A Joint DNA Encoding Approach Based on LZW and Arithmetic Encoding

Zhongyang Cheng, *Student Member, IEEE,*Qiang Liu*, *Member, IEEE,*Kun Yang, *Fellow, IEEE*
*corresponding author

***Abstract*—Molecular communication (MC) represents a novel approach to communication that employs nanoengineering and bioengineering technology to establish transient communication links in challenging environments. Deoxyribonucleic acid (DNA) molecular communication can transmit more and faster data than traditional molecular communication. Deoxyribonucleic acid (DNA) has been demonstrated to offer significant advantages over traditional information carriers, including its excellent storage density and structural stability, which renders it an ideal medium for information transmission. It is therefore imperative to investigate methods of increasing the data information density of DNA in order to reduce costs and enhance overall performance. LZW encoding is Lempel–Ziv–Welch encoding which creates a string table with shorter codes representing longer strings. Arithmetic coding is a compression process that involves the continuous refinement of probabilities of the input stream within an interval. A notable drawback of LZW coding is its suboptimal compression efficiency and the presence of data redundancy after dictionary mapping. Conversely, arithmetic coding attains compression efficiency that approaches the Shannon limit. In this study, we propose a novel DNA encoding method which is capable of adaptively generating coding streams in accordance with the characteristics of the stored content. The contribution of this paper is as follows: 1) A bespoke coding dictionary is constructed, which is capable of intelligently generating the corresponding coding stream in accordance with the specific characteristics of the file to be stored. 2) Utilising arithmetic coding techniques, these coding streams are converted into the final DNA sequence by means of compression techniques. Following comprehensive verification, it has been established that the information density of this encoding method is markedly superior to that of the prevailing mainstream encoding schemes.**

***Index Terms*—DNA encoding,arithmetic encoding, Lempel–Ziv–Welch encoding, molecular communication information density**

I. Introduction

PRESENTLY, optical and electromagnetic communication have become integral aspects of our daily lives, with each technological advancement significantly impacting human society. However, the prevailing communication methods are contingent upon a stable electromagnetic environment, which is ineffectual in certain contexts, including underwater, in battle, within mountain tunnels and in metal-confined spaces. In light of these considerations, molecular communication (MC) technology has emerged as a prominent area of interest and development. Among the various avenues of research, the utilisation of short sequences for the efficient transmission of information has emerged as a particularly active area of focus. Globally, the overwhelming majority of data transmission continues to rely on electromagnetic waves and opticalmedia. Nevertheless, even optimised disc technology necessitates the utilisation of millions of storage units, thereby occupying a considerable quantity of physical space for the storage of a single zettabyte of data. In the long term, the continued use of existing transmission media may give rise to a number of issues, including increased land and energy consumption, elevated data migration and maintenance costs, and heightened data security risks. Deoxyribonucleic acid is short for DNA. In light of the mounting challenge posed by the exponential growth in data volume, it has become imperative to identify a novel transmission medium. In this regard, DNA data transmission represents a pioneering solution. The four bases of DNA offer a high degree of flexibility in encoding any letter, number, or character, akin to the capacity of a binary number [1]. In the domain of data transmission, DNA has demonstrated exceptional capabilities, with a superior information density and stability compared to the current silicon-based transmission media. The primary work focus in the field of molecular communication is DNA-based molecular communication . The primary objective of this paper is to explore the efficient use of DNA as a medium for data transmission. The application of digital coding to DNA coding has revealed the necessity for refinement in mainstream coding schemes, particularly with regard to the development of DNA sequence data compression algorithms [2]. The existing LZW encoding algorithm is found to have certain deficiencies, including its low compression efficiency and its inability to achieve an adaptive expansion of the dictionary size. Concurrently, LZW encoding algorithm is continually generating a dictionary sequence, resulting in data redundancy in the final result, which is a disadvantage in its own right. The present paper proposes a methodology for the implementation of dynamic dictionary expansion, whereby the size of the input stream is calculated in order to ensure that the final dictionary is as small as possible. This approach obviates the need for blind expansion of the dictionary

This paper was supported by the National Natural Science Foundation of China under Grant No.62071101 and the Municipal Government of Quzhou under Grant No. 2023D008 and No. 2024D068.

Zhongyang Cheng is with Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, Zhejiang, China(e-mail:chengzy@std.uestc.edu.cn)

Qiang Liu is with Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China, Quzhou, Zhejiang, China(liuqiang@uestc.edu.cn)

Kun Yang is with University of Essex, Colchester, Essex, UK.(e-mail: kunyang@essex.ac.uk)

when the input flow is substantial. In addition, this paper proposes the integration of arithmetic coding for the purpose of enhancing the compression efficiency of the data following LZW encoding algorithm processing, with the objective of approaching the Shannon limit. The four-base coding process of DNA itself is more conducive to probability calculation and final data transmission, because in the final DNA sequence, it is necessary to attach the probability of different symbols [3]. The efficiency of processing DNA sequences is significantly superior to that of processing data due to the limited number of bases in DNA.

II. Research background

A. Introduction of DNA encoding

As an ancient and highly efficient information storage medium in living organisms. Deoxyribonucleic acid is short for DNA, has become a highly promising storage medium due to its high density, excellent durability and nearly infinite storage time. The storage of digital information in the form of DNA molecules, which is the basis of DNA encoding technology, involves encoding the core digital data in a sequence of nucleotides (A, C, G, T). These sequences can then be synthesised into DNA molecules for long-term storage. This chapter will provide a comprehensive examination of the two principal methodologies employed in the domain of DNA data coding. This chapter will introduce two main approaches to DNA data coding: lzw encoding and arithmetic encoding. It will then present an innovative scheme that improves and merges these two methods, achieving a higher information density coding effect than a single method [4] [5].

