

Article

Two-Tier PSO Based Data Routing Employing Bayesian Compressive Sensing in Underwater Sensor Networks

Xuechen Chen ^{1,*} , Wenjun Xiong ² and Sheng Chu ²

¹ School of Computer Science and Engineering, Central South University, Changsha 410083, China

² School of Electronics and Information Technology, Sun Yat-Sen University, Guangzhou 510275, China; xiongwj3@mail2.sysu.edu.cn (W.X.); chusheng@mail.sysu.edu.cn (S.C.)

* Correspondence: chenxuec@csu.edu.cn

Received: 10 September 2020; Accepted: 15 October 2020; Published: 21 October 2020

Abstract: Underwater acoustic sensor networks play an important role in assisting humans to explore information under the sea. In this work, we consider the combination of sensor selection and data routing in three dimensional underwater wireless sensor networks based on Bayesian compressive sensing and particle swarm optimization. The algorithm we proposed is a two-tier PSO approach. In the first tier, a PSO-based clustering protocol is proposed to synthetically consider the energy consumption and uniformity of cluster head distribution. Then in the second tier, a PSO-based routing protocol is proposed to implement inner-cluster one-hop routing and outer-cluster multi-hop routing. The nodes selected to constitute i -th effective routing path decide which positions in the i -th row of the measurement matrix are nonzero. As a result, in this tier the protocol comprehensively considers energy efficiency, network balance and data recovery quality. The Bayesian Cramér-Rao Bound (BCRB) in such a case is analyzed and added in the fitness function to monitor the mean square error of the reconstructed signal. The experimental results validate that our algorithm maintains a longer life time and postpones the appearance of the first dead node while keeps the reconstruction error lower compared with the cutting-edge algorithms which are also based on distributed multi-hop compressive sensing approaches.

Keywords: Bayesian compressive sensing; particle swarm optimization; three dimensional underwater wireless sensor network; Bayesian Cramér-Rao Bound

1. Introduction

1.1. Motivation

In recent decades, the research on wireless sensor networks (WSNs) has been a hotspot for numerous applications, such as, environment monitoring [1], smart cities [2,3], military monitoring [4], maritime resource gathering [5] and health-care surveillance [6].

As maritime rights and interests are paid more and more attention, underwater wireless sensor networks (UWSNs), an important extension of WSNs into ocean, is of great value in various application fields, such as, oceanographic information collection, hydrological and environmental monitoring, resources exploration, disaster forecast, underwater navigation, military defense, and etc. [7]. For UWSNs, there exist two kinds of topological structures, two dimensional network and three dimensional network. In this work, we study the three dimensional UWSNs, with which all nodes are deployed at different depths in the ocean.

Generally, in WSNs, including UWSNs, the information from all sensor nodes, which are equipped with limited power battery, are gathered to the sink node. When the power of the node is used up, the

node expires. While most nodes died, the sink node is no longer able to gain necessary information from common nodes. Considering the fact that the nodes are hard to recharge, energy consumption remains a key factor that affects the whole network lifetime. On the other hand, in underwater scenario, as the network usually occupies large area and the effective transmission semidiameter is limited, those nodes who might not transfer the signal to sink node in one hop will need relay nodes. As a result, to design an efficient data routing algorithm for UWSN, it is vital to reduce the processing complexity and traffic overhead and meanwhile ensure that the data are collected efficiently. Wang et al. [8] render a layering algorithm which divides nodes to different parts according to their depth and data are forwarding based on opportunistic directional forwarding strategy. Guan et al. [9] improved opportunistic routing by adding distance-vector. The algorithm uses a query mechanism to establish the distance vectors for sensor nodes, then opportunistic routing is developed to forward packet based on the distance vectors. Further studies tend to cluster all nodes in the network, and only transmit the merged message in the cluster head to the sink node. Common nodes just need to transmit their data to their cluster heads (CHs) in one hop. For instance, clustering is added as one stage in [10] to save energy better. This strategy cuts down the energy-consumption efficiently. In [11], clustering and CH transmission are introduced as well. After the formation of cluster and selection of CH, the courier nodes are deployed into the clusters to collect the aggregated data from the CHs, and the visiting tour is scheduled based on the Decisional Welzl's algorithm.

