Mathematical Biosciences and Engineering

MBE, 20(6): 11482–11501.
DOI: 10.3934/mbe.2023509
Received: 12 January 2023
Revised: 02 April 2023
Accepted: 15 April 2023
Published: 28 April 2023

http://www.aimspress.com/journal/MBE

Research article

Adaptive and blind audio watermarking algorithm based on dither modulation and butterfly optimization algorithm

Qiuling Wu[1], Dandan Huang[1], Jiangchun Wei[2] and Wenhui Chen[1]

[1] School of Cyber Security, Jinling Institute of Technology, Nanjing, Jiangsu 211169, China
[2] Jiangsu Liyuan Power Engineering Co., Ltd., Nanjing, Jiangsu 211100, China

* **Correspondence:** Email: Wuqiuling@jit.edu.cn; Tel: +8613951032810.

Abstract: How to improve the robustness to resist attacks and how to adaptively match the key parameters of the watermarking algorithm with the performance requirements to achieve the best performance in different applications are two hot issues in the research of audio watermarking algorithms. An adaptive and blind audio watermarking algorithm based on dither modulation and butterfly optimization algorithm (BOA) is proposed. Based on the convolution operation, a stable feature is designed to carry the watermark, which will improve the robustness by means of the stability of this feature to prevent the watermark loss. Blind extraction will be achieved only by comparing the feature value and the quantized value without the original audio. The BOA is used to optimize the key parameters of the algorithm which can be matched with the performance requirements by coding the population and constructing the fitness function. Experimental results confirm that this proposed algorithm can adaptively search for the optimal key parameters that match the performance requirements. Compared with other related algorithms in recent years, it exhibits strong robustness against various signal processing attacks and synchronization attacks.

Keywords: audio watermarking; butterfly optimization algorithm; robustness; dither modulation; optimal parameter

1. Introduction

1.1. Related work

In the past two decades, digital watermarking technology has played an increasingly important role in the field of information security. By embedding specific watermarks into digital works such as images [1,2], audio [3,4] or video [5], it can achieve the purposes of copyright tracking, integrity protection, content authentication, medical security and so on.

With the wide application of audio on the Internet, people are paying more and more attention to the copyright protection of audio, which attracts many scholars to research audio watermarking technology. Salah et al. [6] presented an audio watermarking algorithm by using a discrete Fourier transform, which has high transparency but poor robustness. Bhat et al. [7] proposed an audio watermarking algorithm based on a discrete cosine transform (DCT). The algorithm used singular value decomposition to achieve blind watermark extraction, and it had strong robustness to some signal processing operations, but its payload capacity was low. Hu and Hsu [8] proposed a sufficient audio watermarking algorithm in the discrete wavelet transform domain by applying the spectrum shaping technology into vector modulation. Authors claimed the payload capacity reached 818.26 bits per second (bps). Hwang et al. [9] designed an audio watermarking algorithm with singular value decomposition and quantization index modulation in order to reach blind extraction. This algorithm applied singular value decomposition on the stereo signal to achieve strong robustness against amplitude scaling, MP3 compression and resampling, but its transparency was low. Merrad [10] developed a robust audio watermarking algorithm based on the strong correlation between two continuous samples in a hybrid domain that consisted of a discrete wavelet transform and DCT. With the increasingly widespread application of audio watermarking algorithms, people have put forward higher and higher requirements for the performance of the algorithm. How to resist malicious attacks on audio has always been a challenging issue in the research of audio watermarking algorithms. Yamni et al. [11] proposed a blind and robust audio watermarking algorithm by combining the discrete Tchebichef moment transform, the chaotic system of the mixed linear–nonlinear coupled map lattices and a discrete wavelet transform. This algorithm achieved good results in terms of robustness and payload capacity, but no experimental results against synchronous attacks were found. A robust and blind audio watermarking scheme based on the dual tree complex wavelet transform and the fractional Charlier moment transform was proposed in paper [12]. It also obtained high imperceptibility and robustness against most common audio processing operations. Synchronous attacks may seriously destroy the structure of audio data in the embedding process, which will make the extracting algorithm unable to accurately search the location of a watermark in the carried audio [13,14]. Therefore, how to resist synchronization attacks is the bottleneck in improving the robustness of algorithms [15]. A robust audio watermarking algorithm for overcoming synchronous attacks was proposed in paper [16]. This algorithm took the audio frame sequence number as a global feature to carry the watermark, and it could resist partial synchronization attacks. Hu et al. [17] explored the distributive feature of the approximate coefficients to develop an audio watermarking algorithm with a self-synchronization mechanism in a discrete wavelet transform. This algorithm reconstructed and reshaped the wavelet coefficients for tracking the locations of the watermark. It had strong robustness to attacks, but its transparency only was a low level. An audio watermarking algorithm for resisting during de-synchronization and recapturing attacks was developed in a previous paper [18].

In this algorithm, the logarithmic mean feature was constructed to design the embedding and extracting algorithm according to the residuals of the two sets of features. He et al. [19] proposed a novel audio watermarking by embedding watermarks into the frequency domain power spectrum feature to resist recapturing attacks. From the analysis of the above literatures, it can be seen that embedding a watermark on some stable features can effectively improve the robustness of the algorithm. The main reason is that these features will not change much due to the stable performance when the audio is attacked, so the embedded watermark will not be easily lost.

The performance of an audio watermarking algorithm is not only related to the embedding and extracting rules, but it is also related to the setting of algorithm parameters, so how to choose the parameters in the application is particularly important. When different applications put forward new requirements for payload capacity, transparency and robustness, the watermarking algorithm usually cannot accurately adjust its parameters to meet these performance requirements. Nowadays, parameters of most audio watermarking algorithms are chosen by the users according to their experience in application, or are adjusted by the designers according to the performance achieved by the algorithm in experiments. These methods lack an effective parameter adjustment mechanism and cannot effectively stimulate the performance of the algorithm. Robustness, transparency, and payload capacity are three important indicators of audio watermarking algorithms, and these indicators are determined by multiple algorithm parameters. Therefore, how to set these parameters so that all three indicators can meet performance requirements is a multi-parameter and multi-objective combinatorial optimization problem.

In order to solve the above problems, some scholars have used meta-heuristic algorithms to optimize the parameters of watermarking algorithms. Meta-heuristic algorithms are self-organized and decentralized algorithms used for solving complex problems using team intelligence [20]. Wu et al. [21] proposed an audio watermarking algorithm based on a genetic algorithm for parameter optimization. This algorithm had high transparency and a large payload capacity, but it was not robust against attacks due to the lack of a synchronization mechanism. Kaur et al. [22] also proposed an audio watermarking method with a genetic algorithm which was used to find the optimal number of audio samples needed to conceal the watermark. Some scholars have attempted to apply sine and cosine algorithms to the design of image watermarking algorithms [23,24]. With the deepening of the research on watermarking technology, more and more watermarking algorithms based on meta-heuristic algorithms were explored. They all play a positive role in improving the performance of watermarking algorithms, but there are still many problems to be solved in the practical application.

1.2. Contributions

Based on the above analysis, weak robustness and a multi-parameter optimization problem are still urgent issues in the current research and application of audio watermarking algorithms. In our research, an adaptive and blind audio watermarking algorithm based on dither modulation and a BOA is proposed. The main contributions are as follows.

