

Received April 1, 2019, accepted May 3, 2019, date of publication May 22, 2019, date of current version June 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2918261

JamSys: Coverage Optimization of a Microphone Jamming System Based on Ultrasounds

HAO SHEN, WEIMING ZHANG¹, HAN FANG¹, ZEHUA MA, AND NENGHAI YU

School of Information Science and Technology, University of Science and Technology of China, The Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, Hefei 230026, China

Corresponding author: Weiming Zhang (zhangwm@ustc.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant U1636201 and Grant 61572452, and in part by the Anhui Initiative in Quantum Information Technologies under Grant AHY150400.

ABSTRACT With the development of digital devices, the recording process has become increasingly easier to conduct. However, the portability of the recording devices has also made recording difficult to monitor. If private conversations are illegally recorded, it will cause serious secret-leakage events. Therefore, it is imperative to prevent unauthorized recordings. Recent works have demonstrated that the nonlinearity effect of microphones can be leveraged to interfere with microphone recording using ultrasounds. However, an ultrasonic array has a limited jamming area. The design of an anti-recording system composed of multiple ultrasonic arrays remains to be addressed. In this paper, a jamming system, JamSys, is presented to prevent eavesdropping in a given region. We propose a new scheme composed of the angle coverage model and the modified harmony search algorithm (MHSA) to optimize the deployment of ultrasonic arrays, which achieves the maximum jamming area with the given number of arrays. In the simulation and experiments, three different optimization algorithms, the MHSA, the genetic algorithm (GA), and the regular coverage algorithm (RCA) are compared. The MHSA is demonstrated to provide the best results.

INDEX TERMS Privacy protection, anti-recording, microphones, ultrasounds, optimization algorithms.

I. INTRODUCTION

Electronic eavesdropping is a means of stealing important information. Recording is a typical type of eavesdropping. Due to the small size of recording devices, attackers can easily record secret information in confidential meetings and private conversations. Therefore, protecting confidential and private activities from illegal recording is very important for personal communication secrets, commercial trade, and even national security.

The current protection methods can be categorized into two broad categories: detection and jamming techniques. The detection techniques rely on metal detectors or X-ray scanners to detect electronic devices. However, such techniques are unreasonable if electronic devices are not allowed in some situations, such as cinemas. Some electronic devices, such as laptops, cannot be blocked from being carried because they may be used in meetings. The above drawbacks limit the use of detection techniques.

The associate editor coordinating the review of this manuscript and approving it for publication was Cristina Rottondi.

Many solutions based on jamming techniques have been proposed in both industry and academia. In industry, some companies have developed audio jammers based on white noise [1], but these jammers will produce audible noise, which is intolerable in conversations and live concerts. In academia, Kune *et al.* [2] used electromagnetic interference (EMI) signal injection to attack analog sensors. Then, Wu *et al.* [3] prevented unauthorized audio recording by utilizing the principle of EMI injection. However, this approach can cause pacemakers to stop operating and threaten people's lives [2]. In addition, EMI may seriously influence the operation of other electronic devices.

To solve the above problems, the more recent work [4] leveraged ultrasounds and white noise to jam recordings. Roy *et al.* [5] proposed an ultrasonic jamming technique called BackDoor. In BackDoor, by utilizing the nonlinearity of microphones, noise can be injected into microphones over ultrasounds. This means that the designed ultrasonic noise signals can be recorded by microphones, but people cannot hear them. As long as the amplitude of noise is greater than that of people's voice, recording devices can only record noise, and people's voice can hardly be recognized.

TABLE 1. Comparison of ultrasonic-based methods and other anti-recording methods.

Method	Inaudible	Coverage in all directions	Harmfulness
White noise	×	✓	Medium
EMI	✓	✓	Severe
Ultrasounds	✓	×	Low

In Table 1, we compare the differences of three jamming methods. As shown, ultrasonic-based methods have the characteristics of inaudibility and low harmfulness when compared to other methods. However, the existing ultrasonic-based method [5] only uses two ultrasonic arrays, which can only cover limited directions and areas. In this paper, to jam all directions, we propose a jamming system named JamSys, which is composed of multiple arrays.

There are two key problems behind this ultrasonic jamming system:

- 1) **How can the maximum jamming area with a given number of ultrasonic arrays be achieved?** This question can be known as a coverage problem. Coverage is one of the crucial issues for the quality of service in wireless sensor networks (WSN) [6], which refers to the ability to detect events occurring in the monitored entity [7]. The coverage problem is to determine a minimum number of sensors to achieve maximum coverage [8]. This inspires us to solve the coverage problem using the model in WSN. A simple coverage model assumes that a sensor is able to cover a point if the distance between them is less than a radius in WSN [9]. However, traditional coverage models are not applicable to the ultrasonic coverage problem because of the directionality of ultrasounds. Therefore, a new coverage model called the angle coverage model is proposed to calculate the coverage.
- 2) **What are the optimal locations to place these arrays?** The angle coverage model can also be considered as an optimization problem. The optimized target is the coverage. The solution vector is the optimal location of all ultrasonic arrays. In the present paper, to solve the angle coverage model, a modified harmony search algorithm is proposed.

The simulation and experimental results verify the effectiveness of JamSys.

In summary, our core contributions are listed as follows.

- 1) We pioneer the design of a jamming system JamSys, which prevents unauthorized recording in a given region by taking advantage of multiple ultrasonic arrays.
- 2) To solve the ultrasonic coverage problem, we present a new angle coverage model and the modified harmony search algorithm.
- 3) Multiple recording devices have been tested in experiments and demonstrate the effectiveness of JamSys.

The subsequent sections are organized as follows. In Section II, we briefly review inaudible voice command attacks, the principle of ultrasonic jamming, and methods to optimize coverage in WSN. Then, we present and analyze

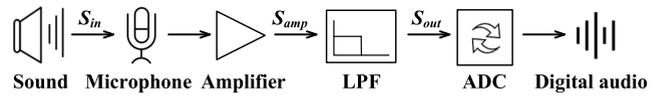


FIGURE 1. The processing flow of a microphone system.

the angle coverage model in Section III. Section IV details the modified harmony search algorithm to solve the angle coverage model. The simulation and experimental results are discussed in Section V. Finally, we draw the conclusion and discuss future work in Section VI.

II. RELATED WORK

A. INAUDIBLE VOICE ATTACKS

Inaudible voice command attacks have been widely researched [10]–[12]. Song and Mittal [11] showed that an adversary can exploit the microphone’s nonlinearity and play well-designed inaudible ultrasounds. Then, the microphone will record normal voice commands, which are hidden in the ultrasounds, and thus, the adversary can control the victim device inconspicuously. BackDoor [5] is the first system to jam recording by exploiting nonlinearities in microphone hardware.

We will introduce the principle of inaudible voice attacks in this subsection. A microphone system is composed of a microphone, an amplifier, a low-pass filter (LPF), and an analog-to-digital converter (ADC), as shown in Fig. 1. The cutoff frequency of the LPF is generally 24 kHz; thus, all signals higher than 24 kHz will be filtered by the LPF.

To understand how inaudible voice command attacks work, we model the nonlinearity of the microphone system as follows. The input ultrasonic sound can simply be denoted by S_{in} , and the output signals of the amplifier and the LPF can be denoted as S_{amp} and S_{out} , respectively, as illustrated in Fig. 1. Due to the nonlinear phenomena of the amplifier in the microphone system, S_{amp} can be modeled as follows:

$$S_{amp} = \sum_{i=1}^{\infty} (A_i S_{in}^i) = A_1 S_{in} + A_2 S_{in}^2 + A_3 S_{in}^3 + \dots \quad (1)$$

where the third and higher order terms can be ignored because they are weak.

Hence, Eq. (1) can be simply expressed as:

$$S_{amp} = A_1 S_{in} + A_2 S_{in}^2 \quad (2)$$

Next, we show the inaudible voice command attacks. For instance, we assume that S_{in} is composed of two inaudible ultrasonic sounds, i.e., $S_{in} = \cos(2\pi f_1 t) + \cos(2\pi f_2 t)$, where $f_1 = 45 \text{ kHz}$ and $f_2 = 40 \text{ kHz}$.

After passing through the amplifier, S_{amp} can be expressed as follows:

$$\begin{aligned} S_{amp} &= A_1 S_{in} + A_2 S_{in}^2 \\ &= A_1 [\cos(2\pi f_1 t) + \cos(2\pi f_2 t)] \\ &\quad + A_2 [\cos(2\pi f_1 t) + \cos(2\pi f_2 t)]^2 \\ &= A_1 [\cos(2\pi f_1 t) + \cos(2\pi f_2 t)] + A_2 \cos^2(2\pi f_1 t) \\ &\quad + A_2 \cos^2(2\pi f_2 t) + 2A_2 \cos(2\pi f_1 t) \cos(2\pi f_2 t) \quad (3) \end{aligned}$$

Due to the LPF, A_1S_{in} will be filtered. Therefore, only the quadratic term $A_2S_{in}^2$ will remain. We expand this term (omitting amplitude A_2) as follows:

$$S_{in}^2 = \frac{1}{2} [1 + \cos(4\pi f_1 t)] + \frac{1}{2} [1 + \cos(4\pi f_2 t)] + \cos(2\pi(f_1 + f_2)t) + \cos(2\pi(f_1 - f_2)t) \quad (4)$$

where $f_1 - f_2 = 5 \text{ kHz} < 24 \text{ kHz}$. This means that only $\cos(2\pi(f_1 - f_2)t)$ will remain after the LPF. Other terms lie out of the cutoff of the LPF and will be filtered. Therefore, S_{out} can be represented as

$$S_{out} = \cos(2\pi(f_1 - f_2)t) \quad (5)$$

Hence, a new frequency has been recorded by the microphone, but remains inaudible to humans.