However, the works above do not consider the facts that in UWSN, while the sensor nodes are deployed to measure environmental data, for example, maritime temperatures, the signals they collect usually possess spatial correlations that induce sparsity into practical application. Based on such a fact, we comprehensively consider the routing strategy and the data sparsity to save energy even further. In such contexts, compressive sensing (CS) is adopted to reduce communication cost while also preserve information precisely. In this paper, we propose a CS based schedule which simultaneously considers routing and nodes selection for overall data reconstructions at the sink node for 3-D UWSNs. We use Particle Swarm Optimization (PSO) to optimize both the clustering and routing protocols design while the factors consisting of fitness functions consist of signal recovery quality, energy consumption, network balance and etc..

1.2. Related Work

CS, which unifies sampling and compression, has been enthusiastically promoted and studied [12,13]. It states that when the original signal is sparse itself or sparse on some basis, we can reconstruct it from compressed measurements. Making use of its advantages on simple encoding and efficient compressing, many researchers apply CS in WSN to achieve advanced compression compared with conventional compression algorithms [14–16]. CS could also be utilized for sensor selection; that is, only part of nodes will be chosen to transmit the product of its data with a random coefficient to the sink [17–19]. In the context that all nodes transmitted their data directly to the sink node in one hop, Hwang et al. [20] propose both centralized and decentralized algorithms for sensor selection under multi-variated noise condition. In [21], a tree-based algorithm named CDG is designed to reduce the payload falling on nodes close to the BS. In [22,23], a WSN is partitioned into clusters. Sensor readings are sent to CHs and the CHs send the received data to the BS. However, the measurement matrices they adopt are all full Gaussian ones which consume more power than the sparse measurement matrix used in our work. In the other word, they do not combine the routing and nodes selection with the help of CS.

PSO is a computational method that can be used to improve the optimization of candidate solutions through iteration. The way to adopt PSO for data routing [24] has been studied in several literatures [25–27]. Ref. [25] proposes a novel bi-velocity discrete PSO approach which extends PSO from the continuous version to the binary or discrete domain to find routing paths for WSN. Tukisi et al. [26] presents a PSO based method to find routings for the network with energy harvesting. Targeting on the problem that some of the nodes would be left out during cluster formation, ref. [27]

introduces to apply the concepts of PSO and gravitational search algorithm to prevent residual nodes. All of these studies consider the application of PSO either in the clustering or in the routing, i.e., only one tier. And they do not consider the application of CS.

To the best of our knowledge, there are few works besides the following literatures that have considered the application of CS into combination of data routing and nodes selection in WSNs, especially in UWSNs. In [28], the network is partitioned into clusters. Then each CH collects the sensor readings within its cluster and generates CS measurements to be forwarded directly to the base station. As a result, the overall CS measurement matrix at the base station is a block diagonal matrix. Later on, in [29], an improved method is introduced which allows the multi-hop routing of generated CS measurements through intermediate CHs. The authors generalize the two methods in [30]. Compared to our work, they obtain CS measurements in each CH and then forward to sink node while we postpone this operation to be done in the sink node to reduce the burden in CHs. On the other hand, while choosing multi-hop routing paths from the CH to the sink node, they use traditional tree-based algorithm while we design a PSO-optimization based scheme. Specifically targeting on three dimensional UWSNs as well, distributed multi-hop CS has been proposed by Gong et al. in [31]. This algorithm randomly chooses a subset with size M of N nodes in the first place, then each chosen node finds a tour to the sink node. During the traveling of the message through the tour, each node computes the product of its sensing data and a random weighted coefficient and adds the product to the intermediate result it received. Finally the sink node obtains M measurements and reconstructs the whole data. Compared to our work, they ignore the noise and select the nodes which participate in sensing by Bernoulli generator. To conclude, though these works use sparse measurement matrices similar to what we do, they only consider energy efficiency while choosing appropriate routs without considering the sensing quality. Furthermore, they do not consider the optimization of clustering.

In this work, we introduce a two-tier PSO protocol for both clustering and routing. Specifically, our contributions are listed as below.