1) We propose a robust and blind audio watermarking algorithm based on convolution and dither modulation. A stable feature is designed using convolution operations, and dither modulation is performed on this feature to design embedding and extracting algorithms. The stability of this feature improves the robustness of the algorithm to prevent watermark loss. The algorithm has the

capability of blind extraction, and the watermark can be extracted only by comparing the feature value and quantized value, which will be very convenient for the algorithm to be applied in practice.

2) We propose a method for setting the parameters to solve the multi-parameter and multi-objective problem of audio watermarking algorithms, which can adaptively adjust the algorithm parameters with the performance requirements. The BOA is used to optimize the key parameters of the algorithm which can be adaptively matched for the performance requirements by coding the population and constructing the fitness function. In the case of meeting the performance requirements of transparency and payload capacity, the fitness function of the BOA is constructed by the total bit error ratio (BER), which is a comprehensive evaluation of the watermark extracted from the carried audio after it has been subjected to multiple malicious attacks. Through global search and local search, the population is continuously optimized to search for the global optimal butterflies, so as to improve the robustness under specific performance requirements.

2. Audio watermarking algorithm based on dither modulation

In this section, the embedding and extracting principle of the proposed algorithm will be described in detail. A feature which is closely linked to the change of the intermediate frequency coefficient is designed by convolving the low frequency coefficient and the intermediate frequency coefficient. When embedding the watermark, the feature will be quantized by dither modulation, and the direction of dither modulation is controlled by the value of a binary watermark. When extracting the watermark, the feature will be calculated and uniformly quantized, and the binary watermark will be obtained by comparing the feature value and the quantized value.

2.1. Principle of the embedding algorithm

Based on the energy concentration characteristics of the DCT and the bidirectional quantization characteristics of dither modulation [25,26], a feature is explored to carry the watermark in the DCT domain, and then the binary watermark can be embedded into the audio by modifying the feature with dither modulation.

The original audio with N sample-points can be supposed as $x(n)$ $(1 \le n \le N)$. The binary watermark W that will be embedded into the audio can be expressed as the formula (1).

$$W = \{w_{\text{in}}(l,m), 1 \le l \le L_1, 1 \le m \le L_2\} \tag{1}$$

where $w_{\text{in}}(l,m) \in \{0,1\}$. Divide $x(n)$ into L_1 audio fragments, and use the synchronization mechanism proposed in a previous paper [27] to select the voiced frame with the highest energy $x_l(n_0)$ $(1 \le n_0 \le N_1)$ with N_1 sample-points from each audio fragment to carry the watermark. $x_l(n_0)$ will be processed by the DCT using the formulas (2) and (3).

$$X_l(0) = \sqrt{\frac{1}{N_1}} \sum_{n_0=0}^{N_1} x_l(n_0), \ k = 0 \tag{2}$$

$$X_l(k) = \sqrt{\frac{2}{N_1}} \sum_{n_0=0}^{N_1} x_l(n_0) cos \frac{(2n_0+1)k\pi}{2N_1}, \ k \neq 0 \tag{3}$$

where $X_l(0)$ is the component with a frequency of 0 Hz, and $X_l(k)$ is the harmonic component with f_k Hz. f_k is the frequency of each harmonic component, calculated by using the formula (4),

and f_s is the sampling-rate.

$$f_k = \frac{kf_s}{2N_1} \qquad (k \neq 0) \tag{4}$$

Assumed that $X_{l0}(k)$ and $X_{lm}(k)$ respectively represent the low frequency-band and intermediate frequency-band containing N_2 spectral lines from $X_l(k)$. r_0 and r_1 are the positions of the first spectral line of $X_{l0}(k)$ and $X_{l1}(k)$ in $X_l(k)$. The watermark is embedded into audio fragments by modifying $X_{lm}(k)$, and the carried frequency-band $X'_{lm}(k)$ which carries the L_2 bit watermark can be represented by the formula (5), where ρ_m is a constant, indicating the change proportion of the intermediate frequency coefficients $X_{lm}(k)$.

$$X'_{lm}(k) = \rho_m X_{lm}(k) \tag{5}$$

The feature CF_{lm} shown in the formula (6) can be used to represent the change of the intermediate frequency-band relative to the low frequency-band.

$$CF_{lm} = \frac{X_{l0}(k) \otimes X_{lm}(k) \big/ 2N_2-1}{|X_{l0}(k)|^2 \big/ N_2} \tag{6}$$

where $\otimes$ represents the convolution operation on $X_{l0}(k)$ and $X_{lm}(k)$. The numerator of this formula refers to the average value of the convolution result, and the denominator means the average value of the square of the magnitude of $X_{l0}(k)$. Quantize CF_{lm} at an equal interval δ, and the quantized value CFQ_{lm} can be shown in the formula (7).

$$CFQ_{lm} = round(\frac{CF_{lm}}{\delta}) \tag{7}$$

$round()$ means that the data point in the brackets is equal to its nearest integer. Modulate $w_{\text{in}}(l,m)$ into a bipolar bitstream $w(l,m)$ according to the formula (8).

$$w(l,m) = \begin{cases} 1 & w_{\text{in}}(l,m) = 1 \\ -1 & w_{\text{in}}(l,m) = 0 \end{cases} \tag{8}$$

The embedding rule for embedding L_2 bits watermark into $x_l(n_0)$ can be expressed as the formula (9).

$$CF'_{lm} = \delta CFQ_{lm} + \frac{\delta w(l,m)}{4} \tag{9}$$

According to the formulas (5) and (6), the carried feature CF'_{lm} can also be showed in the formula (10).

$$CF'_{lm} = \frac{X_{l0}(k) \otimes X'_{lm}(k) \big/ 2N_2-1}{|X_{l0}(k)|^2 \big/ N_2} = \rho_m CF_{lm} \tag{10}$$

It can be seen that CF_{lm} changes in equal proportion similar to the change of $X_{lm}(k)$, so $X_{lm}(k)$ can be changed by modifying CF_{lm} in order to embed L_2 bits watermark into the audio fragment $x_l(n)$. The change proportion ρ_m can be expressed as the formula (11).

$$\rho_m = \frac{CF'_{lm}}{CF_{lm}} = \frac{X'_{lm}(k)}{X_{lm}(k)} = \frac{\delta CFQ_{lm} + \frac{\delta w'(l,m)}{4}}{\frac{N_2}{2N_2-1}\frac{X_{l0}(k) \otimes X_{lm}(k)}{|X_{l0}(k)|^2}} \tag{11}$$

