l]}} \frac{N(\boldsymbol{w})}{mn} \log_2 \frac{N(\boldsymbol{w})}{mn}.$$

Proof. When $|\mathcal{T}(k,l)|$ is calculated, $|\mathcal{T}(k-1,l)|$ and $|\mathcal{T}(k,l-1)|$ are known. Therefore, $N(\boldsymbol{u})$ and $N(\boldsymbol{v})$ for $\boldsymbol{u} \in \mathcal{X}^{[k-1,l]}$ and $\boldsymbol{v} \in \mathcal{X}^{[k,l-1]}$ are also known.

We show the statement based on the following claim.

Claim: For any $1 \leq k \leq m$ and $1 \leq l \leq n$,

$$|\mathcal{T}(k,l))|^{kl} \leq \frac{(mn)!}{\prod_{\boldsymbol{w} \in \mathcal{X}^{[k,l]}} N(\boldsymbol{w})!}. \tag{27}$$

Once the claim is shown, then we have from (27) that

$$kl \log_2 |\mathcal{T}(k,l))| \leq \log_2 \frac{(mn)!}{\prod_{\boldsymbol{w} \in \mathcal{X}^{[k,l]}} N(\boldsymbol{w})!} \tag{28}$$

Hence, from Lemma A,

$$\log_2 |\mathcal{T}(k,l)| \leq -\frac{mn}{kl} \sum_{\boldsymbol{w} \in \mathcal{X}^{[k,l]}} \frac{N(\boldsymbol{w})}{mn} \log_2 \frac{N(\boldsymbol{w})}{mn} \tag{29}$$

as desired. Thus, we focus on showing the claim in accordance with the following steps.

Step 1: For $k = 1$ and $1 \leq l \leq n$ or for $1 \leq k \leq m$ and $l = 1$, show

$$|\mathcal{T}(k,l)| \leq \left(\prod_{\boldsymbol{y} \in \mathcal{X}^{[k,l-1]}} \binom{N(\boldsymbol{y})}{N(\boldsymbol{y} : c_0), \ldots, N(\boldsymbol{y} : c_{|\mathcal{X}|^k - 1})} \right)^{\frac{1}{k}} \tag{30}$$

$$|\mathcal{T}(k,l)| \leq \left(\prod_{z \in \mathcal{X}^{[k-1,l]}} \binom{N(z)}{N(z/r_0), \ldots, N(z/r_{|\mathcal{X}|^l - 1})} \right)^{\frac{1}{l}} \tag{31}$$

where $\mathcal{X}^{[k,1]} = \{\boldsymbol{c}_0, \ldots \boldsymbol{c}_{|\mathcal{X}|^k - 1}\}$ and $\mathcal{X}^{[1,l]} = \{\boldsymbol{r}_0, \ldots \boldsymbol{r}_{|\mathcal{X}|^l - 1}\}$.

Step 2: Fix m and n. For any $1 \leq k \leq m$, show

$$|\mathcal{T}(k,1)|^k \leq \frac{(mn)!}{\prod_{\boldsymbol{w} \in \mathcal{X}^{[k,1]}} N(\boldsymbol{w})!}. \tag{32}$$

Step 3: For any $1 \leq k \leq m$ and $1 \leq l \leq n$, show (30) and (31) hold.

Step 4: Show that the claim holds.

Proof of Step 1: For $k = 1$ and $1 \leq l \leq n$, $|\mathcal{T}(k,l)|$ never exceeds the product of these possible combinations over all substrings of length l, so that (30) holds. Similarly, for $1 \leq k \leq m$ and $l = 1$, (31) holds.