- (1) In the first tier, a PSO-based clustering protocol is proposed to find appropriate CHs with the comprehensive consideration of energy consumption efficiency and CH distribution uniformity.
- (2) In the second tier, a PSO-based routing protocol is introduced to find appropriate routs. A routing path consists of inner-cluster one-hop routing and outer-cluster multi-hop routing and each path corresponds to one cluster. Hence the number of paths equals to the number of clusters. It is worth mention that the multi-hop routing in our scheme is not restricted to consisting of CHs only. Note that these paths correspond to the rows of the measurement matrix for CS. In this way, we implement the combination of routing and nodes selection based on CS. Hence, the fitness function of PSO-based routing protocol comprises of several factors, including the criterions for energy efficiency, network balance, and data recovery qualities. To measure network balance, we divide the nodes in the 3-D model into different layers in accordance with their horizontal distance to the sink node. As we adopt Bayesian CS (BCS), Bayesian Cramér-Rao Bound (BCRB) is added in the fitness function to reflect the recovery performance.
- (3) With optimization in CH and routing chosen progress, the network could survive longer with relatively lower measurement error as demonstrated by the simulation results. What's more, taking energy-balancing into consideration contributes to form a more balanced energy-consuming network every round cycle, and also prolongs the network lifetime and reduces the sensing error.

The remainder of this paper is organized as follows. In Section 2 we present preliminary knowledge. In Section 3, we introduce our system model for UWSN scenario. The detailed descriptions of the clustering and routing algorithms based on CS and PSO are given in Section 4. The simulation results are presented in Section 5. Finally, Section 6 concludes the paper, and introduces the future work.

The notation used in this paper is according to the convention. Symbols for matrices and vectors are in boldface.

2. Preliminary

2.1. Compressive Sensing

CS theory proves that if a signal \mathbf{z} of dimension N is sufficiently sparse in a certain domain; that is $\mathbf{z} = \mathbf{\Psi}\mathbf{x}$ and $\|\mathbf{x}\|_0 = K, K \ll N$, where $\mathbf{\Psi} \in \mathbb{R}^{N \times N}$ is an orthogonal basis and $\mathbf{x} \in \mathbb{R}^N$, it is efficient to recover \mathbf{z} from a random measurement vector that is obtained by $\mathbf{y} = \mathbf{\Phi}\mathbf{z}$. M , the length of \mathbf{y} is much smaller than N . $\mathbf{\Phi}$ is named as a measurement matrix or sensing matrix. In the sequel, the random non-zero coefficients in $\mathbf{\Phi}$ obey Gaussian distributions.

In general, it is an ill-posed problem to recover signal \mathbf{z} from \mathbf{y} since \mathbf{y} has much smaller dimensions than \mathbf{z} . We have to minimize the solution's ℓ_0 norm to find the sparsest result. However, it is an NP-hard problem, which is not capable to be represented by mathematical formulas. Candés et al. [13] state that as long as a measurement matrix satisfies the restricted isometry property (RIP) condition, the ℓ_0 norm can be replaced by ℓ_1 norm. A measurement matrix satisfies RIP of order of K when

$$(1 - \epsilon) \|\mathbf{x}\|_2 \leq \|\mathbf{\Phi}\mathbf{\Psi}\mathbf{x}\|_2 \leq (1 + \epsilon) \|\mathbf{x}\|_2 \quad (1)$$

for all $\|\mathbf{x}\|_0 \leq 2K$ and some $0 < \epsilon < 1$. Afterwards, many methods for such a convex optimization problem were introduced to solve the ℓ_1 norm optimization problem [32–34]. Later on, iterative-greedy-pursuit recovery algorithms [35–37] are proposed to reduce the time complexity.

2.2. Bayesian Estimation

In the presence of noise, the noisy measurement vector \mathbf{y} is

$$\mathbf{y} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{x} + \mathbf{n} = \mathbf{\Theta}\mathbf{x} + \mathbf{n}. \quad (2)$$

where $\mathbf{n} \in \mathbb{R}^M$ denotes the noise.

