Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8894852, 11 pages
https://doi.org/10.1155/2020/8894852

Research Article

A Random Walk-Based Energy-Aware Compressive Data Collection for Wireless Sensor Networks

Keming Dong,[1] **Chao Chen**,[2] **and Xiaohan Yu**[2]

[1]*School of Information, Yunnan University of Finance and Economics, Kunming 650221, China*
[2]*School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China*

Correspondence should be addressed to Xiaohan Yu; yuxiaohan188@126.com

Received 28 August 2020; Revised 17 October 2020; Accepted 24 October 2020; Published 30 November 2020

Academic Editor: Xingsi Xue

The energy efficiency for data collection is one of the most important research topics in wireless sensor networks (WSNs). As a popular data collection scheme, the compressive sensing- (CS-) based data collection schemes own many advantages from the perspectives of energy efficiency and load balance. Compared to the dense sensing matrices, applications of the sparse random matrices are able to further improve the performance of CS-based data collection schemes. In this paper, we proposed a compressive data collection scheme based on random walks, which exploits the compressibility of data vectors in the network. Each measurement was collected along a random walk that is modeled as a Markov chain. The Minimum Expected Cost Data Collection (MECDC) scheme was proposed to iteratively find the optimal transition probability of the Markov chain such that the expected cost of a random walk could be minimized. In the MECDC scheme, a nonuniform sparse random matrix, which is equivalent to the optimal transition probability matrix, was adopted to accurately recover the original data vector by using the nonuniform sparse random projection (NSRP) estimator. Simulation results showed that the proposed scheme was able to reduce the energy consumption and balance the network load.

1. Introduction

This paper considers the energy efficiency issue of compressive data collection in wireless sensor networks (WSNs). A WSN is consisted of low-cost, low-power, and energy-constrained sensors which acquires and transmits information to the sink through wireless links [1–5]. In the area of Internet of Things (IoT), a WSN is regarded as a key technology for the data sensing and collection [6, 7]. One of the most important factors that affects the performance of WSNs is the energy limitation of sensors [8, 9]. A sensor will cease to operate if it depletes its battery energy. We intend to design an energy-efficient data collection scheme by applying the compressive sensing (CS) technology and random walks.

A popular approach to the data collection problem is the application of the CS technology [10, 11]. In the CS technology, data are assumed to be sparse or sparse under some basis, which is very appropriate for data in WSNs [12, 13]. The key idea behind CS is that, by exploiting the sparsity of the original data vector, a high dimensional data vector can be reliably recovered from a significantly lower number of measurements. Early works mainly focus on applying the CS technology with a dense sensing matrix [14–16]. Luo et al. [14] proposed the Compressive Data Gathering (CDG) scheme in which the sink collects linear combinations of the original data vector instead of the individual data sample. The sink is able to recover the original data vector through solving an ℓ_1-based convex optimization as long as a sufficient number of linear combinations are collected. Compared with traditional schemes, the CDG scheme not only reduces the energy consumption but also evenly distributes loads across the network. The reference [15] improved the CDG scheme and proposed a hybrid-CS scheme in which data are only encoded at overloaded nodes. This significantly reduces the load of nodes which are far away from the sink. The authors showed that, compared with the CDG scheme, the hybrid-CS scheme can further improve the throughput of networks. Adopting the idea of the CDG scheme, the reference [16] considered not only the energy efficiency but also the delay of data collection. The authors proposed a joint optimization problem which aims to minimize the delay of data collection with bounded transmissions. The NP-hardness of the joint optimization problem was proved. Thus, the authors proposed an approximation solution which decomposes the joint optimization problem into a forwarding tree construction subproblem and a link-scheduling subproblem.

Without the sacrifice of recovery fidelity, sparse random matrices have been proven to give better energy efficiency than the dense random matrices [17, 18]. Under the CDG framework of WSNs, the sparse random matrix can be either uniform [17–20] or nonuniform [21–24]. In the uniform sparse random matrix, each entry is equal to zero with an identical probability. However, in the nonuniform sparse random matrix, entries in different columns are equal to zero with variational probabilities. Wang et al. [17] proposed a class of uniform sparse random matrices that do not compromise the recovery performance when compared to a Gaussian sensing matrix. In their scheme, each node aggregates one measurement as a linear combination of the original data vector. The sink collects measurements from nodes through different shortest paths. Zheng et al. [18] proposed a random walk-based data collection scheme and provided mathematical foundations from the perspectives of the CS and graph theory. They showed that uniform sparse random matrices which are constructed from the proposed random walk scheme satisfy the expansion property of expander graphs. Singh et al. [19] proposed an On-Demand Explosion-Based Compressive Sensing (ODECS) technology to reduce the required number of measurements for the recovery of data vector by exploiting the rate of change of the data vector. The ODECS technology is able to adapt itself to the occurrence of events. It has very low communication rate when events are absent. Considering problems in existing schemes, such as the semidynamic routing, the nonuniform sampling, and the dependence on global coordinate information, Zhang et al. [20] proposed a dual random walk-based compressive data collection scheme. In the proposed scheme, a dual random walk, which does not rely on coordinate information, was first designed to achieve a uniform sampling. Then, depending on the dual random walk, a dynamic and distributed CDG-based scheme was proposed to enhance the network dynamic adaptability.

Recently, nonuniform sparse random matrices were proved to give similar performance as the uniform sparse random matrices [21–24]. Liu et al. [21] proposed a novel compressive data collection scheme which compresses data under an opportunistic routing. The proposed scheme requires fewer compressed measurements and allows a simpler routing strategy without excessive computation and overheads. Moreover, the authors proposed the nonuniform sparse random projection (NSRP) algorithm to recover the original data vector. They proved that the NSRP-based estimator can achieve the optimal estimation error bound. Considering the large transmission energy consumption and low recovery accuracy problem in traditional schemes, Zhang et al. [22] proposed a ring topology-based compressive sensing data collection scheme. In the proposed scheme, the total number of hops is reduced by a ring topology-based random walk, and the recovery accuracy is improved by the dual compensation-based compressive sensing measurements. Huang and Soong [23] proposed a cost-aware stochastic compressive data collection scheme, where the cost diversity and the stochastic data collection process are considered by using the Markov chain model. The proposed scheme is aimed at minimizing the expected cost of a random walk subjected to constraints on the global degree of randomness and recovery error. Without loss of the recovery accuracy, the proposed scheme not only reduces the expected cost but also prolongs the network lifetime due to the load balance feature. The reference [24] proposed a mobile CDG scheme including a random walk-based algorithm and a kernel-based method for sparsifying sensory data from an irregular deployment. The sensing matrix, which is constructed from the proposed random walk algorithm combined with a kernel-based sparsity basis, was proved to satisfy the restricted isometry property. Moreover, the authors proved that $O(k \log (n/k))$ measurements, which can be collected within $O(k \log (n/k))$ steps, were sufficient for the accurate recovery of k-sparse signals in a network with n nodes.