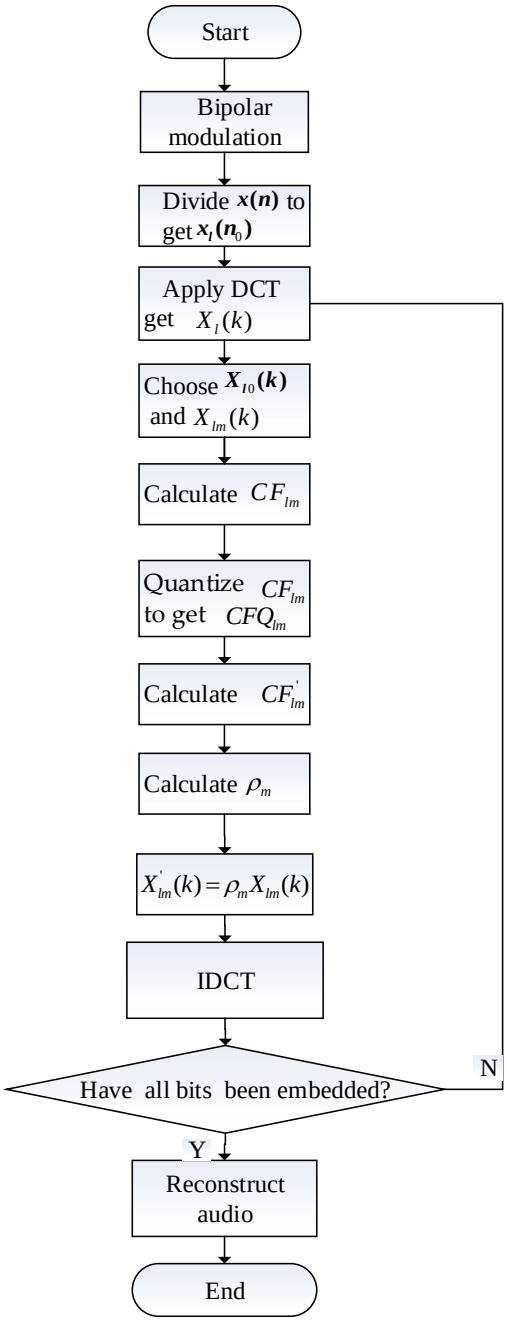


Figure 1. Flow diagram of the embedding algorithm.

Therefore, watermarks can be concealed into an audio fragment by modifying the intermediate frequency-band coefficients $X_{lm}(k)$, and the change proportion ρ_m can be calculated according to the formula (11).

Figure 1 shows the flow diagram of the embedding algorithm, and the embedding steps can be described as follows in detail.

Step 1: Convert the watermark into a binary-string $w_{\text{in}}(l,m)$ and modulate it to obtain a bipolar bit-stream $w(l,m)$.

Step 2: Divide $x(n)$ into L_1 fragments to obtain $x_l(n_0)$.

Step 3: Apply a DCT to $x_l(n_0)$ to obtain the DCT coefficients $X_l(k)$.

Step 4: Select $X_{l0}(k)$ and $X_{lm}(k)$ from $X_l(k)$.

Step 5: Calculate CF_{lm} according to the formulas (6).

Step 6: Quantize CF_{lm} to get CFQ_{lm} according to the formulas (7).

Step 7: Embed L_2 bits watermark into $x_l(n_0)$, and get the carried feature CF'_{lm} according to the formulas (9).

Step 8: Calculate ρ_m according to the formulas (11).

Step 9: Calculate the carried frequency-band $X'_{lm}(k)$ according to the formulas (5), and Substitute $X_{lm}(k)$ to obtain the carried spectrum $X'_l(k)$.

Step10: Obtain the carried audio fragment $x'_l(n_0)$ by applying an inverse DCT to $X'_l(k)$.

Step 11: Repeat step 3 to step 10 until all bits of the watermark are concealed into the audio.

Step 12: Reconstruct all $x'_l(n_0)$ to obtain the carried audio $x'(n)$.

2.2. Principle of the extracting algorithm

According to the embedding principle described in Section 2.1, the binary watermark can be concealed into the audio by applying dither modulation to the feature. In the extracting process, the feature will also be quantized at the same interval as the embedding process, and then the binary watermark can be extracted without the original audio by comparing the feature value with the quantized value.

Divide the carried audio $x'(n)$ to get L_1 audio fragments $x'_l(n_0)$ which will be applied in the DCT to obtain $X'_l(k)$. Calculate CF'_{lm} with the formula (6), and quantize CF'_{lm} at δ to obtain CFQ'_{lm} with the formula (7). The quantized value CF''_{lm} can be calculated with the formula (12).

$$CF''_{lm} = \delta CFQ'_{lm} \tag{12}$$

The extracting rule for obtaining L_2 bits watermark $w_{\text{out}}(l,m)$ from $x'_l(n_0)$ can be expressed as the formula (13).

$$w_{\text{out}}(l,m) = \begin{cases} 1 & CF''_{lm} \le CF'_{lm} \\ 0 & CF''_{lm} > CF'_{lm} \end{cases} \tag{13}$$

Figure 2 shows the flow diagram of the extracting algorithm, and the extracting steps can be described as follows in detail.

Step 1: Divide the carried audio $x'(n)$ into L_1 audio fragments to obtain $x'_l(n_0)$.

Step 2: Apply a DCT to $x'_l(n_0)$ to obtain the DCT coefficients $X'_l(k)$.

Step 3: Select $X'_{l0}(k)$ and $X'_{lm}(k)$ from $X'_l(k)$.

Step 4: Calculate CF'_{lm} with the formulas (6).

Step 5: Quantize $CF'_{lm}$to get CFQ'_{lm} with the formulas (7).

Step 6: Calculate CF''_{lm} with the formula (12).

Step 7: Extract the L_2 bits watermark from $x'_l(n_0)$ with the formula (13).

Step 8: Repeat step 2 to step 7 until all bits of the watermark are extracted.

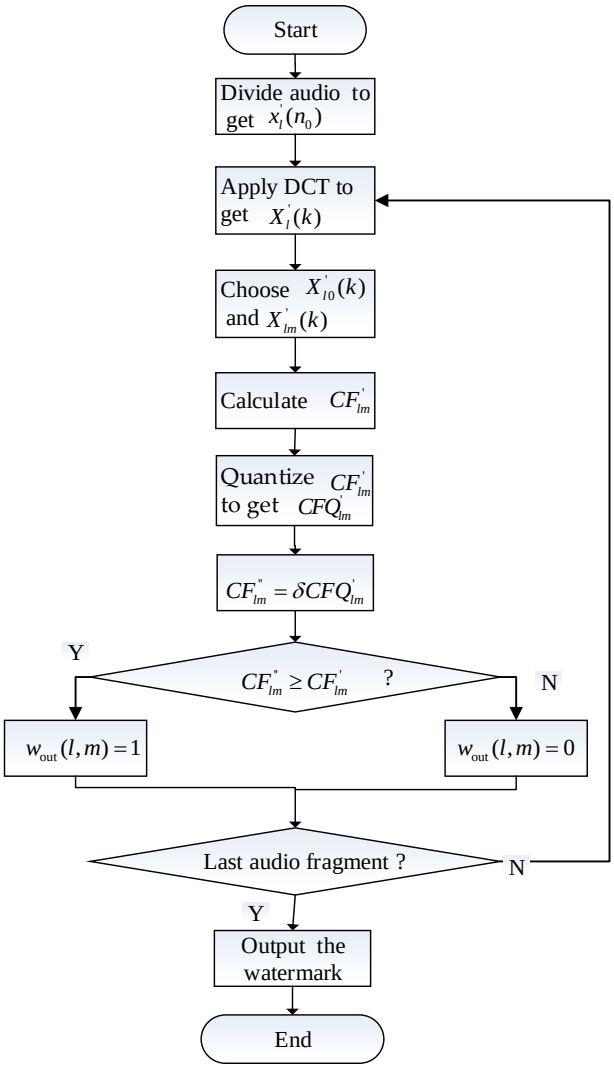


Figure 2. Flow diagram of the extracting algorithm.

3. Parameters optimization based on BOA

In order to stimulate the performance in different applications, the parameters of the algorithm must be set adaptively to meet the different performance requirements. The BOA is a new nature-inspired optimization algorithm developed in 2019. It can be used to solve the global optimization problem by imitating the food-searching and mating behavior of butterflies, and it has the advantages of fast convergence and strong searching ability [28]. There are four important key parameters (r_0, r_1, N_2, δ) in the proposed algorithm, which have a significant impact on the overall performance of the algorithm.