In the underwater acoustic channel, the overall noise at the receiver contains both the ambient noise and residual intersymbol interference (ISI) after equalization. Though the ambient noise is not AWGN in the underwater environment, the use of Gaussian approximation is often convenient and adequate [38]. In the presence of Gaussian ambient noise, according to [39] and [40], the overall noise can be treated as Gaussian, although it is not independent with regard to time step, since the signal after CS transform, i.e., transmitted signal is treated as Gaussian distributed. Hence, in the sequel, \mathbf{n} is approximated as Gaussian distributed.

In [41], BCS is proposed to reconstruct a sparse signal from noisy measurements, if a statistical characterization of the signal is available. The computation time of BCS is comparable to or even faster than other classical greedy iteration algorithms. In this work, we adopt bayesian learning to recover the signal in the sink node. Therefore in this part, we make a brief review of BCS method.

n_i indicates the i -th element of \mathbf{n} in Equation (2), and is approximated as a zero mean Gaussian variable with variance σ^2 . From the Bayesian point of view, to obtain the restoration of the original signal is to seek a full posterior probability function for \mathbf{x} . Usually, a Laplacian distribution is a widely used prior for sparse signals. But considering it is not conjugate to Gaussian likelihood, in BCS, a hierarchical prior has been defined for \mathbf{x} ; that is, a zero-mean Gaussian prior is defined on each element of \mathbf{x}

$$p(\mathbf{x}|\boldsymbol{\alpha}) = \prod_{i=1}^N \mathcal{N}(x_i|0, \alpha_i^{-1}), \quad (3)$$

where α_i is the inverse-variance of x_i and $\boldsymbol{\alpha}$ satisfies a Gamma prior with parameters a and b with $p(\boldsymbol{\alpha}) = \prod_{i=1}^N \Gamma(\alpha_i|a, b)$. By marginalizing over the hyperparameter $\boldsymbol{\alpha}$, the overall prior is given by

$$p(\mathbf{x}|a, b) = \prod_{i=1}^N \int_0^{\infty} \mathcal{N}(x_i|0, \alpha_i^{-1}) \Gamma(\alpha_i|a, b) d\alpha_i. \quad (4)$$

Likewise, the hyperparameter $\alpha_0 = 1/\sigma^2$ satisfies a Gamma prior $\Gamma(\alpha_0|c, d)$. Note that a, b, c , and d are all set to zero to obtain a uniform hyperprior.

The goal of BCS is to estimate three parameters $\boldsymbol{\alpha}$, α_0 , and the most important one \mathbf{x} . The posterior function of all unknowns is

$$p(\mathbf{x}, \boldsymbol{\alpha}, \alpha_0|\mathbf{y}) = p(\mathbf{x}|\mathbf{y}, \boldsymbol{\alpha}, \alpha_0) p(\boldsymbol{\alpha}, \alpha_0|\mathbf{y}). \quad (5)$$

The posterior of \mathbf{x} is

$$p(\mathbf{x}|\mathbf{y}, \boldsymbol{\alpha}, \alpha_0) = (2\pi)^{-\frac{N}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \mathbf{x} - \boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} \boldsymbol{\mu}\right), \quad (6)$$

with the mean matrix $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$,

$$\begin{aligned} \boldsymbol{\mu} &= \alpha_0 \boldsymbol{\Sigma} \boldsymbol{\Theta}^T \mathbf{y} \\ \boldsymbol{\Sigma} &= (\alpha_0 \boldsymbol{\Theta}^T \boldsymbol{\Theta} + \mathbf{A}^{-1})^{-1}. \end{aligned} \quad (7)$$

For the case of uniform hyperpriors,

$$p(\boldsymbol{\alpha}, \alpha_0|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\alpha}, \alpha_0). \quad (8)$$

Finally, we can estimate all unknown parameters by maximizing the posterior in Equation (5). To find a approximate solution, we use EM algorithm to achieve the solution iteratively. In the algorithm, $\boldsymbol{\alpha}$ and α_0 are re-estimated by every iteration with the form

$$\begin{aligned} \alpha_i^{new} &= \frac{\gamma_i}{\mu_i^2}, \\ \frac{1}{\alpha_0^{new}} &= \frac{\|\mathbf{y} - \boldsymbol{\Phi} \boldsymbol{\mu}\|_2^2}{M - \sum_i \gamma_i}, \end{aligned} \quad (9)$$

where μ_i is the i th posterior mean weight from $\boldsymbol{\mu}$ and γ_i is defined as $\gamma_i \triangleq 1 - \alpha_i \boldsymbol{\Sigma}_{ii}$, with $\boldsymbol{\Sigma}_{ii}$ the i -th diagonal element of the posterior covariance weight from $\boldsymbol{\Sigma}$. Then new $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are obtained. The iterations terminate until the values of $\boldsymbol{\alpha}$ and α_0 converges. Finally, $\hat{\mathbf{x}}_{\text{MAP}} = \boldsymbol{\mu}$.