Proof of Step 2: When $k = 1$, the right-hand side of (32) is given by

$$\frac{(mn)!}{\prod_{\boldsymbol{w} \in \mathcal{X}^{[1,1]}} N(\boldsymbol{w})!} = \prod_{z \in \mathcal{X}^{[0,1]}} \binom{N(z)}{N(z/r_0), \ldots, N(z/r_{|\mathcal{X}|-1})}. \quad (33)$$

From Step 1, (32) holds for $k = 1$. We assume that (32) holds when $k = i \geq 1$. When $k = i + 1$, from Step 1,

$$|\mathcal{T}(i+1,1)| \leq \prod_{z \in \mathcal{X}^{[i,1]}} \binom{N(z)}{N(z/r_0), \ldots, N(z/r_{|\mathcal{X}|-1})}. \quad (34)$$

Since $|\mathcal{T}(k+1,l)| \leq |\mathcal{T}(k,l)|$, we have from (34) that

$$\begin{aligned} &|\mathcal{T}(i+1,1)|^{i+1} \\ &\leq |\mathcal{T}(i,1)|^{i} \prod_{z \in \mathcal{X}^{[i,1]}} \binom{N(z)}{N(z/r_0), \ldots, N(z/r_{|\mathcal{X}|-1})} \quad (35) \\ &\leq \frac{(mn)!}{\prod_{z \in \mathcal{X}^{[i,1]}} N(z)!} \frac{\prod_{z \in \mathcal{X}^{[i,1]}} N(z)!}{\prod_{z \in \mathcal{X}^{[i,1]}} N(z/r_0)! \cdots N(z/r_{|\mathcal{X}|-1})!} \quad (36) \\ &= \frac{(mn)!}{\prod_{\boldsymbol{w} \in \mathcal{X}^{[i+1,1]}} N(\boldsymbol{w})!}. \quad (37) \end{aligned}$$

Therefore, (32) holds for any $1 \leq k \leq m$.

Proof of Step 3: Observe that (32) is equivalent to

$$\begin{aligned} |\mathcal{T}(k,1)| &\leq \left(\frac{(mn)!}{\prod_{\boldsymbol{y} \in \mathcal{X}^{[k,1]}} N(\boldsymbol{y})!} \right)^{\frac{1}{k}} \\ &= \left(\prod_{\boldsymbol{y} \in \mathcal{X}^{[k,0]}} \binom{N(\boldsymbol{y})}{N(\boldsymbol{y}:c_0), \ldots, N(\boldsymbol{y}:c_{|\mathcal{X}|^k-1})} \right)^{\frac{1}{k}}. \quad (38) \end{aligned}$$

From (38), (30) holds for $1 \leq k \leq m$ and $l = 1$. We then prove that (30) holds for any $1 \leq k \leq m$ and $1 \leq l \leq n$ by mathematical induction on l for fixed $k \geq 1$, m and n.

Assume that (30) holds when $l = j \geq 1$. For $l = j$, from the assumption,

$$\begin{aligned} &|\mathcal{T}(k,j)| \leq \\ &\left(\prod_{\boldsymbol{y} \in \mathcal{X}^{[k,j-1]}} \binom{N(\boldsymbol{y})}{N(\boldsymbol{y}:c_0), \ldots, N(\boldsymbol{y}:c_{|\mathcal{X}|^k-1})} \right)^{\frac{1}{k}} \\ &= \left(\frac{\prod_{\boldsymbol{y} \in \mathcal{X}^{[k,j-1]}} (N(\boldsymbol{y})!)}{\prod_{\boldsymbol{y}' \in \mathcal{X}^{[k,j]}} (N(\boldsymbol{y}')!)} \right)^{\frac{1}{k}} \quad (39) \end{aligned}$$

From (39), $|\mathcal{T}(k,j)|^k$ never exceeds the product of possible combinations over all subblocks $\boldsymbol{y}' = \boldsymbol{y} : c_a$ $(0 \leq a \leq |\mathcal{X}|^k - 1)$ for all $\boldsymbol{y}$. Moreover, since $N(\boldsymbol{y}') = N(\boldsymbol{y}':c_0) + \cdots + N(\boldsymbol{y}':c_{|\mathcal{X}|^k-1})$, $|\mathcal{T}(k,j+1)|^k$ never exceeds the product of possible combinations over all subblocks $\boldsymbol{y}' : c_a$ $(0 \leq a \leq |\mathcal{X}|^k - 1)$ for all $\boldsymbol{y}'$. Therefore,