B. THE PRINCIPLE OF ULTRASONIC JAMMING

The principle of ultrasonic jamming is similar to inaudible voice attacks. BackDoor [5] has demonstrated that frequency modulation (FM) is more suitable for carrying signals than amplitude modulation (AM) and phase modulation (PM), so FM modulated signals are adopted to jam the spy microphones in this paper. We assume that $\cos(2\pi f_m t)$ is the message signal and its frequency is f_m . Then, the message signal is modulated by FM:

$$S_{fm} = \cos(2\pi f_{c1} t + A_0 \sin(2\pi f_m t)) \quad (6)$$

where $f_{c1} \geq 24 \text{ kHz}$ is the frequency of an ultrasonic carrier. The phase of the FM signal, namely $\sin(2\pi f_m t)$, is the integral of the message signal [5].

S_{fm} is transmitted by an ultrasonic transducer. Another transducer transmits $\cos(2\pi f_{c2} t)$ over the air, where $f_{c2} \geq 24 \text{ kHz}$. Therefore, S_{in} can be represented as

$$S_{in} = S_{fm} + \cos(2\pi f_{c2} t) = \cos(2\pi f_{c1} t + A_0 \sin(2\pi f_m t)) + \cos(2\pi f_{c2} t) \quad (7)$$

Then, according to Eq. (2), S_{amp} is composed of A_1S_{in} and $A_2S_{in}^2$. Because of the LPF, frequency components above 24 kHz will be removed. Therefore, A_1S_{in} will be filtered out by the LPF. The second order term (omitting amplitude A_2) can be calculated as follows:

$$S_{in}^2 = \cos^2(2\pi f_{c1} t + A_0 \sin(2\pi f_m t)) + \cos^2(2\pi f_{c2} t) + 2 \cos(2\pi f_{c1} t + A_0 \sin(2\pi f_m t)) \cos(2\pi f_{c2} t) \quad (8)$$

Then, we expand Eq. (8).

$$S_{in}^2 = \frac{1 + \cos(4\pi f_{c1} t + 2A_0 \sin(2\pi f_m t))}{2} + \frac{1 + \cos(4\pi f_{c2} t)}{2} + \cos(2\pi f_{c1} t + 2\pi f_{c2} t + A_0 \sin(2\pi f_m t)) + \cos(2\pi f_{c1} t - 2\pi f_{c2} t + A_0 \sin(2\pi f_m t)) \quad (9)$$

If $|f_{c1} - f_{c2}| \leq 24 \text{ kHz}$, the only remaining term is $\cos(2\pi f_{c1} t - 2\pi f_{c2} t + A_0 \sin(2\pi f_m t))$ after the LPF, and the other terms will also be cut off with A_1S_{in} . Therefore, $S_{out} = \cos(2\pi f_{c1} t - 2\pi f_{c2} t + A_0 \sin(2\pi f_m t))$. Then,

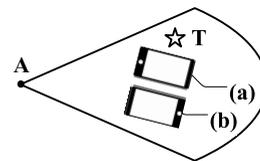


FIGURE 2. The illustration of the traditional directional coverage model.

S_{out} will be processed by the ADC and recorded by the microphone system.

Therefore, the ultrasonic signal carrying messages can be recorded by the spy microphone but remains inaudible to people. If the message signal is white noise, S_{in} will be recorded by a microphone and interfere with recording. To ensure that the noise generated by ultrasounds can cover people's voice, a number of ultrasonic transducers need to be used to form ultrasonic arrays.

C. COVERAGE MODELS

There are two types of coverage models. One typical type of model is the omnidirectional model, where the angle argument is not included in the coverage function [13]. Another coverage model is called the directional coverage model, which is used to calculate the sensing coverage of directional sensor networks. Directional sensors mainly include video sensors, infrared sensors, and ultrasound sensors [14]. The directional sensor has a finite angle of view and thus cannot sense the whole circular region [15]. Similar to [16], the traditional directional coverage model can be formulated as follows:

$$C(x, y) = \begin{cases} 1, & \text{if } (x, y) \in \Gamma \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

where Γ is the sensing range of the directional sensors. $C(x, y)$ measures whether a point (x, y) can be covered by sensors.

For instance, as shown in Fig. 2, a camera A has a finite angle of view. Γ is denoted by the sector area. If the coordinates of the point T are (x_0, y_0) , $C(x_0, y_0) = 1$ because T is in the sector area.

However, the above directional coverage models cannot address the problem of jamming recordings. For instance, we assume that A is an ultrasonic array sending jamming signals, as shown in Fig. 2. The primary microphones of phones are located at the bottom of the phones. Hence, phone (a) is facing A , but phone (b) is not. Due to the directional property of ultrasounds, ultrasonic signals can be recorded by phone (a), but the signals recorded by the phone (b) are weak. This means that when recording, only phone (a) will be jammed, for phones (a) and (b). Therefore, the previously mentioned directional coverage model does not work when solving the ultrasonic coverage problem, and we need a new coverage model that takes the orientations of microphones into account.

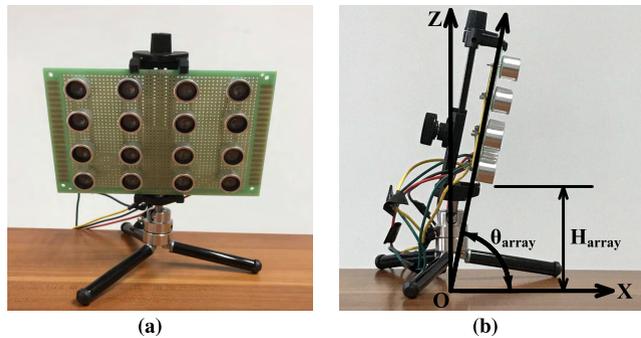


FIGURE 3. An ultrasonic array for jamming. (a) The front view. (b) The side view.

D. OPTIMIZATION ALGORITHMS

Approaches that have been studied to solve coverage models in WSN can be classified as the Voronoi diagram [17], virtual potential-field [18] and optimization techniques. The first two methods are not suitable for solving the ultrasonic coverage problem, so we focus on optimization algorithms. Optimization techniques can be divided into the genetic algorithm (GA) [19] and the harmony search algorithm (HSA) [20].

Wang *et al.* [19] discussed the priority-based target coverage problem. They aimed to select a minimum subset of directional sensors that can monitor all targets to satisfy the prescribed priorities. First, the priority-based target coverage model and the objective function are formulated. Then, by executing the genetic algorithm, the minimum subset of sensors is calculated. In [20], the harmony search algorithm was used to solve the sensing model. At each iteration, new solution vectors are evaluated by the sensing model and replace the worst solution. This process is repeated until the maximum iteration is reached.

III. PROPOSED ANGLE COVERAGE MODEL

A. PRELIMINARIES

Before the angle coverage model is proposed, preliminaries are given. These preliminaries consist of the definitions of ultrasonic arrays, the jamming region, and the working region. All the notations are defined in Table 2.

Ultrasonic arrays. Figs. 3(a)-(b) show the front and side views of an ultrasonic array, respectively. Every ultrasonic array is composed of 16 ultrasonic transducers [21], which are arranged in a 4×4 array. The row and column spacings are 2.5cm and 3.5cm, respectively, and the total size is $15\text{cm} \times 10\text{cm}$.

θ_{array} is the inclination angle of an ultrasonic array, and H_{array} denotes the height between the bottom of the array and the plane. An ultrasonic array is denoted as A_i , $1 \leq i \leq N$, where N is the number of arrays. The location of A_i can be uniquely represented as a 3-tuple $L_i(x_i, y_i, \alpha_i)$, $1 \leq i \leq N$, $-180^\circ < \alpha_i \leq 180^\circ$, where α_i is the rotation angle of A_i in the X-O-Y plane, as shown in Fig. 4.

The ultrasonic signals sent from A_i can jam microphones in some areas. We simply define the area where the phone can be jammed by A_i as A_{cover}^i . A more precise definition of A_{cover}^i

TABLE 2. Parameter definitions.

Ultrasonic arrays	
A	An ultrasonic array
L	The location of an array (cm, cm, degree)
N	The number of ultrasonic arrays
A_{cover}^i	The jamming area of A_i
θ_{array}	The inclination angle of an array (degree)
H_{array}	The height of an array (cm)
α_i	The rotation angle of A_i (degree)
$\gamma(x, y, A_i)$	The array rotation angle of A_i (degree)
Region	
W_0	The width of the working region (cm)
H_0	The height of the working region (cm)
W	The width of the jamming region (cm)
H	The height of the jamming region (cm)
I_0	The interval between the working region and the jamming region (cm)
Phone	
PH_{top}	The top of the phone
PH_{bottom}	The bottom of the phone
P	The position of PH_{bottom} (cm, cm, degree)
β	The rotation angle of a phone (degree)
$D(x, y)$	The allowable deviation angle of a phone at (x, y) (degree)
D_{max}	The maximum allowable deviation angle of a phone (degree)

will be given in Section III-C. The covered region surrounded by solid lines is simply used to estimate the shape of A_{cover}^i in Fig. 4. In addition, A_{cover}^i is symmetrical about a line, and the angle between the line and the X-axis is α_i . For example, in Fig. 4, $\alpha_i = 60^\circ$, and the covered region is symmetrical about A_iC_3 .

The area of A_{cover}^i is related to $\theta_{array} \in [0^\circ, 90^\circ]$, $H_{array} \geq 0$, and the power of the signals. To simplify the problem, we assume that θ_{array} , H_{array} and the transmitted power of every array are the same, so that the shape of A_{cover}^i of every array are the same.