2.3. Particle Swarm Optimization

PSO was inspired by the social behavior of bird flocking or fish schooling. PSO obtains a set of candidate solutions according to the mathematical formulas, which are based on the positions and velocities of the particles. These particles are iterated in the search space to solve the problem. The motion of each particle is not only affected by its local best known position, but is also guided toward the best known position in the search-space, which are updated by better positions found by other particles. PSO is a meta-heuristic algorithm (Algorithm 1) because it makes little or no assumptions about the problem being optimized and is able to search within a very large candidate solutions space.

Initially, each particle is randomly assigned with a position vector $\mathbf{x}_l = [x_{l,1}, x_{l,2}, \dots, x_{l,D}]$ as well as a velocity vector $\mathbf{v}_l = [v_{l,1}, v_{l,2}, \dots, v_{l,D}]$, where D is the dimension of a particle and $l \in (1, 2, \dots, S)$ with S representing the total particles number. Then each particle keeps track of its personal best position

\mathbf{P}_l and the global best one \mathbf{G} in the whole search space. After finding the two best fitness values for the l -th particle, it updates the position and velocity by the formulas

$$\begin{aligned} \mathbf{v}_l^{m+1} &= \omega^m \mathbf{v}_l^m + a_1 [r_{1,l} (\mathbf{P}_l - \mathbf{x}_l^m)] + a_2 [r_{2,l} (\mathbf{G} - \mathbf{x}_l^m)], \\ \mathbf{x}_l^{m+1} &= \mathbf{v}_l^{m+1} + \mathbf{x}_l^m. \end{aligned} \quad (10)$$

where m represents the current number of iteration, and $r_{1,l}, r_{2,l}$ are random variables between $[0, 1]$. a_1 and a_2 are the learning factors while ω is a weight factor that controls the velocity of the particle.

Algorithm 1 PSO algorithm

```

for each particle do
  initialize particle
end for
while target fitness or maximum epoch is not attained do
  for each particle do
    calculate fitness
    if current fitness value better than (pbest) then
      pbest=current fitness
    end if
  end for
  set gbest to the best one among all pbest
  for each particle do
    update velocity
    update position
  end for
end while

```

3. Proposed System Model

We consider the scenario that sensor nodes are randomly deployed in a three dimensional undersea area and sink node (base station) is placed on top of the area as shown in Figure 1. As introduced before, CS can only be used when the signal has sparsity in some domain. In the next, we would prove that the monitored oceanic data such as temperature, hardness, and salinity are indeed sparse in Fourier transform domain because of their spatial correlation. Take the ocean underwater temperature data rendered by National Aeronautic and Space Administration (NASA) (<https://www.nasa.gov/specials/ocean-worlds>) for example, we transform the data into its fast Fourier transform (FFT) domain and the results validate the sparsity as show in Figure 2.

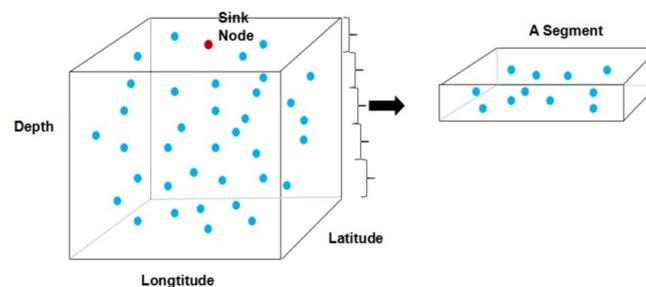


Figure 1. The segment-divided model of the 3-D network.

