Gao *et al.*
EURASIP Journal on Advances in Signal Processing (2023) 2023:33
https://doi.org/10.1186/s13634-023-00995-3

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Optimal deployment of large-scale wireless sensor networks based on graph clustering and matrix factorization

Hefei Gao, Qianwen Zhu and Wei Wang*

*Correspondence: weiwang@tjnu.edu.cn

Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, West Binshui Road, Tianjin 300387, China

Abstract

It is undeniable that there are a large number of redundant nodes in a wireless sensor network. These redundant nodes cause a colossal waste of resources and seriously threaten the life of the sensor network. In this paper, we provide a sensor nodes optimization selection algorithm based on a graph for a large-scale wireless sensor network. Firstly, we propose a representation-clustering joint algorithm based on Graph Neural Network to partition the large-scale graph into several subgraphs. Then, we use Singular-Value-QR Decomposition for the node selection of each subgraph and achieve the optimal deployment for a large-scale wireless sensor network. We conduct the experiments on the CIMIS dataset. The results show that the mean square error between the reconstructed network and the original network is as low as 0.02433. Meanwhile, we also compare our algorithm with the classical optimization algorithm. The results imply that the mean square error of the proposed algorithm is lower and the distribution is more uniform. Further, we verify the scalability of the algorithm for the optimal deployment of the large-scale wireless sensor network.

Keywords: Large-scale wireless sensor network, Deep embedding clustering, Graph attention auto encoder, Singular-value-QR decomposition

1 Introduction

A large-scale wireless sensor network (LWSN) is a wireless network composed of a mass of sensors and is used to sense, collect, process, and transmit the information of the monitoring area [1]. An LWSN is characterized by a large number, wide distribution, dense deployment, frequent communication, and big data traffic [2, 3], which leads to huge communication loss among nodes, high energy consumption, bandwidth resource waste, and data redundancy [4]. It seriously affects the network life and costs expensive maintenance. Therefore, it is very necessary to optimize the deployment of nodes for LWSN.

In recent years, there are much research on the optimal deployment of the wireless sensor network(WSN) with an intelligence optimization algorithm, such as the bat algorithm [5], the improved whale algorithm [6], the virtual spring force algorithm [7] and the improved grey wolf optimization algorithm [8], et al.. For the optimal deployment of

LWSN, Zhang et al. proposed an optimization strategy of LWSN based on matrix completion. Representative nodes were selected by defining the data information of nodes, and matrix completion was used to further reduce redundancy [9]. Jiang et al. proposed a mean filtering algorithm based on graphical node data to reduce the energy consumption and the data redundancy of LWSN [10]. For a network with a large number of device nodes, Bin Cao et al. divided the original high-dimensional multi-objective optimization problem into several low-dimensional multi-objective optimizations and used a group-based multi-objective evolutionary algorithm to close some nodes for saving resources [11].

As a common non-Euclidean data structure, the graph has the characteristics of diversity, irregularity, and large scale, which can efficiently and conveniently represent LWSN. In this paper, we proposed an optimal deployment algorithm in LWSN based on graph clustering and matrix decomposition to further optimize the sensor network randomly deployed in the monitoring area. Our contributions can be summarized as follows:

(1)We improve Deep Embedding Clustering with Graph Embedded Representation. It can learn the node's embedded representation and the graph clustering jointly to enhance the learning of embedding and clustering at the same time.

(2)We adopt Singular-Value-QR Decomposition to select a subset of optimized nodes on each sub-network, which achieves the optimal deployment and retains the crucial information of the LWSN.

The rest of this paper is organized as follows: Section 2 introduces the related work on the optimal deployment of sensor networks in recent years. Section 3 systematically expounds on our proposed theory and algorithm. Section 4 describes the experimental verification. Section 5 summarizes the work of this paper and suggests future research directions.

2 Related work

Some scholars use graphs as the medium and apply graph signal processing theory to the optimal deployment of sensor networks. Akie et al. proposed a deterministic graph signal sampling method, which used local operators implemented by Chebyshev polynomials to limit the coverage and select the optimal vertex subset based on the largest information [12]. On the basis of Akie's research, Huan Xu considered the maximum information and residual energy of nodes and proposed a sampling set selection based on energy-aware to achieve the balance of network optimization and energy constraints [13]. Paolo et al. introduced a probability sampling mechanism based on the adaptive least mean square algorithm so that each node of the graph was sampled with a given probability at each moment to obtain the optimal nodes [14]. Diego applied compressive sensing theory to graph signals and proposed an algorithm with local sampling and compressive sensing. This method calculated the signal coefficients random linear combination of signal coefficients between the node and its neighborhood and selected the optimal subset by comparing the coefficients [15].

These node optimization deployment methods are analyzed and processed on the entire graph. However, with the continuous increase in the number of sensor nodes, the execution efficiency of the algorithm also decreases sharply.

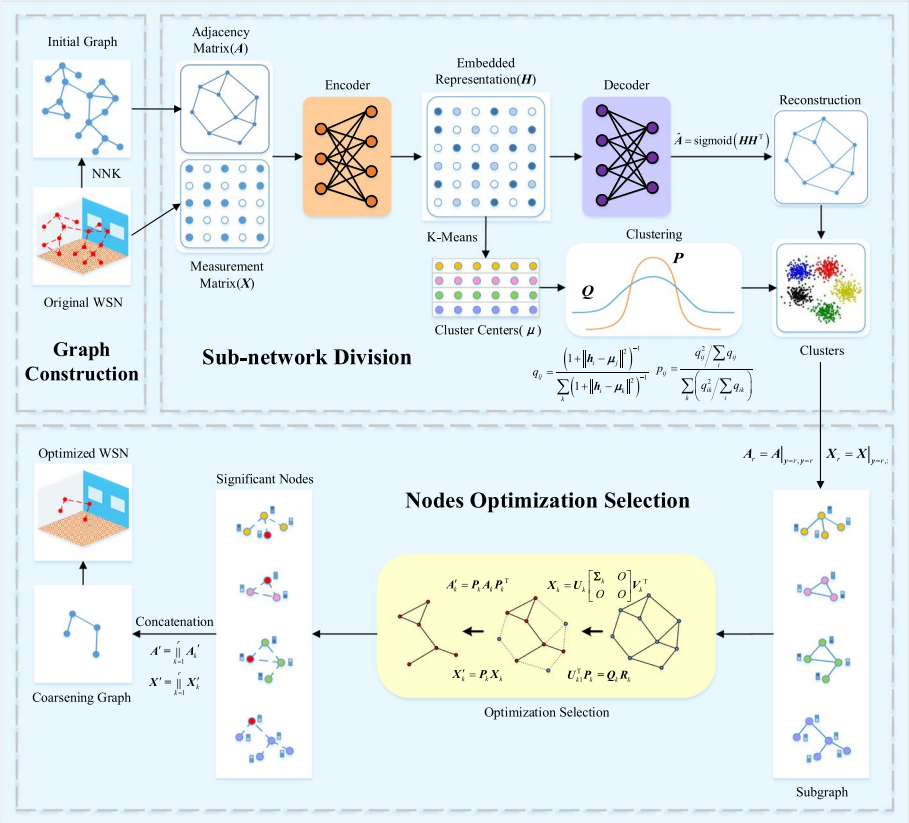

Fig. 1 An illustration of our algorithm architecture

Some research is devoted to identifying crucial nodes in the graph based on the Graph Neural Network (GNN). Zhao G et al. proposed a method that transforms the crucial node identification in the graph into node classification, learns the representation of nodes through a graph convolution network, and then classifies influential nodes and nodes with less influence with a multi-layer perception [16]. Ma M et al. proposed a crucial node identification model based on graph attention networks and reinforcement learning, where a graph attention network is used to obtain the embedded representation of each node, and combined with reinforcement learning, the node embedding is mapped to the corresponding node quality score. Then the ranking of crucial nodes is obtained [17].