It is assumed that the initial population POP has M butterflies, and the position of each butterfly consists of four key parameters, as shown in the formula (14).

$$\text{POP} = \begin{bmatrix} B_1 \\ \vdots \\ B_i \\ \vdots \\ B_M \end{bmatrix} = \begin{bmatrix} r_{01} & r_{11} & N_{21} & \delta_1 \\ \vdots & \vdots & \vdots & \vdots \\ r_{0i} & r_{1i} & N_{2i} & \delta_i \\ \vdots & \vdots & \vdots & \vdots \\ r_{0M} & r_{1M} & N_{2M} & \delta_M \end{bmatrix} \tag{14}$$

where $B_i = (r_{0i}\ r_{1i}\ N_{2i}\ \delta_i)\ (1 \le i \le M)$ represents the i^{th} butterfly, and $r_{0i}, r_{1i}, N_{2i}, \delta_i$ mean that they take random values on their respective ranges [Min(r_0),Max(r_0)], [Min(r_1),Max(r_1)], [Min(N_2),Max(N_2)] and [Min(δ),Max(δ)]. Min( ) and Max( ) represent the minimum and maximum values of the variables in brackets respectively. Each butterfly emits a certain intensity of fragrance f_i, which can be expressed in the formula (15).

$$f_i = cI_i^{\propto} \tag{15}$$

where c is the perceptual form, $\propto$ is the power index, and I is the stimulus factor. Normally, c and $\propto$ are constants, and I_i is related to the fitness function of this butterfly. Fitness function Fit_i comprehensively considers three indicators, including payload capability, transparency and robustness under various attacks in the proposed algorithm, as shown in the formula (16).

$$Fit_i = \frac{1}{I_i} = \sum_{j=1}^{A} a_j\, BER_j,\ 1 \le j \le A \tag{16}$$

The boundary conditions of the above formula are $SNR > SNR_0$ and $Cap > Cap_0$, where SNR is the signal-to-noise ratio (SNR), as expressed as the formula (17). Cap is the payload capacity of this algorithm. SNR_0 and Cap_0 respectively indicate the thresholds of transparency and payload capacity that need to be provided. A indicates the total number of attacks, and BER_j means the BER of the extracted watermark after applying the j^{th} attack on the carried audio, as expressed in the formula (18). a_j indicates the importance of the j^{th} attack in total attack types, and $\sum_{j=1}^{A} a_j = 1$.

$$SNR = 10\lg\left(\frac{\sum_{n=1}^{N} x^2(n)}{\sum_{n=1}^{N}(x'(n)-x(n))^2}\right) \tag{17}$$

$$BER = \frac{\sum_{l=1}^{L_1}\sum_{m=1}^{L_2} \mathrm{w}_{\text{in}}(l,m) \otimes w_{\text{out}}(l,m)}{L_1 L_2} \times 100\% \tag{18}$$

A butterfly can conduct a random local search near its self-position, or they can move towards the butterfly with the highest fragrance value and conduct a global search. Assume that there is a switch probability p. When there is a need to update the position of the butterfly B_i^t in the t^{th} iteration, a random number r is generated. If $r \le p$, then the butterfly performs a local search, and its new position B_i^{t+1} will be updated according to the formula (19).

$$B_i^{t+1} = B_i^t + (r^2 \times B_{i_0}^t - B_{i_1}^t) \times f_i\ ,\ 1 \le i_0, i_1 \le M \tag{19}$$

where $B_{i_0}^t$ and $B_{i_1}^t$ represent the positions of the $i_0{}^{\text{th}}$ butterfly and the $i_1{}^{\text{th}}$ butterfly in the t^{th}

iteration. Else, the butterfly will perform a global search, and its new position B_i^{t+1} will be updated according to the formula (20).

$$B_i^{t+1} = B_i^t + (r^2 \times g^* - B_i^t) \times f_i \tag{20}$$

where g^* represents the position of the best butterfly with the highest fragrance value in the t^{th} iteration. The optimization process can be described as follows in detail.

Step 1: Initialize the population and parameters. Set the perceptual form c, the power index $\propto$, the switch probability p, the population size M, the maximum number of iterations *MaxG*, SNR_0 and Cap_0, and then produce an initial population POP_0.

Step 2: Put four parameters from each butterfly into the embedding algorithm in order to get the carried audio, and then calculate *SNR* with the formula (17).

Step 3: Select all qualified butterflies with performance that meets the boundary conditions, and run the embedding algorithm to get the carried audio.

Step 4: Perform attack. Apply malicious attacks to the carried audio respectively, and then carry out the extracting algorithm to calculate BER_j with the formula (18).

Step 5: Calculate Fit_i with the formula (16) to obtain the best butterfly in the current population.

Step 6: Calculate f_i of each butterfly with the formula (15).

Step 7: Generate r and compare it with p. If $r \leq p$, update the position according to the formula (19); else, update the position with the formula (20).

Step 8: Repeat Step 2 to Step 7 until the maximum number of iterations reaches *MaxG* or the same global best butterfly occurs in five consecutive iterations.

4. Performance evaluation

This section will evaluate the performance of the proposed algorithm in terms of payload capacity, transparency, robustness and complexity. Transparency is measured using the SNR and the object difference grade (ODG) which is the key output of the perceptual evaluation of audio quality. In addition, the transparency can be evaluated by observing the audio changes before and after embedding the watermark from the waveform and spectrogram. Robustness can be evaluated with the BER, normalized correlation (NC) which can be expressed as the formula (21) and structural similarity (SSIM) proposed by the laboratory for image and video engineering of the university of Texas at Austin to reflect the similarity between the extracted watermark and the original watermark. If the extracted watermark is very similar to the original watermark, NC and SSIM all will be very close to 1, which indicates that the robustness is strong. Complexity can be measured by the elapsed time consumed by the embedding algorithm and the extracting algorithm.

$$NC = \frac{\sum_{l=1}^{L_1}\sum_{m=1}^{L_2} \mathrm{w}_{\text{in}}(l,m)\mathrm{w}_{\text{out}}(l,m)}{\sqrt{\sum_{l=1}^{L_1}\sum_{m=1}^{L_2} \mathrm{w}_{\text{in}}(l,m)^2 \sum_{l=1}^{L_1}\sum_{m=1}^{L_2} \mathrm{w}_{\text{out}}(l,m)^2}} \tag{21}$$

Here, we will list the experimental parameters and conditions in our test: 1) Algorithm parameters: $M = 50$, $c = 0.1$, $\propto = 0.1$, $p = 0.8$, MaxG = 500 , $N_1 = 4096$, $a_j = 0.1$, $(j = 1,2,\ldots 10)$, Min(r_0) = 1, Max(r_0) = 100, Min(r_1)=100,Max(r_1) = 1000, Min(N_2) = 1, Max(N_2) = 20, Min(δ) = 0, Max(δ) = 2; 2) Twenty 64-second audio signals which come from the TIMIT standard database including popular and symphony music were tested, and they were formatted by WAV,

sampled at 44,100 Hz and quantized at 16 bits; 3)There were two groups of experiments according to the different watermarks. The first watermark was a binary image shown as Figure 3(a) with the size of 43 × 64, $Cap_0 = 40\,bps$, and $SNR_0 = 27\,dB$; The second watermark is shown as Figure 3(b) with the size of 86 × 64, $Cap_0 = 80\,bps$ and $SNR_0 = 26\,dB$; 4) Computer system: 64-bit Microsoft Windows 10; 5) Programming language: Matlab 2016R.