Another property of oceanic data is that its sparsity varies in different depth. As a result, we divide the network into different segments in line with its depth. Note that data with higher sparsity needs more information for recovery. Therefore, segments with higher sparsity are given higher

cluster-heads chosen probability so that for these segments, more measurements are obtained. In our model, the data in different segments are transmitted to the sink node and reconstructed separately, hence every segment can be considered as a sub-model. The segment-divided network model is presented as Figure 1.

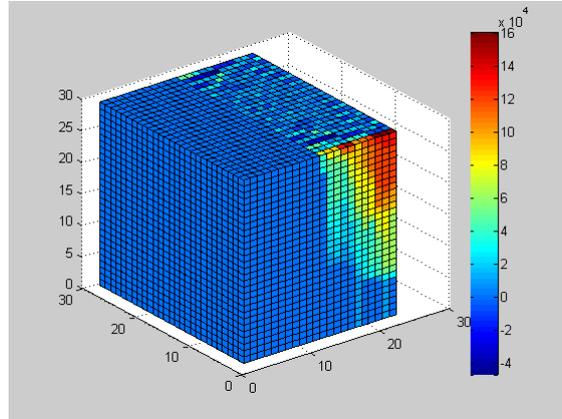


Figure 2. The FFT domain signal of temperature data.

In reality, the data is collected in a 3-dimensional network. We need to transform them into one-dimensional data in accordance with its horizontal position and depth first. Hence the data in the i -th segment is represented as $\mathbf{z}_i = [z_{i,1}, z_{i,2}, \dots, z_{i,N}]$, where N is the total number of nodes in the i -th segment. However, for simplicity, in the sequel, we take every sub-model as an integral model and the data are regarded as $\mathbf{z} = [z_1, z_2, \dots, z_N]$. Its sparse form is denoted as $\mathbf{z} = \Psi\mathbf{x}$, where \mathbf{x} is a sparse vector and Ψ is the FFT transform matrix.

In each segment, we use CS to reduce the transmission burden to the sink node. Therefore, energy is saved and the network lifetime is prolonged. Clustering is also applied in our model to further economize on energy. The noisy measurement vector collected in the sink node is $\mathbf{y} = \Phi\mathbf{z} + \mathbf{n} = \Phi\Psi\mathbf{x} + \mathbf{n}$. For the purpose of making network energy-consuming balanced and easier for a node to find the next relay node, we divide all nodes into different layers according to their distances to the sink node. In the following work, we divided them into four distinct layers, those who are closer to the sink node are given smaller layer number, which is depicted in Figure 3.

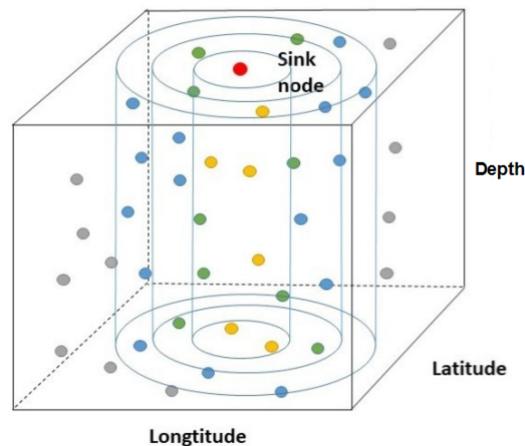


Figure 3. The layer-divided model of the 3-D network.

4. Proposed Algorithm

The overall algorithm contains three main steps illustrated as follows.

Firstly, we use PSO algorithm to choose M CHs for each segment and divide the nodes in each segment into M clusters. The second step is to select one routing path for each cluster using PSO algorithm, hence we obtain M routs to compose an integral routing matrix $\Phi \in \mathbb{R}^{M \times N}$. Figure 4 presents the flowchart of the whole procedure. In the flowchart, E represents the remaining energy of the whole network and f_1, f_2 are the fitness functions of clustering and routing steps respectively. f_1 considers energy consumption and uniformity of cluster head distribution while f_2 focuses on energy consumption state and BCRB.

The detailed description about how the CHs and routing paths, i.e., the measurement matrix are chosen will be presented in the next subsections. After we get the measurement matrix, the sink node receives a M dimensional measurement \mathbf{y} and its measurement matrix Φ . Finally, at the sink node, we use BCS algorithm to reconstruct the sparse signal \mathbf{x} and then transform it to the original temperature signal \mathbf{z} . After obtaining signals for all the segments one by one, the base station merges them together to recover the complete original signal.