The above methods focus on the supervised learning paradigm and use abundant ground-truth labels to train specific downstream tasks, which is hard to apply to real-world scenarios (real-world scenarios are usually unlabeled). More importantly, these methods are prone to fall into the situation of over-fitting and weak robustness [18].

3 Theory and algorithm

The proposed algorithm is shown in Fig. 1. First, we introduced the Non-Negative Kernel (NNK) [19] Regressive to construct the LWSN into a sparse graph topology driven by the measurement data. It fuses two classical graph construction methods, similarity-based and locality-inducing [20], by the kernel method, where similarity-based methods

rely on a kernel metric to measure the similarity among sensor nodes, and locality-inducing methods depend on a regression objective to obtain the edge weights. And then we partition the large-scale graph topology into several subgraphs through an end-to-end representation-clustering joint model based on GNN. Lastly, node selection is performed on each subgraph to achieve the optimal deployment of the LWSN based on singular-value-QR decomposition.

3.1 Sub-network division

In the paper, we proposed the improved graph deep embedding clustering, an end-to-end representation-clustering joint algorithm, to partition the large-scale graph into several subgraphs. In the improved graph deep embedding clustering, we propose the graph auto encoder coupled with Graph Attention Network(GAT) to learn the embedded representation of sensor nodes

$$\boldsymbol{H} = \mathrm{GAT}(\boldsymbol{X}, \boldsymbol{A}) \tag{1}$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_i, \cdots, \boldsymbol{x}_N]^{\mathrm{T}} \in \mathbf{R}^{N \times d_{\mathrm{in}}}$ is the measurement matrix, where each row is the measurement values $\boldsymbol{x} \in \mathbf{R}^{d_{\mathrm{in}}}$ on a sensor node, and d_{in} is the dimension of the measurement; $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_i, \cdots, \boldsymbol{h}_N]^{\mathrm{T}} \in \mathbf{R}^{N \times d_{\mathrm{out}}}$ is the representation of all sensor nodes, where each row $\boldsymbol{h} \in \mathbf{R}^{d_{\mathrm{out}}}$ is the embedded representation of a sensor node, and d_{out} is the dimension of the embedded representation. Specifically, the embedded representation of the sensor node i at the l-th layer is denoted as

$$\boldsymbol{h}_i^{(l)} = \sigma\left(\sum_{j \in \mathcal{N}} \alpha_{ij}^{(l)} W^{(l)} \boldsymbol{h}_j^{(l-1)}\right) \tag{2}$$

where σ is an activation function; $\mathcal{N}$ is the neighborhoods of sensor node i; $W^{(l)}$ is the weight matrix; $\alpha_{ij}^{(l)}$ is the attention coefficient between sensor nodes i and j

$$\alpha_{ij}^{(l)} = \frac{\exp(\mathrm{LReLU}(\boldsymbol{a}^{(l)\mathrm{T}}[W^{(l)}\boldsymbol{h}_i^{(l)} \| W^{(l)}\boldsymbol{h}_j^{(l)}]))}{\sum_{k \in \mathcal{N}} \exp(\mathrm{LReLU}(\boldsymbol{a}^{(l)\mathrm{T}}[W^{(l)}\boldsymbol{h}_i^{(l)} \| W^{(l)}\boldsymbol{h}_k^{(l)}]))} \tag{3}$$

where $\boldsymbol{a}^{(l)}$ is a parameter vector; $\|$ is the concatenation operation; L*ReLU* is Leaky Rectified Linear Unit, a non-saturating activation function.

The decoder is used to reconstruct the adjacency matrix $\hat{\boldsymbol{A}}$

$$\hat{\boldsymbol{A}} = \mathrm{sigmoid}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}) \tag{4}$$

The reconstruction loss function $\mathcal{L}_r$ is defined as Binary Cross Entropy (BCE)

$$\mathcal{L}_r = -\sum_{i,j=1}^{N} A_{ij} \log \hat{A}_{ij} + (1 - A_{ij}) \log(1 - \hat{A}_{ij}) \tag{5}$$

where A_{ij} is an element of the adjacency matrix; $\hat{A}_{ij}$ is an element of the reconstruction adjacency matrix.

While obtaining a suitable node representation vector, we propose a Deep Embedding Clustering (DEC) algorithm for optimal sensor network division. First, we employ K-means on the initial node's embedded representations $\boldsymbol{H}^{(0)}$ to get R initial cluster centers $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_r, \cdots, \boldsymbol{\mu}_R]$. Then, the t distribution is used to measure the similarity between the node's embedded representation vector $\boldsymbol{h}_i$ and the subnetwork cluster center representation vector $\boldsymbol{\mu}_r$. So the soft cluster assignment Q of sensor nodes is obtained

$$q_{ir} = \frac{\left(1 + \|\boldsymbol{h}_i - \boldsymbol{\mu}_r\|^2\right)^{-1}}{\sum_s \left(1 + \|\boldsymbol{h}_i - \boldsymbol{\mu}_s\|^2\right)^{-1}} \tag{6}$$

where q_{ir} can be seen as the probability of assigning node i to cluster r, if the probability is higher, the confidence of the assignment is higher. In other words, the sensor node is closer to the cluster center.

The auxiliary target distribution P is constructed based on Q distribution to improve cluster effect. First, we raise the assignment probability q_{ir} to the second power to emphasize the role of high-confidence nodes and reduce the influence of low-confidence nodes. And then normalizing by frequency per cluster to prevent cluster degradation (all nodes belong to the same cluster)

$$p_{ir} = \frac{q_{ir}^2 / \sum_i q_{ir}}{\sum_s \left(q_{is}^2 / \sum_i q_{is} \right)} \tag{7}$$

where $\sum_i q_{ir}$ is soft cluster frequency and is the sum of the probabilities that the nodes belong to the r-th cluster.

