Distributed Sensor Selection for Speech Enhancement With Acoustic Sensor Networks

De Hu, *Member, IEEE*, Qintuya Si, Rui Liu, *Member, IEEE*, and Feilong Bao

***Abstract*—In distributed acoustic sensor networks, only a few nodes make a significant contribution to speech enhancement tasks. Using these most informative nodes instead of the entire network not only avoids unnecessary energy consumption but also prolongs the lifetime of sensors. To this end, a sensor selection method for distributed speech enhancement is proposed. The best subset of microphone nodes is determined by maximizing the signal-to-noise ratio (SNR), while keeping the activated nodes connected with each other. The above criterion involves an integer and non-linear programming, which is linearized with multiple base-3 sub-optimization problems, and each of them is solved by a state-of-the-art steepest descent (SD) algorithm. In addition, a greedy searching strategy is presented to select sensors rapidly. Finally, a distributed SD algorithm is further derived, which is more suitable for distributed sensor networks. The proposed method can obtain the optimal subnetwork in noisy and reverberant environments. Unlike the existing approaches, it can select nodes from a microphone network with arbitrary communication graphs. Moreover, it requires only local communications among nodes without an external central processor. Experimental results confirm the validity of the proposed method.**

***Index Terms*—Boolean optimization, distributed microphone networks, sensor selection, speech enhancement.**

I. Introduction

With the rapid advances in integrated circuits, wireless communications, and sensors, mobile devices with microphones are ubiquitous around us. If they constitute a wireless acoustic sensor network (WASN), it can boost the performance of many speech processing tasks, e.g., speech enhancement [1], [2], [3], speech recognition [4], speaker localization and tracking [5], [6], [7], etc. Recently, there has been a growing interest in distributed speech enhancement with WASNs [8], [9], [10], which can break the performance bottleneck of approaches using a single acoustic sensor (or acoustic sensor array).

In a WASN, each node consists of a signal processing unit, a wireless communication link, and one or more microphones. Compared with the traditional microphone array, the WASN is no longer constrained to any specific array configuration, and the number or position of sensor nodes is not fixed anymore. Such a sensor network covers a wider area and captures audio signals with high signal-to-noise ratios (SNRs). For instance, there is a higher probability that at least one node is close to the target source(s) and yields a higher SNR signal. Although WASNs possess many advantages, they also suffer from some challenges, such as sampling rate mismatch among nodes [11] and unforeseeable node positions [12], which can be estimated by our previous calibration works [13], [14], [15].

Since the WASN provides higher-quality recordings, several speech enhancement methods using WASNs are developed in a centralized way [16], [17] or in a distributed way [18], [19]. In a centralized manner, each node transmits its measurements to a fusion center (FC) that physically connects all nodes, and then the FC performs all computations, which brings a large amount of transmission and computation burdens for the FC. In contrast, in a distributed manner, the calculations are implemented in parallel over all nodes, where the FC is unnecessary and the information is transmitted among neighboring nodes only. It overcomes the limitations of centralized methods and has attracted great interest in recent years. Specifically, Zeng and Hendriks [20] proposed a distributed delay and sum (D-DS) beamformer for speech enhancement, where the spatial filter coefficients are obtained in each node based on the randomized gossip algorithm [21]. Afterward, in [22], they further proved that the D-DS beamformer converges to the optimal solution of the centralized one. This approach requires no restriction on the network topology and is robust for distributed WASNs. That same year, a distributed minimum variance distortionless response (MVDR) beamformer [23] was presented, in which a quadratic optimization problem is formulated and decomposed into multiple local cost functions, and then the filter coefficients are estimated iteratively based on message passing. Another approach was based on the diffusion adaptation paradigm, where a distributed objective function is constructed using the partial covariance matrix observed in each node [24]. Since the assumption of uncorrelated noises was relaxed in MVDR methods, they can outperform the D-DS beamformer. In [25], Bertrand and Moonen proposed a distributed linearly constrained minimum

Manuscript received 24 July 2022; revised 29 November 2022 and 29 December 2022; accepted 27 January 2023. Date of publication 13 February 2023; date of current version 23 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grants 62201297 and 62206136, and in part by Inner Mongolia University through High-level Talents Introduction Project under Grants 10000-22311201/002 and 10000-22311201/003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Daniele Salvati. *(Corresponding author: Qintuya Si.)*

De Hu, Rui Liu, and Feilong Bao are with the College of Computer Science, Inner Mongolia University, Hohhot 010021, China, also with the National and Local Joint Engineering Research Center of Intelligent Information Processing Technology for Mongolian, Inner Mongolia University, Hohhot 010021, China, and also with the Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, Inner Mongolia University, Hohhot 010021, China (e-mail: cshood@imu.edu.cn; liurui_imu@163.com; csfeilong@imu.edu.cn).

Qintuya Si is with the College of Electronic and Information Engineering, Inner Mongolia University, Hohhot 010021, China (e-mail: siqty@imu.edu.cn).

Digital Object Identifier 10.1109/TASLP.2023.3244525

variance (LCMV) beamformer for the fully connected networks, but they gave the details on how to extend this approach for more general networks. However, we may find that the above distributed speech enhancement methods use all nodes in the WASN, which introduces costly overheads regarding transmission power. In fact, the data transmission among nodes will reduce the battery lifetime of sensor nodes. Moreover, a node with lower SNR (or sensitivity mismatch) may play a negative role in noise reduction tasks. Therefore, it is questionable whether using all nodes in the WASN is beneficial when taking all these factors into account.

Instead of using the entire network, selecting a subnetwork that is most informative for speech processing tasks [26], [27], [28] at hand can prolong the lifetime of the WASN. In general, however, the sensor selecting problem is computationally expensive due to its combinatorial nature. To this end, a sensor selection approach via convex relaxation was developed, which achieves a sub-optimal choice of a subnetwork from the WASN [29]. In [30], a minimum mean squared error (MMSE) signal estimation with sensor selection was exploited, where the link failure of nodes is also considered. Alternatively, a 0-1 knapsack problem was formulated for sensor selection in [31], which is then solved by a greedy searching strategy. Recently, Zhang et al. [32] proposed a subset selection scheme for the MVDR beamformer, by minimizing the transmission cost from nodes to the FC while constraining the output SNR. This approach achieves a better balance between the signal enhancement and the total energy consumption. However, these existing methods require a specific network structure in which an additional FC connects with all nodes. In such a manner, a powerful FC, as well as a large amount of data communications (between FC and nodes) are required. Although O'Connor et al. [33] implemented a sparse MVDR beamformer in a distributed manner, it was hard to put all the lower SNR nodes, which nearly capture none of the signal of interest, to sleep.

This paper proposes a novel sensor selection method for distributed speech enhancements with WASN, which is suitable for arbitrary network structures. An SNR-based Boolean optimization problem is first formulated, where a graph-related constraint is constructed to guarantee the connectivity of the selected subnetwork. Such a problem contains binary and high-order nonlinear optimization. Thus, it is linearized with multiple base-3 binary optimization problems, which are solved by a state-of-the-art steepest descent (SD) algorithm [34]. Next, a greedy searching scheme is presented to select nodes rapidly. Finally, to carry out the distributed sensor selection, a distributed SD algorithm is developed. Compared with the existing literatures [29], [30], [31], [32], the proposed method has no limitation on the structure of WASNs. Moreover, it works in a distributed processing manner, requiring only local communications among neighboring nodes and not needing an external central processor.

The contributions of this paper are as follows:1) A novel cost function is constructed for sensor selection, which can guarantee the connectivity of selected subnetworks; 2) The above complex cost function is linearized with multiple sub-optimization problems; 3) A state-of-the-art SD algorithm is employed to solve sub-optimization problems; 4) A greedy searching algorithm is presented to select sensors effectively; 5) The distributed SD algorithm is derived to carry out the distributed sensor selection.

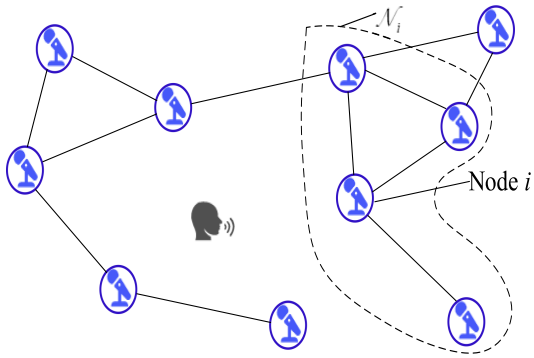

Fig. 1. Speech enhancement using microphone network with arbitrary graphs.

The paper is structured as follows. The reviews of speech enhancement with WASN and sensor selection are described in Section II. A binary programming for sensor selection as well as a greedy searching strategy are presented in Section III. Section IV introduces a novel SD algorithm, and then derives its distributed implementation. Numerical experiments are carried out in Section V, followed by conclusions in Section VI.

Notations: In this paper, we adopt lower cases for scalar, and denote vectors and matrices in bold lowercase letters and bold uppercase letters, respectively. Superscript T and H denote the transpose and conjugate transpose, respectively, and $\boldsymbol{u}_I$ denotes an $I \times 1$ vector with all elements one. The operator $[\cdot]_i$ denotes the ith element of a vector while $[\cdot]_{ij}$ denotes the (i, j)th element of a matrix.

II. Background

A. Distributed Microphone Networks and Signal Model

Consider a speech enhancement problem in a noisy and reverberant environment, where I microphone nodes constitute a spatially distributed microphone network, as depicted in Fig. 1. The communication topology of the network is modelled as an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, 2, \ldots, I\}$ is the vertex set, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set. Each vertex $i \in \mathcal{V}$ denotes a unique node, and each edge $(i, j) \in \mathcal{E}$ represents a communication link between nodes i and j. Namely, the neighbors of node i are defined as the subset $\mathcal{N}_i = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\} \cup \{i\}$, i.e., a node is a neighbor of itself certainly. Usually, each node shares data with its neighbors only.

Since speech enhancement is usually required to operate in real time, all received signals are processed in frames. Without loss of generality, we suppose that each node in the network contains a single microphone. Accordingly, the discrete Fourier transform (DFT) of the collected signal of node i at time frame l, say $y_i(k, l)$, for $i = 1, 2, \ldots, I$, is given by

$$y_i(k,l) = \underbrace{a_i(k)s(k,l)}_{x_i(k,l)} + v_i(k,l), \tag{1}$$

where k is the discrete-frequency index, $s(k,l)$ is the frequency response of the target signal, $a_i(k)$ is the acoustic transfer function (ATF) between the source location of interest and the ith microphone node, and $v_i(k,l)$ is the ambient noise at node i.

For notational convenience, both k and l will be omitted now onwards bearing in mind that the processing takes place in the DFT domain. Using vector notation, (1) can be compactly written as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{v}, \tag{2}$$

where $\boldsymbol{y} = [y_1, y_2, \ldots, y_I]^T$ is the stacked vector consisting of received signals from all nodes, $\boldsymbol{v} = [v_1, v_2, \ldots, v_I]^T$ is the vector form of noises, and $\boldsymbol{x} = \boldsymbol{a}s$ with $\boldsymbol{a} = [a_1, a_2, \ldots, a_I]^T$ indicates the steering vector, which can be obtained from the ATFs. Alternatively, $\boldsymbol{a}$ can also be inferred from the node and source positions, after utilizing suitable geometry calibration methods (e.g., [13]) and source localization methods (e.g., [35]).

Assume that the background noises are uncorrelated with the target signal. Then, the correlation matrix of the received signals is given by

$$\begin{aligned}\boldsymbol{R}_{yy} &= \mathbb{E}\{\boldsymbol{y}\boldsymbol{y}^H\} \\ &= \boldsymbol{R}_{xx} + \boldsymbol{R}_{vv} = \mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^H\} + \mathbb{E}\{\boldsymbol{v}\boldsymbol{v}^H\},\end{aligned} \tag{3}$$

where $\mathbb{E}\{\cdot\}$ denotes the statistical expectation operation. In general, $\boldsymbol{R}_{vv}$ can be estimated during the frames containing pure noises, while $\boldsymbol{R}_{xx}$ can be estimated by subtracting $\boldsymbol{R}_{vv}$ from $\boldsymbol{R}_{yy}$ during the speech-plus-noise frames. Similar to [32], we assume that a perfect voice activity detector (VAD) is available, which classifies the noise-only frames accurately.

B. Distributed Speech Enhancement Algorithms

Many different methods for centralized speech enhancement have been proposed, including spectral subtraction methods [36], [37], subspace methods [38], [39], beamformer-based methods [16], [17], etc. Recently, the distributed implementation of these methods is becoming increasingly popular in the sensor networks community. Because of space limitations, only some distributed beamforming methods are discussed here.