B. DNA molecular communication process

DNA is a biological macromolecule comprising four distinct nucleotides, which are themselves composed of phosphoric acid, sugar, and nitrogenous bases. In the context of data storage and transmission, DNA displays the potential to surpass traditional electronic storage media, exhibiting high information density and long-term stability as its most notable characteristics. The utilisation of DNA in data transmission can be conceptualised as an organic digital signal carrier, whereby digital information is converted into a chemical form, thereby facilitating the storage and transmission of information. The process of DNA molecular communication is contingent upon the accurate encoding and transmission of information [6]. This process entails the conversion of a digital signal into a chemical signal through the encoding of a stream of data into a DNA sequence. The digital nature of DNA renders it a distinctive digital medium, wherein the nucleotide bases (adenine, A; thymine, T; cytosine, C; guanine, G) function as symbols for the digital code [7]. Current mainstream DNA encoding uses an efficient mapping mechanism, shown in Table 1 [8], that pairs binary data with DNA bases: A (adenine) for 00, T (thymine) for 01, C (cytosine) for 10, G (guanine) for 11. This mapping reduces the number of binary symbols by half, since each DNA base can represent two binary symbols. The prevailing approach to DNA encoding technology is to map binary code based on the four bases adenine (A), thymine (T), cytosine (C), and guanine (G), which provides a more stable and dense storage solution than traditional data transmission media. The process of encoding data into DNA entails the conversion of diverse forms of data, including multimedia formats such as movies, books, and images, into nucleotide sequences [9]. This process must be accomplished through chemical DNA synthesis techniques, thereby ensuring that each data sequence can be written with precision.

TABLE I
DNA encoding mapping rule.

Base type	Binary symbol
A	00
T	01
C	10
G	11

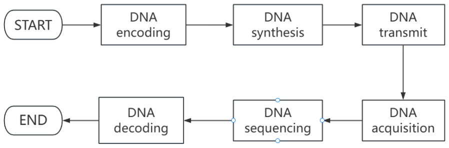

Fig. 1. DNA molecular communication process

The entire process of DNA molecular communication can be summarised in the following six steps: The coding stage involves the conversion of raw data into a format that can be interpreted by the DNA molecular communication system. The process of transforming raw data into a sequence of DNA bases. In this phase, the binary data is mapped to the four bases of DNA [10]. The synthesis phase involves the production of the actual DNA molecule, which is created from the encoded sequence using chemical synthesis methods. The actual DNA molecule is synthesised from the encoded sequence using chemical synthesis methods. The subsequent stage is transmission. The synthesised DNA molecules are transmitted by a variety of means, including test tubes and microfluidic devices [11]. The next stage is the acquisition of the DNA sequence. At the point of reception, the DNA molecules are extracted from the conveyance medium in preparation for the subsequent sequencing stages. The sequence information of extracted DNA molecules is read using DNA sequencing technology [12]. The process of decoding is initiated. The DNA sequence obtained by sequencing is converted back to the original digital data, thus completing the transmission of information. Figure 1 illustrates the complete pathway of DNA molecular communication, from the encoding to the decoding of data. Each step is meticulous and highly specialised, ensuring the accurate transmission of information at the molecular level [13]. As technology progresses, the utilisation of DNA as a data transmission medium is poised to become a significant field of study in the domain of information science.

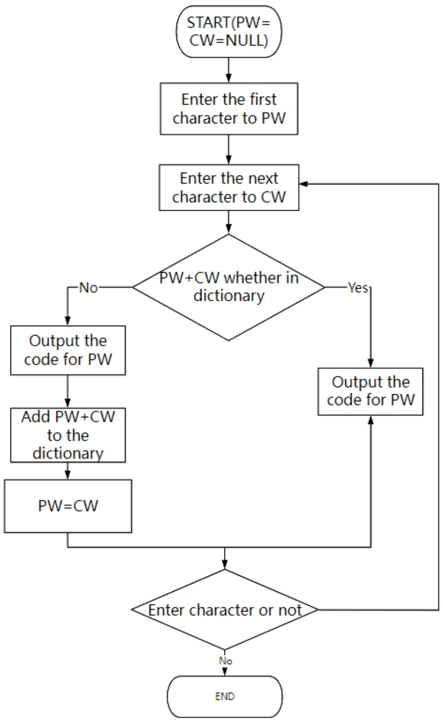
START(PW=
CW=NULL)
Enter the first character to PW
Enter the next character to CW
PW+CW whether in dictionary
No
Yes
Output the code for PW
Add PW+CW to the dictionary
PW=CW
Output the code for PW
Enter character or not
No
END

Fig. 2. LZW encoding flowchart.

III. DNA ENCODING BASED ON LZW ALGORITHM

A. Basic conception

LZW (Lempel-Ziv-Welch) encoding is an adaptive dictionary encoding technique that does not depend on the probability distribution of the input data, thus ensuring a relatively fast encoding speed [14]. LZW (Lempel-Ziv-Welch) encoding will be referred to as LZW encoding in the following. This technology is capable of effectively compressing data into a shorter encoding form while ensuring the integrity of the information without any loss of content. The fundamental premise of lzw encoding is the utilisation of repetitive patterns within data in order to achieve compression. The algorithmic process commences with the construction of an initial dictionary table comprising all potential symbols. A symbol may be either a single character present in the input data or a string of multiple characters. As the algorithm progresses, it performs a scan of the input data, seeking to identify any entries that appear within the dictionary table [15]. Upon identifying a matching entry, the algorithm proceeds to output the corresponding encoding. In the event of a lookup failure, this indicates the presence of a novel string pattern. In such an instance, the algorithm proceeds to incorporate the new entry into the dictionary table and subsequently outputs the corresponding encoding. LZW encoding is distinctive in its capacity to adaptively extend the dictionary table in order to accommodate novel patterns that emerge continuously in the input data [16]. As the encoding process continues, the number of entries in the dictionary increases, beginning with simple two-character combinations and subsequently expanding to longer strings. Each new entry comprises an existing prefix string and a new tail character. LZW encoding is rapid in part because its dictionary can be maintained with an efficient hash table structure, which allows both encoding and decoding operations to be conducted expeditiously. Furthermore, because LZW encoding employs a fixed-length symbol encoding mode, its encoding and decoding process is relatively straightforward, which further enhances the processing speed.