With the help of an auxiliary target distribution P, high-confidence assignments are used as soft labels to supervise the learning of Q to refine the clusters iteratively. To be specific, the training Q distribution is fitted to the P distribution by minimizing the Kullback–Leibler (KL) divergence between the P and Q

$$\mathcal{L}_c = \mathrm{KL}(P\|Q) = \sum_i \sum_r p_{ir} \log \frac{p_{ir}}{q_{ir}} \tag{8}$$

For a better sub-network division, we joint the optimizations of representation learning and cluster assignment and define the total loss with reconstruction loss and clustering loss for optimization

$$\mathcal{L} = \mathcal{L}_r + \gamma \mathcal{L}_c \tag{9}$$

where γ is the clustering coefficient for the balance of reconstruction loss and clustering loss.

In the following training, three parameters are needed to optimize and update: cluster centers $\boldsymbol{\mu}$, encoder's weights W and auxiliary target distribution P.

First, fixing the auxiliary target distribution P, we use Stochastic Gradient Descent (SGD) to optimize the cluster centers and encoder weights jointly. The gradients of the clustering loss $\mathcal{L}_c$ based on the embedded representation $\boldsymbol{h}_i$ of sensor node i and cluster center representation $\boldsymbol{\mu}_r$ can be computed as [21]:

$$\frac{\partial \mathcal{L}_c}{\partial \boldsymbol{h}_i} = 2\sum_r \left(1 + \|\boldsymbol{h}_i - \boldsymbol{\mu}_r\|^2\right)^{-1} (p_{ir} - q_{ir})(\boldsymbol{h}_i - \boldsymbol{\mu}_r) \tag{10}$$

$$\frac{\partial \mathcal{L}_c}{\partial \boldsymbol{\mu}_r} = 2\sum_i \left(1 + \|\boldsymbol{h}_i - \boldsymbol{\mu}_r\|^2\right)^{-1} (q_{ir} - p_{ir})(\boldsymbol{h}_i - \boldsymbol{\mu}_r) \tag{11}$$

Then the cluster center $\boldsymbol{\mu}_r$ is updated by

$$\boldsymbol{\mu}_r = \boldsymbol{\mu}_r - \lambda \frac{\partial \mathcal{L}_c}{\partial \boldsymbol{\mu}_r} \tag{12}$$

where λ is the learning rate.

According to the chain rule, the gradient of the encoder weights can be computed as

$$\frac{\partial \mathcal{L}_c}{\partial W} = \frac{\partial \boldsymbol{h}_i}{\partial W} \frac{\partial \mathcal{L}_c}{\partial \boldsymbol{h}_i} \tag{13}$$

Then the encoder weights W are updated by

$$W = W - \lambda \left(\frac{\partial \mathcal{L}_r}{\partial W} + \gamma \frac{\partial \mathcal{L}_c}{\partial W} \right) \tag{14}$$

While the auxiliary target distribution acts as a "ground-truth" label during training, it also depends on the current soft assignment distribution. In iterative training, the constant change of the target will hinder learning and convergence. Therefore, the auxiliary target distribution P will not be updated at every iteration but only updated at every T iteration by Eq.(7) to enhance the stability of the clustering process.

After the above learning and training, the cluster assignment probability vector tends to be one-hot. Therefore, according to the last learned Q distribution, the cluster assignment label (i.e. the cluster assignment result of each sensor node i) of each sensor node can be obtained

$$y_i = \arg\max_r q_{ir} \tag{15}$$

We summarize the proposed improved graph deep embedding clustering in Algorithm 1.

Algorithm 1 Improved Graph Deep Embedding Clustering

Input: Measurement matrix $\boldsymbol{X}$, Adjacency matrix $\boldsymbol{A}$, Number of clusters R, Number of iterations $Iter$, Upadte interval T, Clustering coefficient γ.
Output: Clustering results.

Initialize node representation $\boldsymbol{H}^{(0)} = \boldsymbol{X}$ and cluster centers $\boldsymbol{\mu}$ by K-means
for $i = 1, 2, \cdots, Iter$ **do**
  Learn node embedded representation $\boldsymbol{H}$ by Eq.(1)
  Calculate soft assignment distribution Q by Eq.(6)
  if $i\%T = 0$ **then**
    Calculate target distribution P by Eq.(7)
  end if
  Calculate clustering loss $\mathcal{L}_c$ by Eq.(8)
  Reconstruct adjacency matrix $\hat{\boldsymbol{A}}$ by Eq.(4)
  Calculate reconstruction loss $\mathcal{L}_r$ by Eq.(5)
  Calculate total loss $\mathcal{L}$ by Eq.(9)
  Update the node embedded representation and cluster centers by Eq.(10)-(14)
end for
Obtain Clustering results by Eq.(15)

3.2 Node optimization selection

For the sub-network *r*, we construct the corresponding measurement matrix and adjacency matrix with $\boldsymbol{X}_r = \boldsymbol{X}\big|_{\boldsymbol{y}=r,:}$ and $\boldsymbol{A}_r = \boldsymbol{A}\big|_{\boldsymbol{y}=r,\boldsymbol{y}=r}$, where $r \in \boldsymbol{y}$, $\boldsymbol{y}$ is the clustering results in Eq.(15), and perform Singular Value Decomposition(SVD) on $\boldsymbol{X}_r$

$$\boldsymbol{X}_r = \boldsymbol{U}_r \begin{bmatrix} \boldsymbol{\Sigma}_r & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} \boldsymbol{V}_r^{\mathrm{T}} \tag{16}$$

where $\boldsymbol{U}_r \in \mathbf{R}^{N_r \times N_r}$ is the left singular vector matrix of $\boldsymbol{X}_r$, and N_r is the number of sub-network nodes; $\boldsymbol{V}_r \in \mathbf{R}^{d_{\text{in}} \times d_{\text{in}}}$ is the right singular vector matrix of $\boldsymbol{X}_r$; $\boldsymbol{\Sigma}_r$ is the singular value diagonal matrix.

Then we partition $\boldsymbol{U}_r$ as follows

$$\boldsymbol{U}_r = \begin{bmatrix} \boldsymbol{U}_{r1} & \boldsymbol{U}_{r2} \end{bmatrix} \tag{17}$$

where $\boldsymbol{U}_{r1} \in \mathbf{R}^{N_r \times M_r}$, $\boldsymbol{U}_{r2} \in \mathbf{R}^{N_r \times (N_r - M_r)}$, M_r is the number of nodes after optimization selection.

Next, performing column pivoting QR decomposition on $\boldsymbol{U}_{r1}$ and obtaining a permutation matrix $\boldsymbol{P}_r$

$$\boldsymbol{U}_{r1}^{\mathrm{T}} \boldsymbol{P}_r = \boldsymbol{Q}_r \boldsymbol{R}_r \tag{18}$$

where $\boldsymbol{Q}_r$ is a unitary matrix and $\boldsymbol{R}_r$ is an upper triangular matrix.