The well-known centralized MVDR beamformer is derived by minimizing the following cost function

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \boldsymbol{w}^H \boldsymbol{R}_{vv} \boldsymbol{w}, \quad \text{s.t.} \;\; \boldsymbol{w}^H \boldsymbol{a} = 1, \tag{4}$$

where $\boldsymbol{w}$ is the filter coefficients, and $\hat{\boldsymbol{w}}$ is the estimate of $\boldsymbol{w}$. An interpretation of (4) is that it minimizes the total output power while simultaneously keeping the gain of the source of interest fixed. Using the method of Lagrange multipliers, the solution of (4) is given by

$$\hat{\boldsymbol{w}} = \frac{\boldsymbol{R}_{vv}^{-1}\boldsymbol{a}}{\boldsymbol{a}^H \boldsymbol{R}_{vv}^{-1}\boldsymbol{a}}. \tag{5}$$

Nevertheless, due to the communication constraints among nodes (see Fig. 1), $\boldsymbol{R}_{vv}$ is difficult to obtain in each node directly, which makes the optimization process of (4) becomes more complicated than (5). To overcome this problem, a series of distributed schemes have been presented.

1) Distributed Sum-and-Delay Beamformer [20]: Zeng and Hendriks assume that v_i is uncorrelated with v_j $(i \neq j)$ when the distance among nodes are sufficiently large. In this case, $\boldsymbol{R}_{vv}$ becomes a diagonal matrix, i.e., $\boldsymbol{R}_{vv} = \mathrm{diag}\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_I^2\}$, where σ_i^2 denotes the power spectral density (PSD) of v_i. To this end, the output of MVDR beamformer is then simplified as

$$\hat{\boldsymbol{w}}^H \boldsymbol{y} = \frac{\sum_{i=1}^{I} (\sigma_i^2)^{-1} a_i^* y_i}{\sum_{i=1}^{I} \|a_i\|_2^2 (\sigma_i^2)^{-1}}, \tag{6}$$

where $\|\cdot\|_2$ is the L-2 norm. Define intermediate variables $\nu_i = (\sigma_i^2)^{-1} a_i y_i$ and $\nu_i' = \|a_i\|_2^2 (\sigma_i^2)^{-1}$, both of them can be computed independently in each node. By using gossip algorithms, (6) can be implemented in a distributed manner, i.e.,

$$\nu_i(t_g) = \sum_{j \in \mathcal{N}_i} \boldsymbol{U}_{ij} \nu_i(t_g - 1), \tag{7a}$$

$$\nu_i'(t_g) = \sum_{j \in \mathcal{N}_i} \boldsymbol{U}_{ij} \nu_i'(t_g - 1), \tag{7b}$$

$$\hat{\boldsymbol{w}}^H \boldsymbol{y} = \frac{\nu_i(T_g)}{\nu_i'(T_g)}, \tag{7c}$$

where t_g is the step index of the gossip iteration, T_g is the total iteration round, and $\boldsymbol{U}$ is an $I \times I$ dimensional weight matrix (details see [21]).

2) Diffusion-Based Distributed MVDR Beamformer [24]: A local cost function is first constructed in node i as

$$J_i(\boldsymbol{w}) = \boldsymbol{w}^H \boldsymbol{\Psi} \circ \boldsymbol{R}_{\mathcal{N}_i} \boldsymbol{w}, \tag{8}$$

where $\boldsymbol{R}_{\mathcal{N}_i}$ is the partial covariance matrix computed by the received signals y_j $(j \in \mathcal{N}_i)$ from neighbors of node i, $\circ$ is the Hadamard product, $\boldsymbol{\Psi}$ is a connection matrix that related to the network graph (see Sec. 2 in [24]). The authors show that the sum of these local cost functions is equivalent to the cost function in (4) with increasing the edge number of the network, i.e.,

$$J(\boldsymbol{w}) = \sum_{i=1}^{I} J_i(\boldsymbol{w}) \approx \boldsymbol{w}^H \boldsymbol{R}_{vv} \boldsymbol{w}. \tag{9}$$

Afterward, $J(\boldsymbol{w})$ is minimized in a distributed fashion by minimizing each local cost function $J_i(\boldsymbol{w})$, where the orthogonal projection method [40] is employed in each node to satisfy the unity-gain constraint in (4). This method can approximate the centralized MVDR beamformer when nodes have large transmit ranges.

On the whole, most of these distributed methods use all nodes in the network to show their superior performance. However, it is questionable whether using all nodes in the network is beneficial when comprehensively considering the energy usage and sensor's lifetime. Actually, only a few nodes with higher SNRs can provide significant contributions. Therefore, selecting a subset of nodes that is most informative for speech enhancement tasks at hand can reduce the transmission data and the energy usage.

C. Sensor Selection Problem

Sensor selection aims to determine the best subset of nodes to activate in order to minimize/maximize a cost function, and subject to some constraints, e.g., the number of activated nodes. Define a selection vector as

$$\boldsymbol{z} = [z_1, z_2, \ldots, z_I]^T \ , \tag{10}$$

where $z_i \in \{0, 1\}$, and $z_i = 1$ means that the ith node is activated, and vice versa.

In [29], the subset of nodes is selected by optimizing the following cost function

$$\begin{aligned} \hat{\boldsymbol{z}} &= \arg \max_{\boldsymbol{z} \in \{0,1\}^I} \log\det \left(\sum_{i=1}^{I} z_i \|a_i\|_2^2 \right) \\ \text{s.t.} \quad & \boldsymbol{u}^T \boldsymbol{z} = I_0 \ , \end{aligned} \tag{11}$$

where $I_0 (\leq I)$ is the number of activated nodes, and $\hat{\boldsymbol{z}}$ is the estimate of $\boldsymbol{z}$. By replacing the nonconvex constraints $z_i \in \{0, 1\}$ with the convex constraints $z_i \in [0, 1]$, problem (11) is solved by the interior-point methods [41].

Recently, Zhang et al. [32] consider the setup, where a fusion center (FC) is connected with all nodes using wireless links. Then, the corresponding cost function is formulated as

$$\begin{aligned} \hat{\boldsymbol{z}} &= \arg \min_{\boldsymbol{z} \in \{0,1\}^I} \|\mathrm{diag}(\boldsymbol{z})\boldsymbol{c}\|_1 \\ \text{s.t.} \quad & \boldsymbol{w}_{\boldsymbol{z}}^H \boldsymbol{R}_{vv,\boldsymbol{z}} \boldsymbol{w}_{\boldsymbol{z}} \leq \sigma_0 \\ & \boldsymbol{w}_{\boldsymbol{z}}^H \boldsymbol{a}_{\boldsymbol{z}} = 1 \ , \end{aligned} \tag{12}$$

where $\|\cdot\|_1$ denotes the L-1 norm, $\boldsymbol{c} = [c_1, c_2, \ldots, c_I]^T$ is the pairwise transmission cost between each node and the FC (i.e., c_i is related to the distance between FC and node i), σ_0 denotes the minimum output noise power after beamforming, $\boldsymbol{w}_{\boldsymbol{z}}$ and $\boldsymbol{a}_{\boldsymbol{z}}$ represent the filter coefficients and steering vector of selected nodes, respectively, and $\boldsymbol{R}_{vv,\boldsymbol{z}}$ is the correlation matrix of noises corresponding to selected nodes. Different from (11), a constraint for output noise power is added in (12), resulting in a variable number of activated nodes.

In summary, the existing sensor selection methods require an additional FC that connects all nodes. Next, all nodes transmit their received signals into FC to carry out the sensor selection scheme. In such a manner, a powerful FC as well as a large amount of data communication (between FC and nodes) are required. Furthermore, these methods may be invalid in distributed microphone networks with arbitrary graphs (e.g., the network in Fig. 1).

III. Sensor Selection for Distributed Microphone Networks

In this section, a target function is formulated to select $I_0 (\leq I)$ nodes with higher SNR. In addition, a graph-related constraint is added to maintain the connectivity among nodes in the selected subnetwork, which is then linearized for the base-3 subnetwork to reduce the complexity of the optimization problem for sensor selection. Finally, a greedy searching scheme is presented to carry out the sensor selection task effectively.

A. SNR-Based Target Function

The local SNR in node i can be calculated as

$$\mathrm{SNR}_i = \frac{x_i^H x_i}{v_i^H v_i} = \frac{[\boldsymbol{R}_{xx}]_{ii}}{\sigma_i^2} \ , \tag{13}$$

where σ_i^2 and $[\boldsymbol{R}_{xx}]_{ii}$ are the noise power and speech power of node i, respectively, where σ_i^2 can be estimated in node i during the noise frames and $[\boldsymbol{R}_{xx}]_{ii}$ can be obtained by subtracting σ_i^2 from $[\boldsymbol{R}_{yy}]_{ii}$.

Alternatively, if the source and node positions (or the ATFs from the source to nodes) are known, the SNR in (13) can be expressed with the steering vector as

$$\mathrm{SNR}_i = \frac{P_s a_i^H a_i}{\sigma_i^2} \propto \frac{a_i^H a_i}{\sigma_i^2} \ , \tag{14}$$

where P_s indicates the power of the sound source, which can be omitted in the relative SNR computation.

That is, using (13) or (14), SNR_i can be computed in node i independently. Then, a SNR-based target function can be formulated as

$$\begin{aligned} \max_{\boldsymbol{z}} f(\boldsymbol{z}) &= \sum_{i=1}^{I} z_i \mathrm{SNR}_i \\ \text{s.t.} \quad & \boldsymbol{u}^T \boldsymbol{z} = I_0 \\ & \boldsymbol{z} \in \{0, 1\}^I \ , \end{aligned} \tag{15}$$

where I_0 is the number of activated nodes. However, due to the communication graph in the network, the selection vector solved from (15) can not guarantee the connectivity among selected nodes. Once the selected subnetwork is disconnected, the follow-up distributed speech enhancement algorithms will become ineffective. In order to avoid this phenomenon, another constraint related to the network graph is constructed in the following subsection.

B. Graph Related Constraint

In graph theory [42], the network connectivity can be described by the adjacency matrix $\boldsymbol{A}$, which is defined as

$$\boldsymbol{A}_{ij} = \begin{cases} 1 & (i, j) \in \mathcal{E} \\ 0 & (i, j) \notin \mathcal{E} \end{cases} \ . \tag{16}$$

Some properties can be found from (16): 1) the adjacency matrix is a symmetric matrix; 2) diagonal elements of the adjacency matrix are zeros; 3) if the elements in the ith row are all zeros, node i is an isolated node, and the network is unconnected. Evidently, the 1-hop neighbors of node i can be observed from the non-zero elements in the ith row (or column) of $\boldsymbol{A}$. Namely, $\boldsymbol{A}_{ij}$ is the number of edges whose endpoints are nodes i and j.

Here, we assume that the network graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ (before sensor selection) is connected since there is an i, j-path for every pair $i, j \in \mathcal{V}$ of vertices, where the i, j-path includes one or more edges that connect nodes i and j. Thus, $\boldsymbol{A}_{ij}$ also represents the number of i, j-paths with length 1. According to [43], $\boldsymbol{A}_{ij}^q$ denotes the number of i, j-paths with length q.

Then, a summation matrix $\mathbf{\Gamma}$ is defined as

$$\mathbf{\Gamma} = \boldsymbol{E}_I + \boldsymbol{A} + \boldsymbol{A}^2 + \ldots + \boldsymbol{A}^{I-1} , \tag{17}$$

where $\boldsymbol{E}_I$ denotes the $I \times I$ identity matrix, and $\mathbf{\Gamma}_{ij}$ denotes the number of i,j-paths. Due to the fact that the original network $\mathcal{G}$ is connected, there exist at least one path between nodes i and j, i.e., $\mathbf{\Gamma}_{ij} > 0$.