(a) (b)

Figure 3. Two watermarks: (a) The first image with 43 × 64; (b) The second image with 86 × 64.

4.1. Capacity and transparency

Payload capacity refers to the bit number of the watermark that can be contained in audio per second. In our study, the payload capacity is related to the size of the watermark and the duration T of the audio, so it can be calculated by the formula (22). The duration T of the audio was about 64 seconds, and the size of the first watermark was 43 × 64 bits, so the pay-load capacity in the first group was 43 bps. Similarly, the payload capacity in the second group was 86 bps.

$$Cap = \frac{L_1 L_2}{T} \tag{22}$$

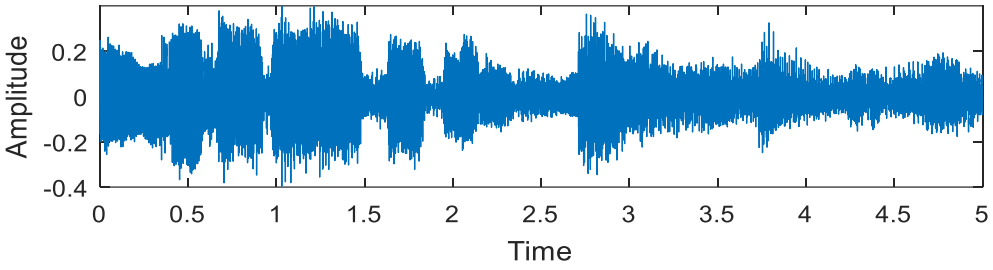
The average experimental results for the SNR (dB), ODG, BER (%), NC, SSIM and Cap (bps) are listed in Table 1. "Yes" in Table 1 indicates the watermarking algorithm with the BOA. "No" indicates the watermarking algorithm without the BOA, and its key parameters (r_0, r_1, N_2, δ) were set as $(20{,}600{,}5{,}0.4)$.

Table 1. Average results under no attack.

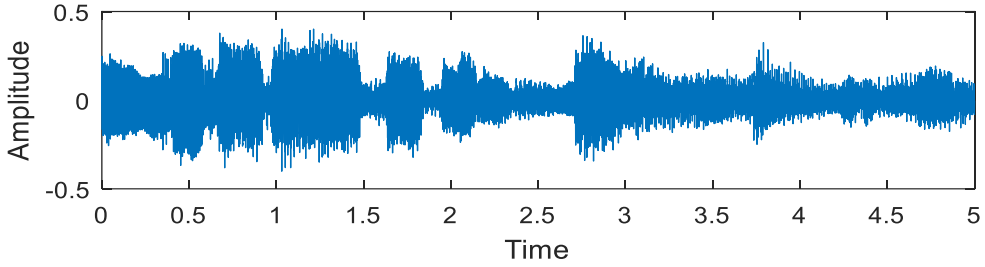
Item	1st group		2nd group		Paper [9]	Paper [13]	Paper [17]	Paper [21]
	Yes	No	Yes	No				
SNR	27	24	26	23	25	31	19	26
ODG	−0.75	−0.85	−1.02	−0.98	−0.81	−0.08	−3.24	−1.18
BER	0.00	0.12	0.05	0.16	0.06	0.00	0.00	0.00
NC	1	0.98	0.99	0.98	0.99	0	1	0
SSIM	1	1	1	1	1	1	1	1
Cap	43	43	86	86	43	43	86	86

According to the standards of the international federation of the phonographic industry (IFPI) for audio watermarking algorithms, the SNR should be more than 20 dB and payload capacity should be greater than 20 bps. It can be seen from the data of two groups in Table 1 that the average SNR values with the BOA were 27 dB and 26 dB, while the average SNR values without the BOA were 24

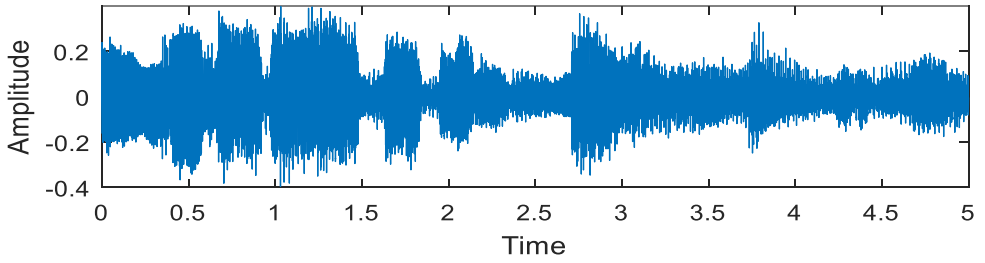
dB and 23 dB, which indicates that the proposed algorithm meets the standards of the IFPI in terms of transparency and payload capacity, and the proposed algorithm achieved good transparency under the payload capacities of 43 bps and 86 bps. Compared with other algorithms with the same payload capacity, the transparency of this proposed algorithm was the same as that of the algorithms in a previous study [21], far superior to the algorithm in [9] and [17], but inferior to the algorithm in [13].



(a)

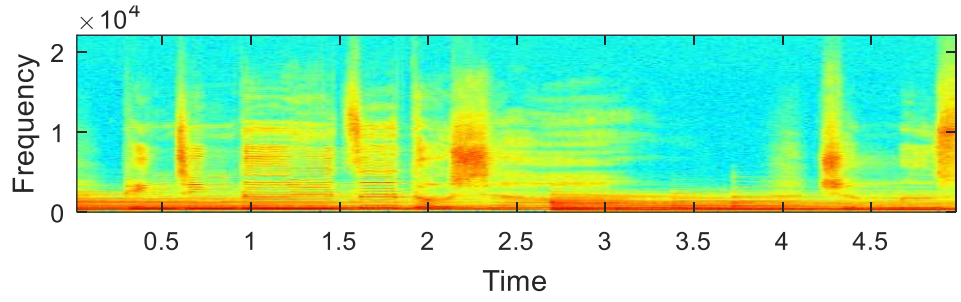


(b)

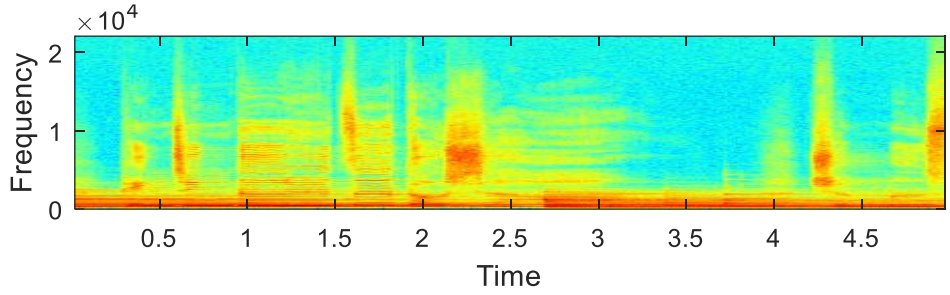


(c)

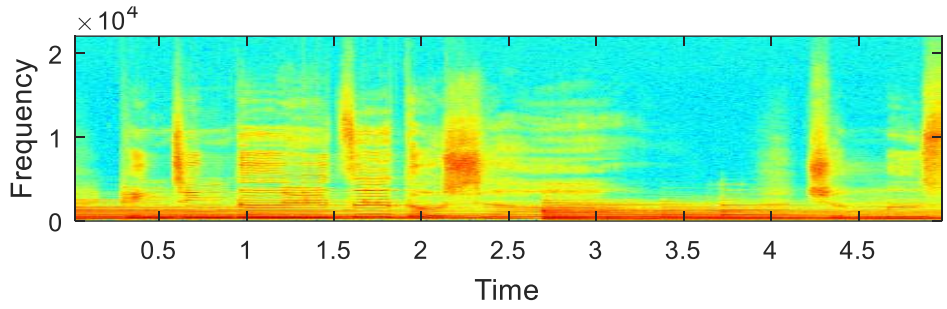
Figure 4. Waveform comparison. (a) Original audio. (b) Carried audio with the first watermark. (c) Carried audio with the second watermark.