In this paper, we propose a data collection scheme for WSNs by integrating the compressive sensing technology and random walks. The total amount of energy consumption is reduced by exploiting the compressibility of the original data vector. Measurements are collected along random walks so that the local energy consumption is balanced. Specifically, each measurement is a linear combination of the original data in nodes which occur in a random walk. Each random walk is formulated as follows. Initially, a node except for the sink is selected as the starting node with a probability that is determined by the residual energy of every node in the network. Then, data are forwarded to the sink in a multihop manner. During the transmission process, each node selects the next hop node from its candidate nodes according to a probability distribution that is determined by the residual energy of its candidate nodes. We can model this stochastic process as an absorbing Markov chain. The key problem is that how to determine the transition probabilities of nodes in every random walk. We formulate this problem as an optimization problem which aims to find the optimal transition probability matrix such that the expected cost of a random walk is minimized. The Minimum Expected Cost Data Collection (MECDC) scheme is proposed to iteratively find the optimal transition probability matrix. After obtaining the optimal transition probability matrix, the sink is able to construct an equivalent sensing matrix based on the optimal transition probability matrix. Eventually, by using the NSRP-based estimator [21], the original data vector can be accurately recovered. For the compressive data collection problem, the reference [23] adopted a similar idea as this paper. However, in their scheme, each random walk starts at a fixed node, which results in the rapid energy expenditure of the fixed node. This paper extends the reference [23] mainly in five aspects: (1) the starting node of random walks is variational; (2) nodes' residual energy is considered for the balance of network load; (3) the MECDC scheme along with its distributed realization is proposed; (4) the process of collecting measurements is accelerated by partitioning the network into layers; (5) computation of optimal transition probability matrix is simplified.

The main contributions of this paper are summarized as follows:

(i) We propose a random walk-based compressive data collection scheme which exploits the compressibility of the original data vector. Random walks are responsible for the collection of measurements. In order to reduce the energy consumption and balance the network load, the residual energy of nodes is considered in the process of data collections

(ii) The absorbing Markov chain model is adopted to characterize the stochastic of a random walk. We formulate an optimization problem to minimize the expected cost of a random walk and propose the MECDC scheme to find the optimal transition probability matrix

(iii) A distributed realization of the MECDC scheme is proposed, where the update of transition probabilities for each node can be obtained only based on the information of its neighbors

(iv) Simulation results are provided to demonstrate that the proposed scheme can both reduce the energy consumption and balance the network load

The remainder of this paper is organized as follows. In Section 2, we present preliminaries of this paper. Next, we introduce the system model and problem formulation in Section 3. The MECDC scheme is proposed in Section 4. In Section 5, we present simulation results. Finally, Section 6 concludes the paper.

2. Preliminaries

We will use boldface letters to denote vectors and matrices. The ith entry of vector $\mathbf{x}$ is denoted by x_i. The entry in the i th row and jth column of matrix $\mathbf{A}$ is denoted by a_{ij}. Denote $[n] \coloneqq \{1, 2, \cdots, n\}$. A vector $\mathbf{x}$ is said to be k-sparse if the number of nonzero entries does not exceed k. Consider the following linear model:

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ $(m \ll n)$ is referred as the *sensing matrix*, each entry y_i in vector $\mathbf{y}$ is referred as a *measurement*, and $\mathbf{x} \in \mathbb{R}^n$ is the data vector to be recovered. It is well known that the Gaussian random matrix can be used as the sensing matrix. When $m \geq O(k \log n)$, a k-sparse data vector can be recovered with high probability via linear programming [25, 26].

It has been shown that sparse random matrices provide similar recovery performance as the Gaussian random matrix [17, 27]. A sparse random matrix is a matrix whose entries are zero with some probability. Importantly, if $a_{ij} = 0$, we will *not* need the data x_j when collecting y_i, because y_i is a linear combination of entries in $\mathbf{x}$. By exploiting the sparsity of the sensing matrix, potential improvement including the reduced energy and data collection delay can be obtained [8, 9]. In this paper, we consider the problem of recovering a compressible data vector by using a nonuniform sparse random matrix. Compressible data vectors can be seen as a subset of sparse data vectors. Specifically, a compressible data vector $\mathbf{x}$ can be represented as $\mathbf{x} = \mathbf{\Psi}\theta$, where Ψ is an $n \times n$ orthonormal basis and θ is a coefficient vector that decays according to the power law [28]. If we rearrange entries of θ according to the magnitude, then the ith largest entry $\theta_{(i)}$ satisfies

$$|\theta_{(i)}| \leq c i^{-1/z}, i \in [n], \tag{2}$$

where c is a constant and z controls the rate of decaying. Throughout this paper, we assume that the data vector $\mathbf{x}$ is compressible in some basis. The best k-term approximation of $\mathbf{x}$ is to keep the largest k coefficients and set the others to zero. Let $\widehat{\theta}_k$ be the coefficient vector of the best k-term approximation of $\mathbf{x}$. Then, we have that [28].

$$\|\mathbf{x} - \widehat{\mathbf{x}}_k\|_2 = \|\theta - \widehat{\theta}_k\|_2 \leq \zeta_r c k^{-1/r+1/2}, \tag{3}$$

where $\widehat{\mathbf{x}}_k = \mathbf{\Psi}\widehat{\theta}_k$ and ζ_r are constant that only depends on r. For a compressible data vector $\mathbf{x}$, the reference [21] proposed a nonuniform sparse random projection-based estimator which gives comparable recovery performance as the best k -term approximation provided that $m = O(k^2 \log n)$ and entries in $\mathbf{A} \in \mathbb{R}^{m\times n}$ are drawn i.i.d. from the following distribution [17, 21, 23].

$$a_{ij} = \begin{cases} +1, & \text{with probability } \dfrac{\pi_j}{2}, \\ -1, & \text{with probability } \dfrac{\pi_j}{2}, \\ 0, & \text{with probability } 1 - \pi_j, \end{cases} \tag{4}$$

where $0 \leq \pi_j \leq 1$ is a probability. Unlike the uniform sparse random matrices, the probability of being zero for entries in different columns of the nonuniform sparse random matrix varies.