The adjacency matrix $\boldsymbol{A}_z$ after sensor selection can be obtained by

$$\boldsymbol{A}_z = \boldsymbol{\Phi}_{\boldsymbol{z}} \boldsymbol{A} \boldsymbol{\Phi}_{\boldsymbol{z}}{}^T , \tag{18}$$

where $\boldsymbol{\Phi}_{\boldsymbol{z}} \in \{0,1\}^{I_0 \times I}$ is a submatrix of $\mathrm{diag}(\boldsymbol{z})$ after all zero rows have been removed. As a result, the $I_0 \times I_0$ summation matrix $\mathbf{\Gamma}_z$ after sensor selection can be constructed as

$$\mathbf{\Gamma}_z = \boldsymbol{E}_{I_0} + \boldsymbol{A}_z + \boldsymbol{A}_z{}^2 + \ldots + \boldsymbol{A}_z^{I-1}. \tag{19}$$

To ensure that the selected subnetwork is connected, i.e., at least one i,j-path is required for every pair $i,j \in \mathcal{V}$ of vertices, i.e.,

$$\mathbf{\Gamma}_{z,ij} \geq 1 \quad i,j = 1,2,\ldots,I_0 , \tag{20}$$

where $\mathbf{\Gamma}_{z,ij}$ is the (i,j)th element of $\mathbf{\Gamma}_z$. Once (20) is satisfied, the signs of the each element of $\mathbf{\Gamma}_z$ is equal to 1, resulting in

$$\mathrm{sign}(\mathbf{\Gamma}_z)\boldsymbol{u}_{I_0} = I_0 \boldsymbol{u}_{I_0}. \tag{21}$$

Combining (15) and (21), a cost function with graph related constraint can be constructed as

$$\begin{aligned} \max_{\boldsymbol{z}} \; & f(\boldsymbol{z}) = \boldsymbol{\beta}^T \boldsymbol{z} \\ \text{s.t.} \; & \boldsymbol{u}^T \boldsymbol{z} = I_0 \\ & \boldsymbol{z} \in \{0,1\}^I \\ & \mathrm{sign}(\mathbf{\Gamma}_z)\boldsymbol{u}_{I_0} = I_0 \boldsymbol{u}_{I_0} , \end{aligned} \tag{22}$$

where $\boldsymbol{\beta} = [\mathrm{SNR}_1, \mathrm{SNR}_2, \ldots, \mathrm{SNR}_I]^T$ is the vector that contains the SNRs of all nodes. However, such an optimization problem of sensor selection for speech enhancement is not trivial. The difficulties are described as follows.

1) A straightforward method for solving the problem (22) is to evaluate the performance for all $\binom{I}{I_0}$ choices for the sensor selection, but evidently this is not practical when I or I_0 is large.
2) $\mathrm{sign}(\mathbf{\Gamma}_z)\boldsymbol{u}_{I_0} = I_0 \boldsymbol{u}_{I_0}$ is a non-linear and non-convex constraint (especially for large I_0), it introduces an $(I_0 - 1)$-degree cost function. Although there exist some binary quadratic programming solvers [44], [45], these methods will be invalid when $I_0 > 3$.
3) The constraint $\mathrm{sign}(\mathbf{\Gamma}_z)\boldsymbol{u}_{I_0} = I_0 \boldsymbol{u}_{I_0}$ (or $\mathbf{\Gamma}_{z,ij} \geq 1$) strictly relies on the assumption that the matrix $\boldsymbol{A}_z$ is Boolean. Therefore, the typical convex relaxation strategy [29], [32], which replaces $\boldsymbol{z} \in \{0,1\}^I$ with the continuous variables $\boldsymbol{z} \in [0,1]^I$, is also unfeasible for solving (22).

To the best of our knowledge, there are no optimization methods to solve (22) effectively. To reduce the difficulty of problem solving, (22) is linearized in the next subsection when $I_0 \leq 3$, then solved by a greedy searching approach in the last subsection.

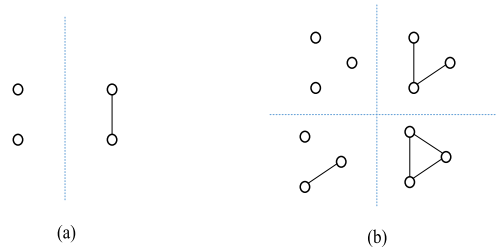

Fig. 2. Isomorphism classes when $I_0 \leq 3$: (a)$I_0 = 2$, (b)$I_0 = 3$.

C. Linearization With Base-3 Subnetwork

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ and $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$ be two graphs. An isomorphism of graphs $\mathcal{G}$ and $\mathcal{G}'$ is a bijection $\hbar : \mathcal{G} \to \mathcal{G}'$ between the vertex sets of $\mathcal{G}$ and $\mathcal{G}'$, such that any two vertices i and j of $\mathcal{G}$ are adjacent in $\mathcal{G}$ if and only if $\hbar(i)$ and $\hbar(j)$ are adjacent in $\mathcal{G}'$ [42]. According to this definition, any two isomorphic graphs have the same connectivity features in the sensor selection problem.

Usually, there exist different isomorphism classes for the same I_0. As depicted in Fig. 2, there are 2 and 4 isomorphism classes for graphs with $I_0 = 2$ and $I_0 = 3$ vertices, respectively. If $I_0 = 2$, (21) is equivalent with

$$\mathrm{sign}(\mathbf{\Gamma}_z)\boldsymbol{u}_{I_0} = \boldsymbol{E}_{I_0}\boldsymbol{u}_{I_0} + \boldsymbol{A}_z\boldsymbol{u}_{I_0} = I_0\boldsymbol{u}_{I_0}, \quad I_0 = 2. \tag{23}$$

By subtracting $\boldsymbol{u}_{I_0}$ from the two sides of (23), we have

$$\boldsymbol{A}_z\boldsymbol{u}_{I_0} = \boldsymbol{u}_{I_0}, \quad I_0 = 2 , \tag{24}$$

which is satisfied with the connected isomorphism class given on the right side of Fig. 2(a). An alternative interpretation of (24) is that each node should has one neighbor to guarantee the subnetwork's connectivity, and $[\boldsymbol{A}_z\boldsymbol{u}_{I_0}]_q$ denotes the number of neighbors corresponding to the qth selected node. For $I_0 = 3$, by observing the connected isomorphism classes shown on the right side of Fig. 2(b), we reasonably conclude that each selected node should have one or more neighbors, i.e.,

$$\boldsymbol{A}_z\boldsymbol{u}_{I_0} \geq \boldsymbol{u}_{I_0}, \quad I_0 = 3. \tag{25}$$

Substituting (18) in (25) and integrating (24) and (25), we get

$$\boldsymbol{\Phi}_{\boldsymbol{z}}\boldsymbol{A}\boldsymbol{\Phi}_{\boldsymbol{z}}{}^T\boldsymbol{u}_{I_0} \geq \boldsymbol{u}_{I_0}, \quad I_0 \leq 3 , \tag{26}$$

where $\boldsymbol{\Phi}_{\boldsymbol{z}}$ meets the relationship $\boldsymbol{\Phi}_{\boldsymbol{z}}{}^T\boldsymbol{\Phi}_{\boldsymbol{z}} = \mathrm{diag}(\boldsymbol{z})$ and $\boldsymbol{\Phi}_{\boldsymbol{z}}{}^T\boldsymbol{u}_{I_0} = \boldsymbol{z}$. By premultiplying the two sides of (26) with a Boolean matrix $\boldsymbol{\Phi}_{\boldsymbol{z}}{}^T$, we can esily get the following relationship

$$\mathrm{diag}(\boldsymbol{z})\boldsymbol{A}\boldsymbol{z} \geq \boldsymbol{z}, \quad I_0 \leq 3 , \tag{27}$$

where $\mathrm{diag}(\boldsymbol{z})\boldsymbol{A}$ clears the rows (to zero) corresponding to the unselected sensors. Because $\boldsymbol{z}_i$ is also equal to zero if the node i is not selected, (27) can be further simplified as

$$\boldsymbol{A}\boldsymbol{z} \geq \boldsymbol{z}, \quad I_0 \leq 3 , \tag{28}$$

which is a linear constraint of $\boldsymbol{z}$. Then, for $I_0 \leq 3$, (22) can be transformed as a linear Boolean optimization problem

$$\begin{aligned} \max_{\boldsymbol{z}} \; & f(\boldsymbol{z}) = \boldsymbol{\beta}^T \boldsymbol{z} \\ \text{s.t.} \; & \boldsymbol{u}^T \boldsymbol{z} = I_0 \\ & \boldsymbol{z} \in \{0,1\}^I \\ & \boldsymbol{A}\boldsymbol{z} \geq \boldsymbol{z}. \end{aligned} \tag{29}$$

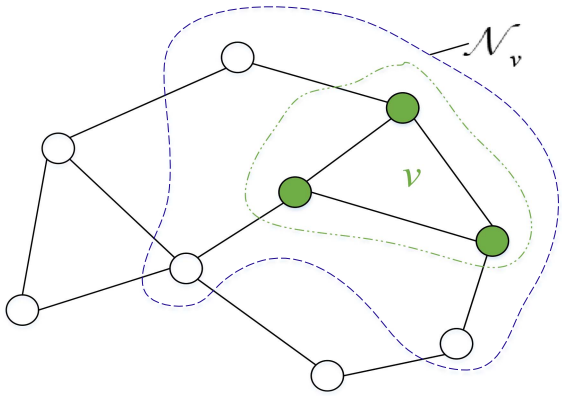

Fig. 3. Virtual node and its neighbors.

Still, $\boldsymbol{z}$ can not be relaxed using continuous variables $\boldsymbol{z} \in [0,1]^I$ in the solving process, since the derivation of (28) critically relies on the assumption that $\boldsymbol{z}$ is the Boolean variable. To this end, a dual convex optimization approach is employed to solve (29) in Section IV.

D. Greedy Searching

For $I_0 \geq 4$, enumerating all isomorphism classes of graphs is a complicated process. For example, there are 11, 34, and 156 isomorphism classes for graphs with $I_0 = 4$, $I_0 = 5$, and $I_0 = 6$ vertices, respectively. In these scenarios, the connectivity of the selected subnetwork can no longer be assured by the constraint in (28). Moreover, when $I_0 \geq 4$, it is hard to find a simple but effective linear constraint to ensure that the selected network is connected. For this reason, a base-3 greedy searching approach is presented to carry out the sensor selection.

As described in Algorithm 1, if $I_0 \leq 3$, the nodes are selected by solving (29). Otherwise, 3 nodes with the highest SNR are firstly selected by solving (29), in which the connectivity of the selected subnetwork can be guaranteed. Then, by merging the selected nodes, a virtual node v is built as

$$v_v = \{i_1, i_2, i_3\}\ , \tag{30}$$

where i_1, i_2 and i_3 denote the indexes of selected nodes in the last round (as shown in Fig. 3). The neighbors of this virtual node are defined as the following subset

$$\mathcal{N}_v = \{\mathcal{N}_{i_1} \cup \mathcal{N}_{i_2} \cup \mathcal{N}_{i_3}\}\ , \tag{31}$$

which contains all neighbors of the selected nodes (see Fig. 3). Besides, the SNR of this virtual node is defined by

$$\mathrm{SNR}_v = \mathrm{SNR}_{i_1} + \mathrm{SNR}_{i_2} + \mathrm{SNR}_{i_3}. \tag{32}$$

In the next round, the virtual node v and other unselected nodes constitute a network, from which the nodes are selected by establishing and solving a new cost function similar to (29). In this round, the adjacency matrix is updated as

$$\boldsymbol{A}' = \begin{bmatrix} \boldsymbol{\Phi}_{\mathcal{G}\backslash v}\boldsymbol{A}\Phi_{\mathcal{G}\backslash v}^T & \mathrm{sign}(\boldsymbol{u}_3^T\boldsymbol{\Phi}_v\boldsymbol{A}\Phi_{\mathcal{G}\backslash v}^T) \\ \mathrm{sign}(\boldsymbol{\Phi}_{\mathcal{G}\backslash v}\boldsymbol{A}^T\boldsymbol{\Phi}_v^T\boldsymbol{u}_3) & 0 \end{bmatrix}\ , \tag{33}$$

where $\boldsymbol{\Phi}_v \in \{0,1\}$ is a submatrix of the identity matrix after leaving only the rows corresponding to selected nodes, while $\boldsymbol{\Phi}_{\mathcal{G}\backslash v}$ is a submatrix of the identity matrix after removing the rows corresponding to selected nodes. In addition, the maximum node index is assigned to the virtual node, i.e., $v = |\mathcal{G}\backslash v| + 1$. The above steps are repeated until the I_0 nodes are selected.