B. DNA encoding based on LZW algorithm code principle

Figure 2 illustrates the LZW encoding processThe comprehensive procedure is outlined as follows.The complete code necessitates the use of two variables, PW and CW, which denote the Previous Code Word and the Current Code Word, respectively. The specific encoding procedures are outlined below: In the event that the initial PW and CW are devoid of data, Step 2: We will read a new character, CW, and combine it with PW to form the string PW+CW. Step 3: The dictionary is then consulted to ascertain whether PW+CW is present. (1) In the event that PW+CW is present within the dictionary, PW is then equal to PW+CW. (2) In the event that PW+CW is not present within the dictionary, the PW symbol is output. Subsequently, a token map is created for PW+CW within the dictionary, and PW=CW is updated. Step 4:One must then return to Step 2 and repeat this process until all the characters in the original string have been read [17].

To illustrate, consider the generation encoding sequence ’abbababac’. In accordance with the aforementioned encoding rules, the initial character of the sequence is designated as a, and the prefix PW is considered to be vacant. Consequently, the current character CW is taken to be a, and the determination is made as to whether“PW+CW” (that is, a) is present in the dictionary. As a is in the dictionary, update PW to a (that is, PW=PW+WC) and continue reading the next character b. The second character is b, where the prefix PW is a, to determine whether “PW+CW” (ab) is in the dictionary. As ab is not present in the dictionary, it is necessary to add ab to the dictionary with an index of 256, and to output the ASCII value of the first character a, which is 97. The prefix PW is then updated to b, and the next character is read. The third character is b, where the prefix PW is b, and this determines whether “PW+CW”(bb) is in the dictionary. As ba is not present in the dictionary, it must be added to the dictionary with an index of 257. The ASCII value of the second character b, which is 98, is then output, PW is updated to b, and the next character is read. The fourth character is a, where the prefix PW is b, and determines whether “PW+CW”(ba) is in the dictionary. As ba is not present in the dictionary, it should be added to the dictionary, index 258, and the ASCII value of the third character b, which is 98, should be output. The prefix PW should then be updated to a, and the reading of the next character continued. The fifth character is b, where the prefix PW is a, in order to ascertain whether “PW+CW”(ab) is present in the dictionary. Given that ab is present in the dictionary, it is not necessary to output and update PW to ab at this time. Instead, the next character should be read. The sixth character is a, where the prefix PW is ab, to determine whether “PW+CW”(aba) is in the dictionary. As aba is not present in the dictionary, it is necessary to add aba to the dictionary with an index of 259. The ASCII value of ab is 256, PW should be updated to a, and the next character should be read. The seventh character is b, where the prefix PW is a, to determine whether "PW+CW"(ab) is in the dictionary. Since ab is not in the dictionary, at this time do not output and update PW to ab, and continue reading the next character. The eighth character is a, where the prefix PW is ab, to determine whether "PW+CW"(aba) is in the dictionary. As aba is present in the dictionary, it is not necessary to output and update PW to aba at this time, and the reading of the next character should continue. The ninth character is c, where the prefix PW is aba, to determine whether "PW+CW"(aba) is in the dictionary. As abac is not present in the dictionary, it should be added to the dictionary with an index of 260, and the index of aba, 259, should be output, and the next character should be read. Upon reading the final character, c, the ASCII table's index 99 is printed. The original sequence is then encoded as follows: ab ba ba a c, encoded 97 (a), 98 (b), 98 (b), 256 (ab), 259 (aba), 99 (c). It can thus be seen that the input used 72 bits, and the output after encoding only used 54 bits.

The LZW decoding principle is analogous to the encoding principle. The LZW decoding algorithm commences with the identical decoding dictionary as the encoding dictionary, comprising all potential prefix roots. The decoding algorithm is as follows: The initial stage of the decoding process entails the utilisation of the dictionary, which encompasses the entirety of potential prefix roots. Step 2: The code word CW is set to the first code word in the code word stream. Step 3: The prefix string CW is then output to the code word stream. Step 4: The preceding code word, PW, is identical to the current code word, CW. Step 5: The current code word, designated as CW, is equal to the subsequent code word in the code word stream. Step 6: Determine whether the prefix string CW is present within the dictionary [18]. (1) If the answer is affirmative, the prefix string CW is then output to the character stream. The present prefix, PW, is identical to the prefixed string, PW. The character currently designated as CW is identical to the initial character of the string of characters that constitutes the current prefix. The prefix string PW+CW should be added to the dictionary. (2) In the event of an affirmative response, the current prefix PW is equal to the prefixed string PW, and the current character CW is equal to the first character of the prefix string CW. The prefix string PW+CW should then be output to the character stream and added to the dictionary. Step 7: Determine whether there are still code words to be translated in the code word stream.(1) If Yes, go back to Step 4. (2) If no, no further action is required.

To illustrate, the final sequence is accepted to be 97, 98, 98, 256, 259. Upon encountering 97 in the base dictionary, it is decoded as A; when 98 is encountered in the basic dictionary, it is decoded as B, expanded as AB, and it can be seen that 256 corresponds to AB, decoded and added to the dictionary. Ultimately, the original character sequence abbababac can be losslessly restored, and the entire decoding process can be completed only by relying on the basic dictionary. The encoded sequence adheres to the mapping rule of four DNA bases. In this paper, the coding rule of A-00, T-01, C-10 and G-11 is adopted as the basis for the proposed methodology. The number of bits in the resulting DNA sequence will be reduced by approximately 50% in comparison to the original binary sequence. In the typical LZW code [19], two additional marker bits must be transmitted for the purpose of clearing the dictionary of the previous transmission and indicating the decoding position of the current one. In DNA encoding based on the LZW algorithm, an additional marker bit is necessary to indicate the number of the extended dictionary in a given transmission [20], representing the output of an N-bit binary code word stream.