Similar to BackDoor [5], multiple ultrasonic transducers are used to boost the jamming power level of an array. However, ultrasounds with the same frequency may induce constructive and destructive interference. If the microphone is placed at some points in A_{cover}^i , it may not be jammed. To simplify the problem, the interference of ultrasounds will not be considered in the present paper. The rationality lies in that only a small region in A_{cover} may not be jammed because of destructive interference, when determining A_{cover} in Section V-B.

Jamming region. We suppose that the jamming region is a $W \times H$ rectangular area, which is shown as the dashed box in Fig. 4. Our purpose is to jam the recording device placed in the jamming region in all directions.

Working region. The width and height of the working region are W_0 and H_0 , respectively, as presented in Fig. 4. In practice, the working region can be a meeting room or a table. Every array is limited to be deployed on the four edges

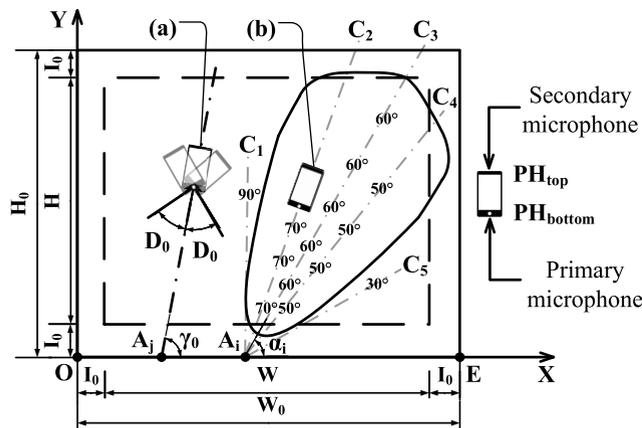


FIGURE 4. The deployment of the phones.

of the working region. Therefore, the location of every array can be represented as

$$\left\{ L_i(x_i, y_i, \alpha_i) | (0 \leq x_i \leq W_0, y_i = 0) \text{ or } (0 \leq x_i \leq W_0, y_i = H_0) \text{ or } (x_i = 0, 0 \leq y_i \leq H_0) \text{ or } (x_i = W_0, 0 \leq y_i \leq H_0) \right\} \quad (11)$$

where $1 \leq i \leq N$ and $-180^\circ < \alpha_i \leq 180^\circ$.

I_0 is defined as the interval between the working region and the jamming region, as shown in Fig. 4. I_0 is set because the distance between A_i and $A_i^{i_{cover}}$ exists if $H_{array} \neq 0$. When the working region is large and the jamming distance of A is far, I_0 can be ignored.

B. ASSUMPTIONS

To solve the ultrasonic coverage problem, some assumptions need to be made.

- 1) We consider jamming in a 2D plane in this paper. The rationality of this assumption lies in two aspects. First, one of the typical applications of JamSys is to jam the recording devices placed on conference tables. A table can be abstracted as a 2D plane. Second, due to the high complexity of the 3D coverage model, most existing works focus on the simplified 2D coverage model in WSN [18]. Similarly, we focus on the 2D model in this paper and will subsequently extend the model to the 3D scene in the future.
- 2) Roy et al. [12] demonstrated that inaudible voice command attacks can achieve an approximately 7.6m range. Therefore, it is possible for JamSys to reach room-level coverage. To reduce the complexity of the experiments, we assume that W_0 and H_0 are approximately 1m. By controlling the power, the furthest jamming range of A_i is set to approximately $\sqrt{W^2 + H^2}$, namely, the diagonal length of the jamming region. According to the above assumption, we can judge the effectiveness of the proposed coverage model and algorithm with low complexity.
- 3) The most common recording devices are smartphones. Therefore, we mainly prevent unauthorized recordings

by smartphones. The jamming effect of other recording devices will also be tested in experiments.

C. PROPOSED ANGLE COVERAGE MODEL

Generally, the primary and secondary microphones of a phone are located at the bottom and top of the phone separately. We denote them as PH_{bottom} and PH_{top} , as shown in the right part of Fig. 4. When a microphone is recording, the primary microphone mainly works. Next, some definitions are given.

The rotation angle of a phone is denoted as β . β is the angle between $PH_{bottom}PH_{top}$ and the X-axis. For example, for phone (b), $\beta = 70^\circ$ in Fig. 4. The position of a phone can be represented as a 3-tuple $P(x, y, \beta)$, $I_0 \leq x \leq I_0 + W$, $I_0 \leq y \leq I_0 + H$, $-180^\circ < \beta \leq 180^\circ$, where (x, y) is the location of PH_{bottom} , namely, the bottom of the phone.

In addition, we define the array rotation angle $\gamma(x, y, A_i)$ as the angle between A_iE and A_iC , where C is a point on the plane and (x, y) is any point on the line A_iC . For instance, in Fig. 4, $\angle C_kA_iE$, $1 \leq k \leq 5$ can be represented as $\gamma(x, y, A_i)$. In a special case, $\gamma(x, y, A_i)$ is equal to β when PH_{top} , PH_{bottom} , and A_i are aligned, and the primary microphone of the phone is facing A_j . For example, phone (b) is placed on the gray dash-dotted line A_iC_2 and facing A_i , so that $\gamma(x_2, y_2, A_i) = \beta = \angle C_2A_iE = 70^\circ$, where (x_2, y_2) is any point on the line A_iC_2 . Note that in the following paper, (x, y) is the location of PH_{bottom} , i.e., the center point of the bottom of the phone.

If the microphone is not placed toward the array, the ultrasonic jamming signals may not be recorded by the phone. Hence, we define $D(x, y) \in [0^\circ, 180^\circ]$ as the allowable deviation angle of a phone at (x, y) . For example, in Fig. 4, it is assumed that phone (a) is located at (x_0, y_0, γ_0) , so $\gamma(x_0, y_0, A_j) = \gamma_0$. $D(x_0, y_0)$ is denoted as D_0 . Then, we define that all the positions of a phone that meet $P(x_0, y_0, \gamma_0 - D_0 \leq \beta \leq \gamma_0 + D_0)$ can be jammed. In the experiments, we rotate the phone to determine $D(x, y)$.

However, $D(x, y)$ varies slightly when (x, y) is different. We define $D_{max} \in [0^\circ, 180^\circ]$ as the maximum allowable deviation angle of a phone for the given θ_{array} and H_{array} . D_{max} is the average value of N_{dmax} points, i.e., $D(x_k, y_k)$, $1 \leq k \leq N_{dmax}$.

$$D_{max} = \frac{1}{N_{dmax}} \sum_{k=1}^{N_{dmax}} D(x_k, y_k) \quad (12)$$

To simplify the problem, we assume that for one phone, the allowable deviation angle of every point is the same, which is denoted as D_{max} .

Therefore, $A_i^{i_{cover}}$ can be precisely defined as the area where for every point (x, y) , the phone can be jammed by A_i when $\beta \in [\gamma(x, y, A_i) - D_{max}, \gamma(x, y, A_i) + D_{max}]$.

When θ_{array} and H_{array} are given, D_{max} is a fixed value for a given phone. Additionally, every phone has its D_{max} . Specifically, if $D_{max} = 180^\circ$ and the phone is placed in $A_i^{i_{cover}}$, the phone can be jammed in all directions.

Finally, the angle coverage model is proposed. Given a point (x, y) in the jamming region, we define $F_i(x, y)$ to depict the range of jamming angles of A_i at (x, y) .

$$F_i(x, y) = \begin{cases} [\gamma(x, y, A_i) - D_{max}, \gamma(x, y, A_i) + D_{max}], & \text{if } (x, y) \in A_{cover}^i \\ \emptyset, & \text{otherwise} \end{cases} \quad (13)$$

where $I_0 \leq x \leq I_0 + W, I_0 \leq y \leq I_0 + H$ and $1 \leq i \leq N$. If (x, y) is out of the A_{cover}^i of $A_i, F_i(x, y) = \emptyset$.

For example, in Fig. 4, A_iC_1 and A_iC_5 are tangent to A_{cover}^i . Therefore, $\gamma(x, y, A_i) \in [30^\circ, 90^\circ]$. For every point on $A_iC_4, F_i(x, y) = [50^\circ - D_{max}, 50^\circ + D_{max}]$, if (x, y) is in A_{cover}^i .

Viewed from the point (x, y) in the jamming region, it is desirable that a point (x, y) is jammed in all directions. We assume that (x, y) is in the A_{cover} of n arrays, $0 \leq n \leq N$. All of the n arrays have a jamming range of angles at (x, y) . If the union of the range of whole n arrays at (x, y) can cover $(-180^\circ, 180^\circ)$, the phone will be jammed at (x, y) .

Therefore, when the jamming region is denoted as a two-dimensional matrix C , the angle coverage model can be represented as follows:

$$C(x, y) = \begin{cases} 1, & \text{if } (-180^\circ, 180^\circ) \subseteq \left(\bigcup_{i=1}^N F_i(x, y) \right) \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where $I_0 \leq x \leq I_0 + W, I_0 \leq y \leq I_0 + H$. If $C(x, y) = 1$, the phone will be jammed at (x, y) , irrespective of β .

However, if (x, y) are continuous, the objective function is difficult to solve. Therefore, we discretize the coordinates with a step length of $\Delta > 0$ to simplify the function, i.e., $x = I_0 + k/\Delta, k = 1, 2, \dots, \Delta \cdot W$ and $y = I_0 + k/\Delta, k = 1, 2, \dots, \Delta \cdot H$.

Therefore, the coverage rate can be calculated as follows:

$$O = \left(\sum_{x=I_0}^{W+I_0} \sum_{y=I_0}^{H+I_0} C(x, y) \right) / (\Delta \cdot W \times \Delta \cdot H) \quad (15)$$

where $0 \leq O \leq 1$, and the larger O is, the more area that can be jammed. With the increase in Δ , more accurate coverage can be calculated.