Algorithm 1: Greedy Sensor Selecting Scheme.

if $I_0 \leq 3$.
  Selecting I_0 nodes by solving (29).
else
  Selecting first 3 nodes by solving (29).
  for $q = 1, 2, \ldots, \lfloor \frac{I_0-3}{2} \rfloor$ **do**
    I_0 is set to 3.
  Construct a virtual node v with (30) and (31).
  Compute the SNR of node v with (32).
  Update the the adjacency matrix with (33).
  Selecting nodes by solving (29) over the network contains the virtual and other unselected nodes.
  end for
  if $\mathrm{mod}(I_0 - 3, 2) = 1$
    repeat step 5 to step 9, but I_0 is set to 2.
  end if
end if

Remark 1: In the above sensor selecting procedure, the virtual node must be selected in the next round. The reason is given as follows. In the first round, according to (13), (29), and (32), the SNR of the virtual node exceeds any other base-3 subneworks. Thus, in the next several rounds, the subnetwork containing the virtual node has a higher SNR than other base-3 subnetworks.

Remark 2: Generally, the nodes with higher SNRs are distributed in the same spatial region. For instance, these nodes are mainly focused in the area near the sound source. Withal, a network's communication graph usually relies on each node's communication radius. That is, the nodes which are close in space will be more likely to communicate with each other. It means that, except for the virtual node, the nodes with higher SNRs are likely to connect with the virtual node. These features imply that the proposed greedy searching method can approximate the optimal solution for sensor selection.

IV. Dual Convex Optimization and Its Distributed Implementation

In this section, a dual convex optimization method is employed to solve (29) effectively, and then, it is extended to a distributed way suitable for distributed microphone networks.

A. Dual Convex Optimization

Before deriving the dual form of (29), it is transformed into a binary optimization problem with respect to $\boldsymbol{s} \in \{-1, 1\}^I$, where $\boldsymbol{s} = 2(\boldsymbol{z} - 0.5\boldsymbol{u}_I)$. Based on this, $\boldsymbol{z}$ can be expressed by $\boldsymbol{s}$ as

$$\boldsymbol{z} = \frac{\boldsymbol{s}}{2} + \frac{1}{2}\boldsymbol{u}_I. \tag{34}$$

Substituting (34) in (29), we get

$$\begin{aligned} \max_{\boldsymbol{s}} \; & f(\boldsymbol{s}) = \boldsymbol{\beta}^T(\tfrac{1}{2}\boldsymbol{s} + \tfrac{1}{2}\boldsymbol{u}_I) \\ \text{s.t.} \; & \boldsymbol{u}_I^T(\tfrac{1}{2}\boldsymbol{s} + \tfrac{1}{2}\boldsymbol{u}_I) = I_0 \\ & \boldsymbol{s} \in \{-1, 1\}^I \\ & \boldsymbol{A}(\tfrac{1}{2}\boldsymbol{s} + \tfrac{1}{2}\boldsymbol{u}) \geq (\tfrac{1}{2}\boldsymbol{s} + \tfrac{1}{2}\boldsymbol{u}) \end{aligned} \quad \propto \quad \begin{aligned} \max_{\boldsymbol{s}} \; & f(\boldsymbol{s}) = \boldsymbol{\beta}^T \boldsymbol{s} \\ \text{s.t.} \; & \boldsymbol{u}^T \boldsymbol{s} = 2I_0 - I \\ & \boldsymbol{s} \in \{-1, 1\}^I \\ & \dot{\boldsymbol{A}}\boldsymbol{s} \leq \dot{\boldsymbol{u}} \; , \end{aligned} \tag{35}$$

where $\dot{\boldsymbol{A}} = -(\boldsymbol{A} - \boldsymbol{E}_I)$ and $\dot{\boldsymbol{u}} = (\boldsymbol{A} - \boldsymbol{E}_I)\boldsymbol{u}$. By introducing the Lagrange multipliers L_{eq} and $\boldsymbol{L}_{ineq}$, the dual form of (35) can be given by

$$\begin{aligned} & \min_{L_{eq}, \boldsymbol{L}_{ineq} \leq \boldsymbol{0}} \; \max_{\boldsymbol{s} \in \{-1,1\}^I} \boldsymbol{s}^T \boldsymbol{\beta} + L_{eq}[\boldsymbol{u}^T \boldsymbol{s} - (2I_0 - I)] \\ & \quad + \boldsymbol{L}_{ineq}^T [\dot{\boldsymbol{A}}\boldsymbol{s} - \dot{\boldsymbol{u}}] \\ & = \min_{L_{eq}, \boldsymbol{L}_{ineq} \leq \boldsymbol{0}} \; \max_{\boldsymbol{s} \in \{-1,1\}^I} \boldsymbol{s}^T [\boldsymbol{\beta} + \boldsymbol{u} L_{eq} + \dot{\boldsymbol{A}}^T \boldsymbol{L}_{ineq}] \\ & \quad - [L_{eq}(2I_0 - I) + \boldsymbol{L}_{ineq}^T \dot{\boldsymbol{u}}]. \end{aligned} \tag{36}$$

After considering the maximization part of the problem, it can be simplified as

$$\begin{aligned} & \max_{\boldsymbol{s} \in \{-1,1\}^I} \boldsymbol{s}^T [\boldsymbol{\beta} + \boldsymbol{u} L_{eq} + \dot{\boldsymbol{A}}^T \boldsymbol{L}_{ineq}] \\ & \quad - [L_{eq}(2I_0 - I) + \boldsymbol{L}_{ineq}^T \dot{\boldsymbol{u}}] \\ & = \| \boldsymbol{\beta} + \boldsymbol{u} L_{eq} + \dot{\boldsymbol{A}}^T \boldsymbol{L}_{ineq} \|_1 - [L_{eq}(2I_0 - I) + \boldsymbol{L}_{ineq}^T \dot{\boldsymbol{u}}]. \end{aligned} \tag{37}$$

Therefore, the Lagrange multipliers can be solved by the following cost function

$$\begin{aligned} & \min_{L_{eq}, \boldsymbol{L}_{ineq} \leq \boldsymbol{0}} h(L_{eq}, \boldsymbol{L}_{ineq}) \\ & = \min_{L_{eq}, \boldsymbol{L}_{ineq} \leq \boldsymbol{0}} \{ \| \boldsymbol{\beta} + \boldsymbol{u} L_{eq} + \dot{\boldsymbol{A}}^T \boldsymbol{L}_{ineq} \|_1 \\ & \quad - [L_{eq}(2I_0 - I) + \boldsymbol{L}_{ineq}^T \dot{\boldsymbol{u}}] \} \; , \end{aligned} \tag{38}$$

which is obviously a convex optimization problem and the global optimal solutions of L_{eq} and $\boldsymbol{L}_{ineq}$ can be obtained. With the above dual problem, we can obtain the binary vector $\boldsymbol{s}$ by the following formula

$$\boldsymbol{s} = \text{sign}(\boldsymbol{\beta} + \boldsymbol{u} L_{eq} + \dot{\boldsymbol{A}}^T \boldsymbol{L}_{ineq}) \; , \tag{39}$$

which is derived from the fact that the optimal solution $\boldsymbol{s}$ of the problem $\arg\max_{\boldsymbol{s} \in \{-1,1\}^I} \boldsymbol{s}^T \boldsymbol{v}$ is $\boldsymbol{s} = \text{sign}(\boldsymbol{v})$, where $\boldsymbol{v} = [\boldsymbol{\beta} + \boldsymbol{u} L_{eq} + \dot{\boldsymbol{A}}^T \boldsymbol{L}_{ineq}]$.

B. Steepest Descent Algorithm

A state-of-the-art steepest descent (SD) algorithm [34] is employed to find the optimal solution of (38). In [34], the SD algorithm is proposed to obtain an intermediate solution of the binary quadratic programming problems, while it can be used here to obtain the optimal L_{eq} and $\boldsymbol{L}_{ineq}$ rapidly, from which $\boldsymbol{s}$ can be computed analytically based on (39).

The SD algorithm is summarized in Algorithm 2. In the initialization step, the time index t is introduced, then L_{eq}^t and $\boldsymbol{L}_{ineq}^t$ are set to zeros. During the iteration process, $\boldsymbol{s}$ is firstly updated with (39). Next, the negative gradient directions of the function $h(L_{eq}, \boldsymbol{L}_{ineq})$ with respect to L_{eq} and $\boldsymbol{L}_{ineq}$ are computed. Afterward, the optimal step size $\hat{\gamma}$ for the descent direction is obtained by solving the following optimization problem

$$\begin{aligned} \hat{\gamma} = & \arg\min_{\gamma} \phi(\gamma) \\ = & \arg\min_{\hat{\gamma}} h(L_{eq}^t + \gamma \mathrm{d}L_{eq}, \boldsymbol{L}_{ineq}^t + \hat{\gamma} \mathrm{d}\boldsymbol{L}_{ineq}) \; , \end{aligned} \tag{40}$$

where the details will be explained later. Finally, L_{eq} and $\boldsymbol{L}_{ineq}$ are updated in step 6. To satisfy the constraint $\boldsymbol{L}_{ineq} \leq \boldsymbol{0}$, the operations $\hat{\gamma} = \max(\hat{\gamma}, 0)$ and $\min(\hat{\gamma} \mathrm{d}\boldsymbol{L}_{ineq}, 0)$ are used here.

The solving process of (40) is given as follows. By substituting $L_{eq} = L_{eq}^{t-1} + \hat{\gamma} \mathrm{d}L_{eq}$ and $\boldsymbol{L}_{ineq} = \boldsymbol{L}_{ineq}^{t-1} + \hat{\gamma} \mathrm{d}\boldsymbol{L}_{ineq}$ to (38), (40) can be rewritten as

$$\hat{\gamma} = \arg\min_{\gamma} \phi(\gamma) = \arg\min_{\gamma} \|\tilde{\boldsymbol{q}} + \gamma \tilde{\boldsymbol{d}}\|_1 + \gamma \tilde{b} \; , \tag{41}$$

where $\tilde{\boldsymbol{q}}$, $\tilde{\boldsymbol{d}}$, and $\tilde{b}$ are given by

$$\tilde{\boldsymbol{q}} = \boldsymbol{\beta} + \boldsymbol{u} L_{eq}^{t-1} + \dot{\boldsymbol{A}}^T \boldsymbol{L}_{ineq}^{t-1} \; , \tag{42a}$$

$$\tilde{\boldsymbol{d}} = \boldsymbol{u} \mathrm{d}L_{eq} + \dot{\boldsymbol{A}}^T \mathrm{d}\boldsymbol{L}_{ineq} \; , \tag{42b}$$

$$\tilde{b} = -(2I_0 - I)\mathrm{d}L_{eq} - \mathrm{d}\boldsymbol{L}_{ineq}^T \dot{\boldsymbol{u}}. \tag{42c}$$

Obviously, (41) is also a convex problem, which is solved in Algorithm 3 (for the theoretical proof, see Theorem 1 in [34]). Specifically, define $\tilde{q}_i$ and $\tilde{d}_i$ as the ith elements of $\tilde{\boldsymbol{q}}$ and $\tilde{\boldsymbol{d}}$, respectively. Then, the pairs $(\tilde{q}_i, \tilde{d}_i)$ with zero $\tilde{d}_i$ are removed, and the remaining I' pairs are sorted in step 2. After that the index i_* is found by minimizing the equation $i_* = \arg\min_i |\tilde{b} - \sum_{i=1}^{I'} |\tilde{d}_i| + 2\sum_{i=i_*}^{I'} |\tilde{d}_i||$. Finally, the optimal step size is obtained by $\hat{\gamma} = \frac{\tilde{q}_{i_*}}{\tilde{d}_{i_*}}$.

Algorithm 2: Steepest Descent Algorithm.

1: Introduce time index t, and for $t = 0$, set $L_{eq}^t = 0$, $\boldsymbol{L}_{ineq}^t = \boldsymbol{0}$.
2: **for** $t = 1, 2, \ldots, T_0$ **do**
3: Compute $\boldsymbol{s} = \text{sign}(\boldsymbol{\beta} + \boldsymbol{u} L_{eq}^{t-1} + \dot{\boldsymbol{A}}^T \boldsymbol{L}_{ineq}^{t-1})$.
4: Compute $\mathrm{d}L_{eq} = (I - 2I_0) - \boldsymbol{u}^T \boldsymbol{s}$ and $\mathrm{d}\boldsymbol{L}_{ineq} = \dot{\boldsymbol{u}} - \dot{\boldsymbol{A}}\boldsymbol{s}$.
5: Compute $\hat{\gamma} = \arg\min_{\gamma} \phi(\gamma)$ and $\hat{\gamma} = \max(\hat{\gamma}, 0)$ (Details see Algorithm 3).
6: Compute $L_{eq}^t = L_{eq}^{t-1} + \hat{\gamma} \mathrm{d}L_{eq}$ and $\boldsymbol{L}_{ineq}^t = \boldsymbol{L}_{ineq}^{t-1} + \min(\hat{\gamma} \mathrm{d}\boldsymbol{L}_{ineq}, 0)$.
7: **end for**
8: Output $\boldsymbol{s}$, $L_{eq} = L_{eq}^{T_0}$ and $\boldsymbol{L}_{ineq} = \boldsymbol{L}_{ineq}^{T_0}$

Algorithm 3: Optimal Step Size Searching Approach.