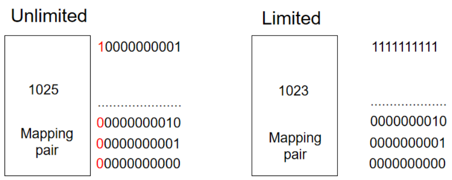

Fig. 3. Unlimited and limited dictionaries.

C. Improvements in LZW encoding

Once the LZW encoding has formed a mapping dictionary, the size of the dictionary will influence the size of the final transmission of the data [21]. This is because the LZW encoding represents the final transmission of a string of binary numbers, with the number dependent on the size of the dictionary to be binary encoded. To illustrate, if the dictionary contains 1025 mapping pairs, each mapping pair requires 11 bits to represent, resulting in a total of 11 bits needed for each transmission number. Conversely, if we limit the dictionary to 1024 mapping pairs, we can reduce the number of bits needed for each transmitted number to 10 bits, thereby saving more bits. Nevertheless, if the number of mapping pairs in the final dictionary amplification is 1023 and we limit the dictionary amplification to 512, it will frequently result in a greater number of required bits as Figure 3. Consequently, in order to accommodate different scenarios, we utilise both unlimited and limited dictionary amplification in the final comparison, employing the minimum number of bits for the subsequent compression. It is also noteworthy that LZW encoding employs dictionary-amplified mapping pairs for encoding. Consequently, even after the final formation of the number string to be transmitted, there will be repeated occurrences of the same number. In such cases, the use of arithmetic encoding can facilitate further compression of the data [22].

IV. DNA ENCODING BASED ON ARITHMETIC ENCODING ALGORITHM

A. Basic conception

The arithmetic encoding correlation algorithm is primarily employed in the field of data compression. The fundamental premise of this concept is that the lower the probability of a given symbol occurring in a given information set, the greater the amount of information that symbol carries, and vice versa. To illustrate, a symbol with probability $p(0 < p \leq 1)$ transmits

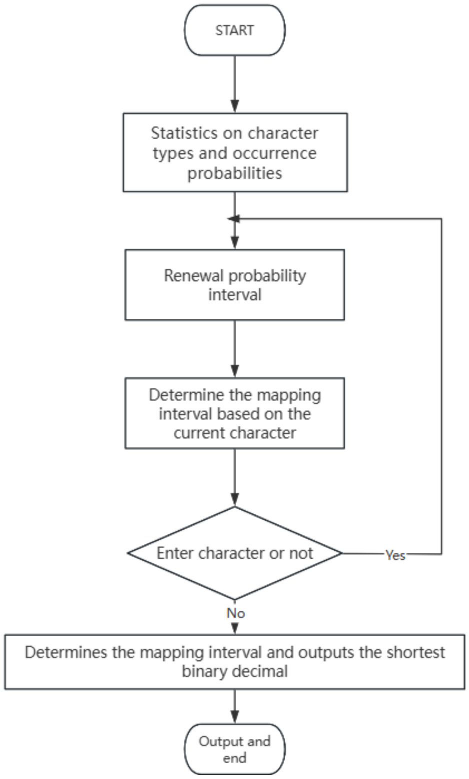
START
Statistics on character types and occurrence probabilities
Renewal probability interval
Determine the mapping interval based on the current character
Enter character or not
Yes
No
Determines the mapping interval and outputs the shortest binary decimal
Output and end

Fig. 4. The arithmetic encoding flowchart.

$\log_2(\frac{1}{p})$ bits of information. By selecting a symbol from a set of symbols with a probability distribution of $p(s)(0 \leq s < K)$, the average amount of information that can be transmitted is $H(s)$ bits, where $H(s)$ is Shannon entropy [23].

$$H(s) = \sum_{s=0}^{k-1} p(s) \log_2 \left(\frac{1}{p(s)} \right). \tag{1}$$

The output of a data compression algorithm typically exhibits a uniform or near-uniform frequency distribution for each symbol. This is due to the fact that the average information rate (in bits per symbol) is optimised when symbols are distributed uniformly.

B. arithmetic encoding code principle

The flow chart illustrating the arithmetic encoding process is presented in Figure 4. The following steps constitute the coding process: The initial step is to set the input symbol string, denoted by s, and to define the set of symbols in s, which we will denote by $S = \{a1, a2, ..., an\}$. An initial coding interval is then established, which is defined $[L0, L0 + R0)$.Step 2: The statistical symbol occurrence probability $p(an) = \{p1, p2, ...pn\}$. Step 3: The cumulative probability of updating the symbol is the lower limit of the probability interval, given by the following equation:

$$P(an) = \sum_{i=0}^{n} p(i). \tag{2}$$

Step 4: The size of the coding interval is updated:

$$R_{i+1} = R_i \times p(an). \tag{3}$$

Step 5: The size of the coding interval is updated as follows:

$$L_{i+1} = L_i + R_i \times P(an). \tag{4}$$

Step 6: The final mapping interval is obtained by repeating steps 3, 4, and 5 and continuously updating the mapping interval until the symbol coding is complete. The shortest binary decimal in the final mapping interval is then selected.

To illustrate, consider the sequence aabbc, where the probabilities of a, b, and c are 0.4, 0.4, and 0.2, respectively. In this case, we update the size of our encoding interval, i.e., a:[0, 0.4), b:[0.4, 0.8), and c:[0.8, 1.0), given that the first character is A. Consequently, we select the mapping interval [0, 0.4). Subsequently, the mapping interval is continuously subdivided in accordance with the order of characters, and the size of the encoding interval is updated in line with the string probability. Thus, a is mapped to the interval [0, 0.16), b to [0.16, 0.32), and c to [0.32, 0.4). Given that the second character is ”a,” the mapping interval selected is [0, 0.16) [24]. According to the recursive interval mapping, the final aabbc sequence mapping interval is [0.111008, 0.1152]. It is necessary to select a decimal within this mapping interval to represent the interval, as the computer encoding decimal will result in a repeated binary sequence. Consequently, the shortest binary sequence within this interval must be chosen as the final encoding. The arithmetic decoding process is analogous to the encoding process, with the exception of the necessity to ascertain the interval in which the representative decimal is situated, according to the probability of the string, in order to determine the input symbol sequence [25]. In arithmetic encoding, a marker bit is typically set to terminate the infinite decoding process, preventing an infinite loop. This will not be discussed further here. In DNA encoding combined with arithmetic encoding, an additional sequence must be transmitted. Consequently, the decoding process must refer to the probability of the string, which is masked after encoding. The additional sequence is used to represent the probability of occurrence of four bases for decoding. The following proof demonstrates the mathematical derivation of the compression effect of arithmetic encoding.