Finally, we obtain the objective function in Eq. (16).

$$\max O, \text{ subject to } N \leq \tau \quad (16)$$

where N is the number of arrays and τ is a constant. Generally, to solve the best solution, we set $N = \tau$.

Note that the dual problem of Eq. (16) is to minimize N given O . Both of the above objective functions can solve the ultrasonic coverage problem. We will focus on Eq. (16) in the following paper. In Section IV, the modified harmony search algorithm is proposed to solve Eq. (16).

Finally, we summarize the advantages of the angle coverage model.

- 1) The angle coverage model can depict the directionality of ultrasounds and microphones, and then transform the directions into a Boolean value, which indicates whether

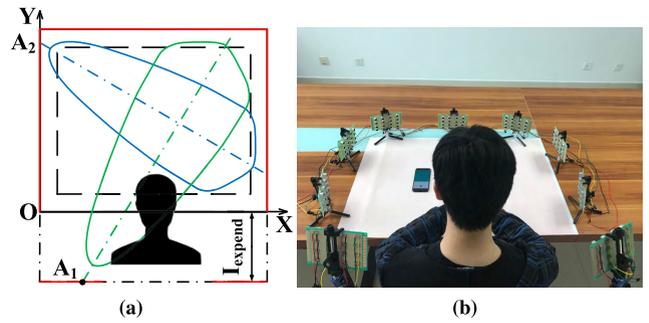


FIGURE 5. Deployment in the practical situation.

a microphone can be jammed at (x, y) , such that the model can be solved by mathematical formulas.

- 2) The model has good generality. The generality lies in three aspects. First, if both the primary and secondary microphones work when recording, our model still works as long as both of the microphones are in the jamming region. Second, regardless of the recording device, the objective function is the same. The differences of the recording devices only affect D_{max} and A_{cover} . Third, in different situations, the shape and size of the jamming region may be different. The angle coverage model does not require that the jamming region be a regular region, and any shape of the jamming region can be solved by the model.
- 3) Due to the generality of the angle coverage model, the jamming system can be customized. For example, all arrays can be mounted on the ceiling. Once H_{array} and θ_{array} are given, we can determine D_{max} and A_{cover} . Then, the angel coverage model and the modified harmony search algorithm (MHSA, which will be proposed in Section IV) can be used to optimize the coverage.

IV. MODIFIED HARMONY SEARCH ALGORITHM

A. PRACTICAL CASES

We mainly consider two practical cases in this paper. First, in case 1, the working region may be a room in reality, such as a meeting room. Every array can be placed on the four edges of the working region, i.e., the red lines in Fig. 6(a).

Second, in case 2, a small region needs to be jammed, such as a conference table. We divide the entire conference table into several identical parts. Each part is shown in Fig. 5. A person sits at the table and occupies a part of the table. In this way, a large and complex problem can be decomposed into many identical subproblems. Each subproblem can be solved by the method in Fig. 5. In this case, to not interfere with humans, any array placed on the X-axis, i.e., $\{L_i(x_i, y_i, \alpha_i) | 0 \leq x_i \leq W_0, y_i = 0, -180^\circ < \alpha_i \leq 180^\circ, 1 \leq i \leq N\}$ originally should be placed behind the person as A_1 in Fig. 5(a). The power and A_{cover} of these arrays should be slightly larger than the other arrays.

Considering that the person may block the ultrasonic signals transmitted by these arrays, we set $0 \leq x \leq W/4$ or $3W/4 \leq x \leq W$ when the arrays' Y coordinates

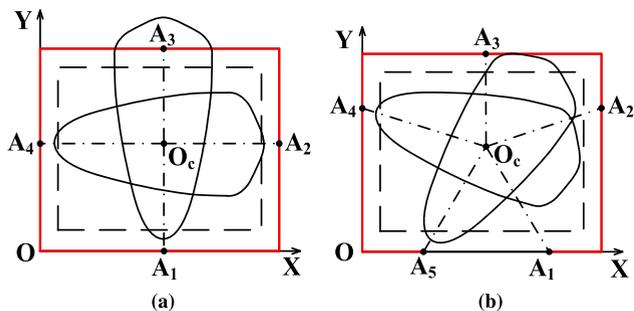


FIGURE 6. Regular coverage algorithm in the simulation. (a) Case 1. (b) Case 2.

are $-I_{\text{expend}}$ in case 2. Therefore, in reality and in actual experiments, all the arrays can only be placed on the red lines in Fig. 5(a). However, in the simulation, to simplify the problem, the arrays behind the person are still placed on the X-axis. All arrays are allowed to be placed on the red lines in Fig. 6(b).

B. REGULAR COVERAGE ALGORITHM

Intuitively, the uniform distribution is the simplest deployment strategy. Therefore, we first use the regular coverage algorithm (RCA) to deploy the arrays.

In case 2, O_c is the center of the jamming region, as shown in Fig. 6(b). The total array number is N , and A_1 and A_N are fixed at the starting point and end point of the red line, respectively. Then, we divide $360^\circ - \angle A_1 O_c A_N$ equally. $A_i, 1 < i < N$ is placed at the intersection between the angular bisector and the working region. The included angle can easily be calculated as

$$\angle A_i O_c A_{i+1} = (360^\circ - \angle A_1 O_c A_N) / (N - 1) \quad (17)$$

where $1 \leq i \leq N - 1$.

According to the included angle, we can infer the locations of all N arrays that are placed on the red lines.

For example, in Fig. 6(b), $N = 5$, and we divide $360^\circ - \angle A_1 O_c A_5$ equally. This means that $\angle A_1 O_c A_2 = \angle A_2 O_c A_3 = \angle A_3 O_c A_4 = \angle A_4 O_c A_5 = (360^\circ - \angle A_1 O_c A_5) / (5 - 1)$. If the width and height of the working region, namely, W_0 and H_0 , are given, the location of every array can easily be calculated.

Similarly, as shown in Fig. 6(a), we can infer that $\angle A_1 O_c A_2 = \angle A_2 O_c A_3 = \angle A_3 O_c A_4 = \angle A_4 O_c A_1 = 90^\circ$ in case 1.

C. MODIFIED HARMONY SEARCH ALGORITHM

The regular coverage algorithm is a simple method. Many optimization algorithms, such as the harmony search algorithm (HSA) can be utilized to find a better solution to Eq. (16). However, the previous HSA [20] cannot be directly used. Therefore, the modified harmony search algorithm (MHSA) is proposed to compensate for the limitation of the HSA.

We improve the HSA in the following three aspects. First, the previous HSA can only address regular sensing shapes,

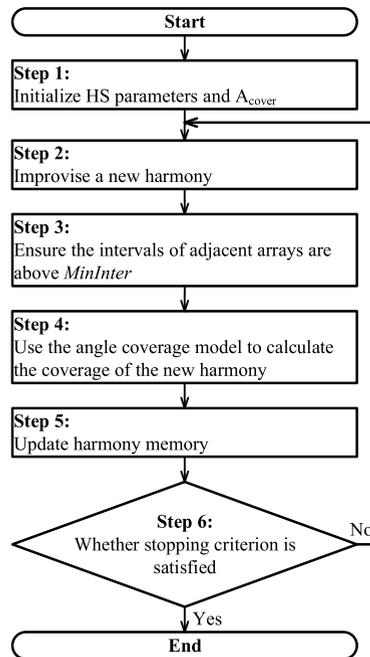


FIGURE 7. An overview of the modified harmony search algorithm.

such as circular shapes. However, the shape of the jamming area, i.e., $A_{\text{cover}}^i, 1 \leq i \leq N$, may be irregular in practice. Hence, a new method is proposed to determine whether a point is in the irregular A_{cover}^i . Second, considering the size of the arrays, the distance of adjacent arrays has a minimum value, denoted as $MinInter$. $MinInter$ is set to ensure that the deployment can be realized in practice. Third, a parallel computing strategy is adopted to accelerate the program. The following six steps detail the MHSA, as presented in Fig. 7.

Step 1: Initialize HS parameters and A_{cover}

As mentioned in Section III-A, the location of A_i can be represented as a 3-tuple $L_i(x_i, y_i, \alpha_i), 1 \leq i \leq N$. Therefore, the deployment of the whole arrays is a $1 \times 3N$ vector, which can also be called the solution vector. The harmony memory (M_{hm}) is a matrix composed of some solution vectors, as shown in Eq. (18). The harmony memory size (N_{hm}) is the number of solution vectors in M_{hm} .

$$M_{hm} = \begin{bmatrix} x_1^1 & y_1^1 & \alpha_1^1 & \dots & x_N^1 & y_N^1 & \alpha_N^1 \\ x_1^2 & y_1^2 & \alpha_1^2 & \dots & x_N^2 & y_N^2 & \alpha_N^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ x_1^{N_{hm}} & y_1^{N_{hm}} & \alpha_1^{N_{hm}} & \dots & x_N^{N_{hm}} & y_N^{N_{hm}} & \alpha_N^{N_{hm}} \end{bmatrix} \quad (18)$$

where M_{hm} is an $N_{hm} \times 3N$ matrix and each row of M_{hm} is a potential deployment of arrays. M_{hm} is randomly generated in this initial step.

The other parameters are described as follows.

- 1) I_{max} , the maximum permissible number of iterations.
- 2) R_{hmc} (Harmony memory considering rate), where $R_{hmc} \in [0, 1]$. The new solution vector inherits the values from M_{hm} with a probability of R_{hmc} .