1: Remove the pairs $(\tilde{q}_i, \tilde{d}_i)$ with zero $\tilde{d}_i$.
2: Sort the remaining sequence so that $\frac{\tilde{q}_1}{\tilde{d}_1} \leq \frac{\tilde{q}_2}{\tilde{d}_2} \leq \ldots \leq \frac{\tilde{q}_{I'}}{\tilde{d}_{I'}}$.
3: Solving problem $i_* = \arg\min_i |\tilde{b} - \sum_{i=1}^{I'} |\tilde{d}_i| + 2\sum_{i=i_*}^{I'} |\tilde{d}_i||$.
4: Output $\hat{\gamma} = \frac{\tilde{q}_{i_*}}{\tilde{d}_{i_*}}$.

C. Distributed Implementation

Note that the utilized SD algorithm is a centralized optimization method, thus, it can only be applied in distributed microphone networks in a centralized manner, which requires an external powerful central processor. Besides, all audio signals received by microphone nodes must be transmitted into the central processor, leading to a large communication burden. With consideration of the above problems, a distributed SD (DSD) method is presented in this subsection.

In the beginning, we assign the variables L_{eq}, s_i and $L_{ineq,i}$ to node i, where s_i and $L_{ineq,i}$ denote the ith elements of $\boldsymbol{s}$ and $\boldsymbol{L}_{ineq}$, respectively. To distinguish the same variable L_{eq} at different nodes, we use $L_{eq,i}$ to indicate the local form of L_{eq} at node i. Besides, define a weight matrix $\mathbf{W}$ that satisfies

$$\boldsymbol{u}_I^T\mathbf{W}=\boldsymbol{u}_I^T,\ \mathbf{W}\boldsymbol{u}_I=\boldsymbol{u}_I, \Upsilon(\mathbf{W}-\boldsymbol{u}_I\boldsymbol{u}_I^T/I)<1\ , \tag{43}$$

where $\Upsilon(\cdot)$ denotes the spectral radius of a matrix [46]. Typically, $\mathbf{W}_{i\kappa}=0$ when $\kappa\notin\mathcal{N}_i$.

Different from the step 3 in Algorithm 2, the elements of $\boldsymbol{s}$ are computed in parallel at different nodes. At the ith node, s_i is calculated as

$$s_i=\text{sign}\left(\beta_i+L_{eq,i}^{t-1}-\sum_{\kappa\in\mathcal{N}_i\backslash i}L_{ineq,\kappa}^{t-1}+L_{ineq,i}^{t-1}\right)\ , \tag{44}$$

where $\beta_i=\text{SNR}_i$ is the SNR of node i itself, which can be estimated in parallel at each node, and $\dot{\boldsymbol{A}}^T\boldsymbol{L}_{ineq}$ is alternatively implemented by summing the local variables corresponding the neighbors of node i. Subsequently, step 4 in Algorithm 2 can be carried out in parallel at each node as

$$\mathrm{d}L_{eq,i}=2I_0-I-\tilde{s}_i\ , \tag{45a}$$

$$\mathrm{d}L_{ineq,i}=\dot{u}_i+\sum_{\kappa\in\mathcal{N}_i\backslash i}s_\kappa-s_i\ , \tag{45b}$$

where $\dot{u}_i$ is a constant related to graph topology, $\tilde{s}_i$ is the summation of s_i $(i=1,2,\ldots,I)$, which can be estimated by the following consensus stages

$$s_i(m+1)=\mathbf{W}_{ii}s_i(m)+\sum_{\kappa\in\mathcal{N}_i}\mathbf{W}_{i\kappa}s_\kappa(m)\ , \tag{46a}$$

$$\tilde{s}_i=Is_i(M_0)\ , \tag{46b}$$

where m is the step index of the consensus iterations, and M_0 denotes the number of consensus rounds. Next, the step 5 in Algorithm 2 is implemented at each node after obtaining $\tilde{\boldsymbol{q}}$, $\tilde{\boldsymbol{d}}$, and $\tilde{b}$. Specifically, $\tilde{\boldsymbol{q}}$ is computed in a distributed manner as

$$\hat{\boldsymbol{q}}_i(m+1)=\mathbf{W}_{ii}\hat{\boldsymbol{q}}_i(m)+\sum_{\kappa\in\mathcal{N}_i}\mathbf{W}_{i\kappa}\hat{\boldsymbol{q}}_\kappa(m)\ , \tag{47a}$$

$$\tilde{\boldsymbol{q}}=I\hat{\boldsymbol{q}}_i(M_0)\ , \tag{47b}$$

where $\hat{\boldsymbol{q}}_i$ is a vector whose the ith element is s_i and the other elements are zeros. In addition, a local variable $\hat{d}_i$ at node i is calculated as

$$\hat{d}_i=\mathrm{d}L_{eq,i}-\xi_i\ , \tag{48}$$

Algorithm 4: Distributed Steepest Descent Algorithm.

1: Introduce time index t, and for $t=0$, set $L_{eq,i}^t=0$, $\boldsymbol{L}_{ineq,i}^t=\mathbf{0}$.
2: **for** $t=1,2,\ldots,T_0$, **at node** i, **do**
3: Compute s_i with (44).
4: Calculate $\tilde{s}_i$ with (46).
5: Compute $\mathrm{d}L_{eq,i}$ and $\mathrm{d}L_{ineq,i}$ with (45).
6: Calculate ξ_i with (49).
7: Calculate $\tilde{\xi}_i$ with (52).
8: Compute $\tilde{\boldsymbol{q}}$, $\tilde{\boldsymbol{d}}$, and $\tilde{b}$ with (47), (50), and (51).
9: Estimate $\hat{\gamma}$ with Algorithm 3.
10: Update $L_{eq,i}^t$ and $L_{ineq,i}^t$ with (53).
11: **end for**
12: Output s_i at node i, where $i=1,2,\ldots,I$.

where ξ_i is an intermediate variable, which can be computed from the variables corresponding to the neighbors of node i, i.e.,

$$\xi_i=\sum_{\kappa\in\mathcal{N}_i\backslash i}\mathrm{d}L_{ineq,\kappa}-\mathrm{d}L_{ineq,i}\ , \tag{49}$$

where $\mathrm{d}L_{ineq,i}$ is the gradient direction of $L_{ineq,i}$, which is previously obtained in (45b). Afterward, $\tilde{\boldsymbol{d}}$ is calculated in node i as

$$\hat{\boldsymbol{d}}_i(m+1)=\mathbf{W}_{ii}\hat{\boldsymbol{d}}_i(m)+\sum_{\kappa\in\mathcal{N}_i}\mathbf{W}_{i\kappa}\hat{\boldsymbol{d}}_\kappa(m)\ , \tag{50a}$$

$$\tilde{\boldsymbol{d}}=I\hat{\boldsymbol{d}}_i(M_0)\ , \tag{50b}$$

where $\hat{\boldsymbol{d}}_i$ is a vector whose the ith element is $\hat{d}_i$ and the other elements are zeros. Besides, $\tilde{b}$ is calculated at node i as

$$\tilde{b}=-(2I_0-I)\mathrm{d}L_{eq,i}-\tilde{\xi}_i\ , \tag{51}$$

where $\tilde{\xi}_i=\mathrm{d}\boldsymbol{L}_{ineq}^T\dot{\boldsymbol{u}}=\mathrm{d}\boldsymbol{L}_{ineq}^T(\boldsymbol{A}-\boldsymbol{E}_I)\boldsymbol{u}=\sum_{i=1}^I\xi_i$, which can be alternatively calculated by the average-consensus method, i.e.,

$$\xi_i(m+1)=\mathbf{W}_{ii}\xi_i(m)+\sum_{\kappa\in\mathcal{N}_i}\mathbf{W}_{i\kappa}\xi_\kappa(m)\ , \tag{52a}$$

$$\tilde{\xi}_i=I\xi_i(M_0). \tag{52b}$$

Finally, $L_{eq,i}^t$ and $L_{ineq,i}^t$ are updated at node i as

$$L_{eq,i}^t=L_{eq,i}^{t-1}+\hat{\gamma}\mathrm{d}L_{eq,i}\ , \tag{53a}$$

$$\boldsymbol{L}_{ineq,i}^t=\boldsymbol{L}_{ineq,i}^{t-1}+\min(\hat{\gamma}\mathrm{d}\boldsymbol{L}_{ineq,i},0). \tag{53b}$$

So far, the SD algorithm can be carried out in parallel at all microphone nodes. Once the distributed SD iterations are finished, the value of s_i at node i directly decides the selection result of node i, i.e., $s_i=1$ means that node i is selected. The proposed DSD method is summarized in Algorithm 4.

D. Analysis of Computational Complexity

For a given I_0, if $I_0\leq 3$, a single round of greedy searching is required; Otherwise, as given in Algorithm 1, $1+\lfloor\frac{I_0-3}{2}\rfloor$ rounds

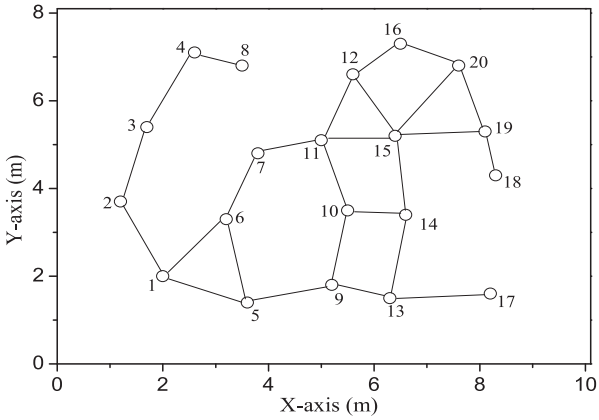

Fig. 4. Communication graph $\mathcal{G}$ of the distributed microphone network (circles represent nodes).

of greedy searching are needed. Noticeably, in each greedy iteration, the sensor selection is carried out in a distributed manner where all nodes perform similarly, thus the computational cost at node i is analyzed as follows.

From Algorithm 4, we can find that a greedy searching stage involves T_0 rounds of steepest descent iterations, in which the main computational cost is from equations (46), (47), (50), and (52). Specifically, (46) and (52) contain $O(|\mathcal{N}_i|M_0)$ multiplications, (47) and (50) contain $O(I|\mathcal{N}_i|M_0)$ multiplications. Therefore, each steepest descent iteration requires about $O(2I|\mathcal{N}_i|M_0 + 2|\mathcal{N}_i|M_0)$ multiplications. But it is worth noting that both $|\mathcal{N}_i|$ and I will decrease with the increasing number of greedy iterations due to the introduced virtual node in Fig. 3. Namely, if $I_0 \leq 3$, the computational load is about $O(2T_0(I+2)|\mathcal{N}_i|M_0)$; Otherwise, the corresponding computation load is lower than $O(2(1+\lfloor\frac{I_0-3}{2}\rfloor)T_0(I+2)|\mathcal{N}_i|M_0)$.

V. Experiments and Result Discussions

To evaluate the performance of the proposed method, some typical simulation experiments are carried out.

A. Simulation Setup

The simulated environment was a room of size 10 m $\times$ 6 m $\times$ 3 m, where $I = 20$ nodes were arranged. For simplicity, the microphone network was deployed on a plane at the height of 1 m, as depicted in Fig. 4. Fig. 4 also showed the considered communication graph of the network with a communication radius $\Re$ of 2 m. The RIRs between the source and the nodes were simulated with the well-known image source method [47]. The sound speed was set to $c_0 = 343$ m/s. The calibration signals were convolved with the above RIRs, and then added by different levels of Gaussian white noises, yielding the final noisy and reverberant audio signals. The sound signal was a continuous female speech utterance sampled at 16 kHz, where the frame length was equal to 1024 samples with 75% overlap.