As is well known, the decimal l_N requires at least $-\log_2(l_N)$binary digits. The length of the subinterval resulting from this is given by l_N. The product of the probabilities of the states, $p(s_k)$, represents the first k coding frequency of the characters. Therefore, the average number of bits per character, where σ is a constant. must be calculated as follows:

$$B_s = \frac{-\log_2 l_N + \sigma}{N}. \tag{5}$$

This is because the character frequency and other related information must be stored in the code, which also occupies space and thus $\mathrm{B_S} \leq \frac{\sigma - \sum_{k=1}^{N} \log_2 p(s_k)}{N}$bits/symbol. Defining $E\{\cdot\}$ as the expected value operator, the expected number of bits per symbol is B as following:

$$
\begin{aligned}
\bar{B} &= \mathrm{E}\left\{B_S\right\} \\
&\leq \frac{\sigma - \sum_{k=1}^{N} \mathrm{E}\left\{\log_2 p\left(s_k\right)\right\}}{N} \\
&= \frac{\sigma - \sum_{k=1}^{N} \sum_{m=0}^{M-1} p(m) \log_2 p(m)}{N} \\
&\leq H(\Omega) + \frac{\sigma}{N}.
\end{aligned}
\tag{6}
$$

Since the average number of bits per symbol cannot be smaller than the entropy, we have $H(\Omega) \leq \bar{\mathrm{B}} \leq H(\Omega) + \frac{\sigma}{N}$, and it follows that

$$\lim_{N \to \infty} B = H(\Omega). \tag{7}$$

$H(\Omega)$ represents the limit of information entropy It can be seen that arithmetic encoding can reach the limit of approximate information entropy [26].

C. Improvements in arithmetic encoding

arithmetic encoding represents a specific form of entropy coding. Its capacity to compress data hinges on the utilisation of a broader decimal interval for high-frequency characters, while ensuring the preservation of the compressed characters' data order. The objective of arithmetic encoding is to identify the shortest decimal number within the final target interval as the final encoding result. As the range of the final target interval increases, the precision of the small numbers within the interval decreases, resulting in a shorter encoding length for the final encoded binary decimal number. The objective of arithmetic encoding is therefore to make the final target decimal space as large as possible.

Accordingly, in arithmetic encoding, the higher the frequency of characters, the larger the corresponding decimal interval; conversely, the lower the frequency of characters, the smaller the corresponding decimal interval. Once all the characters to be encoded have been considered, the target interval for the final iteration will be relatively larger, the decimal precision will be lower, and the binary code length of the decimal number that can be selected will be shorter. The more efficient the coding, the greater the overall efficiency [27].

Subsequently, the arithmetic code will incorporate redundancy to represent the probability of the string. In contrast, the DNA code comprises a mere four characters, namely A, T, C and G. It is thus necessary to add four probabilities to the final encoded DNA sequence in order to represent the content of the four bases. It should be noted that arithmetic encoding is a form of entropy coding. In the event that the probability of the original sequence, ATCG, has been entirely equalised, it is not possible to utilise a coding algorithm in order to achieve the compression function. Subsequently, the number of bases saved by compression and the size of the DNA sequence that was added must be calculated. In the event of an uneven probability distribution, a significant degree of compression is possible. As the probability distribution becomes more uniform, it becomes necessary to alter the methodology employed in the addition of DNA sequences [28]. Rather than utilising concrete numerical transmission, differential transmission must be employed. To illustrate, consider two digits, 00000101 and 00000100, which originally necessitated the transmission of eight bits. When transmitted, a difference transmission approach is employed, whereby the digits are represented by the difference between them, thus further reducing the number of bases required. In the final stage of decoding, it is only necessary to calculate the probability of each base based on the difference between the current value and the minimum value. Furthermore, the probability sum of the four bases, ATCG, is 1. As illustrated in Figure 6, among the 1000 DNA sequences randomly generated by the four bases, ATCG, each sequence contains 40000 bases. Given an 8-bit resolution, the minimum resolution of the difference is thus 2^{-8}. It was determined that the maximum probability difference among the four bases of ATCG is no greater than 4×2^{-8} [29] [30]. It can thus be concluded that differential transmission allows for a reduction in the final transmitted DNA sequence. The occurrence probability of each base can be determined by the difference between the occurrence probability of the four bases. It can be demonstrated that the sum of the probabilities is equal to 1.

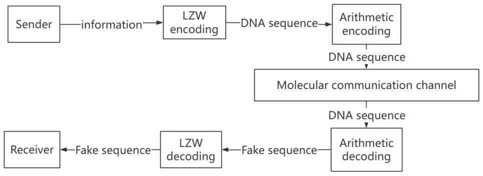

Fig. 5. Joint encoding flowchart.

As illustrated in Figure 7, the statistical analysis of the DNA sequence from 20,000 bases, with an incremental increase of 5,000 bases at regular intervals up to 80,000 bases, revealed that over 99% of the DNA sequences from 40,000 bases exhibited a maximum of four 2^{-8} values between the highest and lowest values of the four bases. The aforementioned analysis is designed to generate four bases with equal probability. However, it is possible to achieve a more pronounced compression effect by modifying the probability distribution. In practice, the degree of compression may vary depending on the specific information in question. However, the overall compression effect can be achieved [31] [32]. The mathematical derivation of arithmetic encoding has been demonstrated above, and the effect of approximate information entropy can be achieved. It can thus be concluded that the 40,000 base probability representation employing the difference representation is suitable for a greater number of base sequences, including those with 60,000 and 80,000 base lengths.