3. System Model and Problem Formulation

3.1. Network Model. In this paper, we consider a multihop wireless sensor network consisting of n nodes with node n being the sink. Sensors are randomly deployed in a sensing field to sense the surrounding environment and then periodically report readings to the sink through multihop transmissions. Define x_i^t as the reading of sensor i at time instant t. The sink aims to collect data $\mathbf{x}^t = [x_1^t, x_2^t, \cdots, x_{n-1}^t]$ for different time instants. Previous works [17, 28] have shown that most natural classes of signals, such as smooth signals with bounded derivatives and bounded variation signals, are compressible in some transform domain. As stated in the previous section, we assume that the data $\mathbf{x}^t$ is compressible.

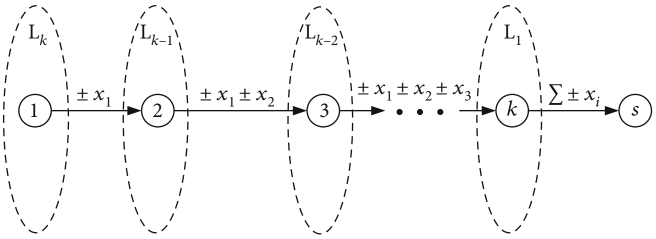

FIGURE 1: The example of collecting a measurement through a random walk. The random walk starts at node $1 \in L_k$ and ends at the sink.

Without loss of generality, we assume that sensors are randomly deployed in a unit square, and each sensor is equipped with an identical battery with the initial power E_0. Any two nodes are able to communicate with each other if the Euclidean distance between these two nodes is no more than the communication range R. The WSN is modeled as a connected graph $G = (V, E)$ with $V = [n]$ the set of nodes including the root/sink n and E the set of edges/wireless links. Each edge is associated with a weight which is related to the residual energy of nodes. Specifically, we define w_{ij}, the *ij*th entry of the weight matrix $\mathbf{W} \in \mathbb{R}^{(n-1)\times(n-1)}$, as the weight of edge (i, j) representing the cost of transmitting data from node i to node j. Note that we omit edges related to the sink. Suppose each node knows information of its neighbors. Expect for the sink, we partition nodes into layers $L_k, k = 1, \cdots, T$, where L_k is consisted of the nodes at distance k from the sink. For any node $i \in L_k$, its neighbors are divided into two disjoint sets: the successors set $S_i := \{j \mid (i, j) \in E, j \in L_{k-1}\}$ and the predecessors set $D_i := \{j \mid (i, j) \in E, j \in L_{k+1}\}$. Let $E_r(i)$ be the residual energy of node i. At the beginning of data collection, each node contains an identical initial energy E_0.

3.2. Opportunistic Routing. In this subsection, we describe how measurements are collected by the sink through random walks. The process of collecting measurements can be modeled as a discrete absorbing Markov chain [29] with the state set $\{s_1, s_2, \cdots, s_n\}$ and the transition probability matrix $\mathbf{P}$. Each node in the network corresponds to a state in the discrete absorbing Markov chain. Specifically, we assume that node $i \in [n]$ corresponds to the state s_i, and s_n is the absorbing state. The *ij*th entry in $\mathbf{P}$, i.e., the state transition probability p_{ij}, corresponds to the probability that the data is transmitted from node i to node j.

Our goal is to collect m measurements through m random walks. Each measurement corresponds to a random walk that starts from a randomly selected node and ends at the sink. Figure 1 shows the process of collecting a measurement, say the *j*th measurement y_j, which corresponds to the *j*th random walk. Initially, node 1 in layer L_k is chosen as the starting node. Then, it transmits data $+x_1$ or $-x_1$ to the randomly selected node $2 \in S_1$. Note that $S_1 \subseteq L_{k-1}$. Subsequently, node 2 adds or subtracts the received value to its own data and transmits the result, i.e., $\pm x_1 \pm x_2$, to a randomly selected node, say node $3 \in S_2$. The above process is repeated until the sink receives the measurement $y_j = \sum_{i=1}^{k} \pm x_i$. We can observe that the length of *j*th random walk is exactly the layer index of the starting node.

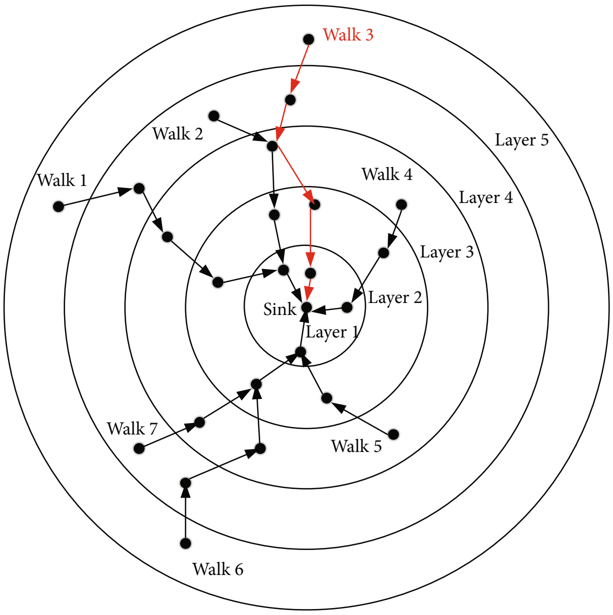

FIGURE 2: The process of collecting seven measurements. Measurements are collected through random walks. Each random walk starts from a randomly selected node.

In general, the process of collecting each measurement starts at a randomly chosen node. In this paper, a node $i (i \neq n)$ is selected as the starting node with probability p_i. Then, node i randomly selects a successor according to a certain probability distribution and subsequently transmits its compressed data to the selected successor. After receiving data, the selected successor adds or subtracts its own data to the received data and transmits the result towards the sink. The process is repeated until the sink collects every measurement. Figure 2 shows the process of collecting seven measurements. In this figure, nodes are partitioned into layers based on its length to the sink, and there are five layers in Figure 2.

3.3. The Transition Probability Matrix and the Sensing Matrix. The long-term behavior of random walks is closely related to the sensing matrix under the CS framework. In order to see this, let us write the transition probability matrix in the canonical form [29].

$$\mathbf{P} = \begin{pmatrix} \mathbf{Q} & * \\ \mathbf{0} & 1 \end{pmatrix}, \quad (5)$$

where $\mathbf{Q} \in \mathbb{R}^{(n-1)\times(n-1)}$ is the transition probability matrix of transient states and $\mathbf{0}$ is a row vector with zero entries. It is well known that the fundamental matrix, i.e., $\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}$, represents the long-term behavior of the discrete absorbing Markov chain. Specifically, the ijth entry of $\mathbf{F}$, f_{ij}, gives the expected number of times that the chain is in the transient state s_j, if it is started in the transient state s_i. In other words, f_{ij} is the expected number of occurrence of node j if the random walk starts at node i. In our formulations, every node occurs at most once in a random walk. Therefore, f_{ij} represents the probability of a random walk that passes the node j if it is started at the node i. Furthermore, excepting for the sink n, node i is selected as the starting node with probability p_i. Then, we have that

$$\pi_j = \sum_{i=1}^{n-1} p_i f_{ij}, \tag{6}$$

where π_j, the probability of node j in a random walk, is referred as the compression probability. As stated in the references [21, 23], π_j is exactly the nonzero probability of entries in the jth column of the sensing matrix $\mathbf{A}$.