TABLE I
Input SNRs for Different Nodes in Fig. 4

Node index	1	2	3	4	5	6	7
SNR (dB)	13.6	12.5	13.4	13.2	16.7	18.8	21.1
Node index	8	9	10	11	12	13	14
SNR (dB)	16.0	20.8	26.0	23.4	19.7	20.0	25.5
Node index	15	16	17	18	19	20	
SNR (dB)	24.2	17.3	17.6	21.4	21.0	17.7	

The weight matrix $\mathbf{W}$ in (43) was defined by a Metropolis scheme [48], i.e.,

$$W_{im} = \begin{cases} \frac{1}{I}, & m \in \mathcal{N}_i, m \neq i \\ 1 - \frac{(|\mathcal{N}_i|-1)}{I}, & m = i \\ 0, & m \notin \mathcal{N}_i \end{cases}, \tag{54}$$

and $M_0 = 100$ consensus iterations were executed. Besides, the iteration number T_0 of the DSD algorithm was set to 100. In the subsequent experiments, the target signals were obtained from the TIMIT database [49].

The VAD was implemented as follows. First, for node i, the power $p_i(l)$ of the lth frame was estimated. Then, the frame with minimum power was found, and its index was defined as l_i. Next, a threshold for node i was constructed as $\mu_i = q * p_i(l_i)$,where q was a constant, and it was set to $q = 1.24$; Afterward, if $p_i(l) < \mu_i$, the lth frame was detected as the noise-only frame for node i. Finally, the correlation matrix of noise was updated in the frame in which all nodes received noise only. Since the VAD is not the contribution of this paper, the above simple VAD method is employed, which may be applicable only for the specific dataset used here. In a more general case, the other complex VAD approaches [50], [51] should be carried out to further improve the performance.

The following simulations were carried out with Matlab(2017a) on an Intel CPU i7-11700F@2.50 GHz with 16 GB RAM. All the experimental results were averaged over 50 Monte Carlo trials under different noise realizations.

B. Simulation Results for Sensor Selection

First, an exhaustive searching (ES) method for (22) was employed to conduct the comparison experiment of sensor selection. Then, in the second experiment, the computational times of the considered algorithms were compared. Next, the rate of guaranteed connectivity was discussed in the third experiment.

1) Comparison of Sensor Selection Results: In this experiment, the ATFs were derived from the source and node positions, and the SNRs were computed with (14). The sensor selection results are shown in Fig. 5 for different numbers of activated nodes, i.e., $I_0 = [3, 4, 5, 6]$. To clearly present the above results, the input SNRs for different nodes are given in Table I. Evidently, the proposed method can ensure the connectivity of activated nodes in all situations. From Fig. 5(a), we can see that both the proposed and ES methods select nodes 10, 14, 15 when $I_0 = 3$, all of these chosen nodes are located near the source and have higher SNRs. This verifies that the linear constraint in (28) is effective and the proposed method can always find the global

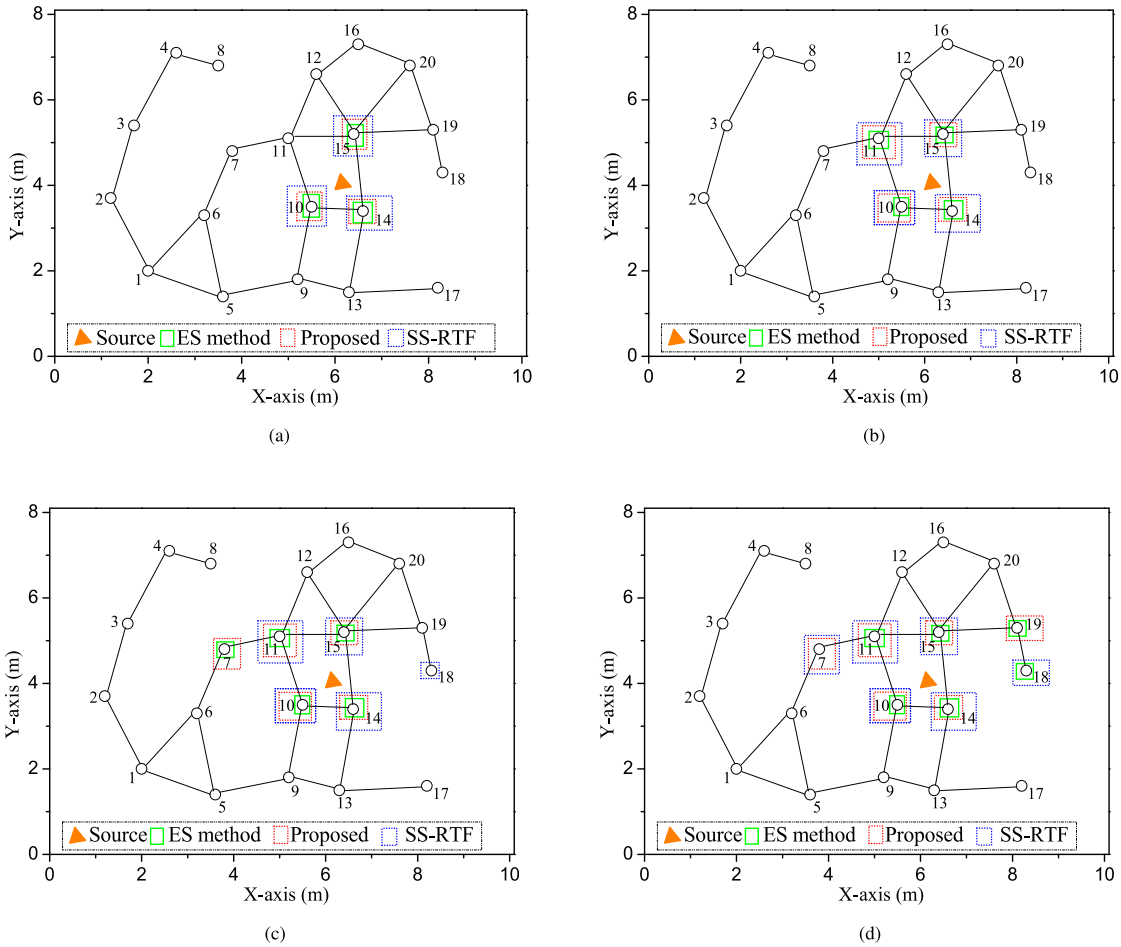

Fig. 5. Sensor selection results: (a) $I_0 = 3$, (b) $I_0 = 4$, (c) $I_0 = 5$, (d) $I_0 = 6$.

optimal solution when $I \leq 3$. Then in Fig. 5(b), based on the first round of sensor selection, the proposed method further chooses the virtual node (that contains nodes 10, 14, and 15) and node 11 through greedy searching. The ES method also selects nodes 10, 14, 15, and 11 after researching all possible solutions in the feasible region. Similarly, in Fig. 5(c), the proposed method selects nodes 11 and 7 in the second round of greedy searching, and the ES method selects nodes 10, 14, 15, 11, and 7 by re-searching all possible and feasible solutions. Fig. 5(b) and 5(c) indicate that the proposed method can approximate the ES method well. This is because the nodes with higher SNRs are located in the same region near the source, and the distance among them is less than the communication radius ($\Re = 2$ m in the considered network), which leads that the global optimal solution of (22) contains the virtual node and its neighbors. However, the different sensor selection results are produced when $I_0 = 6$, as given in Fig. 5(d). As can be readily observed, based on the first two rounds of sensor selection, the proposed method selects node 19 in the third greedy searching round, while the ES method selects nodes 10, 14, 15, 11, 18, and 19 by re-searching all possible and feasible solutions. The reason is analyzed as follows: the SNR of node 18 is higher than that of node 7, but it is not selected by the proposed method, due to the fact that node 18 is not a neighbor of the virtual node (that contains nodes 10, 14, 15, 11, and 7) in the third greedy searching round. Still, the SNRs of the two methods are very close to each other.

Furthermore, a state-of-the-art sensor selection method [52] named SS-RTF was also adopted here. From Fig. 5 we can see that the SS-RTF method tend to choose the nodes with the highest SNRs, but the connectivity of the selected subnetwork can not be guaranteed (e.g. when $I_0 = 5$ or $I_0 = 6$). This is because it considered the scenario where a fusion center connected all sensor nodes. In this manner, the sensor selection was carried out in the fusion center, and the connectivity among nodes was not required. While this paper considers a distributed sensor network without the fusion center, in which both the sensor selection and the speech enhancement are distributed. Therefore, once the selected subnetwork is not connected, the subsequent distributed speech enhancement algorithm will be unable to run. Similar to [52], most of the existing sensor selecting approaches for speech enhancement [29], [30], [31], [32] considered the centralized scenario and were not applicable to the subsequent distributed speech enhancements. For this reason, the comparison with these algorithms is not considered in the next experiments.

2) Comparison of Computation Times: The previous experiment settings were kept, and the computation times of the

TABLE II
COMPUTATIONAL COST COMPARISONS

Running time(s)	I_0=3	I_0=4	I_0=5	I_0=6
Proposed	0.053	**0.092**	**0.093**	**0.146**
ES	**0.042**	0.181	0.922	4.237

TABLE III
RATE OF GUARANTEED CONNECTIVITY

Connectivity	I_0=2	I_0=3	I_0=4	I_0=5	I_0=6	I_0=7
IP	100%	80%	60%	80%	60%	60%
DSD	100%	100%	100%	100%	100%	100%

proposed and ES methods were tested in this experiment. From Table II, we can see that the ES method runs faster only when $I_0 = 3$, and the advantage of the proposed method becomes more evident with the increase of I_0. For $I_0 = 3$, though, the ES method may not necessarily outperform the proposed method when the number I of nodes is larger than 20, since the number of feasible solutions will increase significantly as I increases. Moreover, the computational times for $I_0 = 4$ and $I_0 = 5$ are similar, about twice the running time for $I_0 = 3$. This is because both the scenarios with $I_0 = 4$ and $I_0 = 5$ implemented two rounds of greedy searching algorithms. The computational time for $I_0 = 6$ is 0.146 s, about triple the running time for $I_0 = 3$, since three rounds of greedy searching are required. Unlike the proposed method, the computational time of the ES method increases exponentially with I_0. Thus, it is pretty valuable to develop sensor selecting algorithms.

3) Rate of Guaranteed Connectivity: In this experiment, the proposed method was compared with the sensor selection by using the interior-point (IP) method to solve (29), after replacing the nonconvex constraints $z_i \in \{0, 1\}$ with the convex one, i.e., $z_i \in [0, 1]$. To obtain a more general conclusion, 5 sound sources with different positions were used. Table III shows that the proposed DSD can ensure the connectivity of the selected subnetwork in all I_0 conditions. In comparison, the sensor selection via the IP method reaches 100% rate of guaranteed connectivity only when $I_0 = 2$. Still, it satisfies the connectivity to a certain extent, due to the characteristics of the sound field and node's communication radius discussed in Remark 2. However, once the selected subnetwork is disconnected, the follow-up distributed speech enhancement algorithms will become ineffective. Therefore, the proposed DSD algorithm is a preferable way to choose sensors. Simultaneously, this experiment also verifies the effectiveness of the formulated graph-related constraint.

C. Simulation Results for Distributed Speech Enhancement With Sensor Selection

In the fourth experiment, the power consumption in speech enhancement with sensor selection was discussed. In the fifth experiment, the speech enhancement performance of [24] with sensor selection was studied for different I_0 and SNR conditions.

TABLE IV
POWER CONSUMPTIONS AND OUTPUT SNRS

I_0	5	10	15	20
P_c (mW)	$5P_a$+$23P_0$	$10P_a$+$56P_0$	$15P_a$+$100P_0$	$20P_a$+$135P_0$
oSNR (dB)	25.2	24.4	24.1	24.6

Then, the speech enhancement performance with sensor selection was further tested under different SNR estimation strategies and distributed beamformers. Finally, the speech quality was evaluated under different reverberation times (RT60 s).

1) Power Consumptions for Different I_0 Conditions: To evaluate the power consumption of the activated sensors, a realistic power consumption model [53] was employed. Specifically, the power consumption of the ith node was modeled as $P_i = P_a + P(d_i)$, where P_a was a constant power required to activate node i, and $P(d_i)$ represented the power consumption which was a function of the transmission range. Here, $P(d_i)$ was defined as $P(d_i) = P_0 \sum_{j\in\mathcal{N}_i, z_j=1} d_{ij}^2$, where P_0 was a constant and d_{ij} was the distance between nodes i and j. Therefore, the overall power consumption P_c was obtained as $P_c = \sum_{i,z_i=1} P_i$. Besides, the sensor selection was adopted for the D-MVDR beamformer [24]. The reverberation time RT60 was fixed to 300 ms, and the input SNR was defined by the mean value of the local SNRs corresponding to all nodes, which was fixed to 10 dB. As given in Table IV, the power consumption P_c increases rapidly with the rising number of activated nodes. Concretely speaking, when $I_0 = 5$, P_c reduces more than four times compared to the WASN without sensor selection (i.e., $I_0 = 20$). Moreover, the output SNR (oSNR) does not always increase with I_0. For example, the oSNR of $I_0 = 5$ is slightly higher than that of $I_0 = 20$. In conclusion, the number of activated nodes is not the larger the better, thus the sensor selection is a promising strategy for the speech enhancement using distributed WASNs.