V. Joint encoding based on LZW and arithmetic encoding

A. Joint encoding process

As demonstrated in Figure 5, the original information is initially encoded by LZW into a dictionary sequence string, where the dictionary sequence string is represented by binary, with each digit representing the meaning mapped in the amplified dictionary. Consequently, the final sequence

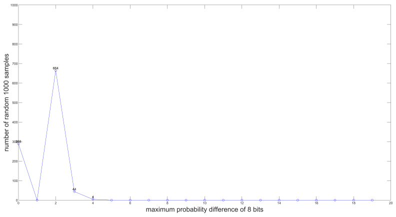

Fig. 6. Maximum probability difference of 4000 bases.

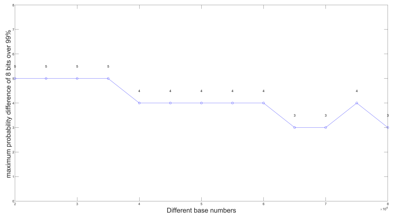

Fig. 7. 99% of the maximum probability difference of the DNA sequence.

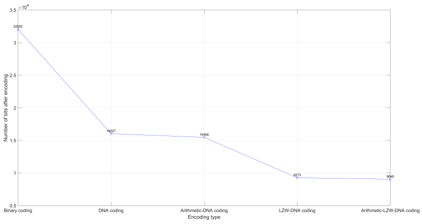

Fig. 8. Comparison of four encoding methods for a single document.

generated by LZW encoding is a long sequence of digits represented by binary. The sequence of numbers is encoded with DNA bases, yet there are still some repetitions of the same number. Following this, the DNA sequence encoded by LZW is subjected to arithmetic encoding to calculate the probability of four bases [33]. This is then compressed further to reduce the number of bits occupied by the original repeating number, and finally generates a decimal within the representative interval. This decimal is represented in binary to carry out the final information transmission. The DNA sequence received by the receiver is first converted into a binary sequence, then arithmetically decoded into a DNA sequence, and then the DNA sequence is decoded through the LZW amplified dictionary to complete the LZW decoding, and finally the original information is restored. Since both encoding methods are lossless compression, the final joint encoding is also lossless compression [32] [34].

B. Simulation results of lzw encoding combined with arithmetic encoding

As illustrated in Figure 8, the encoding of a randomly generated 4549 English character text document requires 32052 bits for conventional computer coding, 16,027 bases for ordinary DNA encoding, and 15,468 bases for arithmetical coding-based DNA encoding. The encoding of DNA based on lzw encoding requires 9,273 bases, while the encoding of DNA based on lzw encoding and arithmetic encoding necessitates 9,045 bases. It can be observed that the combined coding method, based on lzw encoding and arithmetic encoding, exhibits a superior compression effect compared to the individual coding methods. This research will compare the compression effect of four different compression methods for a sample.

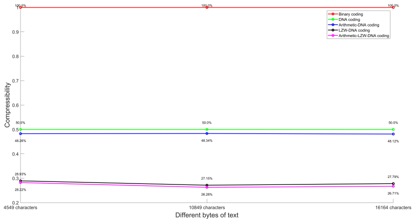

Fig. 9. Comparison of four encoding methods for multiple documents.

As illustrated in Figure 9, the randomly generated 4,549 English characters, 10,849 English characters, and 113,799 English characters were subjected to comparison through the application of ordinary DNA encoding, DNA encoding based on arithmetic encoding, DNA encoding based on LZW, and combined DNA encoding based on lzw encoding and arithmetic encoding [35] [36]. It can be observed that in DNA sequences of varying lengths, the joint DNA encoding based on lzw encoding and arithmetic encoding exhibits the most effective compression, with a potential reduction of up to 30%. The following study will compare the compression effect of four different compression methods for three samples of different sizes.

C. Comparison of computational complexity

This paper presents a detailed analysis of the time complexity of the LZW compression algorithm. In light of the length of the input data, denoted by n, and the maximum number of entries in the dictionary, represented by D, it is reasonable to assume that the average time complexity of each lookup and update operation of the dictionary entry is O(1). For each input character, the algorithm performs a lookup operation in the dictionary. In the optimal case, the time complexity per lookup is (O(1)), thus the total search time complexity is (O(n)) for (n) input characters. Furthermore, the time complexity of each update operation, that is, the addition of a new entry to the dictionary, is also (O(1)). In the least favourable scenario, an update operation may be required for each input character, resulting in a total time complexity of (O(n). In the most extreme cases, if the dictionary is of a considerable size and the new string is not present within it, each lookup may necessitate traversing the entire dictionary, thereby raising the time complexity of each lookup to (O(D)). Nevertheless, this is not a realistic scenario, given that the input data frequently exhibits repeating patterns and the dictionary size is typically constrained. In consideration of the information entropy of the input data, the time complexity of lzw encoding is observed to be greater than O(n) but less than O(n^2) in practical applications. In order to facilitate comparison with arithmetic encoding, the time complexity of

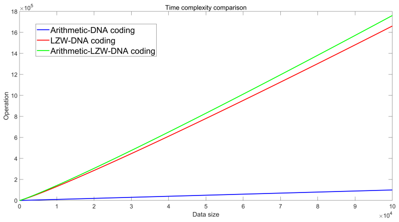

Fig. 10. Comparison of time complexity of three DNA encoding methods.

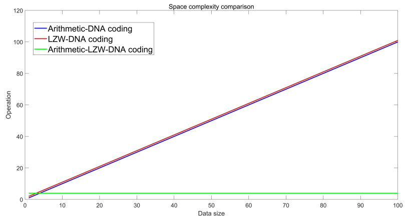

Fig. 11. Comparison of spatial complexity of three DNA encoding methods.

LZW in this paper is expressed as (O(n*$\log_2$ (n))), which does not signify that the actual time complexity of each operation is this value. Rather, it is intended to indicate that its complexity lies between (O(n)) and (O(n^2)).