$$\begin{aligned} &|\mathcal{T}(k,j+1)| \leq \\ &\left(\frac{\prod_{\boldsymbol{y}' \in \mathcal{X}^{[k,j]}} N(\boldsymbol{y}')!}{\prod_{\boldsymbol{y}' \in \mathcal{X}^{[k,j]}} N(\boldsymbol{y}':c_0)! \cdots N(\boldsymbol{y}':c_{|\mathcal{X}|^k-1})!} \right)^{\frac{1}{k}} \quad (40) \\ &= \left(\prod_{\boldsymbol{y}' \in \mathcal{X}^{[k,j]}} \binom{N(\boldsymbol{y}')}{N(\boldsymbol{y}':c_0), \ldots, N(\boldsymbol{y}':c_{|\mathcal{X}|^k-1})} \right)^{\frac{1}{k}} \quad (41) \end{aligned}$$

Hence, (30) holds for $1 \leq k \leq m$ and $1 \leq l \leq n$. Similarly, by swapping the row k and the column l, (31) holds.

Proof of Step 4: From (32), (27) holds for $1 \leq k \leq m$ and $l = 1$. We next prove that (30) holds for any $1 \leq k \leq m$ and $1 \leq l \leq n$ by induction on k for fixed $l \geq 1$, m and n.

Assume that (27) holds when $k = i \geq 1$. For $k = i + 1$, from (31),

$$\begin{aligned} &|\mathcal{T}(i+1,l)|^{l} \leq \\ &\prod_{\boldsymbol{w} \in \mathcal{X}^{[i,l]}} \binom{N(\boldsymbol{w})}{N(\boldsymbol{w}/r_0), \ldots, N(\boldsymbol{w}/r_{|\mathcal{X}|^l-1})}. \quad (42) \end{aligned}$$

From (34),

$$\begin{aligned} &|\mathcal{T}(i+1,l)|^{(i+1)l} \\ &\leq |\mathcal{T}(i,l)|^{il} \prod_{\boldsymbol{w} \in \mathcal{X}^{[i,l]}} \binom{N(\boldsymbol{w})}{N(\boldsymbol{w}/r_0), \ldots, N(\boldsymbol{w}/r_{|\mathcal{X}|^l-1})} \quad (43) \\ &= \frac{(mn)!}{\prod_{\boldsymbol{w} \in \mathcal{X}^{[i,l]}} N(\boldsymbol{w})!} \frac{\prod_{\boldsymbol{w} \in \mathcal{X}^{[i,l]}} N(\boldsymbol{w})!}{\prod_{\boldsymbol{w} \in \mathcal{X}^{[i,l]}} N(\boldsymbol{w}/r_0)! \cdots N(\boldsymbol{w}/r_{|\mathcal{X}|^l-1})!} \quad (44) \\ &= \frac{(mn)!}{\prod_{\boldsymbol{w}' \in \mathcal{X}^{[i+1,l]}} N(\boldsymbol{w}')!}. \quad (45) \end{aligned}$$

Hence, (27) holds for $1 \le k \le m$ and $1 \le l \le n$. From (27),

$$kl \log_2 |\mathcal{T}(k,l))| \le \log_2 \frac{(mn)!}{\prod_{\boldsymbol{w} \in \mathcal{X}^{[k,l]}} N(\boldsymbol{w})!}. \quad (46)$$

Hence, from Lemma A,

$$\log_2 |\mathcal{T}(k,l)| \le -\frac{mn}{kl} \sum_{\boldsymbol{w} \in \mathcal{X}^{[k,l]}} \frac{N(\boldsymbol{w})}{mn} \log_2 \frac{N(\boldsymbol{w})}{mn}. \quad (47)$$

□

Lemma 3. *For* $K = \left\lfloor \sqrt{\log_{|\mathcal{X}|} \log_{|\mathcal{X}|} m} \right\rfloor$ *and* $L = \left\lfloor \sqrt{\log_{|\mathcal{X}|} \log_{|\mathcal{X}|} n} \right\rfloor$,

$$\limsup_{m,n\to\infty} -\frac{1}{KL} \sum_{\boldsymbol{w} \in \mathcal{X}^{[K,L]}} \left(E\left[\frac{N(\boldsymbol{w} \mid X^{[m,n]})}{mn}\right] \log_2 E\left[\frac{N(\boldsymbol{w} \mid X^{[m,n]})}{mn}\right]\right) = \hat{H}(\mathbf{X}).$$