3.4. Problem Formulation. The energy efficiency is the key issue in this paper. We intend to decrease the energy consumption of data collection and meanwhile balance the load of sensors. Since the opportunistic routing is a stochastic method, a natural idea to minimize the expected cost of a random walk. Specifically, we define c_i, the ith entry in the vector $\mathbf{c} \in \mathbb{R}^{n-1}$, as the expected cost of a random walk if node i is selected as the starting node. The goal is to minimize the expected cost of a random walk which is given by

$$\sum_{i=1}^{n-1} p_i c_i. \tag{7}$$

Furthermore, for any c_i, we have that

$$c_i = \sum_{j\in S_i} q_{ij}\left(w_{ij} + c_j\right). \tag{8}$$

An immediate observation is that $c_i > c_j$ if $i \in L_k, j \in L_r$ with $k > r$. In other words, the expected cost of a random walk with the starting point in a high layer is more than that in a low layer.

Similar to the reference [23], we introduce the concept of randomness for data collection in order to avoid the vulnerability to attack and load unbalance. Specifically, by using the Shannon entropy [30], the local randomness of node i is denoted by

$$h_i = -\sum_{j\in S_i} q_{ij} \log q_{ij}. \tag{9}$$

Obviously, the uniform distribution achieves the maximum local randomness for each node. Let us consider a random walk with starting node k. The randomness of such a random walk is defined as the sum of weighted local randomness of nodes in the random walk:

$$H_k = \sum_{i=1}^{n-1} f_{ki} h_i = -\sum_{i=1}^{n-1} f_{ki} \sum_{j\in S_i} q_{ij} \log q_{ij}. \tag{10}$$

Eventually, the expected randomness of a random walk is given by

$$H = \sum_{k=1}^{n-1} p_k H_k = -\sum_{k=1}^{n-1} p_k \sum_{i=1}^{n-1} f_{ki} \sum_{j\in S_i} q_{ij} \log q_{ij}. \tag{11}$$

The expected randomness of a random walk measures the uncertainty of the measurement that is collected through this random walk.

Recall that the goal is to estimate the transition probability matrix $\mathbf{Q}$ such that the expected cost of a random walk is minimized. Specifically, given the expected randomness of any random walk H and the probabilities of each node being the starting node $\mathbf{p} = [p_1, p_2, \cdots, p_{n-1}]^T$, the problem can be formulated as follows.

$$\min_{\mathbf{Q}} \sum_{i=1}^{n-1} p_i c_i, \tag{12}$$

$$s.t. \sum_{j\in S_i} q_{ij}\left(w_{ij} + c_j\right) = c_i, i \in [n-1], \tag{13}$$

$$-\sum_{k=1}^{n-1} p_k \sum_{i=1}^{n-1} f_{ki} \sum_{j\in S_i} q_{ij} \log q_{ij} = H, \tag{14}$$

$$0 \le q_{ij} \le 1, \tag{15}$$

$$\sum_{j\in S_i} q_{ij} = 1, i \in [n-1], \tag{16}$$

where constraint (13) shows how to compute the cost of a random walk with a given starting node, constraint (14) guarantees the uncertainty of the collected measurements, and constraint (16) states that the sum of the probabilities of selecting successors must be one.

In order to save the energy consumption and balance network loads, we relate the energy efficiency issue to the weight of edges and p_i's. The idea is to assign smaller edge weight and larger starting probability to nodes that contain more residual energy. Specifically, let w_{ij} and p_i be functions of r_i where $r_i := E_r(i)/E_0$ is defined as the proportion of the residual energy of node i normalized by the initial energy E_0. Suppose the starting probability of node i in a random walk is proportional to r_i, i.e., $p_i = \alpha r_i$, where α is a constant. In order to calculate the constant α, let us recall that the sink needs to collect m measurements so that the data vector can be precisely recovered. This means that m random walks are required for the data recovery. Since p_i is also the expected number of random walks that starts at node i, we have that $\sum_{i=1}^{n-1} p_i = m$. Therefore, the constant α is given by

$$\alpha = \frac{1}{m}\sum_{i=1}^{n-1} r_i. \tag{17}$$

The similar idea is also applied to the computation of edge weights. For a node i, we assign larger weight to the edge which is connected to the successor with smaller proportion of the residual energy. The weight of transmitting data from node i to node j is defined as

$$w_{ij} = \frac{1}{r_j}\sum_{j\in S_i} r_j, j \in S_i. \tag{18}$$

4. Minimum Expected Cost Data Collection

In this section, we propose a network layer-based Minimum Expected Cost Data Collection (MECDC) scheme by using the absorbing Markov chain model. The MECDC scheme is consisted of two phases. In the Phase I, the transition probability matrix **Q** is calculated, and the sensing matrix **A** is constructed based on **Q**. In the Phase II, measurements are collected by applying random walks with the transition probability matrix **Q**. After receiving enough number of measurements, the sink is able to recover the original data vector by using the NSRP decoder with the sensing matrix **A** [21, 23].

4.1. Solution of the Optimization Problem. Let us first discuss how to derive the transition probability matrix **Q**. By leveraging the idea in the reference [23], we apply the Lagrange multiplier method to iteratively update transition probabilities. The Lagrange for the optimization problem is given by

$$\begin{aligned} L = {} & \sum_{i=1}^{n-1} p_i c_i + \sum_{i=1}^{n-1} \lambda_i \left[c_i - \sum_{j\in S_i} q_{ij}\left(w_{ij} + c_j\right)\right] + \sum_{i=1}^{n-1} \mu_i \left(\sum_{j\in S_i} q_{ij} - 1\right) \\ & + \eta\left[\sum_{k=1}^{n-1} p_k \sum_{i=1}^{n-1} f_{ki} \sum_{j\in S_i} q_{ij}\log q_{ij} + H\right], \end{aligned} \tag{19}$$

where λ_i, μ_i, and η are the Lagrangian multipliers.