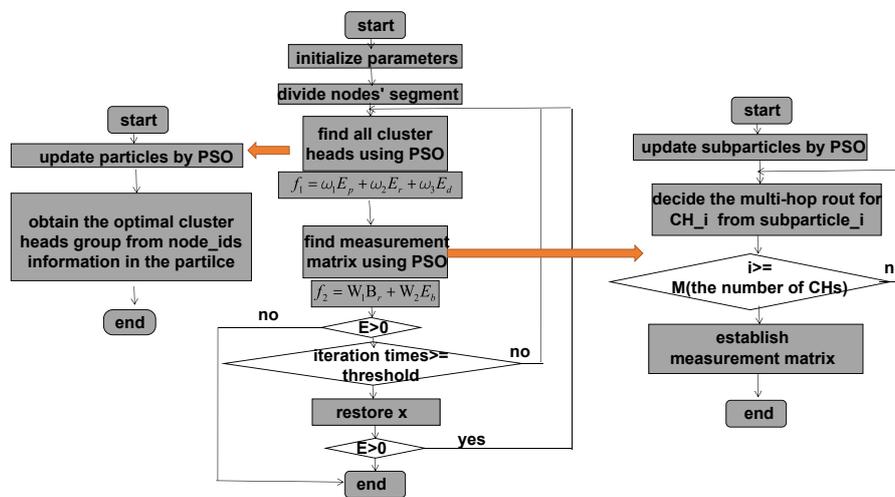


Figure 4. The flowchart of proposed algorithm.

To design the final object, there are three important factors to be taken into consideration: the signal reconstruction accuracy, the remaining energy of the network, and the balance state of the network.

Usually, mean squared error (MSE) is used to measure the reconstruction quality of the signals. If $\hat{\mathbf{x}}(\mathbf{y})$ denotes the recovered signal based on noisy measurements \mathbf{y} represented by Equation (2). The MSE in estimation of the vector \mathbf{x} is given by

$$\text{MSE} = E\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} = \text{Tr}\{E[(\mathbf{x} - \hat{\mathbf{x}}(\mathbf{y}))(\mathbf{x} - \hat{\mathbf{x}}(\mathbf{y}))^T]\} \quad (11)$$

However, the calculation of MSE requires the actual estimator which is impossible for us to get before choosing the measurement matrix. To solve this problem, we introduce Bayesian Cramér-Rao Bound (BCRB) to measure the lower bound of MSE; that is, $\text{MSE} \geq \text{BCRB}$. Hence, adding BCRB into the fitness function is beneficial for us to find the solution with lower MSE, since minimizing BCRB is equal to obtaining minimum acceptable MSE. Subsequently, we will derive the formula of BCRB for our problem.

4.1. BCRB Derivation

The BCRB is computed as $\text{BCRB} = \text{Tr}\{\mathbf{J}_B^{-1}\}$ with $\mathbf{J}_B \in \mathbb{R}^{N \times N}$ being the Bayesian Fisher information matrix (FIM) [42].

The i, j -th element of \mathbf{J}_B , \mathbf{J}_{ij} is equal to $E_{XY}[\nabla_{x_i} \log p(x, y) \nabla_{x_j}^T \log p(x, y)]$. Herein, $\nabla_v \triangleq [\frac{\partial}{\partial v_1}, \dots, \frac{\partial}{\partial v_q}]$. Then \mathbf{J}_{ij} can be calculated as

$$\begin{aligned} \mathbf{J}_{ij} &\triangleq E_{XY}[\nabla_{x_i} \log p(x, y) \nabla_{x_j}^T \log p(x, y)] \\ &= -E_{XY}[\nabla_{x_i} \nabla_{x_j}^T \log p(x, y)] \\ &= E_{XY}[\nabla_{x_i} \log p(y|x) \nabla_{x_j}^T \log p(y|x)] \\ &\quad + E_X[\nabla_{x_i} \log p(x) \nabla_{x_j}^T \log p(x)] \\ &= -E_{XY}[\nabla_{x_i} \nabla_{x_j}^T \log p(y|x)] - E_X[\nabla_{x_i} \nabla_{x_j}^T \log p(x)]. \end{aligned}$$