Proof. For $\boldsymbol{w} \in \mathcal{X}^{[K,L]}$, $P_{X^{[m,n]}}(\boldsymbol{w})$ can be written as

$$E\left[\frac{|\{(i,j) \text{ s.t. } X_{(i,j)}^{(i+K-1,j+L-1)} = \boldsymbol{w}, 1 \le i \le m', 1 \le j \le n'\}|}{m'n'}\right]$$

where m' and n' are $m-K+1$ and $n-L+1$, respectively, and (i,j) is the coordinate. For $\boldsymbol{p}$, let $N'(\boldsymbol{w} \mid \boldsymbol{p})$ be $|\{(i,j) \text{ s.t. } \boldsymbol{p}_{(i,j)}^{(i+K-1,j+L-1)} = \boldsymbol{w}, 1 \le i \le m', 1 \le j \le n'\}|$. Moreover, $\frac{N(\boldsymbol{w} \mid \boldsymbol{p})}{mn}$ can be written as $\left(\frac{N'(\boldsymbol{w} \mid \boldsymbol{p})+\delta}{m'n'}\right)\left(\frac{m'n'}{mn}\right)$ where $0 \le \delta \le (K-1)(n-L+1) + (L-1)m$ from (2). Because K and L are respectively $\left\lfloor \sqrt{\log_{|\mathcal{X}|} \log_{|\mathcal{X}|} m} \right\rfloor$ and $\left\lfloor \sqrt{\log_{|\mathcal{X}|} \log_{|\mathcal{X}|} n} \right\rfloor$, $\frac{N(\boldsymbol{w}|\boldsymbol{p})}{mn}$ converges to $\frac{N'(\boldsymbol{w}|\boldsymbol{p})}{m'n'}$ as m and n go to infinity. Since $E\left[\frac{N'(\boldsymbol{w}|X^{[m,n]})}{m'n'}\right] = P_{X^{[m,n]}}(\boldsymbol{w})$,

$$\begin{aligned}
&\limsup_{m,n\to\infty} -\frac{1}{KL} \sum_{\boldsymbol{w} \in \mathcal{X}^{[K,L]}} \left(E\left[\frac{N(\boldsymbol{w} \mid X^{[m,n]})}{mn}\right] \log_2 E\left[\frac{N(\boldsymbol{w} \mid X^{[m,n]})}{mn}\right]\right) \\
&= \limsup_{m,n\to\infty} -\frac{1}{KL} \sum_{\boldsymbol{w} \in \mathcal{X}^{[K,L]}} P_{X^{[m,n]}}(\boldsymbol{w}) \log_2 P_{X^{[m,n]}}(\boldsymbol{w}) \\
&= \limsup_{m,n\to\infty} \frac{H(X^{[K,L]})}{KL} = \hat{H}(\mathbf{X}).
\end{aligned}$$

□

We are now in a position of proving Theorem 1.

(Proof of Theorem 1). As for $\ell_0(\boldsymbol{p})$, from the assumption and $m \le n$, we have $\ell_0(\boldsymbol{p}) \le 2(2\lceil \log_2 n \rceil + 1) + \lceil \log_2 mn \rceil$, where $(2\lceil \log_2 n \rceil + 1)$ and $\lceil \log_2 mn \rceil$ are the costs of the Elias code for integers n and $\epsilon(\text{rank}(\boldsymbol{p}))$, respectively. As for $\ell_1(\boldsymbol{p})$, the cost of $N(b_i)$ in Condition (P-i) is $\lceil \log_2 mn \rceil$ bits from (18), so that $\ell_1(\boldsymbol{p}) \le (|\mathcal{X}|-1)\lceil \log_2 mn \rceil$. As for $\ell_2(\boldsymbol{p})$, since $I(\boldsymbol{b}_i) \le mn$ and $I'(\boldsymbol{b}_i) \le mn$, the costs of $I(\boldsymbol{b}_i)$ and $I'(\boldsymbol{b}_i)$ are at most $\log_2 mn$ bits. Moreover, because $m \le n$ and $K \le L$,