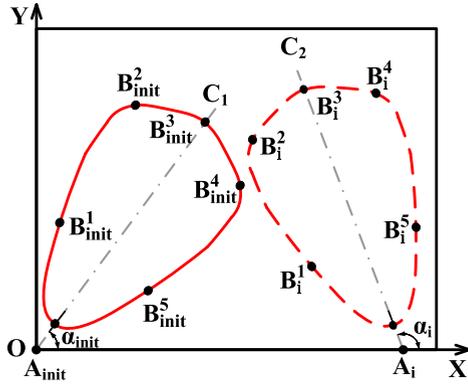


FIGURE 8. The illustration of irregular A_{cover} .

- 3) R_{pa} (Pitch adjusting rate), where $R_{pa} \in [0, 1]$. M_{hm} will be adjusted by probability R_{pa} .
- 4) B_w (Bandwidth), if the new solution vector is adjusted, the maximum modification amplitude is B_w .
- 5) N_{cpu} , the number of CPUs.

In addition, the shape and location of A_{cover}^i should be determined. To make the problem general, we assume that the shape of A_{cover}^i is irregular, as shown in Fig. 8. The location of the array A_{init} is $L_{init}(0, 0, \alpha_{init})$ and N_{point} coordinates of A_{cover}^i 's boundary, i.e., B_{init}^j , $1 \leq j \leq N_{point}$, are given, so A_{cover}^i can be drawn. For example, $N_{point} = 5$. B_{init}^j , $1 \leq j \leq 5$ are points on the boundary of A_{cover}^i in Fig. 8.

Step 2: Improve a new harmony

For each iteration, a new solution vector $v_{new} = (x_1, y_1, \alpha_1, \dots, x_N, y_N, \alpha_N)$ should be generated based on M_{hm} . For every array in v_{new} , (x_i, y_i, α_i) , $1 \leq i \leq N$ inherits the values $(x_i^k, y_i^k, \alpha_i^k)$ from M_{hm} with a probability of R_{hmc} , where $k \in [1, N_{hm}]$ is selected randomly. If the generated random number is in the probability of $1 - R_{hmc}$, (x_i, y_i, α_i) will be generated randomly. After repeating N times, v_{new} will be improvised.

Then, we determine whether v_{new} should be fine-tuned. For every array in v_{new} , a random number $r \in [0, 1]$ is generated. If $r > R_{pa}$, (x_i, y_i, α_i) remains. Otherwise, the array will be adjusted as follows:

$$\begin{cases} x_i = x_i + (2r_0 - 1) \times B_w \\ y_i = y_i + (2r_0 - 1) \times B_w \\ \alpha_i = \alpha_i + (2r_0 - 1) \times B_w \end{cases} \quad (19)$$

where $1 \leq i \leq N$ and $r_0 \in [0, 1]$ is a new random number.

Step 3: Ensure the intervals of adjacent arrays are above MinInter

When v_{new} is generated, we need to check whether every array is placed in the defined area, i.e., the red lines in Fig. 6. Moreover, considering the size of the arrays, we should ensure that the intervals of adjacent arrays are above a constant $MinInter$. If two arrays are too close to each other, we will increase their distance.

Step 4: Use the angle coverage model to calculate the coverage of the new harmony

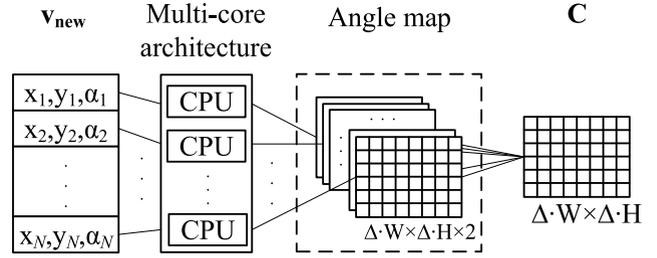


FIGURE 9. The illustration of the parallel computing strategy.

In this step, the coverage of v_{new} is calculated by the proposed angle coverage model according to Eqs. (14) and (15).

First, for a $1 \times 3N$ solution vector, we deal with every array, i.e., a 1×3 vector, separately. The difficulty of solving Eqs. (14) and (15) lies in judging whether a point (x, y) is in A_{cover}^i , $1 \leq i \leq N$, particularly when the shape of A_{cover}^i is irregular. According to the assumption in Section III-A, the shape of A_{cover}^i is the same for every A_i , $1 \leq i \leq N$. However, the locations of the arrays are different and determined by A_i . Therefore, based on the known $L_{init}(0, 0, \alpha_{init})$, $L_i(x_i, y_i, \alpha_i)$ and $B_{init}^j(x_{init}^j, y_{init}^j)$, the coordinate of $B_i^j(x_{Bi}^j, y_{Bi}^j)$, $1 \leq i \leq N$, $1 \leq j \leq N_{point}$, where B_i^j is the point on the boundary of A_{cover}^i , can be calculated by movement and rotation, as shown in Fig. 8. Generally, the coordinate of B_i^j can be calculated as follows:

$$\begin{bmatrix} x_{Bi}^j \\ y_{Bi}^j \end{bmatrix} = \begin{bmatrix} \cos(\alpha_i - \alpha_{init}) & -\sin(\alpha_i - \alpha_{init}) \\ \sin(\alpha_i - \alpha_{init}) & \cos(\alpha_i - \alpha_{init}) \end{bmatrix} \begin{bmatrix} x_{init}^j \\ y_{init}^j \end{bmatrix} + \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (20)$$

where $1 \leq i \leq N$, $1 \leq j \leq N_{point}$.

Hence, given $L_i(x_i, y_i, \alpha_i)$, $1 \leq i \leq N$, the edge and location of A_{cover}^i can be determined by $B_i^j(x_{Bi}^j, y_{Bi}^j)$, $1 \leq j \leq N_{point}$. As long as N_{point} is sufficiently large, the shape of A_{cover}^i can be depicted precisely.

When calculating the coverage, v_{new} is divided into N parts, and every part is a 1×3 vector. Then, for every 1×3 vector, because the boundary of A_{cover}^i can be determined by the above method, A_{cover}^i , $1 \leq i \leq N$ is drawn in a 2D plane and saved as a figure. We downsample the figure into a $\Delta \cdot W \times \Delta \cdot H$ matrix, and every element in the matrix indicates whether a point (x, y) , $x \in I_0 + [1/\Delta, W]$, $y \in I_0 + [1/\Delta, H]$ is in A_{cover}^i . Then, we define a new $\Delta \cdot W \times \Delta \cdot H \times 2$ matrix as the angle map. The angle map indicates $F_i(x, y)$. The angle map can be calculated by the downsampled figure and Eq. (13).

For all the 1×3 vectors, N angle maps can be computed independently by the above method. This inspires us to use a parallel computing strategy based on multiple CPUs to speed up the calculation, as shown in Fig. 9. Because the number of CPUs is N_{cpu} , N_{cpu} angle maps can be calculated at the same time. After the calculation of N angle maps, we compute C by Eq. (14). Finally, the coverage of v_{new} , i.e., O can be calculated.

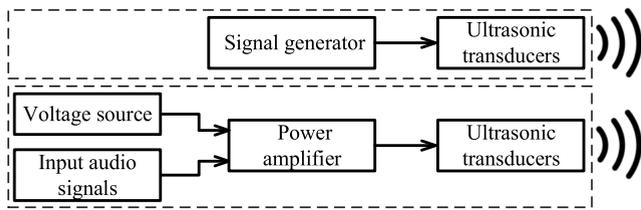


FIGURE 10. The diagram of the jamming transmitter.

Step 5: Update harmony memory

If the coverage of v_{new} is larger than the worst coverage of M_{hm} , the worst solution in M_{hm} will be replaced with v_{new} . Therefore, when iterating, the algorithm can improve the solution.

Step 6: Whether stopping criterion is satisfied

All of steps 2-6 are repeated until I_{max} is reached. Finally, the best solution in M_{hm} is the output of the MHSA.

V. EXPERIMENTAL RESULTS

A. SETUP

We evaluate the proposed methods through simulations and practical experiments. The experimental setup is shown as follows.

- 1) **Jamming transmitter.** As mentioned in Eqs. (6)-(9), the message signal is white noise, and we set the frequency of the noise signal as $f_m = 1\text{ kHz}$. The carrier frequencies are $f_{c1} = 40\text{ kHz}$ and $f_{c2} = 41\text{ kHz}$. Fig. 10 presents a diagram of the jamming transmitter. The signal generator drives the ultrasonic transducers with the signal $\cos(2\pi f_{c2}t)$. The input audio signals are modulated by FM (represented as S_{fm} in Eq. (6)) and then amplified by the power amplifier. Finally, the ultrasonic signals are transmitted by ultrasonic transducers. The jamming signal S_{in} is composed of S_{fm} and $\cos(2\pi f_{c2}t)$.
- 2) **Jamming criterion.** We select 100 words from Google’s Trillion Word Corpus [22]. Then, these words are converted to audio by a text-to-speech (TTS) engine [23]. This audio is played to simulate conversation, and we test whether the audio can be jammed when recording. The volume of the audio is set to 60dB at 1m away from the arrays. Additionally, the arrays transmit inaudible ultrasonic signals to jam microphones. When the location of the phone is $P(x, y, \beta)$, $I_0 \leq x \leq I_0 + W$, $I_0 \leq y \leq I_0 + H$, $-180^\circ < \beta \leq 180^\circ$, the TTS-generated audio is played. We record the audio and recruit 5 volunteers to listen to it. If less than 15% of the words [5] are legible to at least 3 volunteers, we consider that the recording device can be jammed at $P(x, y, \beta)$.
- 3) **Parameters.** The width and height of the working region, i.e., W_0 and H_0 are set to 100cm and 80cm, respectively. For the jamming region, $W = 70\text{cm}$, $H = 50\text{cm}$, and the interval between two regions, i.e., I_0 , is 15cm as shown in Fig. 4. For the convenience of experiments, $I_{extend} = 5\text{cm}$ from Section V-C to V-E. And $I_{extend} = 60\text{cm}$ in Section V-F as shown in Fig. 5(a). In the MHSA, the harmony memory size $N_{hm} = 20$,

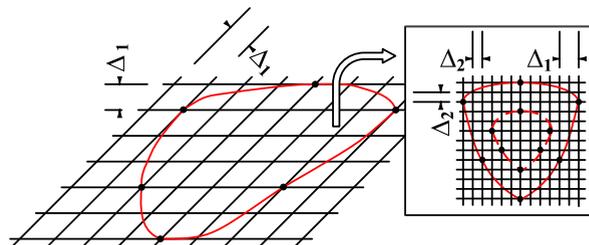


FIGURE 11. The illustration of D_{max} and A_{cover} .

the maximum iteration $I_{max} = 10000$, $R_{hmc} = 0.9$, $R_{pa} = 0.4$, $B_w = 5$, and $N_{cpu} = 13$. In Eq. (15), $\Delta = 3$ in the simulation.