(a)



(b)



(c)

Figure 5. Spectrogram comparison. (a) original audio. (b) Carried audio with the first watermark. (c) Carried audio with the second watermark.

Figure 4 shows the waveform comparison of the original audio and the carried audio respectively. In order to display the details of the audio more clearly, only a 5-second audio clip is shown here. Their spectrograms under different payload capacities are shown in Figure 5. It can be seen that the waveforms and spectrograms of the original audio and the carried audio with different watermarks all have no visible changes, which also indicates that the transparency of this algorithm is high. The main reasons are as follows. Firstly, the watermark is only embedded in the intermediate frequency coefficients, and the location of the watermark can be adjusted by optimizing the key parameters. Second, the algorithm only modifies the DCT coefficient by dither modulation, so the audio data are less damaged. The frequency range with watermarks can be calculated according to

the formula (4).

4.2. Robustness

Table 1 also shows the robustness results under no attack. It can be seen that all algorithms can perfectly extract watermarks from the carried audio without any attacks. The robustness against malicious attacks will be discussed in this section. Two watermarks with different sizes are embedded into the audio respectively, and then different attacks are performed on the carried audio. In the case of meeting the transparency requirements, the BOA is used to adaptively select the algorithm parameters that minimize the fitness function according to the formula (16), so that the algorithm can achieve the strongest robustness against these attacks. Attack types can be shown as follows.

A. Noise addition: Add Gaussian noise with 30 dB into the carried audio.

B. Echo addition: Add an echo with a delay of 50 ms into the carried audio.

C. MP3 compression: Apply MPEG-1 layer 3 compression at a bit rate of 128 kbps.

D. Low-pass filtering: Apply a low-pass filter with a cutoff frequency of 12 kHz.

E. Re-quantization: Re-quantize the carried audio with 8 bits per sample, and back into 16 bits per sample.

F. Re-sampling: Re-sample the carried audio with 22.05 kHz and back into 44.1 kHz.

G. Amplitude scaling: Scale the amplitude at a factor of 0.8.

H. Time scale modification (TSM): Apply TSM with 1% on the carried audio.

I. Jittering: Randomly delete one audio sample from every 1000 samples in the carried audio.

J. Random cropping: Randomly cut out 100 samples from the carried audio.

The above attacks were applied to the carried audio one by one. The average results of the BER (%) are listed in Table 2. The extracted watermarks corresponding to the global best butterfly, NC and SSIM are shown in Figures 6 and 7.

Table 2. Robustness comparison with other algorithms.

Item	**1st group**		**2nd group**		**Paper [9]**	**Paper [13]**	**Paper [17]**	**Paper [21]**
	Yes	No	Yes	No				
A	0.00	0.32	0.78	1.02	11.96	0.49	0.02	1.25
B	0.08	0.39	0.97	1.54	18.64	0.18	0.34	0.16
C	0.53	0.86	0.82	1.41	19.97	0.24	0.01	0.18
D	0.00	0.19	0.76	1.12	0.28	1.27	0.00	0.09
E	0.00	0.62	0.72	1.21	0.76	1.89	0.01	0.25
F	0.55	0.98	1.03	1.57	0.89	0.00	0.01	0.12
G	0.00	0.16	0.46	0.88	0.33	0.05	0.01	0.08
H	10.42	13.03	12.21	16.44	48.25	38.45	5.71	42.89
I	1.64	2.69	2.53	3.87	25.19	28.42	1.78	32.59
J	0.57	1.24	1.57	2.11	22.82	29.17	0.87	46.24

From the experimental results of two groups in Table 2, it can be seen that the proposed algorithm with the BOA shows strong robustness under different payload capacities. After the payload capacity was doubled, the experimental results in the second group became larger than those

in the first group, indicating that the robustness decreases as the payload capacity increases. In addition, the robustness of the algorithm with the BOA was stronger than the algorithm without the BOA, indicating that BOA is effective in improving the robustness by optimizing multiple key parameters.

When the carried audio was subjected to noise addition at 30 dB, echo addition at 50ms, MP3 compression at 128 kbps, low-pass filtering at 12 kHz, re-quantization, re-sampling, amplitude scaling and random cropping, the proposed algorithm with the BOA showed particularly excellent robustness, which can be reflected by the following three points: 1) All BER values are very close to 0 in Table 2. 2) The extracted watermarks are very clear in Figures 6 and 7. 3) All NC and SSIM values are very close to 1 in Figures 6 and 7.

The proposed algorithm with the BOA showed good robustness when a jittering attack was applied to the carried audio. The extracted watermark was very similar to the original watermark, as shown in Figure 6(i) and Figure 7(i). The BER values were 1.64% and 2.53% under two payload capacities, and NC values were higher than 0.96.

Under TSM attack, the BER values in the two groups of experiments reached 10.42% and 12.21% respectively, indicating that the robustness of the proposed algorithm against TSM is weak. However, these results still meet IFPI, and the main information can be distinguished from the extracted images, as seen in the Figure 6(h) and Figure 7(h).

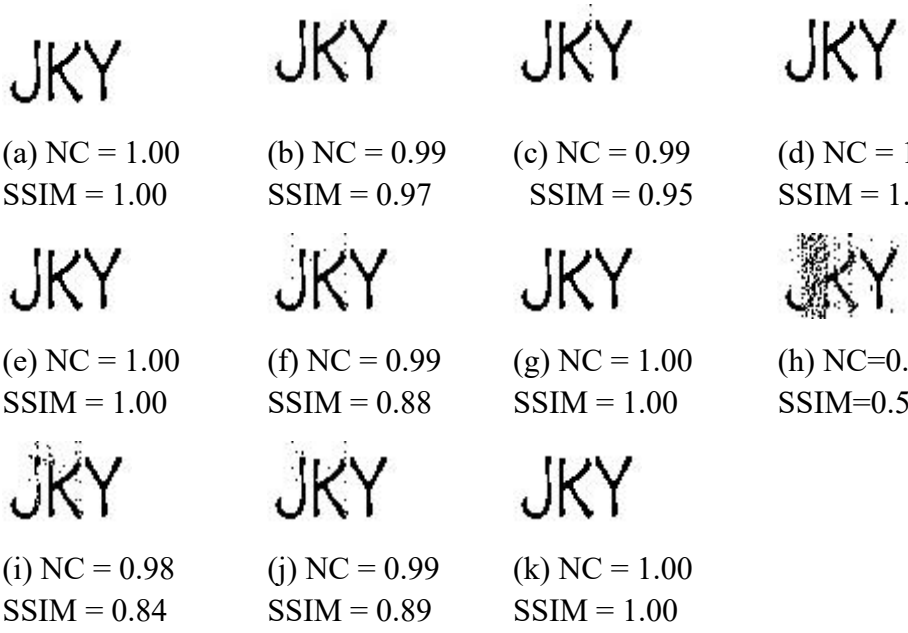


Figure 6. The first extracted watermarks. (a) Noise addition (30 dB). (b) Echo addition (50 s). (c) MP3 compression (128 kbps). (d) Low-pass filtering (12 kHz). (e) Re-quantization. (f) Re-sampling. (g) Amplitude scaling (0.8). (h) TSM (1%). (i) Jittering (1000). (j) Random cropping (100). (k) no attack.