The rows with the number 1 in the permutation matrix $\boldsymbol{P}_r$ correspond to the M_r most important sensor nodes. According to the permutation matrix $\boldsymbol{P}_r$, we can obtain the optimized measurement matrix $\boldsymbol{X}_r^{'}$ and adjacency matrix $\boldsymbol{A}_r^{'}$

$$\boldsymbol{X}_r^{'} = \boldsymbol{P}_r \boldsymbol{X}_r \tag{19}$$

$$\boldsymbol{A}_r^{'} = \boldsymbol{P}_r \boldsymbol{A}_r \boldsymbol{P}_r^{\mathrm{T}} \tag{20}$$

Obviously, the SVD-QR method can select a set of independent nodes for each sub-network to minimize the least square residual and achieve the optimal deployment of the large-scale sensor network. Specifically, SVD can determine the principal components of the sub-network by calculating the singular values of the measurement matrix. Based on the principal component, QR decomposition can distinguish the importance of each node, and choose to keep or discard it. Finally, the optimized measurement matrices and adjacency matrices of all subgraphs are concatenated together, and the optimization deployment of the LWSN is completed

$$\boldsymbol{X}^{'} = \Big\|_{r=1}^{R} \boldsymbol{X}_r^{'} \tag{21}$$

$$\boldsymbol{A}^{'} = \Big\|_{r=1}^{R} \boldsymbol{A}_r^{'} \tag{22}$$

where $\|$ is the concatenation operation; R is the number of sub-network.

According to the description of the above parts, the proposed optimal deployment algorithm for LWSN in the paper is shown in Algorithm 2.

Algorithm 2 Optimal Deployment Algorithm for LWSN based on Graph Clustering and Matrix Decomposition

Input: Original measurement matrix $\boldsymbol{X}$ of the sensor network.

Output: Measurement matrix $\boldsymbol{X}^{'}$ and adjacency matrix $\boldsymbol{A}^{'}$ of optimized sensor network.

1: Initialize graph topology with the NNK graph
2: Obtain the clustering results with Algorithm 1
3: **for** $r = 1, 2, \cdots, R$ **do**
4: $\quad \boldsymbol{X}_r = \boldsymbol{X}\big|_{\boldsymbol{y}=r,:}$, $\boldsymbol{A}_r = \boldsymbol{A}\big|_{\boldsymbol{y}=r,\boldsymbol{y}=r}$
5: $\quad$ Obtain left singular vector matrix $\boldsymbol{U}_r$ with SVD by Eq.(16)
6: $\quad$ Obtain left singular vector submatrix $\boldsymbol{U}_{r1}$ by Eq.(17)
7: $\quad$ Obtain permutation matrix $\boldsymbol{P}_r$ with QR decomposition by Eq.(18)
8: $\quad$ Obtain the optimized sub-network measurement matrix $\boldsymbol{X}_r^{'}$ and adjacency matrix $\boldsymbol{A}_r^{'}$ by Eq.(19)-(20)
9: **end for**
10: Obtain the optimized sensor network measurement matrix $\boldsymbol{X}^{'}$ and adjacency matrix $\boldsymbol{A}^{'}$ with concatenating by Eq.(21)-(22)

3.3 Complexity analysis

Due to the large number of nodes in LWSN, many optimal deployment algorithms have high computational complexity and low execution efficiency. The algorithm proposed in this paper mainly focuses on two parts: improved graph deep embedding clustering and node optimization selection based on the SVD-QR. The computational complexity of these two parts is significantly improved with the NNK graph. Therefore, this paper analyzes the computational complexity.

The graph auto encoder coupled with the GAT for coding learning. First, it needs to calculate the feature maps of all sensor nodes. The l-th layer of the GAT needs to map

the dimension of the representation vector $\boldsymbol{h}_i^{(l-1)}$ from $d^{(l-1)}$ to $d^{(l)}$, and its computational complexity is $O(Nd^{(l-1)}d^{(l)})$. Therefore, the computational complexity of the feature mapping based on the GAT is $O(Nd_{\max}^2)$, where N is the number of nodes, and $d_{\max} = \max_l d^{(l)}$ is the maximum dimension of the hidden layers. Secondly, the attention coefficient needs to be calculated in the l-th layer, and each attention coefficient needs to map a $2d^{(l)}$ dimensional vector into a real number, and its computational complexity is $O(d^{(l)})$. For the whole graph, the attention coefficient between each sensor node and its neighbor nodes needs to be calculated, the number of calculations is the same as the number of edges, so the computational complexity of the attention coefficient is $O(Ed_{\max})$, where E is the number of edges. Only weighted summation is involved in the next step of the GAT, and no complex multiplication operations are involved, so the computational complexity of the node representation part is $O(Nd_{\max}^2 + Ed_{\max})$.

In the deep embedding clustering step, we assign nodes to the clusters according to the probability distribution Q of the cluster centers by minimizing the KL divergence of the Q distribution and the P distribution. So its computational complexity is $O(NRd_{\text{out}})$, where d_{out} is the embedded dimension of the representation vector, and R is the number of clusters. Therefore, the whole computational complexity in the sensor sub-network division is $O(Nd_{\max}^2 + Ed_{\max} + NRd_{\text{out}})$.

For each sub-network, the computational complexity of SVD is $O(N_r^3)$, and the computational complexity of QR decomposition is $O(2N_rM_r^2)$, where N_r is the number of nodes in the sub-network, and M_r is the number of optimized nodes. Since $M_r \ll N_r$, the computational complexity of optimizing selection for one sub-network is $O(N_r^3)$, and the computational complexity of optimizing selection for R sub-networks is $O(\sum_{r=1}^{R} N_r^3)$.

According to the previous analysis, the computational complexity of sub-network division is $O(Nd_{\max}^2 + Ed_{\max} + NRd_{\text{out}})$. For a connected graph topology, the number of edges E is often much larger than the number of nodes N, so the edges number E has a significant influence on the computational complexity. And the edges number E in a sparse NNK graph does not exceed $N \log N$, which effectively controls computational complexity. In addition, compared with directly performing matrix decomposition on the entire sensor network, dividing the sensor network into several sub-networks and separately performing matrix decomposition can reduce the computational complexity effectively. And according to the number of sub-networks, there are $O(N) \leqslant O(\sum_{r=1}^{R} N_r^3) \leqslant O(N^3)$, where R is the sub-networks number.