- 4) **Simplified angle coverage model.** In the simulation, Eqs. (14) and (15) are used to calculate the coverage. However, in actual experiments, we only test 8 directions at a point, i.e., $\beta = -180^\circ + 45^\circ \times k$, $1 \leq k \leq 8$, due to the high complexity of Eq. (14). If a microphone can be jammed in all 8 directions at (x, y) in the experiments, we set $C(x, y) = 1$, where (x, y) is a point in the jamming region. Similarly, the total number of points that we need to test is $\Delta \cdot H \times \Delta \cdot W$ if we adopt the approach in Eq. (15). To simplify the experiments, we determine the area where the phones can be jammed through two rounds. First, we roughly determine the boundary of the jamming area at a fixed step length. The step length is 15cm ($\Delta = 1/15$) on both the X-axis and Y-axis. In the second round, the step length is set to 5cm ($\Delta = 1/5$) to test precisely. Therefore, less than $\lfloor (H/5) \times (W/5) \rfloor$ points need to be tested, and thus, the boundary of the jamming area can be determined.
- 5) **Genetic algorithm.** We compare the MHSA with the genetic algorithm (GA). We used the framework in [24] and the parameter values are set as follows. The coverage model is the angle coverage model. The set of solutions is called chromosomes, and the population consists of N_{ga} sets of chromosomes. $N_{ga} = N_{hm} = 20$. The crossover rate and mutation rate are set to 0.7 and 0.2, respectively. The initial value of the population is also generated randomly. $MinIter = 18\text{cm}$ for both the GA and the MHSA. In the GA, another parallel computing method is used. Because the N_{ga} sets of solution vectors are updated in each iteration, we compute N_{ga} solution vectors in parallel. However, when calculating the coverage of each solution vector, every 1×3 vector will no longer be calculated in parallel. Hence, to achieve the best performance, $N_{cpu} = N_{ga} = 20$ in the GA. The number of iterations will influence the running time of the GA. To conduct a fair comparison between the MHSA and the GA, we ensure that the running time of the MHSA is close to that of the GA. Therefore, I_{max} of the GA is equal to 2000 when I_{max} of the MHSA is 10000. Note that in the MHSA, the running time when $N_{cpu} = 13$ is similar to that when $N_{cpu} = 20$, so $N_{cpu} = 13$ is adopted in the MHSA to save

computing resources. When $N < 8$, the running time of the MHSA may be longer than that of the GA. However, this is acceptable because the search space is small when $N < 8$, the jamming percent has converged in the GA when $I_{max} = 2000$. Even if the number of iterations in the GA increases to ensure that the running time of the two algorithms is close, the jamming percent of the GA will not increase too much. Hence, for simplicity, I_{max} of the GA is 2000 for all N .

B. THE DETERMINATION OF D_{max} AND A_{cover}

In this experiment, only one array A is used. The receiver is an iPhone 7 smartphone (released in September 2016) running iOS 11.2.6. Note that when θ_{array} and H_{array} are different, D_{max} of a phone may vary. Therefore, we first fix θ_{array} and H_{array} . When determining D_{max} and A_{cover} , our approach is to obtain D_{max} first and then determine according to the fixed D_{max} .

- 1) D_{max} . As shown in Fig. 11, we move the phone in the plane. The step length is Δ_1 . For each position of the phone, i.e., $P(x, y, \gamma(x, y, A))$, where (x, y) is the location of PH_{bottom} and A is an array, we record the coordinates if the phone can be jammed. Therefore, a rough A_{cover} can be determined as the area surrounded by red solid lines in Fig. 11. In the rough A_{cover} , we regularly select N_{dmax} points and calculate D_{max} according to Eq. (12).
- 2) A_{cover} . Because A_{cover} is inside the rough A_{cover} , the search space can be reduced. In this step, the step length is Δ_2 , and $\Delta_2 < \Delta_1$. When the phone is moved to (x, y) , we measure the $D(x, y)$ of this point. If $D(x, y) \geq D_{max}$, (x, y) is in A_{cover} ; otherwise, $(x, y) \notin A_{cover}$. Finally, A_{cover} can be determined, as indicated by the red dashed line in Fig. 11. Note that due to the existence of destructive interference, if placed at a small particular region in A_{cover} , the phone may not be jammed. In the simulation, the small region will be ignored for simplicity.

In addition, note that for any given H_{array} and θ_{array} in reality, the angle coverage model and the MHSA can still optimize the coverage, as long as D_{max} and A_{cover} are determined. In the following experiments, for all ultrasonic arrays, $H_{array} = 7\text{cm}$ and $\theta_{array} = 85^\circ$ are adopted. $N_{dmax} = 10$, $\Delta_1 = 1/20$ (20cm), and $\Delta_2 = 1/10$ (10cm). After the actual experiments, D_{max} of the iPhone 7 smartphone is 52° . Two semiellipses are used to simulate A_{cover} , as shown in Fig. 12. The array is placed at A . In the simulation, the long semiaxis of the ellipse $B_0B_1B_3B_2B_0$, i.e., $B_0B_3 = 40\text{cm}$ and the short axis $B_1B_2 = 43\text{cm}$. Similarly, for semiellipse $B_0B_1B_4B_2B_0$, $B_0B_4 = 65\text{cm}$.

C. PERFORMANCE OF THE PROPOSED ANGLE COVERAGE MODEL

To demonstrate the validity of the angle coverage model, comparisons of the simulation and actual experiments are shown in Fig. 13. The receiver is still the iPhone 7 smartphone. The power of every array is the same as the power in

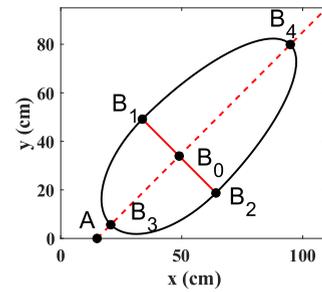


FIGURE 12. The shape of A_{cover} in the simulation.

TABLE 3. The absolute difference of jamming percent between the simulation and actual experiments.

N	Case 1			Case 2		
	MHSA	GA	RCA	MHSA	GA	RCA
4	5.51	7.11	0.61	3.21	7.91	0.32
5	10.04	8.65	11.01	15.83	7.48	11.92
6	1.81	3.41	2.07	0.07	1.31	4.12
7	3.44	4.81	3.62	1.95	5.40	2.41
8	7.89	14.36	9.04	5.89	6.87	2.01
9	4.79	7.72	13.65	2.78	4.95	6.30
10	5.74	7.67	10.16	0.85	2.77	16.10
11	4.59	3.52	15.03	3.77	1.08	14.63
12	1.76	1.58	15.04	3.57	3.04	19.32
13	1.22	3.14	19.19	3.39	6.60	24.07
Average	4.68	6.20	9.94	4.13	4.74	10.12

Section V-B. Because too many points need to be tested in this experiment, we need to reduce the complexity of the experiments. Therefore, we only select 10 words from [22] and then convert the 10 words to an audio to simulate conversation in this experiment. Fig. 13(a) illustrates the results of case 1, where the jamming region represents a meeting room and all arrays are placed at the four edges of the working region. In case 2, the jamming region is a part of a table, and some arrays are deployed behind the person. The experimental results are shown in Fig. 13(b).

As shown in Fig. 13, the X-axis represents the number of arrays, and the Y-axis indicates the percentage of the area where the recording devices can be jammed. When the number of arrays, i.e., N , is less than 8, the jamming percentages of the three algorithms are relatively close. When the number of arrays is greater than 8, the jamming percent of the MHSA increases faster than the other two algorithms in both the simulation and experiments. The reason for the difference is that when N is small, the coverage is limited by N , and the regular coverage algorithm (RCA) can cover the area in all directions, which maximizes the jamming effect of every array. In contrast, when N is larger and the RCA is adopted, many arrays will repeatedly cover the region that has already been jammed, which reduces the efficiency. However, the MHSA and the GA can search uncovered areas and cover these regions. More importantly, our modified method, i.e., the MHSA, has the best performance of all three algorithms in cases 1 and 2 due to its high search efficiency.