$$\begin{aligned}
\ell_2(\boldsymbol{p}) &\le \sum_{h=1}^{K} \sum_{w=1}^{L} |\mathcal{X}|^{hw} \log_2 mn \le L^2 |\mathcal{X}|^{L^2} \log_2 mn \\
&\le 2(\log_{|\mathcal{X}|} \log_{|\mathcal{X}|} n)(\log_{|\mathcal{X}|} n)(\log_2 n).
\end{aligned}$$

Therefore,

$$\lim_{m,n\to\infty} (\ell_0(\boldsymbol{p}) + \ell_1(\boldsymbol{p}) + \ell_2(\boldsymbol{p}))/mn = 0. \quad (48)$$

As for $\ell_3(\boldsymbol{p})$, the cost of $N(\boldsymbol{b}_i)$ is $-\log_2(|\mathcal{T}(i)|/|\mathcal{T}(i-1)|)$ bits from (20). Similarly, the cost of $N(\boldsymbol{b}_{i+1})$ is $-\log_2(|\mathcal{T}(i+1)|/|\mathcal{T}(i)|)$ where $N(\boldsymbol{b}_{i+1})$ is encoded subsequent to $N(\boldsymbol{b}_i)$ which has been encoded. The denominator $|\mathcal{T}(i)|$ for $N(\boldsymbol{b}_{i+1})$ is equal to the previous numerator $|\mathcal{T}(i)|$ for $N(\boldsymbol{b}_i)$, so that they are canceled. In other words,

$$\begin{aligned}
&-\log_2(|\mathcal{T}(i)|/|\mathcal{T}(i-1)|) - \log_2(|\mathcal{T}(i+1)|/|\mathcal{T}(i)|) \\
&= -\log_2(|\mathcal{T}(i+1)|/|\mathcal{T}(i-1)|)).
\end{aligned}$$

On the other hand, $N(\boldsymbol{b}_{i+1})$ may not be encoded when $N(\boldsymbol{b}_{i+1})$ does not satisfy the conditions as shown in (P-ii). We assume that $N(\boldsymbol{b}_j)$ is encoded while $N(\boldsymbol{b}_{i+k})$ $(1 \le k < j)$ are not encoded. The cost of $N(\boldsymbol{b}_j)$ is $-\log_2(|\mathcal{T}(j)|/|\mathcal{T}(j-1)|)$. From Lemma 1, $|\mathcal{T}(j-1)| = |\mathcal{T}(i)|$ holds because any $N(\boldsymbol{b}_{i+k})$ $(1 \le k < j)$ does not satisfy the conditions as shown in (P-ii) from the assumption. Therefore, they are also canceled. In other words,

$$\begin{aligned}
&-\log_2(|\mathcal{T}(i)|/|\mathcal{T}(i-1)|) - \log_2(|\mathcal{T}(j)|/|\mathcal{T}(j-1)|) \\
&= -\log_2(|\mathcal{T}(j)|/|\mathcal{T}(i-1)|)).
\end{aligned}$$

Moreover, since $|\mathcal{T}(|\mathcal{B}(\boldsymbol{p})|)| = 1$,

$$\ell_3(\boldsymbol{p}) = \log_2 |\mathcal{T}(S-1)|, \quad (49)$$

where S is the index of the first block $\boldsymbol{b}_S \in \mathcal{B}_3(\boldsymbol{p})$ which is encoded by an arithmetic coder or something like that. From Lemma 1, $|\mathcal{T}(S-1)| = |\mathcal{T}(K,L)|$. Therefore,

$$\ell_3(\boldsymbol{p}) = \log_2 |\mathcal{T}(K,L)|. \quad (50)$$

From (50) and Lemma 2,

$$\ell_3(\boldsymbol{p}) \le -\frac{mn}{KL} \sum_{\boldsymbol{w} \in \mathcal{X}^{[K,L]}} \frac{N(\boldsymbol{w})}{mn} \log_2 \frac{N(\boldsymbol{w})}{mn}. \quad (51)$$