As LZW does not carry dictionary information during transmission, but instead builds the dictionary during decoding, the spatial complexity of the output data is given by the following equation: O(n).

In the context of arithmetic encoding, the input data length is typically assumed to be (n), with the average time complexity of each probability update and interval calculation assumed to be (O(1)). In the context of DNA encoding as presented in this paper, the static model is adopted, whereby the type of fixed symbol and its probability are taken as given. Consequently, the time complexity is low. arithmetic encoding necessitates probabilistic updates and interval calculations for each symbol, resulting in a total time complexity of (O(n)).

In the static model, the arithmetic encoder requires a fixed amount of space to store the symbol probabilities, typically (O(M)), where (M) is the total number of different symbols. The arithmetic encoder is required to store the low and high values of the current encoding in the requisite space, which is of the order of (O(1)). In consequence, the spatial complexity of arithmetic encoding in the fixed model presented in this paper is (O(1)).

In conclusion, we undertake an analysis of the complexity of joint DNA encoding. The time complexity of the joint coding is (O(n*$\log_2$(n)+n)) when LZW and arithmetically encoded DNA encoding are combined. The space complexity is (O(n + 1)). The time and space complexity of the three DNA encoding modes are illustrated in Figures 10 and 11, respectively.

As illustrated in the figure, the time complexity of DNA joint coding is marginally higher than that of lzw encoding. However, the time complexity of both is considerably higher than that of arithmetic encoding. The spatial complexity of DNA joint coding is slightly higher than that of arithmetic encoding; however, the spatial complexity of both is considerably higher than that of LZW. In other words, the use of DNA joint coding entails a certain degree of compromise in terms of complexity, with the objective of achieving enhanced data compression.

VI. Conclusions and prospects

This paper presents a comparative analysis of four DNA encoding methods: ordinary DNA encoding, LZW-based DNA encoding, arithmetical DNA encoding, and combined DNA encoding based on LZW encoding and arithmetical encoding. The simulation results demonstrate that the combined DNA coding method based on LZW coding and arithmetic coding exhibits superior compression efficacy in comparison to the two primary DNA coding methods, namely LZW coding and arithmetic coding. However, it is important to note that the combined DNA coding methods based on LZW encoding and arithmetic encoding exhibit a higher time and space complexity compared to the two principal DNA coding techniques, as illustrated in Figures 10 and 11. The following two areas are identified as promising avenues for future research. 1. The subsequent stage is to enhance and utilise multi-layer arithmetic encoding, thereby enabling the nesting of multi-layer arithmetic encoding to achieve an optimal compression effect. Furthermore, the potential for further derivation of the compression range and compression capacity for the approximate equal probability of four bases should be investigated.

References

[1] L. Ceze, J. Nivala, and K. Strauss, "Molecular digital data storage using dna," *Nature Reviews Genetics*, vol. 20, no. 9, pp. 456–466, 2019.

[2] C. P. Hodgson, "A dna text code," *Biotechniques*, vol. 9, no. 3, p. 312, 1990.

[3] R. Shibata and H. Kaneko, "Shift-interleave coding for dna-based storage: Correction of ids errors and sequence losses," 2024.

[4] K. Katherine, B. Shounak, Z. Lisha, T. Derrick, B. Jessica, Z. W. Jim, S. Saurabh, and W. D. Fakhouri, "Sequence-to-expression approach to identify etiological non-coding dna variations in p53 and cmyc-driven diseases," *Human Molecular Genetics*, no. 19, p. 19, 2024.

[5] N. Rambaldi Migliore, D. Bigi, M. Milanesi, P. Zambonelli, R. Negrini, S. Morabito, A. Verini-Supplizi, L. Liotta, F. Chegdani, and S. Agha, "Mitochondrial dna control-region and coding-region data highlight geographically structured diversity and post-domestication population dynamics in worldwide donkeys," *PLoS ONE*, vol. 18, no. 8, 2024.

[6] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in dna," *Science*, vol. 337, no. 6102, p. 1628, 2012.

[7] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. Leproust, B. Sipos, and E. Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized dna," *Nature*, no. Feb.7 TN.7435, p. 494, 2013.

[8] W. Jing, N. Mengli, N. Wimms, and N. Guodong, "The application research of dna multiple-compression," in *Conference of Biomathematics*, 2008.

[9] C. I. Ragan, A. Matsuno-Yagi, Y. Hatefi, R. F. Doolittle, G. Attardi, A. Chomyn, P. Mariottini, and M. W. J. Cleeter, "Six unidentified reading frames of human mitochondrial dna encode components of the respiratory-chain nadh dehydrogenase," *Nature*, vol. 314, no. 6012, pp. 592–597, 1985.

[10] M. E. M. Campbell, J. W. Palfreyman, and C. M. Preston, "Identification of herpes simplex virus dna sequences which encode a trans-acting polypeptide responsible for stimulation of immediate early transcription," *Journal of Molecular Biology*, vol. 180, no. 1, pp. 1–19, 1984.

[11] J. Rothberg, W. Hinz, T. M. Rearick, J. Schultz, W. Mileski, M. Davey, J. H. Leamon, K. L. Johnson, M. J. Milgrew, M. Edwards, J. Hoon, J. F. Simons, D. F. Marran, J. W. Myers, J. F. Davidson, A. Branting, J. R. Nobile, B. P. Puc, D. Light, T. A. Clark, M. Huber, J. T. Branciforte, I. B. Stoner, S. E. Cawley, M. R. Lyons, Y. Fu, N. Homer, M. Sedova,

X. Miao, B. D. Reed, J. Sabina, E. Feierstein, M. A. Schorn, M. Alanjary, E. Dimalanta, D. C. Dressman, R. W. Kasinskas, T. D. Sokolsky, J. A. Fidanza, E. Namsaraev, K. J. McKernan, A. Williams, G. T. Roth, and J. M. Bustillo, "An integrated semiconductor device enabling non-optical genome sequencing," *Nature*, vol. 475, pp. 348–352, 2011.