By setting $\partial L/\partial q_{kl} = 0$, we have that

$$-\lambda_k(w_{kl} + c_l) + \eta(\log q_{kl} + 1)\sum_{i=1}^{n-1} p_i f_{ik} + \mu_k + \eta\sum_{i=1}^{n-1} p_i \sum_{j=1}^{n-1} \frac{\partial f_{ij}}{\partial q_{kl}} h_j = 0, \tag{20}$$

where $h_j = -\sum_{k\in S_j} q_{jk}\log q_{jk}$. After some simple manipulations, we obtain that

$$q_{kl} = \exp\left\{\frac{\lambda_k(w_{kl} + c_l)}{\eta\beta_k} - \frac{\xi_{kl}}{\beta_k} - \frac{\mu_k}{\eta\beta_k} - 1\right\}, \tag{21}$$

where $\beta_k = \sum_{i=1}^{n-1} p_i f_{ik}$ and

$$\xi_{kl} = \sum_{i=1}^{n-1} p_i \sum_{j=1}^{n-1} \frac{\partial f_{ij}}{\partial q_{kl}} h_j. \tag{22}$$

Applying equation (21) for $l \in S_k$ to the fact that $\sum_{l\in S_k} q_{kl} = 1$, we have that

$$\exp\left\{-\frac{\mu_k}{\eta\beta_k} - 1\right\} = \left(\sum_{l\in S_k}\exp\left\{\frac{\lambda_k(w_{kl} + c_l)}{\eta\beta_k} - \frac{\xi_{kl}}{\beta_k}\right\}\right)^{-1}. \tag{23}$$

Substituting equation (23) into equation (21), we obtain that

$$q_{kl} = \frac{\exp\left\{(\lambda_k(w_{kl} + c_l)/\eta\beta_k) - \xi_{kl}/\beta_k\right\}}{\sum_{l\in S_k}\exp\left\{(\lambda_k(w_{kl} + c_l)/\eta\beta_k) - \xi_{kl}/\beta_k\right\}}. \tag{24}$$

In order to update q_{kl} for a given **Q**, we need to compute parameters λ_k and ξ_{kl}. Setting $\partial L/\partial c_k = 0$, we have that

$$\lambda_k = \begin{cases} -p_k, & if k \in L_t, \\ -p_k - \sum_{r\in D_k} \lambda_r q_{rk}, & \text{otherwise.} \end{cases} \tag{25}$$

Thus, the Lagrange multiplier λ_k can be computed layers by layers.

Next, we compute $\xi_{kl} = \sum_{i=1}^{n-1} p_i \sum_{j=1}^{n-1}(\partial f_{ij}/\partial q_{kl})h_j$. Denote $L(i)$ the layer of node i. For different nodes i, j, and k, three cases may occur: (1) $L(k) > L(i) > L(j)$; (2) $L(i) > L(j) > L(k)$; (3) $L(i) > L(k) > L(j)$. Note that $f_{ij} = 0$ if $L(i) \le L(j)$. We observe that $\partial f_{ij}/\partial q_{kl} = 0$ if cases (1) and (2) occur. Under the case (3), we have that

$$\frac{\partial f_{ij}}{\partial q_{kl}} = f_{ik} f_{lj}. \tag{26}$$

By substituting equation (26) into equation (22), we obtain that

$$\xi_{kl} = \sum_{i=1}^{n-1} p_i f_{ik} \sum_{j=1}^{n-1} f_{lj} h_j. \tag{27}$$

Given a guess of **Q**, transition probabilities can be updated based on equation (24). Note that it is impossible to obtain an analytical expression of the Lagrange multiplier η [23]. It controls the degree of randomness of a random walk. Larger value of η implies larger degree of randomness. Algorithm 1 shows how to compute the transition probability matrix **Q** iteratively. In line 5 of the Algorithm 1, ε represents the threshold of the stopping criterion.

The sensing matrix **A** can be constructed based on **Q**. Given **Q**, the fundamental matrix is given by $\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}$. Then, each entry in **A** is identically and independently drawn from the following distribution

```
Input the graph $G = (V, E)$, the weight matrix $\mathbf{W}$, the starting probabilities of nodes $p_1, p_2, \cdots, p_{n-1}$ and the randomness of random walks $\eta$.
Compute layers $L_1, L_2, \cdots, L_T$, the successors set $S_i$ and the predecessors set $D_i$ for every node $i \in [n-1]$.
Output an estimator of $\mathbf{Q}$.
Initialize the step index $t = 0$ and $\mathbf{Q} = \mathbf{Q}^0$ such that.
    $q_{ij}^0 = \{(1/|S_i|),\ if\, i \notin L_1, j \in S_i, 0,\ otherwise,$
where $q_{ij}^0$ is the $ij$-th entry of $\mathbf{Q}^{0.}$
while $\max_{q,l} |q_{kl}^t - q_{kl}^{t-1}/q_{kl}^{t-1}| \geq \varepsilon$ do
    Compute $\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}$.
    Compute $c_i = \sum_{j \in S_i} q_{ij}(w_{ij} + c_j)$ for any $i \in [n-1]$.
    Compute $\lambda_k$ based on equation (25) in a layer-by-layer manner.
    Compute $\beta_j = \sum_{i=1}^{n-1} p_i f_{ij}$ for any $j \in [n-1]$.
    Compute $h_j = -\sum_{k \in S_j} q_{jk} \log q_{jk}$ for any $j \notin L_1$.
    Compute $\xi_{kl}$ based on equation (27) for any $k \in [n-1], l \in S_k$.
    Update $q_{kl}^t$ based on equation (24) for any $k \in [n-1], l \in S_k$.
    Update the step index $t = t + 1$.
end while
```

Algorithm 1: Iteratively solve for transition probabilities.

$$a_{ij} = \begin{cases} +1, & \text{with probability } \dfrac{\pi_j}{2}, \\ -1, & \text{with probability } \dfrac{\pi_j}{2}, \\ 0, & \text{with probability } 1 - \pi_j, \end{cases} \tag{28}$$

where $\pi_j = \sum_{i=1}^{n-1} p_i f_{ij}$.

After obtaining the transition probability matrix $\mathbf{Q}$, Phase II collects measurements through random walks. Except for the sink, any node i starts a random walk with probability p_i. Then, packets are transmitted towards the sink in a layer-by-layer manner as stated in Section 3.2. Given $\mathbf{A}$, $m = O(k^2 \log n)$ measurements are sufficient for the sink to recover the original data vector by using the NSRP-based estimator [21, 23] .