2) Effect of I_0 for Different SNR Conditions: In this part, the proposed sensor selection method was employed for the typical D-MVDR beamformer, from which the impact of I_0 was studied on the speech enhancement performance for different ambient noises. Besides, we also used 5 sound sources with different positions when RT60 = 100 ms, and depicted the statistical average of these sources in Fig. 6. Since the proposed method allows $I_0 \geq 2$ scenarios, the single sensor selection was implemented by the ES method when $I_0 = 1$. To present the sensor selection results in more detail, the index of selected nodes under different source positions are given in Table V. By observing Fig. 4 and Table V simultaneously, we can see that the proposed method always selects the subnetwork closest to the given source. Moreover, the connectivity of the selected subnetwork can also be guaranteed, which is the necessary condition for the subsequent D-MVDR beamformer. As shown in Fig. 6, the distributed beamformer performs better when $I_0 \geq 2$ compared with $I_0 = 1$, and the oSNRs of $I_0 = 2$ were roughly 4 dB higher than that of $I_0 = 1$ when SNR $\leq$ 25 dB. Nevertheless, the rising rate of oSNR decreases when $I_0 \geq 3$, and the peaks did not always appear at the largest number of activated nodes, i.e., $I_0 = 6$. According to this phenomenon, it

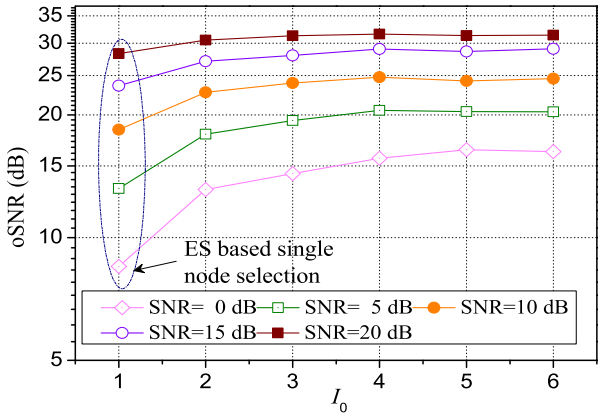

Fig. 6. Output SNR for different numbers of activated nodes and SNRs.

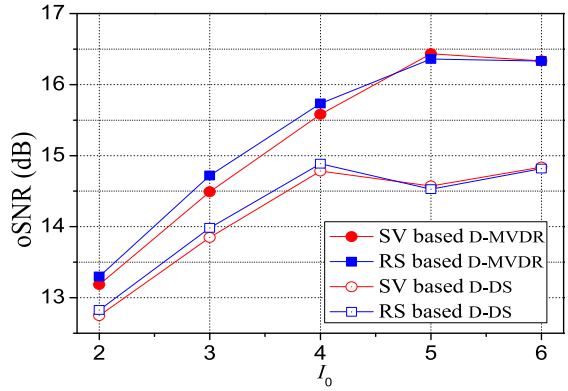

Fig. 7. Output SNR for different beamformers and for different SNR estimation strategies.

TABLE V
SENSOR SELECTION RESULTS FOR DIFFERENT SOURCE POSITIONS

Pos. \ I_0	2	4	6
(6.0,4.0) m	{10,14}	{10,14,15,11}	{10,14,15,11,7,19}
(6.5,5.0) m	{11,15}	{11,15,19,14}	{11,15,19,14,18,10}
(4.5,3.0) m	{6,7}	{6,7,11,10}	{6,7,11,9,10,5}
(2.0,5.0) m	{2,3}	{2,3,4,8}	{2,3,4,8,1,6}
(3.0,2.0) m	{1,5}	{1,5,6,7}	{1,5,6,7,9,2}

TABLE VI
PESQ AND STOI FOR D-MVDR BEAMFORMER UNDER DIFFERENT RT60 S

RT60 (ms)	100			300			500		
I_0	2	4	6	2	4	6	2	4	6
PESQ	2.90	2.83	2.74	2.58	2.52	2.49	2.30	2.29	2.23
STOI	0.90	0.89	0.89	0.84	0.84	0.83	0.79	0.78	0.77

can be inferred that the number of activated nodes is not the larger, the better. This is because the different nodes collect audio signals with different SNRs. If the nodes with poor speech quality are selected, the speech enhancement performance will be reduced. Therefore, when we use the microphone network to enhance speech signals, maybe a few nodes are required to be selected, which can obtain superior performance as well as a resource-saving framework. Based on this, the importance of sensor selection in microphone networks is further explained.

3) Effects of Different SNR Estimation Strategies and Different Distributed Beamformers: The previous experiment settings were kept, but the difference was that the input SNR was fixed to 0 dB. In this experiment, the SNRs were estimated based on the received signals (RS) with (13) and the steering vectors (SV) with (14), respectively. As shown in Fig. 7, the oSNRs corresponding to two different SNR estimation strategies are comparable (the difference is less than 0.1 dB). The reason is that the cost function (29) constructed with (13) or (14) always produces the same optimal solution. However, the RS-based strategy may be more suitable for sensor selection than the SV-based strategy, since the steering vector estimation needs a good RIR estimation method (or a perfect source localization approach) that increases the complexity of sensor selection. Besides, Fig. 7 also shows the speech enhancement performance of two distributed beamformers, namely D-DS [20] and D-MVDR [24], after sensor selection. We can see that the output SNRs of the D-MVDR method are always higher than that of the D-DS beamformer, since the assumption of a diagonal $\boldsymbol{R}_{vv}$ was relaxed in the D-MVDR method. But this is at the cost of more information transmission among nodes. A better algorithm still achieves a better performance after sensor selection through the proposed method. Actually, the proposed sensor selection method is completely independent of the follow-up speech enhancement approaches. Thus, it possesses a good versatility and can be adopted for any centralized or distributed speech enhancement approaches, including some future works.

4) Speech Quality Evaluation for D-MVDR Beamformer With Sensor Selection: In this experiment, the quality of enhanced speech was tested under different RT60 s, i.e., RT60 = {100, 300, 500} ms, when SNR = 15 dB. Here, the evaluation metrics were the perceptual evaluation of speech quality (PESQ) [54] and the short-time objective intelligibility (STOI) [55]. The scores of the PESQ metric range from -0.5 to 4.5, with higher scores indicating better speech quality. The scores of STOI range from 0 to 1, and a higher score indicates better speech intelligibility. To obtain the averaged results, 5 sound sources with different positions (as given in Table V) were used, where each source emitted 10 segments of speech signals. From Table VI we can see that both the PESQ and the STOI become lower as RT60 increases, indicating that the speech quality degrades as the reverberation time increases. Besides, both the PESQ and the STOI degrades with the raising of I_0, where the decline is relatively small. For instance, the PESQ decreases from 2.90 to 2.74 when I_0 is increased from 2 to 6, but it decreases from 2.90 to 2.30 when RT60 is increased from 100 ms to 500 ms. This is because the array aperture of selected nodes becomes larger as I_0 increases, which breaks down the spatial Nyquist sampling theorem gradually. In this case, the spatial aliasing is introduced and the speech fidelity is decreased

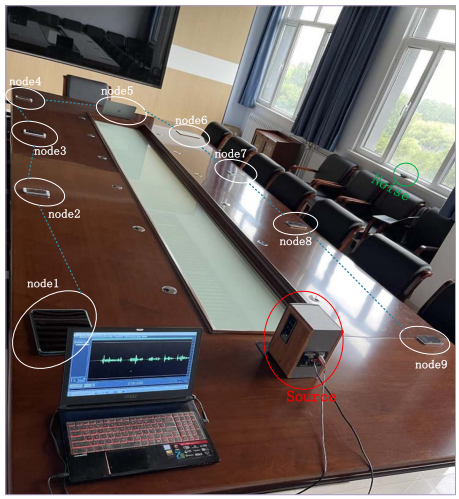

Fig. 8. Real Laboratory environment. The conference table was embedded with an acoustic sensor network consists of 9 nodes.

TABLE VII
LOCAL SNRS FOR DIFFERENT NODES

Node index	1	2	3	4	5
SNR (dB)	17.17	10.11	12.39	13.89	14.49
Node index	6	7	8	9	
SNR (dB)	10.34	24.52	9.96	14.83	

TABLE VIII
SPEECH ENHANCEMENT PERFORMANCE WITH SENSOR SELECTION

I_0	index of Selected nodes	oSNR (dB)
3	{5,6,7}	26.56
4	{5,6,7,4}	**27.87**
5	{5,6,7,4,8}	26.66
6	{5,6,7,4,8,9}	27.22
7	{ 5,6,7,4,8,3,9}	26.9

slightly. This phenomenon further proves the importance of sensor selection.

D. Real-World Experiment

To further investigate the validation of the proposed method, the real-world experiment is carried out.

The real-world environment was a conference room of size 9.5 m $\times$ 7.5 m $\times$ 3 m, where $I = 9$ mobile devices constituted a distributed WASN, as shown in Fig. 8. Besides, we also defined a communication graph for this WASN, which was depicted by the light blue dotted lines in Fig. 8. The source signal was the female speech with a length of 25 s, and it was emitted by a loudspeaker (Model: Edifier R1200II) on the conference table, while another loudspeaker (Model:DS-65VA300B) on the windowsill emitted white noise to simulate the interference outside the window. The reverberation time RT60 was about 260 ms, which was tested previously by measuring the 60 dB decay period of acoustic source energy. The sampling rate is 44.1 kHz, but these devices have individual oscillators and central processors, resulting in more or fewer differences among their sampling rates [56], [57], [58]. Therefore, a state-of-the-art sampling rate calibration method [59] was employed here to compensate for the sampling rate difference among nodes.

Generally, the node nearest to the target source yields a recording with the highest SNR. However, in our real-world scenario, such an argument may not be hold. The reason is given as follows: Each mobile device consists of multiple microphones, and for some nodes, the device manufacturers design and adopt specific far-field speech enhancement algorithms. Here, the local SNRs for 9 nodes are measured in Table VII. Obviously, node 7 has the highest SNR, although it is not the node closest to the source.

The speech enhancement performance with sensor selection is given in Table VIII, where the D-MVDR beamformer [24] was employed. We can see that all the selected subnetworks satisfy the network connectivity. Besides, the subnetwork with 4 activated nodes achieves the best performance. Moreover, it can save hardware or software resources compared to the subnetwork containing $I \geq 5$ nodes. This real-world experiment further demonstrates the importance of sensor selection.

VI. CONCLUSION

A distributed sensor selection approach is proposed for speech enhancement with WASNs. Specifically, the subnetwork is selected by maximizing the sum of local SNRs, where a non-convex non-linear constraint is added to satisfy the connectivity among activated nodes. After splitting the formulated cost function into multiple linear base-3 binary optimization problems, the steepest descent method is employed to solve them. To select the sensors effectively and rapidly, a greedy searching strategy is presented based on the base-3 virtual node. Finally, a distributed steepest descent algorithm is derived to further implement the decentralized sensor selection. The proposed method can successfully select the subnetwork with higher SNR nodes in noisy and reverberant environments. Moreover, it is suitable for the wireless acoustic sensor network with arbitrary communication graphs and can be adopted for any distributed speech enhancement tasks.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful remarks and constructive suggestions. Some MATLAB code for this paper is available at https://github.com/ttslr/DSS-SE.

REFERENCES

[1] S. Markovich-Golan, S. Gannot, and I. Cohen, "Distributed beamforming using the relative transfer function," in *Proc. Eur. Signal Process. Conf.*, 2012, pp. 27–31.
[2] A. I. Koutrouvelis, T. W. Sherson, R. Heusdens, and R. C. Hendriks, "Low-cost robust distributed linearly constrained beamformer for wireless acoustic sensor networks with arbitrary topology," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 8, pp. 1434–1448, Aug. 2018.

[3] M. Gogate, K. Dashtipour, A. Adeel, and A. Hussain, "CochleaNet: A robust language-independent audio-visual model for real-time speech enhancement," *Inf. Fusion*, vol. 64, pp. 273–285, Nov. 2020.

[4] D. C. Moore and I. A. McCowan, "Microphone array speech recognition: Experiments on overlapping speech in meetings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2003, pp. V497–V500.