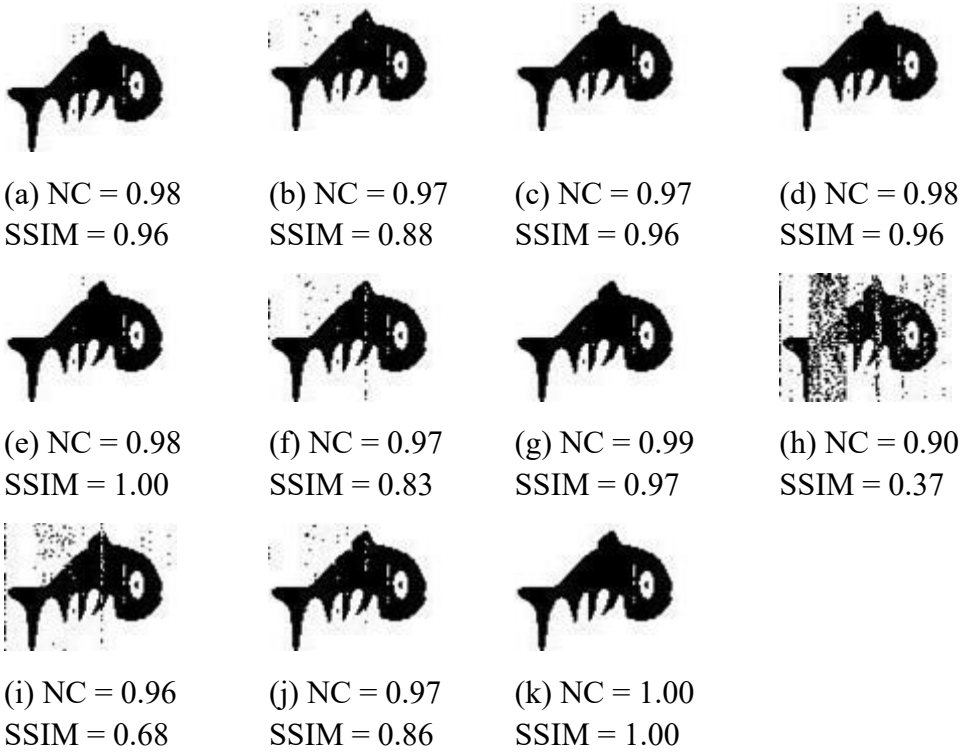


Figure 7. The second extracted watermarks. (a) Noise addition (30 dB). (b) Echo addition (50 s). (c) MP3 compression (128 kbps). (d) Low-pass filtering (12 kHz). (e) Re-quantization. (f) Re-sampling. (g) Amplitude scaling (0.8). (h) TSM (1%). (i) Jittering (1000). (j) Random cropping (100). (k) no attack.

From the comprehensive results of transparency, hiding capacity, and robustness, the proposed algorithm with the BOA has stronger robustness than those in [9] and [13] under the payload capacity with 43 bps when resisting most attacks. When the payload capacity reaches 86 bps, this proposed algorithm has higher transparency, but worse robustness against attacks than that in [17]. This is mainly because the SNR of the algorithm in [17] is only 19 dB, which does not meet the IFPI standard, so it traded for strong robustness by reducing transparency. The proposed algorithm with the BOA has the same payload capacity and transparency as that in [21], and it is more robust when resisting noise addition, amplitude scaling, TSM, jittering and random cropping. It can be viewed from the above analysis that the robustness and transparency of this algorithm are excellent under different payload capacities. This is mainly because of the following two reasons: 1) The feature designed by using convolution is relatively stable, which makes the watermark embedded in it also very stable and will not easily lost when the carried audio is attacked. 2) With the minimum total BER as the optimization goal, the BOA can adaptively search the most suitable key parameters according to the performance requirements, which makes the proposed algorithm have the strong robustness in resisting various attacks.

4.3. Complexity

Complexity is an important indicator for evaluating the performance of a watermarking algorithm. The lower the complexity, the less time it takes for the algorithm to embed and extract the

watermark. Table 3 lists the average runtime (seconds) of the proposed algorithm and four related algorithms in embedding and extracting process.

Table 3. Complexity comparison with other algorithms (seconds).

Item	1st group		2nd group		Paper [9]	Paper [13]	Paper [17]	Paper [21]
	Yes	No	Yes	No				
Embed	856	1.80	1147	1.91	2.95	3.42	2.89	1526
Extract	0.92	0.92	1.08	1.08	1.79	2.59	1.84	1.89

According to the experimental results, when embedding watermark, the running time of the algorithm with the BOA is much higher than that of the algorithm without the BOA, mainly because the BOA needs to run the embedding program and extraction program repeatedly when optimizing the parameters of the watermark algorithm. The extracting time of the two groups is basically the same, which indicates that the algorithm with the BOA does not increase the complexity in the extracting process. Compared with papers [9,13,17], the proposed algorithm without the BOA has lower complexity due to its shorter running time. The algorithm proposed in [21] costs 1526 seconds to embed the watermark, which is much higher than that of our proposed algorithm with the BOA. The main reason is that the BOA is simpler than the genetic algorithm used in [21] and can quickly jump out of the local optimal solution.

Based on the experimental results of the above four indicators, the following points can be summarized: 1) The algorithm has stronger robustness by embedding watermarks on the stable feature. 2) The algorithm can adaptively search for the optimal parameters to meet the requirements of transparency and payload capacity in practical applications, thereby improving the overall performance of the algorithm. 3) Under the same payload capacity and transparency, the algorithm with the BOA has stronger robustness than the algorithm without the BOA, but the BOA increases the complexity in the embedding process.

5. Conclusions

An adaptive audio watermarking algorithm based on dither modulation and the BOA has been proposed to improve the poor robustness and optimize the key parameters of audio watermarking. Based on convolutional operation and dither modulation, a watermark will be embedded into the stable feature to prevent watermark loss. When extracting the watermark, a binary watermark can be extracted by comparing the feature value and the quantized value without the original audio, which is very convenient for practical application. In order to match the key parameters of the algorithm with the performance requirements in different applications, the BOA is used to optimize the key parameters of the algorithm. Under the condition of meeting the two indicators of payload capacity and transparency, a fitness function composed of the BER under various attacks is constructed. In the process of continuous iteration, the key parameters of the algorithm are adaptively optimized by searching for the position of the butterfly with the largest fragrance.

Experimental results demonstrate that the proposed algorithm with the BOA has good transparency, strong robustness, and the ability to search for the optimal parameters. Our research provides a solution to the multi-parameter and multi-objective optimization problem formed between the parameters and performance of watermarking algorithms. The population coding method and the

construction scheme for the fitness function can also provide an example for other meta heuristic algorithms to be applied for the parameter optimization of watermarking algorithms. Compared with other related watermarking algorithms, although the proposed algorithm has achieved better results in terms of robustness and overall performance improvement, it still has problems, such as high complexity and weak robustness in resisting TSM. In future research, we will further explore the methods to overcome TSM, reduce the complexity, and focus on using more intelligent optimization algorithms to improve the overall performance of the watermark algorithm.