[12] Y. Ogawa, H. Nishimoto, and K. Nakao, "Molecular cloning of the complementary dna and gene that encode mouse brain natriuretic peptide and generation of transgenic mice that overexpress the brain natriuretic peptide gene." *Journal of Clinical Investigation*, vol. 93, 1994.

[13] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, and K. Strauss, "A dna-based archival storage system," *IEEE Micro*, vol. PP, no. 4, pp. 637–649, 2016.

[14] M. Varol Arsoy, "Lzw-cie: a high-capacity linguistic steganography based on lzw char index encoding," *Neural Computing and Applications*, vol. 34, no. 21, pp. 19 117–19 145, 2022.

[15] T. P. Niedringhaus, D. Milanova, M. B. Kerby, M. P. Snyder, and A. E. Barron, "Landscape of next-generation sequencing technologies." *Analytical chemistry*, vol. 83 12, pp. 4327–4341, 2011.

[16] T. Kida, M. Takeda, A. Shinohara, M. Miyazaki, and S. Arikawa, "Multiple pattern matching in lzw compressed text," in *Data Compression Conference, 1998. DCC '98. Proceedings*, 1998.

[17] M. R. Nelson, "Lzw data compression," *Dr Dobbs Journal*, vol. 14, no. 10, pp. 29–36, 1989.

[18] J. H. E. Iii, "Hardware-based, lzw data compression co-processor," 2003.

[19] P. M. Nishad and R. M. Chezian, "A vital approach to compress the size of dna sequence using lzw (lempel-ziv-welch) with fixed length binary code and tree structure," *International Journal of Computer Applications*, vol. 43, no. 1, pp. 7–9, 2012.

[20] A. J. Pinho, A. J. R. Neves, C. A. C. Bastos, and P. J. S. G. Ferreira, "Dna coding using finite-context models and arithmetic coding," *IEEE*, 2009.

[21] L. Feng and G. Dong-Mei, "Dna algorithm of verifiable secret sharing," *IEEE*, 2009.

[22] F. Weindel, A. L. Gimpel, R. N. Grass, and R. Heckel, "Embracing errors is more efficient than avoiding them through constrained coding for dna data storage," *IEEE*, 2023.

[23] A. Singh and R. Gilhotra, "Data security using private key encryption system based on arithmetic coding," *International Journal of Network Security Its Applications*, vol. 3, no. 3, 2011.

[24] A. Cirillo, "Arithmetic code to encode a string of char." *Arithmetic*.

[25] D. Kim, D. Cho, and S. Lee, "Context-based adaptive binary arithmetic coding method and apparatus," 2006.

[26] B. Anglès, F. Pellarin, F. Tavares Ribeiro, and F. Demeslay, "Arithmetic of positive characteristic l-series values in tate algebras," *Compositio Mathematica*, vol. 152, no. 01, pp. 1–61, 2016.

[27] W. Chen, Z. Zhang, Z. Liu, and F. Xu, "Convolutional codes based index-free coding strategy forhigh-density dna storage," in *International Conference on Bio-Inspired Computing: Theories and Applications*, 2024.

[28] G. Pradhan, B. R. Dawadi, A. Chaulagain, A. L. Joshi, and P. G. Vaidya, "Chaos and dna coding technique for image cryptography," *Security Privacy*, vol. 7, no. 2, 2024.

[29] S. Zhou, Y. Wei, Y. Zhang, and L. Teng, "Novel chaotic image cryptosystem using dynamic dna coding," *International Journal of Modern Physics, C. Physics and Computers*, no. 6, p. 35, 2024.

[30] F. Q. Meng and G. Wu, "A color image encryption and decryption scheme based on extended dna coding and fractional-order 5d hyperchaotic system," *Expert Systems With Applications*, vol. 254, 2024.

[31] L. Li, "Image encryption algorithm based on hyperchaos and dna coding," *IET Image Processing (Wiley-Blackwell)*, vol. 18, no. 3, 2024.

[32] S. De Braganc, C. Aicart-Ramos, and A. J. P.-M. L.-B. M. P. O.-H. F. Arribas-Bosacoma, RaquelRivera-Calzada, "Aplf and long non-coding rna nihcole promote stable dna synapsis in non-homologous end joining," *cell reports*, vol. 42, no. 1, 2023.

[33] Y. Zhao, Q. Shi, and Q. Ding, "Cryptanalysis of an image encryption algorithm using dna coding and chaos," 2025.

[34] C. Rimaz, Y. Uslan, and A. Hareedy, "Protecting the future of information: Loco coding with error detection for dna data storage," *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, 2023.

[35] N. N. P. Cerize, "Coding, decoding and retrieving a message using dna: An experience from a brazilian center research on dna data storage," *Micromachines*, vol. 15, 2024.

[36] R. Sokolovskii, P. Agarwal, L. A. Croquevielle, Z. Zhou, and T. Heinis, "Coding over coupon collector channels for combinatorial motif-based dna storage," 2024.

VII. Biographies and Author Photos

Liu Qiang The following research interests are pursued: Internet of Things/Wireless Sensor Networks, Low Power Wide Area Network, Molecular Communication Research overview: The individual in question is responsible for the following projects: one key project subproject of the National Natural Science Foundation, two projects of the National Natural Science Foundation (surface), one national science and technology support plan and a number of horizontal projects. In addition, he is a leading researcher who has participated in a number of national scientific research projects, including National 863, major science and technology projects, and key projects of Natural Science Foundation.

Kun Yang Distinguished Professor at Nanjing University, a National-level selection of high-level talent, and a doctoral supervisor. His primary research encompasses the domains of wireless networks and communication, the integration of communication, computing and perception, and the concept of A! The advent of technology has facilitated networking and communication. He has published extensively in international core journals and major conferences, with over 400 papers, and holds more than 30 multinational patents. He has undertaken several European Union and British National Science Fund projects, led three Chinese NSFC key projects, and has won numerous awards, including provincial and ministerial science and technology first prizes.

Zhongyang Cheng University of Electronic Science and Technology of China Master, molecular communication research enthusiast.