In both the simulation and actual experiments, the jamming percent is higher in Fig. 13(a) than in Fig. 13(b) under most circumstances when the same algorithm is used. This result

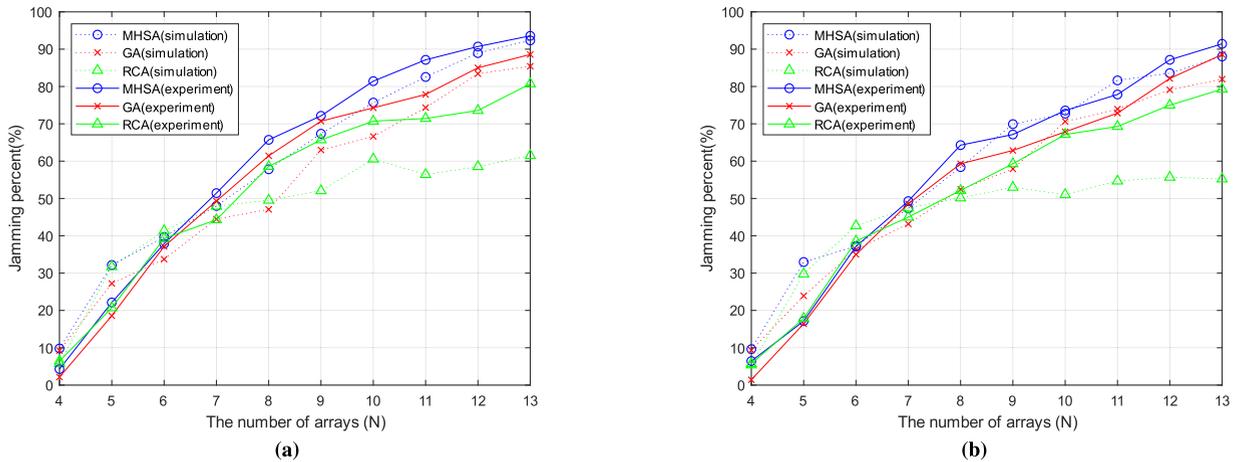


FIGURE 13. A comparison between the simulation and experiments of case 1 and case 2 using iPhone 7. (a) Case 1. (b) Case 2.

occurs because in case 1, more area is available to place arrays. Therefore, a better solution can be searched.

When $N = 13$, more than 90% of the area can be jammed in cases 1 and 2, and this deployment is satisfactory. Hence, in the following experiments, 13 arrays are deployed to form a jamming system.

The absolute difference of jamming percent between the simulation and actual experiments is presented in Table 3. The average difference of absolute values is from 4.13% to 10.12%. Therefore, the simulation fits the results well and further demonstrates that the angle coverage model is a good objective function.

However, the simulated coverage for the RCA is significantly lower than the measured coverage when the number of arrays is large, as shown in Table 3. Two reasons account for this phenomenon.

First, as shown in Eq. (14), only if the union of the range of whole N arrays at (x, y) can cover $(-180^\circ, 180^\circ]$ can the phone be jammed at (x, y) in the simulation. However, in the experiment, due to the high complexity of Eq. (14), we only test the jamming percent in 8 directions at (x, y) . Therefore, the calculation of coverage in the simulation is more rigorous than that in the experiment. As shown in Fig. 13, when the number of arrays is larger than 7, the measured jamming percentages of all three algorithms, i.e., the MHSA, GA, and RCA, are higher than the simulated jamming percentages. Hence, the gap between the simulated coverage and the measured coverage of all the algorithms, including the RCA, can be reduced if more directions are tested at (x, y) in the experiment.

Second, the simulated A_{cover} is slightly smaller than the real A_{cover} . When determining A_{cover} in the experiment, as shown in Fig. 11, a part of the area between the red solid line and the red dashed line can also be jammed with smaller jamming ranges of angles. Therefore, the area of simulated A_{cover} is smaller, and some points in that area may also be jammed in the experiments. Meanwhile, the RCA only has one deployment strategy when N is given. Therefore, the coverage of the RCA is mainly determined by the shape of A_{cover}

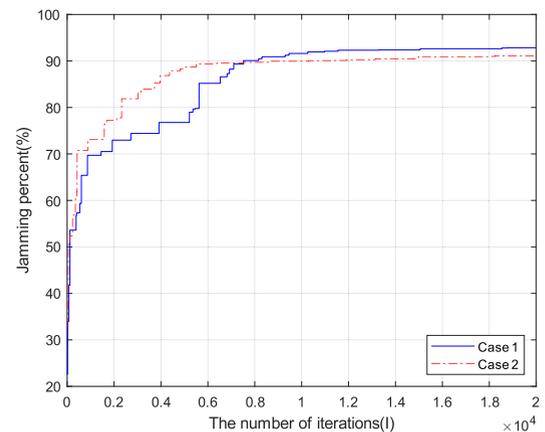


FIGURE 14. The convergence of the MHSA in case 1 and case 2 when $N = 13$.

in the simulation. The smaller A_{cover} will lead to smaller coverage of the RCA in the simulation. However, the MHSA and the GA can search for better deployment strategies such that the smaller A_{cover} will not affect the coverage too much.

Based on the above two reasons, the coverage of the RCA in the simulation is lower than the measured coverage. In future work, we will solve the above problems in two aspects. First, we will test the jamming percent in more directions in the experiments. Second, in the simulation, a more precise A_{cover} will be adopted.

D. THE CONVERGENCE OF THE MHSA

In Section V-C, $N = 13$ is adopted to form a jamming system. In this subsection, the convergence of the MHSA is studied in cases 1 and 2 when $N = 13$. As shown in Fig. 14, each curve represents the change in the jamming percent with the increase in the number of iterations, which is denoted as I . The maximum number of iterations is 20000.

As shown in Fig. 14, when I is near 20000, the jamming percent remains almost unchanged in both cases. Therefore, even if a much larger I is used in the MHSA, the jamming percent in the simulation will not increase too much. In addition,

TABLE 4. Jamming percent comparison of the MHSA, the GA, and the RCA in experiments using different devices ($N = 13$).

Device	Case 1			Case 2		
	MHSA	GA	RCA	MHSA	GA	RCA
iPhone 7	93.57	88.57	81.43	91.43	88.57	80.71
iPhone 8	100.00	99.29	97.14	97.86	96.43	95.71
Xiaomi 6	60.00	59.29	56.43	53.57	54.29	52.14
Sony RX100m3	72.86	71.43	65.71	72.50	69.29	62.14

the MHSA can converge to the high jamming percent when I is large.

Considering the tradeoff between the jamming percent and the running time of the algorithm, we adopt $I_{max} = 10000$ in Section V. The rationality of $I_{max} = 10000$ lies in three aspects. First, when I increases from 10000 to 20000, the jamming percent increases approximately 1.22% in case 1 and 1.17% in case 2, but the running time doubles. Therefore, we can save considerable time without sacrificing too much jamming percent when $I_{max} = 10000$. Second, the deployment strategy calculated when $I = 10000$ is similar to that when $I = 20000$. In actual experiments, the difference in coverage can be ignored when the above two deployment strategies are adopted. Hence, $I_{max} = 10000$ is reasonable in both case 1 and case 2. Third, because the size of the solution vector, i.e., v_{new} , is $1 \times 3N$, with a decrease in N , the search space will be reduced in cases 1 and 2. Therefore, when $N < 13$, the jamming percent will converge to a stable value at a faster rate, and $I_{max} = 10000$ is sufficiently large.

In addition, the area allowed to place arrays in case 1 is larger than that in case 2; thus, the search space is larger in case 1. Therefore, the jamming percent converges more slowly in case 1 than in case 2, as shown in Fig. 14. The jamming percent is higher in case 1 than in case 2 when $I \geq 8000$ because more area can be searched in case 1, so that better solutions can be calculated. Therefore, Fig. 14 is reasonable.

E. PERFORMANCE OF THE PROPOSED MHSA

In reality, the difficulty lies in the fact that the deployment strategy is fixed, but the recording devices vary. To demonstrate the universality of MHSA, the deployment of arrays is the same as in Section V-C ($N = 13$). We compare the performances of three algorithms on different recording devices in cases 1 and 2, as shown in Table 4. When a recording device is tested, the power of the arrays is the same. However, the power is different when testing different devices. The reason is that the jamming percent may be 100% for all three algorithms to some recording devices if the power is the same, which cannot distinguish the performance of the algorithms.

The data in Table 4 indicate that in most cases, the MHSA has a better jamming percentage than the other two methods. In case 1, the MHSA performs better than the GA by 0.71%-5.00%, while the improvement is 2.86%-12.14% when compared with the RCA. In case 2, the improvement of the MHSA can reach 3.21% and 10.71% when compared with

the GA and the RCA, respectively. From another perspective, the jamming area is larger using the MHSA than the other algorithms in most cases. We attribute this phenomenon to the strong generalization of the angle coverage model and the MHSA.

Then, we discuss the reason why the MHSA is superior to the GA and the RCA in Section V-C and V-E. First, we compare the MHSA with the RCA. The RCA is also an algorithm that we propose. In the RCA, given the working region and the number of arrays N , we can calculate the deployment strategy quickly using geometric methods. Then, based on the above deployment strategy, the coverage can be computed. However, the MHSA is an optimization method that searches for the optimal solution of the objective function until the iteration stops. With the increase in the number of iterations, the coverage calculated by the MHSA continues to increase until it converges. Therefore, the MHSA has better performance than the RCA.

In addition, the differences between the MHSA and the GA mainly lie in two aspects. First, the frameworks of the two algorithms are different. For each iteration of the MHSA, a new solution vector v_{new} is generated. If the coverage of v_{new} is larger than the worst coverage of M_{hm} , the worst solution in M_{hm} will be replaced with v_{new} . However, for the GA, the whole population, which consists of N_{ga} sets of solution vectors, is updated in every iteration. Therefore, the calculation time of the GA is several times longer than that of the MHSA in every iteration. The method for generating new solution vectors in the two algorithms is different. The framework of the MHSA is more suitable to solve Eq. (16); thus, in most cases, the jamming percent is higher when the MHSA is used.

Second, the parallel computing methods of the two algorithms are different. For the MHSA, we process N arrays in a solution vector in parallel. After summarizing all the results of N arrays, the coverage can be calculated. In the GA, we compute N_{ga} solution vectors in parallel. As shown in Fig. 13, when N is large, the parallel computing strategy of the MHSA is better than that of the GA because the MHSA can find better deployment methods when the same time is spent and fewer CPUs are used.