Acknowledgments

This work was funded by the High-Level Talent Scientific Research Foundation of Jinling Institute of Technology, China (Grant No. jit-b-201918), Industry-university-research Cooperation Project of Jiangsu Province in 2022 (BY2022654), National Natural Science Foundation of China, China (Grant No. 11601202), Collaborative Education Project of the Ministry of Education (Grant no.202102089002) and Jiangsu Provincial Vice President of Science and Technology Project in 2022 (Grant no. FZ20220114).

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. M. Yamni, H. Karmouni, H. Qjidaa, Robust zero-watermarking scheme based on novel quaternion radial fractional Charlier moments, *Multimedia Tools Appl.*, **80** (2021), 21679–21708. https://doi.org/10.1007/s11042-021-10717-2
2. S. Z. Xiao, X. Y. Zuo, Z. G. Zhang, F. F. Li, Large-capacity reversible image watermarking based on improved DE, *Math. Biosci. Eng.*, **19** (2022), 1108–1127. https://doi.org/10.3934/mbe.2022051
3. D. Singh, S. K. Singh, DWT-SVD and DCT based robust and blind watermarking scheme for copyright protection, *Multimedia Tools Appl.*, **76** (2017), 13001–13024. https://doi.org/10.1007/s11042-016-3706-6
4. Y. Hong, J. Kim, Autocorrelation modulation-based audio blind watermarking robust against high efficiency advanced audio coding, *Appl. Sci.*, **9** (2019), 1–17. https://doi.org/10.3390/app9142780
5. S. Q. Zhang, X. Y. Guo, X. H. Xu, A video watermark algorithm based on tensor decomposition, *Math. Biosci. Eng.*, **16** (2019), 3435–3449. https://doi.org/10.3390/app9142780
6. E. Salah, A. Khaldi, K. Redouane, A Fourier transform based audio watermarking algorithm, *Appl. Acoust.*, **172** (2021), 1–7. https://doi.org/10.1016/j.apacoust.2020.107652
7. K. V. Bhat, A. K. Das, J. H. Lee, Design of a blind quantization-based audio watermarking scheme using singular value decomposition, *Concurr. Comput. Pract. Exper.*, **32** (2020), 1–11. https://doi.org/10.1002/cpe.5851

8. H. T. Hu, L. Y. Hsu, Incorporating spectral shaping filtering into DWT based vector modulation to improve blind audio watermarking, *Wireless Pers. Commun.*, **94** (2017), 221–240. https://doi.org/10.1007/s11277-016-3178-z
9. M. J. Hwang, J. S. Lee, M. S. Lee, SVD based adaptive QIM watermarking on stereo audio signals, *IEEE Trans. Multimedia*, **20** (2017), 45–54. https://doi.org/10.1109/TMM.2017.2721642
10. A. Merrad, S. Saadi, Blind speech watermarking using hybrid scheme based on DWT/DCT and sub-sampling, *Multimedia Tools Appl.*, **77** (2018), 27589–27615. https://doi.org/10.1007/s11042-018-5939-z
11. M. Yamni, H. Karmouni, M. Sayyouri, H. Qjidaa, Efficient watermarking algorithm for digital audio/speech signal, *Digital Signal Process.*, **120** (2022), 103251. https://doi.org/10.1016/j.dsp.2021.103251
12. M. Yamni, H. Karmouni, H. Qjidaa, Robust audio watermarking scheme based on fractional Charlier moment transform and dual tree complex wavelet transform, *Expert Syst. Appl.*, **203** (2022), 117325. https://doi.org/10.1016/j.eswa.2022.117325
13. M. M. Yao, Q. Z. Du, LWT domain adaptive audio watermarking algorithm based on norm ratio, *Commun. Technol.*, **54** (2021), 1478–1485.
14. A. A. Attari, A. A. B. Shirazi, Robust audio watermarking algorithm based on DWT using Fibonacci numbers, *Multimedia Tools Appl.*, **77** (2018), 25607–25627. https://doi.org/10.1007/s11042-018-5809-8
15. M. Mahdi, S. Saeed, B. Behrang, High-capacity, transparent and robust audio watermarking based on synergy between DCT transform and LU decomposition using genetic algorithm, *Analog Integr. Circuits Signal Process.*, **100** (2019), 513–525. https://doi.org/10.1007/s10470-019-01464-4
16. W. Z. Jiang, X. H. Huang, Y. J. Quan, Audio watermarking algorithm against synchronization attacks using global characteristics and adaptive frame division, *Signal Process.*, **162** (2019), 153–160. https://doi.org/10.1016/j.sigpro.2019.04.017
17. H. T. Hu, J. R. Chang, S. J. Lin, Synchronous blind audio watermarking via shape configuration of sorted LWT coefficient magnitudes, *Signal Process.*, **147** (2018), 190–202. https://doi.org/10.1016/j.sigpro.2018.02.001
18. Z. H. Liu, Y. K. Huang, J. W. Huang, Patchwork-based audio watermarking robust against de-synchronization and recapturing attacks, *IEEE Trans. Inf. Foren. Sec.*, **14** (2019), 1171–1180. https://doi.org/10.1109/TIFS.2018.2871748
19. J. J. He, Z. H. Liu, K. Lin, Q. Qian, A novel audio watermarking algorithm robust against recapturing attacks, *Multimedia Tools Appl.*, **80** (2022).
20. V. Sharma, A. K. Tripathi, A systematic review of meta-heuristic algorithms in IoT based application, *Array*, **14** (2022), 100164. https://doi.org/10.1016/j.array.2022.100164
21. Q. L. Wu, A. Y. Qu, D. D. Huang, Robust and blind audio watermarking scheme based on genetic algorithm in dual transform domain, *Math. Problems Eng.*, **8** (2021), 1–14. https://doi.org/10.18280/mmep.080101
22. A. Kaur, M. K. Dutta, An optimized high payload audio watermarking algorithm based on LU-factorization, *Multimedia Syst.*, **24** (2018), 341–353. https://doi.org/10.1007/s00530-017-0545-x

23. A. Daoui, H. Karmouni, O. Ogri, M. Sayyouri, H. Qjidaa, Robust image encryption and zero-watermarking scheme using SCA and modified logistic map, *Expert Syst. Appl.*, **190** (2022), 116193. https://doi.org/10.1016/j.eswa.2021.116193
24. A. Daoui, H. Karmouni, M. Sayyouri, New robust method for image copyright protection using histogram features and sine cosine algorithm, *Expert Syst. Appl.*, **177** (2021), 114978. https://doi.org/10.1016/j.eswa.2021.114978
25. T. K. Tewari, V. Saxena, J. P. Gupta, A digital audio watermarking algorithm using selective mid band DCT coefficients and energy threshold, *Int. J. Audio Technol.*, **17** (2014), 365–371. https://doi.org/10.1007/s10772-014-9234-8
26. Z. H. Liu, D. Luo, J. W. Huang, Tamper recovery algorithm for digital speech signal based on DWT and DCT, *Multimedia Tools Appl.*, **76** (2017), 12481–12504. https://doi.org/10.1007/s11042-016-3664-z
27. Q. L. Wu, R. Y. Ding, J. C. Wei, Audio watermarking algorithm with a synchronization mechanism based on spectrum distribution, *Secur. Commun. Networks*, **9** (2022), 1–13. https://doi.org/10.1155/2022/2617107
28. Y. Q. Fan, J. P. Shao, G. T. Sun, X. Shao, A self-adaption butterfly optimization algorithm for numerical optimization problems, *IEEE Access*, **8** (2020), 88026–88041. https://doi.org/10.1109/ACCESS.2020.2993148

AIMS Press