4.2. Distributed Realization of MECDC. In this subsection, we show that the update of transition probabilities can be realized locally and distributively. In other words, q_{kl} can be computed only using information of neighbors. In order to see this, let us consider the node k. Suppose each node knows the probability of being the starting node. Then, all of the parameters which are required to update q_{kl} can be computed based on the neighboring nodes as follows.

(i) λ_k. From the equation (25), λ_k can be calculated by using the information of predecessors. Specifically, $\lambda_k = -p_k$ for any $k \in L_T$. Then, λ_k for any $k \in L_i$ can be computed based on nodes in $D_k \subseteq L_{i+1}$. In such a manner, the values of λ_k can be computed layers by layers

(ii) β_k. Similar to λ_k, the value of β_k can be computed based on predecessors of node k. Recall that $\beta_k = \sum_{i=1}^{n-1} p_i f_{ik}$, which can be rewritten as

$$\beta_k = \begin{cases} 0, & if\, k \in L_T, \\ p_k + \sum\limits_{i \in D_k} \beta_i q_{ik}, & \text{otherwise.} \end{cases} \tag{29}$$

Therefore, starting from the layer L_T, the values of β_k can be obtained layers by layers.

(i) h_k. Recall that $h_k = -\sum_{l \in S_k} q_{kl} \log q_{kl}$, which can be computed based on successors of node k

(ii) ξ_{kl}. Denote $g_l = \sum_{j=1}^{n-1} f_{lj} h_j$. In order to obtain ξ_{kl}, we first compute g_l for each $l \in [n-1]$. Based on the information of successors, we obtain that $g_l = \sum_{j \in S_l} q_{lj} g_j$. Thus, starting from the layer L_1, the parameter g_l can be computed layers by layers as follows:

$$g_l = \begin{cases} 0, & if\, l \in L_1, \\ h_l, & if\, l \in L_2, \\ \sum\limits_{j \in S_l} q_{lj} g_j, & \text{otherwise.} \end{cases} \tag{30}$$

Based on the values of g_l and the equation (27), we have that $\xi_{kl} = \beta_k g_l$.

In summary, the computation of transition probabilities can be realized distributively by saving the information of parameters λ_k, β_k, h_k, and g_k in every node.

5. Simulation Results

In this section, we numerically evaluate the performance of the proposed scheme with the baseline scheme. Suppose n nodes are uniformly and randomly deployed in a unit square area. The sink is located at the top right corner. There exists

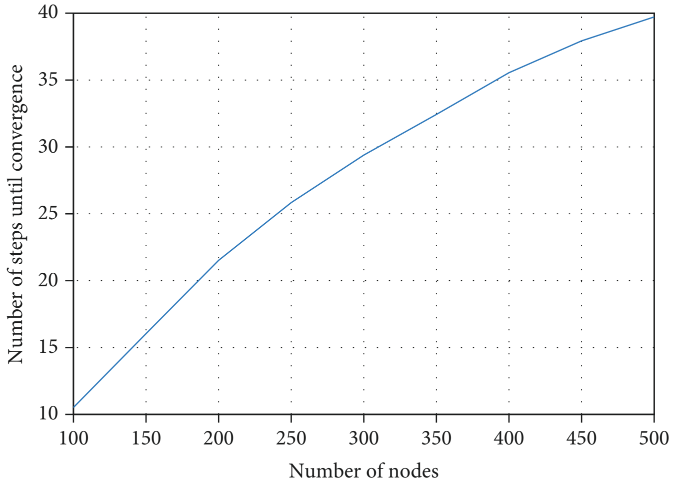

Figure 3: Number of iterations until the Algorithm 1 converges versus the number of nodes in the network. The threshold of the stopping criterion is set to $\varepsilon = 0.1$.

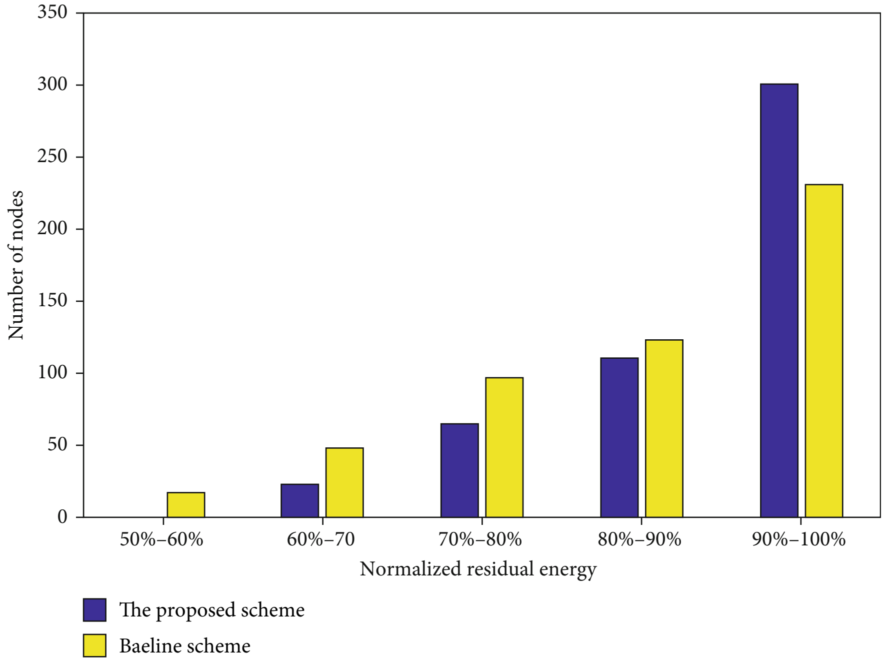

Figure 4: Distribution of the normalized residual energy of nodes in the network.

an edge between two nodes if the distance between these two nodes is not greater than the communication range 0.2. We assume that $m = k^2 \log n$ measurements are required to recover the data vector. The sparsity of the data vector is set to $k = 5$. Initially, each node is equipped with an identical battery that contains 100 joules of the energy. For simplicity, we assume that a packet transmission consumes 0.1 joules of the energy. In the baseline scheme, except for the sink, each node is selected as the starting node of a random walk with probability $p = m/(n-1)$. In a random walk, each node transmits data to its successors with an identical probability, i.e., $q_{ij} = 1/|S_i|$ for any $j \in S_i$. In the proposed scheme, we first compute the starting probability of every node and the transition probability matrix $\mathbf{Q}$ at the beginning of collecting every sample. Then, measurements are collected through random walks with the obtained parameters.