[5] P. Aarabi and S. Zaky, "Robust sound localization using multi-source audiovisual information fusion," *Inf. Fusion*, vol. 2, no. 3, pp. 209–223, Sep. 2001.

[6] Q. Zhang, Z. Chen, and F. Yin, "Distributed marginalized auxiliary particle filter for speaker tracking in distributed microphone networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 11, pp. 1921–1934, Nov. 2016.

[7] Q. Zhang, W. Xu, W. Z., J. Feng, and Z. Chen, "Multi-hypothesis square-root cubature Kalman particle filter for speaker tracking in noisy and reverberant environments," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1183–1197, 2020.

[8] Q. Wang, S. Guo, and K. F. C. Yiu, "Distributed acoustic beamforming with blockchain protection," *IEEE Trans. Ind. Inform.*, vol. 16, no. 11, pp. 7126–7135, Nov. 2020.

[9] J. Zhang, A. I. Koutrouvelis, R. Heusdens, and R. C. Hendriks, "Distributed rate-constrained beamforming," *IEEE Signal Process. Lett.*, vol. 26, no. 5, pp. 675–679, May 2019.

[10] R. V. Rompaey and M. Moonen, "Distributed adaptive signal estimation in wireless sensor networks with partial prior knowledge of the desired sources steering matrix," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 7, pp. 478–492, 2021.

[11] R. Wang, Z. Chen, and F. Yin, "Active sampling rate calibration method for acoustic sensor networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 65, pp. 3095–3107, 2020.

[12] A. Plinge, F. Jacob, R. Haeb-Umbach, and G. A. Fink, "Acoustic microphone geometry calibration: An overview and experimental evaluation of state-of-the-art algorithms," *IEEE Signal Process. Mag.*, vol. 33, no. 4, pp. 14–29, Jul. 2016.

[13] D. Hu, Z. Chen, and F. Yin, "Passive geometry calibration for microphone arrays based on distributed damped Newton optimization," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 118–131, 2021.

[14] D. Hu, Z. Chen, and F. Yin, "Geometry calibration for acoustic transceiver networks based on network newton distributed optimization," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1023–1032, 2021.

[15] D. Hu, Z. Chen, and F. Yin, "Analytical geometry calibration for acoustic transceiver arrays," *IEEE Signal Process. Lett.*, vol. 27, pp. 1979–1983, 2020.

[16] J. Zhang, R. Heusdens, and R. C. Hendriks, "Rate-distributed spatial filtering based noise reduction in wireless acoustic sensor networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 11, pp. 2015–2026, Nov. 2018.

[17] J. Zhang, H. Chen, L. R. Dai, and R. C. Hendriks, "A study on reference microphone selection for multi-microphone speech enhancement," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 671–683, 2021.

[18] A. Hassani, J. Plata-Chaves, M. H. Bahari, M. Moonen, and A. Bertrand, "Multi-task wireless sensor network for joint distributed node-specific signal enhancement, beamforming and estimation," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 518–533, Apr. 2017.

[19] A. Hassani, A. Bertrand, and M. Moonen, "Distributed node-specific direction-of-arrival estimation in wireless acoustic sensor networks," in *Proc. IEEE 21st Eur. Signal Process. Conf.*, 2013, pp. 1–5.

[20] Y. Zeng and R. C. Hendriks, "Distributed delay and sum beamformer for speech enhancement in wireless sensor networks via randomized gossip," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*., 2012, pp. 4037–4040.

[21] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[22] Y. Zeng and R. C. Hendriks, "Distributed delay and sum beamformer for speech enhancement via randomized gossip," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 1, pp. 260–273, Jan. 2014.

[23] R. Heusdens, G. Zhang, R.C. Hendriks, Y. Zeng, and W. B. Kleijn, "Distributed MVDR beamforming for (wireless) microphone networks using message passing," in *Proc. Int. Workshop Acoust. Signal Enhancement*, 2012, pp. 1–4.

[24] M. O'Connor and W. B. Kleijn, "Diffusion-based distributed beamformer," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 810–814.

[25] A. Bertrand and M. Moonen, "Distributed beamforming in a wireless sensor network with single-channel per-node signal transmission," *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3447–3459, Jul. 2013.

[26] K. Yamada, Y. Saito, T. Nonomura, and K. Asai, "Greedy sensor selection for weighted linear least squares estimation under correlated noise," *IEEE Access*, vol. 10, pp. 79356–79356, 2022.

[27] M. Courcoux-Caro, C. Vanwynsberghe, C. Herzet, and A. Baussard, "Sequential sensor selection for the localization of acoustic sources by sparse Bayesian learning," *J. Acoust. Soc. Amer.*, vol. 152, pp. 1695–1708, 2022.

[28] C. Liu, K. Di, T. Li, and V. Elvira, "A sensor selection approach to maneuvering target tracking based on trajectory function of time," *EURASIP J. Adv. Signal Process.*, vol. 1, pp. 1–14, 2022.

[29] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[30] A. Bertrand and M. Moonen, "Efficient sensor subset selection and link failure response for linear MMSE signal estimation in wireless sensor networks," in *Proc. IEEE 18th Eur. Signal Process. Conf.*, 2010, pp. 1092–1096.

[31] J. Szurley, A. Bertrand, M. Moonen, P. Ruckebusch, and I. Moerman, "Energy aware greedy subset selection for speech enhancement in wireless acoustic sensor networks," in *Proc. IEEE 20th Eur. Signal Process. Conf.*, 2012, pp. 789–793.

[32] J. Zhang, S. P. Chepuri, R. C. Hendriks, and R. Heusdens, "Microphone subset selection for beamformer based noise reduction," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 3, pp. 550–563, Mar. 2018.

[33] M. O'Connor, W. B. Kleijn, and T. Abhayapala, "Distributed sparse beamforming using the bi-alternating direction method of multipliers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 106–110.

[34] B. S. Y. Lam and A. W. C. Liew, "A fast binary quadratic programming solver based on stochastic neighborhood search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 32–49, Jan. 2022.

[35] A. Brendel and W. Kellermann, "Distributed source localization in acoustic sensor networks using the coherent-to-diffuse power ratio," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 1, pp. 61–75, Mar. 2019.

[36] M. T. Islam, C. Shahnaz, and S. A. Fattah, "Speech enhancement based on a modified spectral subtraction method," in *Proc. IEEE 57th Int. Midwest Symp. Circuits Syst.*, 2014, pp. 1085–1088.

[37] B. Kirubagari, S. Palanivel, and N. Subathra, "Speech enhancement using minimum mean square error filter and spectral subtraction filter," in *Proc. IEEE Int. Conf. Inf. Commun. Embedded Syst.*, 2014, pp. 1–7.

[38] S. Gazor and A. Rezayee, "An adaptive approach for speech enhancement," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 2, pp. 87–95, Feb. 2001.

[39] J. Chensheng, Z. Xueying, and J. Hairong, "A speech enhancement method based on signal subspace and hearing masking effect," in *Proc. IEEE Int. Forum Comput. Sci.- Technol. Appl.*, 2009, pp. 15–18.

[40] X. Zhao, J. Chen, and A. H. Sayed, "Beam coordination via diffusion adaptation over array networks," in *Proc. IEEE 13th Int. Workshop Signal Process. Adv. Wireless Commun.*, 2012, pp. 105-109.

[41] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press., 2004.

[42] R. Diestel, *Graph Theory*. New York, NY, USA: Springer, 2000.

[43] D. Wang, F. Liu, H. Peng, and L. Wang, "Research on the algorithm of connectivity analysis for power system based on spectral graph theory," in *Proc. IEEE 24th Chin. Control Decis. Conf.*, 2012, pp. 19–22.

[44] G. R. Mauri and L. Lorena, "Lagrangean decompositions for the unconstrained binary quadratic programming problem," *Int. Trans. Oper. Res.*, vol. 18, pp. 257–270, 2011.

[45] C. Olsson, A.P. Eriksson, and F. Kahl, "Solving large scale binary quadratic problems: Spectral methods versus semidefinite programming," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[46] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. IEEE 4th Int. Symp. Inf. Process. Sensor Netw.*, 2005, pp. 63–70.

[47] E. A. Lehmann, A. M. Johansson, and S. Nordholm, "Reverberation-time prediction method for room impulse responses simulated with the image-source model," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2007, pp. 159–162.

[48] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Proc. IEEE 42nd Int. Conf. Decis. Control*, 2003, pp. 4997–5002.

[49] J. S. Garofolo, "Acoustic-phonetic speech database," *Nat. Inst. Standards Thechnol.*, vol. 15, pp. 29–50, 1988.

[50] X. L. Zhang and D. Wang, "Boosting contextual information for deep neural network based voice activity detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 2, pp. 252–264, Feb. 2016.

[51] H. Dinkel, S. Wang, X. Xu, M. Wu, and K. Yu, "Voice activity detection in the wild: A data-driven approach using teacher-student training," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1542–1555, 2021.
[52] J. Zhang, J. Du, and L. R. Dai, "Sensor selection for relative acoustic transfer function steered linearly-constrained beamformers," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1220–1232, 2021.
[53] Q. Wang, M. Hempstead, and W. Yang, "A realistic power consumption model for wireless sensor network devices," in *Proc. IEEE 3rd Annu. Commun. Soc. Sensor Ad Hoc Commun. Netw.*, 2006, pp. 286–295.
[54] A. W. Rix, J. G. Beerends, M.P. Hollier, and A. P. Hekstra, "Perceptual evaluation ofspeech quality (PESQ) - A new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, pp. 749–752.
[55] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2125–2136, Sep. 2011.
[56] A. Chinaev, P. Thüne, and G. Enzner, "Double-cross-correlation processing for blind sampling-rate and time-offset estimation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1881–1896, 2021.
[57] A. Chinaev and G. Enzner, "Distributed synchronization for ad-hoc acoustic sensor networks using closed-loop double-cross-correlation processing," in *Proc. IEEE Int. Workshop Acoust. Signal Enhancement*, 2022, pp. 1–5.
[58] D. Hu, H. Zhang, F. Bao, and R. Wang, "Distributed sampling rate offset estimation over acoustic sensor networks based on asynchronous network newton optimization," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 301–312, 2023.
[59] R. Wang, Z. Chen, and F. Yin, "Active sampling rate calibration method for acoustic sensor networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 3095–3107, 2020.

De Hu (Member, IEEE) was born in Inner Mongolia Autonomous Region, China, in 1993. He received the B.E. degree in communication engineering from Liaoning University, Shenyang, China, in 2014, and the Ph.D. degree in signal and information processing from the Dalian University of Technology, Dalian, China, in 2021. He is currently a Professor with the Department of Computer Science, Inner Mongolia University, Hohhot, China. His research interests include speech processing, sound localization, microphone arrays, acoustic SLAM, and natural language processing.

Qintuya Si received the B.E. degree from Liaoning University, Shenyang, China, in 2016, and the Ph.D. degree from the School of Information and Communication Engineering, Dalian University of Technology, Dalian, China, in 2022. She is currently a Lecturer with the School of Electronic Information Engineering, Inner Mongolia University, Hohhot, China. Her research interests include MIMO, spatial modulation, wireless communication, and signal processing.

Rui Liu (Member, IEEE) received the bachelor's degree from the Taiyuan University of Technology, Taiyuan, China, in 2014, and the Ph.D. degree from Inner Mongolia University, Hohhot, China, in 2020. He is currently a Professor with the National and Local Joint Engineering Research Center of Mongolian Intelligent Information Processing, Inner Mongolia University. From 2019 to 2020, he has been an exchange Ph.D. candidate with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, funded by China Scholarship Council. From 2020 to 2022, he was a Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore. He has authored or coauthored more than 20 papers in top-tier NLP/ML/AI conferences and journals, including IEEE/ACM-TASLP, Neural Networks, ICASSP, COLING, and INTERSPEECH. His research interests include audio, speech, and natural language processing, which include expressive Text-to-Speech, expressive voice conversion, speech emotion recognition, prosody structure prediction, grapheme-to-phoneme conversion, and syntax parsing. He was the recipient of the Best Paper Award at the 2021 International Conference on Asian Language Processing. He is a Member of ISCA and CCF, and the Reviewer for many major referred journal and conference papers.

Feilong Bao received the Ph.D. degree in computer application technology from Inner Mongolia University, Hohhot, China, in 2013. He is currently a Professor with the Department of Computer Science, Inner Mongolia University. His research interests include speech signal processing, natural language processing, speech synthesis, speech recognition, and neural machine translation.