F. JAMMING PERFORMANCE OF JamSys

In this subsection, we demonstrate the effectiveness of JamSys. The deployment strategy is the MHSA, and $N = 13$. The smallest D_{max} is 52° among iPhone 7, iPhone 8, Xiaomi 6, and Sony RX100m3. Hence, $D_{max} = 52^\circ$ is adopted such that JamSys will work for all the above four devices. For every recording device, 20 different locations of the device $P(x_k, y_k, \beta_k)$, $I_0 \leq x_k \leq I_0 + W$, $I_0 \leq y_k \leq I_0 + H$, $-180^\circ < \beta_k \leq 180^\circ$ where $1 \leq k \leq 20$ are selected randomly in the jamming region. This experiment is more practical because more directions can be tested instead of the limited 8 directions.

Similar to the validation methodology of [5], the recorded audios are played back to volunteers and the speech

TABLE 5. Jamming percent of JamSys recognized by volunteers and DeepSpeech ($N = 13$).

Device	Case 1		Case 2	
	Volunteers	DeepSpeech	Volunteers	DeepSpeech
iPhone 7	99.90	100.00	99.25	100.00
iPhone 8	100.00	100.00	99.95	100.00
Xiaomi 6	100.00	100.00	100.00	100.00
Sony RX100m3	100.00	100.00	100.00	100.00

recognition system DeepSpeech [25]. Then, we count how many words can be jammed in the whole 20×100 words for every recording device. The average number of legible words by 5 volunteers is recorded.

Table 5 shows the jamming percent of JamSys recognized by volunteers and DeepSpeech when $N = 13$. To make the evaluation more objective, DeepSpeech is adopted to evaluate the jamming percent. To volunteers, JamSys jams all the tested recording devices, and more than 99% of words are illegible. Moreover, 100% of words can be jammed for all recording devices when the audios are recognized by DeepSpeech. The experimental results verify the effectiveness of JamSys.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we designed JamSys, a jamming system based on ultrasounds. Due to the nonlinearity of microphones, well-designed ultrasonic noise signals can be recorded by microphones, but remain inaudible to humans. To jam a given area, some ultrasonic arrays need to be leveraged to form a jamming system. We propose the angle coverage model and the modified harmony search algorithm (MHSA) to maximize the coverage ratio and calculate the deployment scheme given the number of ultrasonic arrays. The efficiency and superiority of the modified harmony search algorithm have been demonstrated compared with the genetic algorithm (GA) and the regular coverage algorithm (RCA) through both the simulation and actual experiments.

In addition, further research can be performed to improve JamSys. We outline a few directions here.

1) **3D coverage model.** As the first paper to study the jamming system, our work focuses on the 2D coverage model. In reality, the recording devices could be placed anywhere. Therefore, a 3D coverage model is more practical. We will generalize the proposed angle coverage model and the MHSA to solve the coverage problem in 3D space.

2) **The theoretical analysis of acoustics.** The ultrasonic coverage problem can also be considered from the perspective of acoustic propagation. We can calculate the boundary of A_{cover} based on the acoustic method such that the angle coverage model can be improved. In addition, different deployment methods of transducers can build different ultrasonic arrays. Theoretical acoustics can guide us to design a new array to reduce the effect of destructive interference.

ACKNOWLEDGMENT

The authors would like to thank all the anonymous reviewers for their helpful suggestions and valuable comments.

REFERENCES

- [1] *Aj-34 Audio Jammer Noise Generato*. Accessed: Mar. 21, 2019. [Online]. Available: <https://proofpronto.com/privacy-protection/conversation-protection/aj-34-audio-jammer-noise-generator>
- [2] D. F. Kune, J. Backes, S. S. Clark, D. Kramer, M. Reynolds, K. Fu, Y. Kim, and W. Xu, "Ghost talk: Mitigating EMI signal injection attacks against analog sensors," in *Proc. IEEE Symp. Secur. Privacy*, May 2013, pp. 145–159.
- [3] Y. Wu, Z. Ma, and W. Xu, "Research of anti-eavesdropping technology based on electromagnetic interference against analog sensors," *Electron. Technol.*, vol. 45, no. 4, pp. 47–51, 2016. doi: 10.3969/j.issn.1000-0755.2016.04.014.
- [4] *Ultrasonic Microphone Suppressor + White Noise Generator Bughunter BDA-3*. Accessed: Mar. 21, 2019. [Online]. Available: https://www.alibaba.com/product-detail/Ultrasonic-microphone-suppressor-white-noise-generator_50036200189.html
- [5] N. Roy, H. Hassanieh, and R. R. Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *Proc. 15th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2017, pp. 2–14.
- [6] A. Sangwan and R. P. Singh, "Survey on coverage problems in wireless sensor networks," *Wireless Pers. Commun.*, vol. 80, no. 4, pp. 1475–1500, 2015.
- [7] I. Khoufi, P. Minet, A. Laouiti, and S. Mahfoudh, "Survey of deployment algorithms in wireless sensor networks: Coverage and connectivity issues and challenges," *Int. J. Auton. Adapt. Commun. Syst. (IJAAACS)*, vol. 10, no. 4, pp. 341–390, 2017.
- [8] C. Zhu, C. Zheng, L. Shu, and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 619–632, Mar. 2012.
- [9] A. Boubrima, W. Bechkit, and H. Rivano, "Optimal WSN deployment models for air pollution monitoring," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2723–2735, May 2017.
- [10] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, pp. 103–117, Nov. 2017.
- [11] L. Song and P. Mittal, "Poster: Inaudible voice commands," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2017, pp. 2583–2585.
- [12] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible voice commands: The long-range attack and defense," in *Proc. 15th USENIX Symp. Networked Syst. Design Implement. (NSDI)*, 2018, pp. 547–560.
- [13] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Comput. Surv.*, vol. 43, no. 4, p. 32, Oct. 2011.
- [14] M. A. Guvensan and A. G. Yavuz, "On coverage issues in directional sensor networks: A survey," *Ad Hoc Netw.*, vol. 9, no. 7, pp. 1238–1255, Sep. 2011.
- [15] D. Tao and T.-Y. Wu, "A survey on barrier coverage problem in directional sensor networks," *IEEE Sensor J.*, vol. 15, no. 2, pp. 876–885, Feb. 2014.
- [16] Y. Yoon and Y.-H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1473–1483, Oct. 2013.
- [17] T.-W. Sung and C.-S. Yang, "Voronoi-based coverage improvement approach for wireless directional sensor networks," *J. Netw. Comput. Appl.*, vol. 39, pp. 202–213, Mar. 2014.
- [18] H. Ma, X. Zhang, and A. Ming, "A coverage-enhancing method for 3D directional sensor networks," in *Proc. INFOCOM*, Apr. 2009, pp. 2791–2795.
- [19] J. Wang, C. Niu, and R. Shen, "Priority-based target coverage in directional sensor networks using a genetic algorithm," *Comput. Math. Appl.*, vol. 57, nos. 11–12, pp. 1915–1922, 2009.
- [20] O. M. Alia and A. Al-Ajouri, "Maximizing wireless sensor network coverage with minimum cost using harmony search algorithm," *IEEE Sensors J.*, vol. 17, no. 3, pp. 882–896, Feb. 2017.
- [21] *16 mm 40MHz Split Ultrasonic Transceiver and Receiver Ultrasonic Sensor Ultrasonic Probe TCT40-16RT*. Accessed: Mar. 21, 2019. [Online]. Available: https://www.alibaba.com/product-detail/16MM-40MHz-split-ultrasonic-transceiver-and_60830365835.html?spm=a2700.7724838.2017121.10.42f18ed6e0sERU
- [22] *Top 10000 Words from Google's Trillion Word Corpus*. Accessed: Mar. 21, 2019. [Online]. Available: <https://github.com/first20hours/google-10000-english>
- [23] *Free Online Text to Speech (TTS) Service With Natural Sounding Voices*. Accessed: Mar. 21, 2019. [Online]. Available: <http://www.fromtexttospeech.com/>

- [24] K. S. Yildirim, T. E. Kalayci, and A. Uğur, "Optimizing coverage in a k-covered and connected sensor network using genetic algorithms," in *Proc. 9th WSEAS Int. Conf. Evol. Comput., World Sci. Eng. Acad. Soc. (WSEAS)*, May 2008, pp. 21–26.
- [25] Mozilla. *Project DeepSpeech*. Accessed: Mar. 21, 2019. [Online]. Available: <https://github.com/mozilla/DeepSpeech>



HAO SHEN received the B.S. degree from the Harbin Institute of Technology, in 2017. He is currently pursuing the M.S. degree in electrical engineering with the University of Science and Technology of China. His research interests include privacy protection and audio processing.



WEIMING ZHANG received the M.S. and Ph.D. degrees from the Zhengzhou Information Science and Technology Institute, China, in 2002 and 2005, respectively. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding and multimedia security.



HAN FANG received the B.S. degree from the Nanjing University of Aeronautics and Astronautics (NUAA), in 2016. He is currently pursuing the Ph.D. degree in information security with the University of Science and Technology of China (USTC). His research interests include image watermarking, information hiding, and image processing.



ZEHUA MA received the B.S. degree in information security from the University of Science and Technology of China (USTC), in 2018, where he is currently pursuing the M.S. degree in information security. His research interests include image watermarking, information hiding, and image processing.



NENGHAI YU received the B.S. degree from the Nanjing University of Posts and Telecommunications, in 1987, the M.E. degree from Tsinghua University, in 1992, and the Ph.D. degree from the University of Science and Technology of China, in 2004, where he is currently a Professor. His research interests include multimedia security, multimedia information retrieval, video processing, and information hiding.

...