Therefore,

$$E\left[\frac{\ell_3(X^{[m,n]})}{mn}\right] \le -\frac{1}{KL} \sum_{\boldsymbol{w} \in \mathcal{X}^{[K,L]}} E\left[\frac{N(\boldsymbol{w}|X^{[m,n]})}{mn} \log_2 \frac{N(\boldsymbol{w}|X^{[m,n]})}{mn}\right].$$

From Jensen's inequality,

$$E\left[\frac{N(\boldsymbol{w}|X^{[m,n]})}{mn}\right] E\left[\log_2 \frac{N(\boldsymbol{w}|X^{[m,n]})}{mn}\right] \leq E\left[\frac{N(\boldsymbol{w}|X^{[m,n]})}{mn} \log_2 \frac{N(\boldsymbol{w}|X^{[m,n]})}{mn}\right].$$

Therefore, from Lemma 3,

$$\limsup_{m,n\to\infty} E\left[\frac{\ell_3(X^{[m,n]})}{mn}\right] \leq \hat{H}(\mathbf{X}). \tag{52}$$

From (48) and (52),

$$\limsup_{m,n\to\infty} E\left[\frac{\ell(X^{[m,n]})}{mn}\right] \leq \hat{H}(\mathbf{X}). \tag{53}$$

The proposed code is a prefix code, so Kraft's inequality holds. Therefore,

$$\limsup_{m,n\to\infty} E\left[\frac{\ell(X^{[m,n]})}{mn}\right] \geq \hat{H}(\mathbf{X}).$$

□

From Remark 1.7.3 [17], if $\mathbf{X}$ is a stationary source, $\hat{H}(\mathbf{X})$ can be expressed by $H(\mathbf{X})(:= \lim_{m,n\to\infty} \frac{H(X^{[m,n]})}{mn})$, which is the entropy rate of $\mathbf{X}$. Therefore, if $\mathbf{X}$ is a stationary source, the average codeword length of the proposed algorithm converges to $H(\mathbf{X})$ as m and n go to infinity.

6. Conclusion

We proposed a new CSE for a 2D source which uses the flat torus of the source for reducing the computational time and compress a 2D source without converting to a 1D source. The total number of output blocks of the new CSE is polynomial, while that of the conventional CSE is exponential with respect to the source size. The new CSE encodes the source in a block-by-block fashion, while the conventional CSE does in a line-by-line fashion. Moreover, we proved that an upper bound on the average codeword length of the proposed CSE converges to the sup-entropy rate for a 2D general source as the size of the input source goes to infinity. In other words, we proved the asymptotic optimality of the proposed CSE for a 2D general source. Furthermore, if a 2D general source is a stationary source, then the length converges to the entropy rate of the source as the size goes to infinity.

Acknowledgements

The authors truly thank the anonymous reviewers for their valuable comments. This work was supported by JSPS KAKENHI Grant Numbers JP17K00147 and JP17K00400.