Let us first look at the convergence speed of Algorithm 1. Figure 3 shows the number of iterations until the proposed algorithm converges when the network size increases. In Figure 3, we set the threshold of the stopping criterion $\varepsilon = 0.1$. We observe that Algorithm 1 converges very fast when the network size is not large. The convergence rate increases as the network size increases. One possible reason is that the candidate edge (k, l) with $|(q_{kl}^t - q_{kl}^{t-1})/q_{kl}^{t-1}|$ achieving the maximum value increase as the network size increases. Furthermore, we observe that the slope of the convergence rate

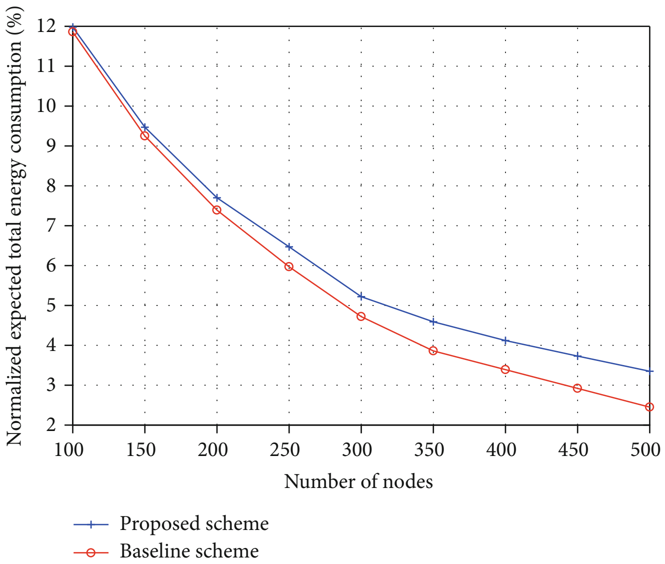

Figure 5: Comparison of the expected total energy consumption between the proposed scheme and the baseline scheme.

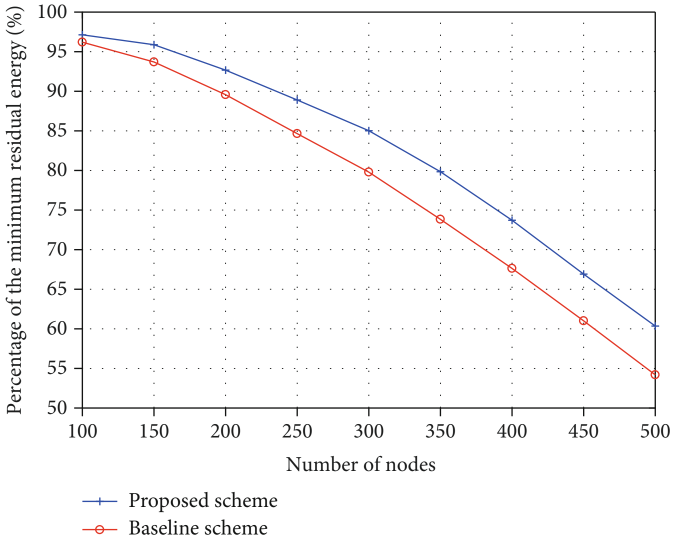

Figure 6: Comparison of the minimum residual energy among nodes when the number of nodes increases.

curve decreases as the network size increases. This implies that there may exist an upper bound for the convergence rate of Algorithm 1.

Next, we compare the energy efficiency between the proposed scheme and the baseline scheme. Figure 4 shows the distributions of the normalized residual energy of nodes in the network. In Figure 4, we set the network size to $n = 500$, and the residual energy is obtained after 50 samples are collected. Note that the residual energy is normalized based on the initial energy. We observe that the residual energy of nodes in both of the two schemes mainly concentrates on the interval 90%-100%. This is because CDG-based schemes can balance network loads. However, the number of nodes with large residual energy in the proposed scheme is more than that in the baseline scheme. This demonstrates that the proposed scheme is able to further reduce the energy consumption and balance the network loads.

Figure 5 compares the normalized expectation of the total energy consumption between the proposed scheme and the baseline scheme. In Figure 5, the residual energy is computed after 50 samples are collected. The expectation of the total energy consumption is normalized based on the initial total energy of nodes in the whole network. We first observe that, for a fixed number of nodes, the normalized expected total energy consumption in the proposed scheme is smaller than that in the baseline scheme. This demonstrates that the proposed scheme is able to reduce the total energy consumption by considering the residual energy of nodes and optimizing the transition probability matrix. Another observation is that, as the number of nodes increases, the normalized

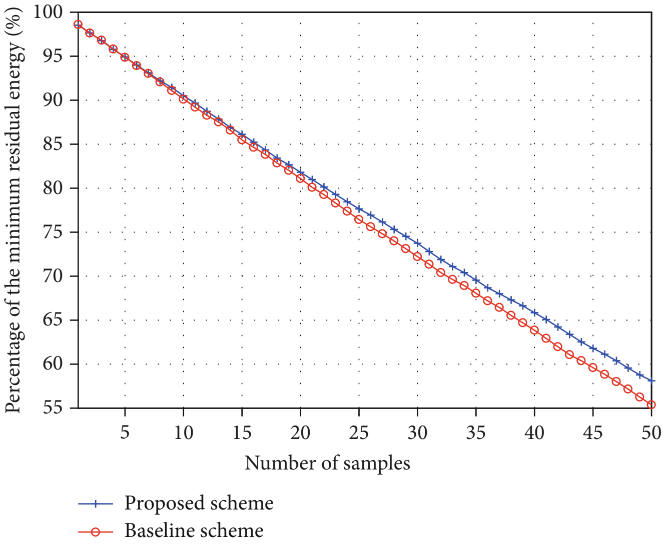

Figure 7: Comparison of the minimum residual energy among nodes when the number of collected samples increases.

expected total energy consumption decreases in both of the two schemes. This implies that collecting a fixed number of samples consumes less normalized total energy for large-scale networks. In other words, the proposed scheme is more suitable for large-scale networks. Finally, we observe that the gap of the normalized expected total energy consumption between the proposed scheme and the baseline scheme increases as the number of nodes increases. This also implies that the proposed scheme performs better in large-scale networks.

Next, let us consider the minimum residual energy of nodes in the network. Large minimum residual energy implies balanced load of the network. Figure 6 compares the minimum residual energy of nodes between the proposed scheme and the baseline scheme. The residual energy is computed after 50 samples are collected. Similar to Figure 5, the residual energy is normalized based on the initial energy. A direct observation is that the minimum residual energy in the proposed scheme is larger than that in the baseline scheme. This demonstrates that the proposed scheme is able to balance the network load. Another observation is that the gap between the proposed scheme and the baseline scheme increases as the number of nodes increases, which suggests that the proposed scheme is more suitable for large-scale networks.