4 Experiments

4.1 Datasets

We use the California Irrigation Management Information System(CIMIS)[1] dataset for the algorithm testing in the paper. The CIMIS composed of more than 140 active meteorological stations, and each station integrates seven sensor types of solar radiation, air temperature, soil temperature, relative humidity, wind speed, wind direction, and

[1] http://www.cimis.water.ca.gov

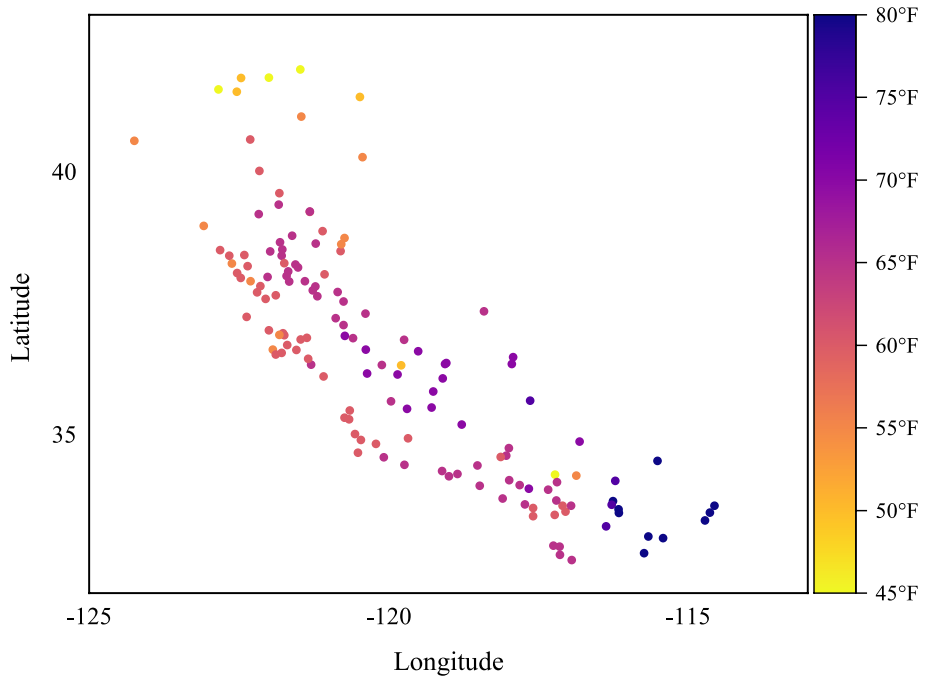

Fig. 2 CIMIS meteorological station locations and temperature values distribution

precipitation. The air temperature sensors were selected to build a WSN which contains 144 active nodes and the temperature values in one day are shown in Fig. 2.

4.2 Parameter setting

In the graph construction section, we set the maximum number of neighbors to 10 for each node and the weight zero threshold to 3×10^{-3}. In the clustering section, we divide the whole WSN into five sub-networks, and set the hidden dimension to 3, the embedded dimension to 5, and the learning rate to 0.0001, the weight decay to 5×10^{-4}. In order to improve the stability of clustering, the target distribution is updated every 5 iterations.

4.3 Evaluation metric

In order to compare the uniformity of the optimized deployment of the sensor network, we define the Optimization-Original Ratio(OOR) and distribution variance. OOR was the ratio of M, the number of optimally selected nodes, to N, the number of nodes in the original sensor network

$$OOR = \frac{M}{N} \tag{23}$$

The distribution variance was the OOR variance of each sub-network, which was used to measure the uniformity of node distribution after optimization. Obviously, the smaller the distribution variance is, the more uniform of the optimally selected nodes are

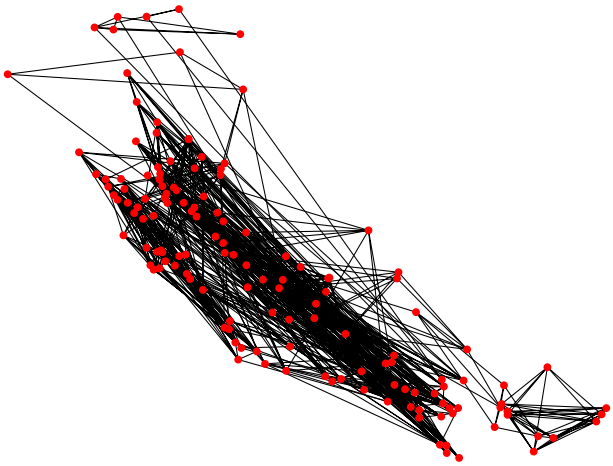

Fig. 3 Constructed Graph for CIMIS dataset. Red dots represent nodes, Black lines represent edges

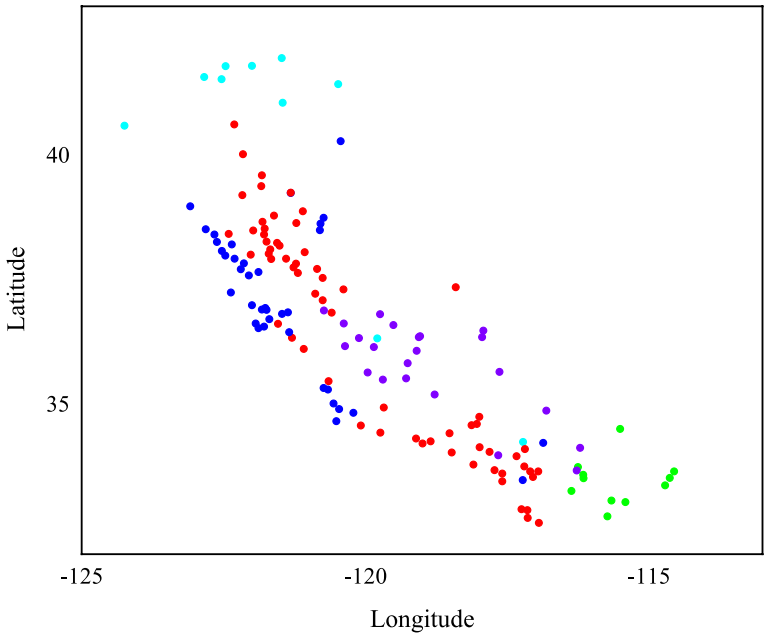

Fig. 4 Clusters of CIMIS dataset

$$s^2 = \frac{1}{R}\sum_{r=1}^{R}(OOR_r - \overline{OOR})^2 \tag{24}$$

where OOR_r is the OOR of sub-network r, $\overline{OOR}$ is the average OOR of each sub-network; R is the number of sub-networks.

4.4 Experimental results and analysis

4.4.1 Basic experiment

We construct a sparse graph with 595 edges for the CIMIS dataset by the NNK algorithm (Fig. 3). The improved graph deep embedding clustering algorithm divides the sensor nodes into five clusters (Fig. 4).