References

[1] D. Dubé and V. Beaudoin, "Lossless data compression via substring enumeration," Proc. Data Compression Conference, pp.229–238, March 2010.
[2] H. Yokoo, "Asymptotic optimal lossless compression via the cse technique," Proc. Data Compression, Communications and Processing, pp.11–18, June 2011.
[3] D. Dubé and H. Yokoo, "The universality and linearity of compression by substring enumeration," Proc. IEEE International Symposium on Information Theory, pp.1619–1623, Aug. 2011.
[4] M. Béliveau and D. Dubé, "Improving compression via substring enumeration by explicit phase awareness," Proc. Data Compression Conference 2014, p.399, March 2014.
[5] S. Kanai, H. Yokoo, K. Yamazaki, and H. Kaneyasu, "Efficient implementation and empirical evaluation of compression by substring enumeration," IEICE Trans. Fundamentals, vol.E99-A, no.2, pp.601–611, Feb. 2016.
[6] M. Burrows and D.J. Wheeler, "A block-sorting lossless data compression algorithm," SRC Research Report, pp.73–93, May 1994.
[7] M. Crochemore, F. Mignosi, A. Restivo, and S. Salemi, "Data compression using antidictionaries," Proc. IEEE, vol.88, no.11, pp.1756–1768, Nov. 2000.
[8] T. Ota and H. Morita, "On antidictionary coding based on compacted substring automaton," Proc. IEEE International Symposium on Information Theory, pp.1754–1758, July 2013.
[9] T. Ota and H. Morita, "On a universal antidictionary coding for stationary ergodic sources with finite alphabet," Proc. International Symposium on Information Theory and its Applications, pp.294–298, Oct. 2014.
[10] K. Iwata and M. Arimura, "Lossless data compression via substring enumeration for k-th order Markov sources with a finite alphabet," IEICE Trans. Fundamentals, vol.E99-A, no.12, pp.2130–2135, Dec. 2016.
[11] T. Ota and H. Morita, "A compact tree representation of an antidictionary," IEICE Trans. Fundamentals, vol.E100-A, no.9, pp.1973–1984, Sept. 2017.
[12] M. Crochemore, G. Fici, R. Marcuş, and S.P. Pissis, "Linear-time sequence comparison using minimal absent words & applications," Proc. Latin American Symposium on Theoretical Informatics, pp.334–346, April 2016.
[13] D. Salomon and G. Motta, Handbook of Data Compression, Springer, 2010.
[14] P. Elias, "Universal codeword sets and representations of the integers," IEEE Trans. Inf. Theory, vol.IT-21, no.2, pp.194–203, March 1975.
[15] T. Ota, H. Morita, and A. Manada, "Compression by substring enumeration with a finite alphabet using sorting," Proc. International Symposium on Information Theory and its Applications, pp.587–591, Oct. 2018.
[16] A. Moffat and A. Turpin, Compression and Coding Algorithms, Kluwer Academic Publishers, 2002.
[17] T.S. Han, Information-Spectrum Methods in Information Theory, Springer-Verlag, 2002.
[18] T. Ota and H. Morita, "Two-dimensional source coding by means of subblock enumeration," Proc. IEEE International Symposium on Information Theory, pp.311–315, June 2017.

Takahiro Ota received the B.E. and Ph.D. degrees from the University of Electro-Communications, Tokyo, Japan, in 1993 and 2007, respectively. In 1997, he joined Nagano Prefectural Institute of Technology, Nagano, Japan, first a Lecturer at Department of Electronic Engineering, where from 2009, he was an Associate Professor. Since 2012, he is an Associate Professor with the Department of Computer & Systems Engineering. His current research interests are in information theory, source coding, and bio-informatics.

Hiroyoshi Morita received the B.E., M.E., and D.E. degrees from Osaka University, Osaka, Japan, in 1978, 1980 and 1983, respectively. In 1983, he joined Toyohashi University of Technology, Aichi, Japan as a Research Associate in the School of Production System Engineering. In 1990, he joined the University of Electro-Communications, Tokyo, Japan, first an Assistant Professor at the Department of Computer Science and Information Mathematics, where from 1992, he was an Associate Professor. Since 1995, he has been with the Graduate School of Information Systems, where from 2005, he is a Professor. He was a Visiting Fellow at the Institute of Experimental Mathematics, University of Essen, Essen, Germany during 1993–1994. His research interests are in combinatorial theory, information theory, and coding theory, with applications to the digital communication systems.

Akiko Manada received the M.S. degree from Tsuda College, Tokyo, Japan, in 2004, and the Ph.D. degree in Mathematics from Queen's University, Kingston, Canada, in 2009. She then worked at Claude Shannon Institute at University College Dublin, Ireland, as a postdoctoral fellow from 2009 through 2011. She was an assistant professor at Graduate School of Information Systems, the University of Electro-Communications, Tokyo, Japan from 2012 to 2018. Since April 2018, she has been a lecturer at Dept. of Information Science, the Shonan Institute of Technology, Kanagawa, Japan. Her research interests are discrete mathematics (especially in graph theory) and its applications towards coding theory.