Figure 7 compares the minimum residual energy of nodes when the number of collected samples increases. In Figure 7, we set the number of nodes $n = 500$. In order to collect a sample/data vector, the sink needs to collect m measurements so that the data vector can be precisely recovered. The residual energy is normalized based on the initial energy. We observe that, for a fixed number of samples, the minimum residual energy of the proposed scheme is larger than that of the baseline scheme. This is because the proposed scheme is able to balance loads of the network. Furthermore, the gap of the minimum residual energy between the proposed scheme and the baseline scheme increases as the number of collected samples increases. This demonstrates that the proposed scheme is more suitable for long-running networks.

6. Conclusions

In this paper, we studied the data collection problem in WSNs. Random walks and the compressive sensing technology with nonuniform sparse random matrices are adopted to collect measurements. Each measurement is collected through a random walk which is modeled as an absorbing Markov chain. By exploiting the residual energy of nodes, we formulate the process of collecting measurements as an optimization problem, which seeks to find optimal transition probabilities of nodes so that the expected cost is minimized. An iterative method, which is referred as the Minimum Expected Cost Data Collection (MECDC) scheme, is proposed to solve this optimization problem and collect measurements. A distributed realization of MECDC, where only local information is needed in the collection of measurements, is proposed. Simulation results show that the proposed scheme not only reduces the energy consumption but also balances the network loads.

Abbreviations

WSNs:	Wireless sensor networks
CS:	Compressive sensing
MECDC:	Minimum Expected Cost Data Collection
IoT:	Internet of Things
CDG:	Compressive data gathering
NSRP:	Nonuniform sparse random projection
ODECS:	On-Demand Explosion-Based Compressive Sensing.

Data Availability

Data settings can be found in the draft.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61701441 and 61801427 and 2020 Undergraduate Training Program for Innovation and Entrepreneurship of Yunnan Province (202010689042).

References

[1] X. Xue and J. Chen, "Optimizing sensor ontology alignment through compact co-firefly algorithm," *Sensors*, vol. 20, no. 7, p. 2056, 2020.

[2] B. Rashid and M. H. Rehmani, "Applications of wireless sensor networks for urban areas: a survey," *Journal of Network and Computer Applications*, vol. 60, pp. 192–219, 2016.

[3] X. Xue and J. Chen, "Using compact evolutionary Tabu search algorithm for matching sensor ontologies," *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.

[4] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[5] X. Xue and J. S. Pan, "A compact co-evolutionary algorithm for sensor ontology meta-matching," *Knowledge and Information Systems*, vol. 56, no. 2, pp. 335–353, 2018.

[6] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in internet of things: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 1–27, 2018.

[7] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of things (IoT): a vision, architectural elements, and security issues," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 1–27, Palladam, India, 2017.

[8] X. Yu and S. J. Baek, "Joint routing and scheduling for data collection with compressive sensing to achieve order-optimal latency," *International Journal of Distributed Sensor Networks*, vol. 13, no. 10, 2017.

[9] X. Yu and S. J. Baek, "Energy-efficient collection of sparse data in wireless sensor networks using sparse random matrices," *ACM Transactions on Sensor Networks*, vol. 13, no. 3, pp. 1–36, 2017.

[10] P. Zhang, S. Wang, K. Guo, and J. Wang, "A secure data collection scheme based on compressive sensing in wireless sensor networks," *Ad Hoc Networks*, vol. 70, pp. 73–84, 2018.

[11] M. Tuan Nguyen, K. A. Teague, and N. Rahnavard, "CCS: energy-efficient data collection in clustered wireless sensor networks utilizing block-wise compressive sensing," *Computer Networks*, vol. 106, pp. 171–185, 2016.

[12] S. Li, L. Da Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2177–2186, 2013.

[13] M. A. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sensors*, vol. 14, no. 2, pp. 2822–2859, 2014.

[14] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*, pp. 145–156, Beijing China, 2009.

[15] J. Luo, L. Xiang, and C. Rosenberg, "Does compressed sensing improve the throughput of wireless sensor networks?," in *2010 IEEE International Conference on Communications*, pp. 1–6, Cape Town, South Africa, 2010.

[16] D. Ebrahimi and C. Assi, "On the interaction between scheduling and compressive data gathering in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2845–2858, 2016.

[17] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," in *2007 6th International Symposium on Information Processing in Sensor Networks*, pp. 331–339, Cambridge, MA, USA, 2007.

[18] H. Zheng, F. Yang, X. Tian, X. Gan, X. Wang, and S. Xiao, "Data gathering with compressive sensing in wireless sensor networks: a random walk based approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 35–44, 2015.

[19] V. K. Singh, S. Verma, and M. Kumar, "ODECS: an on-demand explosion-based compressed sensing using random walks in wireless sensor networks," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2466–2475, 2019.

[20] P. Zhang and J. Wang, "On enhancing network dynamic adaptability for compressive sensing in WSNs," *IEEE Transactions on Communications*, vol. 67, no. 12, pp. 8450–8459, 2019.

[21] X. Y. Liu, Y. Zhu, L. Kong et al., "*CDC*: compressive data collection for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2188–2197, 2015.

[22] P. Zhang, J. Wang, and K. Guo, "Compressive sensing and random walk based data collection in wireless sensor networks," *Computer Communications*, vol. 129, pp. 43–53, 2018.

[23] J. Huang and B. H. Soong, "Cost-aware stochastic compressive data gathering for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1525–1533, 2019.

[24] H. Zheng, W. Guo, and N. Xiong, "A kernel-based compressive sensing approach for mobile data gathering in wireless sensor network systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 12, pp. 2315–2327, 2018.

[25] M. Rani, S. B. Dhok, and R. B. Deshmukh, "A systematic review of compressive sensing: concepts, implementations and applications," *IEEE Access*, vol. 6, pp. 4875–4894, 2018.

[26] Z. Qin, J. Fan, Y. Liu, Y. Gao, and G. Y. Li, "Sparse representation for wireless communications: a compressive sensing approach," *IEEE Signal Processing Magazine*, vol. 35, no. 3, pp. 40–58, 2018.

[27] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 937–947, 2010.

[28] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best $_k$-term approximation," *Journal of the American Mathematical Society*, vol. 22, no. 1, pp. 211–231, 2009.

[29] C. M. Grinstead and J. L. Snell, *Introduction to Probability*, American Mathematical Society, Providence, RI, USA, 2012.

[30] M. Saerens, Y. Achbany, F. Fouss, and L. Yen, "Randomized Shortest-Path problems: two related models," *Neural Computation*, vol. 21, no. 8, pp. 2363–2404, 2009.