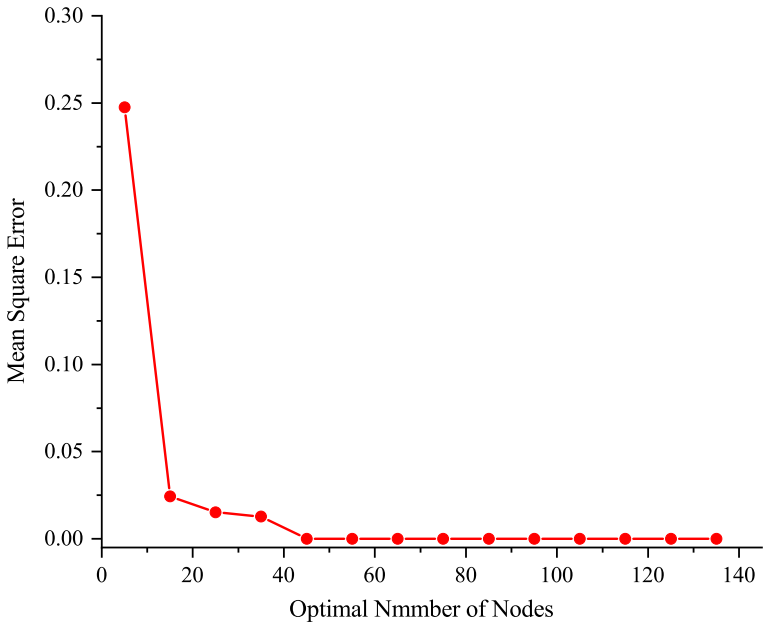

Fig. 5 Nodes with different optimization degrees and performance of CIMIS dataset

For optimization selection, the optimal number of nodes starts from 5 and increases by 10. The relationship between the Mean Square Error (MSE) values of the WSN and the optimal number of nodes is shown in Fig. 5. To maximize the network life and reduce the maintenance cost, we need to reduce the scale of the WSN as much as possible within the acceptable range of data reconstruction precision (i.e. MSE < 0.1). Therefore, according to Fig. 5, we choose the optimal number as 15, and the MSE of the WSN is only 0.02433.

4.4.2 Comparative experiment

We have compared the proposed algorithm with five classic WSN optimization deployment algorithms, namely MinPinv [22], MaxVol [22], MinSpec [23], Entropy [24], and MI [25]. Figure 6 showed the Mean Square Error of the optimization algorithms. It can be seen that the proposed algorithm has a very obvious improvement in the optimal selection of sensor nodes compared with other algorithms.

When the optimal number is 15, we compared the distribution of optimal nodes and the distribution variance for different optimization algorithms, as shown in Figs. 7 and 8. The nodes selected by optimization are marked with color, and the remaining nodes are marked with gray.

It can be easily seen from Figs. 7 and 8 that the distribution of the optimal nodes with the proposed algorithm is more uniform and comprehensive than others, and its distribution variance is as low as 2.68×10^{-4}.

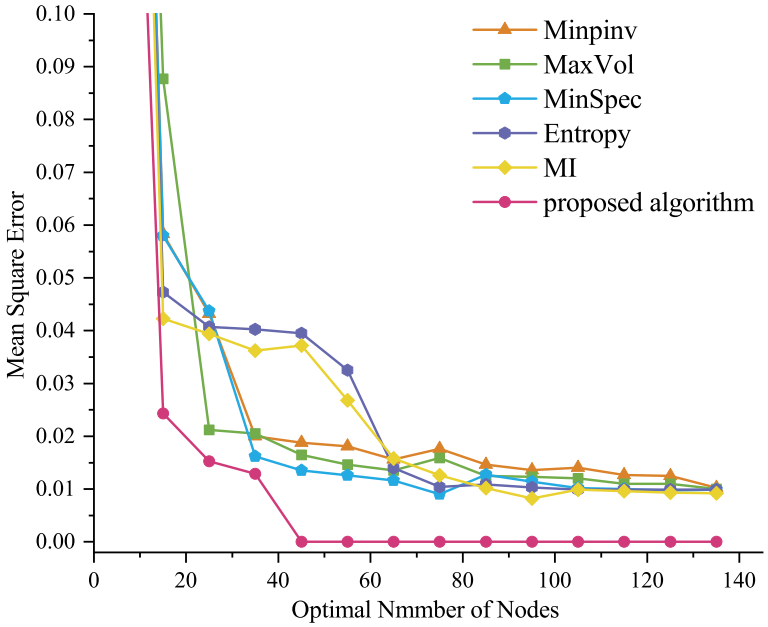

Fig. 6 Performance comparison of different optimal deployment algorithms

4.4.3 Scalability experiment

In order to illustrate the scalability of the proposed algorithm in this paper and its applicability to LWSN, we build several simulation datasets, which are as follows:

(1) Gaussian Simulation Dataset (GSD): In a unit area, sensor nodes are randomly deployed and the node signal values obey the $N(0, 1)$ Gaussian distribution, where the network scales are 500, 1000, 2000, 5000 and, 10000 respectively.

(2) Uniform Simulation Dataset (USD): In a unit area, sensor nodes are randomly deployed and the signal values obey the uniform distribution of $U(0, 1)$, where the network scales are 500, 1000, 2000, 5000 and, 10000 respectively.

The proposed algorithm is applied to the above simulation dataset, and the experimental results are shown in Fig. 9. The bar graph shows the relationship between the optimal number of nodes and the network size, and the line graph corresponds to the mean square error.

The results show that the proposed algorithm can significantly reduce the number of nodes required for LWSN without losing crucial information. It reflects that the proposed algorithm has good adaptability and scalability for LWSN.

5 Conclusion

In this paper, we innovatively built a large-scale graph topology division algorithm based on deep embedding clustering and an optimal selection mechanism for sensor nodes based on SVD-QR for the optimal deployment of LWSN. The experimental results verify the good applicability of the proposed algorithm. In conclusion, the proposed algorithm has the following advantages:

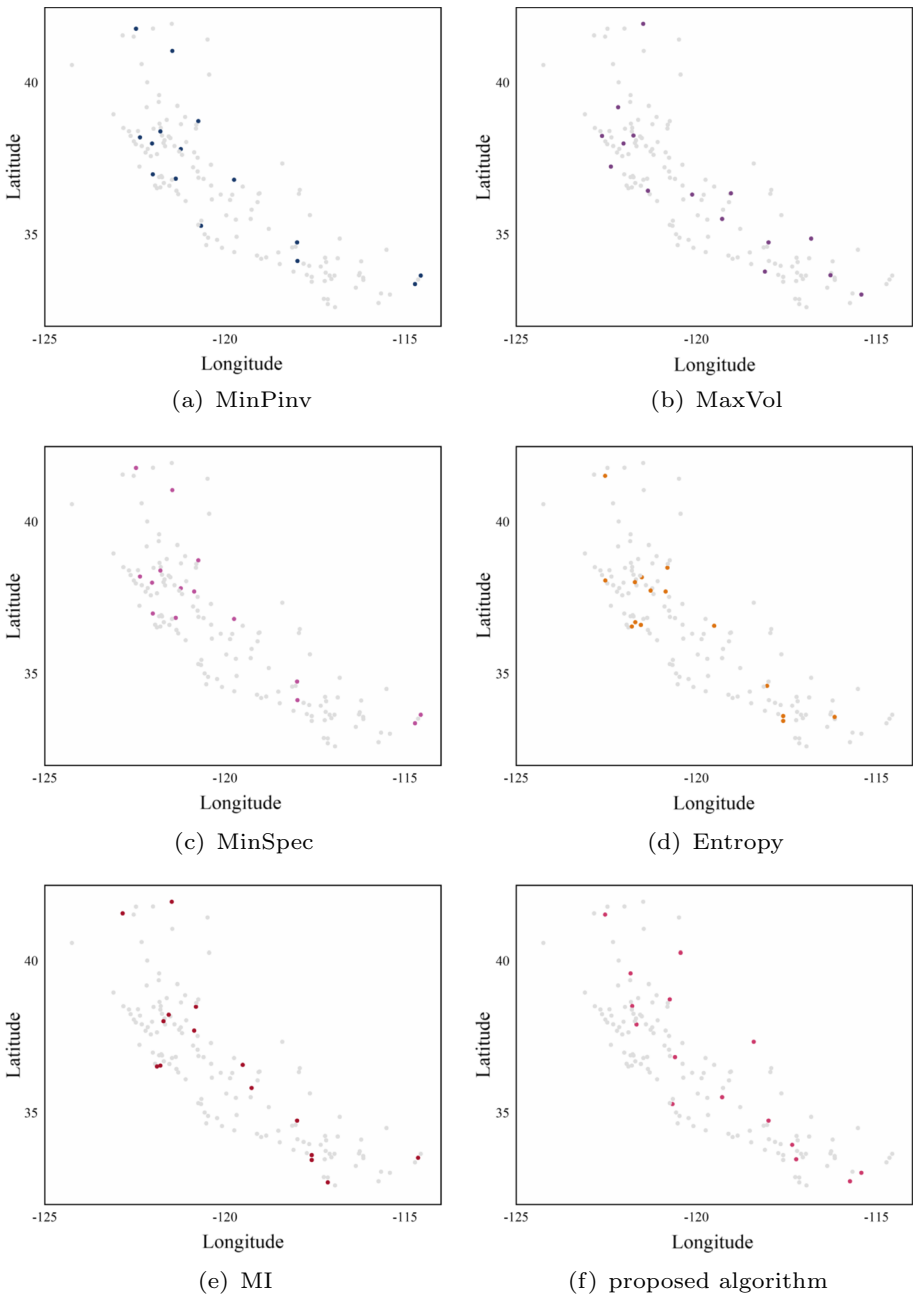

Fig. 7 Optimization deployment and selected node distribution

1. The problem of node optimal deployment was investigated in subgraphs, which improved the algorithm scalability
2. We applied GNN for graph representation and graph clustering in the same framework, which enhanced the adaptability of sensor node representation and clustering
3. It made full use of the attribute information and structure information of the graph to carry out end-to-end learning at the same time, which strengthened the learning quality of the GNN model

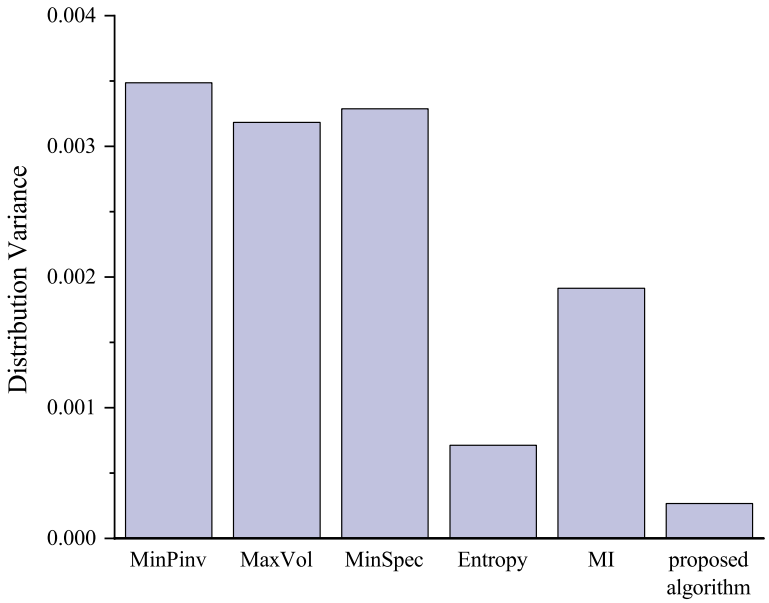

Fig. 8 Comparison of distribution variance of different optimal deployment algorithms

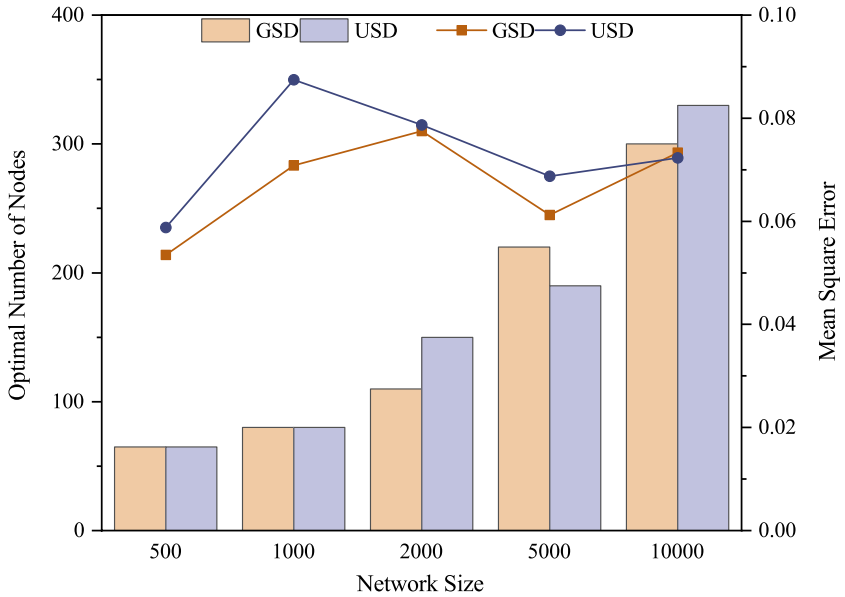

Fig. 9 Scalability experiment and analysis

4. Each step of the proposed algorithm could effectively improve the computational efficiency of subsequent steps, which greatly reduces the whole computational complexity. In the future, we will pay more attention to the optimal deployment of heterogeneous large-scale sensor networks based on our proposed algorithm, and improve the execution efficiency of the proposed algorithm.

Abbreviations

LWSN	Large-scale wireless sensor network
WSN	Wireless sensor network
GNN	Graph neural network
NNK	Non-negative kernel
GAT	Graph attention network
BCE	Binary cross entropy
DEC	Deep embedding clustering
KL	Kullback–Leibler
SGD:	Stochastic gradient descent
SVD	Singular value decomposition
SVD-QR	Singular-value-QR decomposition
CIMIS	California irrigation management information system
OOR	Optimization-original ratio
MSE	Mean square error
GSD	Gaussian simulation dataset
USD	Uniform simulation dataset

Acknowledgements
This work is supported by National Natural Science Foundation of China (No.61731006, 61971310).

Author contributions
WW, as the corresponding author, provides research ideas, oversight, and leadership responsibility for the research activity planning and execution, including mentorship external to the core team. HG provides algorithm design and computer code implementation. QZ analyzes and synthesizes data. All authors read and approved by the final manuscript.

Funding
This work is supported by National Natural Science Foundation of China (No.61731006, 61971310).

Availability of data and materials
The real dataset can be obtained from the link given in the footnote of the text, and simulation datasets can be generated as described in the paper.

Code availability
Not Applicable.

Declarations

Ethics approval and consent to participate
Not Applicable.

Human and animal ethics
Not Applicable.

Consent for publication
There is the consent of all authors.

Conmpeting interests
The authors declare that they have no confict of interest.

Received: 2 August 2022 Accepted: 23 February 2023
Published online: 08 March 2023

References

1. V. Patil, S. Deshpande, Design of fpga soft core based wsn node using customization paradigm. Wireless Pers. Commun. **122**(1), 783–805 (2022). https://doi.org/10.1007/s11277-021-08925-y
2. Y. Sangar, B. Krishnaswamy, WiChronos: energy-efficient modulation for long-range, large-scale wireless networks. Assoc. Comput. Machinery **10**(11453372224), 3380898 (2020)
3. O.A. Khashan, R. Ahmad, N.M. Khafajah, An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks. Ad Hoc. Netw. **115**, 102448 (2021). https://doi.org/10.1016/j.adhoc.2021.102448
4. G. Sahar, K.B.A. Bakar, F.T. Zuhra, S. Rahim, T. Bibi, S.H.H. Madni, Data redundancy reduction for energy-efficiency in wireless sensor networks: a comprehensive review. IEEE Access **9**, 157859–157888 (2021). https://doi.org/10.1109/ACCESS.2021.3128353
5. S.S. Mohar, S. Goyal, R. Kaur, Optimized sensor nodes deployment in wireless sensor network using bat algorithm. Wireless Pers. Commun. **116**(4), 2835–2853 (2021). https://doi.org/10.1007/s11277-020-07823-z
6. M. Tolouei ashtian, M. Golsorkhtabaramiri, S.Y.B. Rad, An improved whale optimization algorithm solving the point coverage problem in wireless sensor networks. Telecommun. Syst. **79**(3), 417–436 (2022). https://doi.org/10.2298/CSIS180103023W

7. X. Deng, Z. Yu, R. Tang, X. Qian, K. Yuan, S. Liu, An optimized node deployment solution based on a virtual spring force algorithm for wireless sensor network applications. Sensors **19**(8), 1817 (2019). https://doi.org/10.3390/s19081817
8. K. Hegde, R. Dilli, Wireless sensor networks: network life time enhancement using an improved grey wolf optimization algorithm. Eng. Sci. (2022). https://doi.org/10.30919/es8d717
9. Z. Xiaohan, Y. Changchuan, W. Huarui, Energy optimization strategy for wireless sensor networks in large-scale farmland habitat monitoring. Smart Agricul. **1**(2), 55 (2019). https://doi.org/10.12133/j.smartag.2019.1.2.201812-SA024
10. P. Jiang, Y. Li, F. Wu, S. Yu, H. Xu, Research on energy consumption and eliminate of large data redundancy method for large scale wireless sensor networks based on mean filter. Adv. Eng. Sci. **49**(2), 145–151 (2017). https://doi.org/10.15961/j.jsuese.201601189
11. B. Cao, Q. Wei, Z. Lv, J. Zhao, A.K. Singh, Many-objective deployment optimization of edge devices for 5G networks. IEEE Trans. Net. Sci. Eng. **7**(4), 2117–2125 (2020). https://doi.org/10.1109/TNSE.2020.3008381
12. A. Sakiyama, Y. Tanaka, T. Tanaka, A. Ortega, Eigendecomposition-free sampling set selection for graph signals. IEEE Trans. Signal Process. **67**(10), 2679–2692 (2019). https://doi.org/10.1109/TSP.2019.2908129
13. H. Xu, G. Li, G. Zhang (2019) Energy-aware sampling set selection for signal reconstruction in internet of things. In: 2019 IEEE 5th international conference on computer and communications (ICCC), pp. 2146–2151. https://doi.org/10.1109/ICCC47050.2019.9064079. IEEE
14. P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, G. Leus, Adaptive graph signal processing: Algorithms and optimal sampling strategies. IEEE Trans. Signal Process. **66**(13), 3584–3598 (2018). https://doi.org/10.1109/TSP.2018.2835384
15. D. Valsesia, G. Fracastoro, E. Magli, Sampling of graph signals via randomized local aggregations. IEEE Trans. Signal Inform. Proc. Over Net. **5**(2), 348–359 (2018). https://doi.org/10.1109/TSIPN.2018.2869354
16. G. Zhao, P. Jia, A. Zhou, B. Zhang, Infgcn: identifying influential nodes in complex networks with graph convolutional networks. Neurocomputing **414**, 18–26 (2020). https://doi.org/10.1016/j.neucom.2020.07.028
17. M. Ma, X. Wang, Y. Li, Q. Zhang, C. Wang. Critical node identification based on attention flow networks. in: 2021 International conference on machine learning and intelligent systems engineering (MLISE), pp. 96–102 (2021). https://doi.org/10.1109/MLISE54096.2021.00025. IEEE
18. X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, J. Tang, Self-supervised learning: generative or contrastive. IEEE Trans. Knowl. Data Eng. **35**(1), 857–876 (2021). https://doi.org/10.1109/TKDE.2021.3090866
19. S. Shekkizhar, A. Ortega. Graph construction from data by non-negative kernel regression. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 3892–3896 (2020). https://doi.org/10.1109/ICASSP40776.2020.9054425. IEEE
20. L. Qiao, L. Zhang, S. Chen, D. Shen, Data-driven graph construction and graph learning: a review. Neurocomputing **312**, 336–351 (2018). https://doi.org/10.1016/j.neucom.2018.05.084
21. J. Xie, R. Girshick, A. Farhadi. Unsupervised deep embedding for clustering analysis. In: international conference on machine learning, pp. 478–487 (2016). PMLR
22. M. Tsitsvero, S. Barbarossa, P. Di Lorenzo, Signals on graphs: uncertainty principle and sampling. IEEE Trans. Signal Process. **64**(18), 4845–4860 (2016). https://doi.org/10.1109/TSP.2016.2573748
23. S. Chen, R. Varma, A. Sandryhaila, J. Kovačević, Discrete signal processing on graphs: sampling theory. IEEE Trans. Signal Process. **63**(24), 6510–6523 (2015). https://doi.org/10.1109/TSP.2015.2469645
24. M.C. Shewry, H.P. Wynn, Maximum entropy sampling. J. Appl. Stat. **14**(2), 165–170 (1987). https://doi.org/10.1080/02664768700000020
25. A. Krause, A. Singh, C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. J. Machine Learn. Res. **9**(2) (2008). https://doi.org/10.1145/1390681.1390689

Publisher's Note