Therefore, the Bayesian FIM is given by

$$\begin{aligned} \mathbf{J}_B &= -E_{XY}[\nabla_x \nabla_x^T \log p(\mathbf{y}|\mathbf{x})] - E_X[\nabla_x \nabla_x^T \log p(\mathbf{x})] \\ &= -E_{(y,x)}\left\{\frac{\partial^2 \mathcal{L}(\mathbf{y}|\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right\} - E_{(x)}\left\{\frac{\partial^2 \mathcal{L}(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right\}. \end{aligned} \tag{12}$$

Firstly, we compute the second part of Equation (12). To solve $\mathcal{L}(\mathbf{x})$, we must obtain the density function of signal \mathbf{x} in the first place. In our model, the sparse signal \mathbf{x} follows the Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$. $\boldsymbol{\Sigma} = \text{diag}(\gamma_1, \dots, \gamma_N)$ is a diagonal matrix with γ_i denoting the variance of x_i . As a result, the density of \mathbf{x} is

$$p(\mathbf{x}; \boldsymbol{\gamma}) = \prod_{i=1}^N (2\pi\gamma_i)^{-1/2} e^{-\frac{(x_i-\mu)^2}{2\gamma_i}}. \tag{13}$$

Hence $\mathcal{L}(\mathbf{x}) = \sum_{i=1}^N -\frac{1}{2} \ln 2\pi\gamma_i - \frac{(x_i-\mu)^2}{2\gamma_i}$.

As the previous part is irrelevant to \mathbf{x} , we abbreviate it as \tilde{g} . Then the formula in matrix form can be written as

$$\mathcal{L}(\mathbf{x}) = \tilde{g} - \frac{1}{2} \mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x}. \tag{14}$$

As a result $\frac{\partial^2 \mathcal{L}(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} = \boldsymbol{\Sigma}^{-1}$.

Secondly, we turn to deal with the first part of FIM. To obtain $-E_{(y,x)}\left\{\frac{\partial^2 \mathcal{L}(\mathbf{y}|\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right\}$, we need to know $p(\mathbf{y}|\mathbf{x})$. Since $\mathbf{y} = \boldsymbol{\Phi} \boldsymbol{\Psi} \mathbf{x} + \mathbf{n}$, the probability density function of \mathbf{y} is related to the noise term \mathbf{n} given the information of \mathbf{x} . The noise can be the Gaussian distribution; that is, $p(\mathbf{n}) = \prod_{i=1}^N (2\pi\sigma^2)^{-1/2} e^{-\frac{n_i^2}{2\sigma^2}}$. Hence, we have

$$p(\mathbf{n}) = (2\pi\sigma^2)^{-N/2} e^{-\frac{\|\mathbf{y} - \boldsymbol{\Phi} \boldsymbol{\Psi} \mathbf{x}\|^2}{2\sigma^2}}. \tag{15}$$

Subsequently,

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= -\frac{N}{2} \times \log 2\pi\sigma^2 - \frac{\|\mathbf{y} - \boldsymbol{\Phi} \boldsymbol{\Psi} \mathbf{x}\|^2}{2\sigma^2} \\ &= -\frac{N}{2} \log 2\pi\sigma^2 - \frac{1}{2\sigma^2} (\mathbf{y} - \boldsymbol{\Phi} \boldsymbol{\Psi} \mathbf{x})^T (\mathbf{y} - \boldsymbol{\Phi} \boldsymbol{\Psi} \mathbf{x}), \end{aligned}$$

and then

$$E_{(y,x)}\left\{\frac{\partial^2 \mathcal{L}(\mathbf{y}|\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}\right\} = -\frac{1}{\sigma^2} \boldsymbol{\Psi}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{\Psi}. \tag{16}$$

Finally, we put the two parts together as the description of FIM,

$$\begin{aligned} \mathbf{J}_B &= -E_{(y,x)} \left\{ \frac{\partial^2 \mathcal{L}(\mathbf{y}|\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} \right\} - E_{(x)} \left\{ \frac{\partial^2 \mathcal{L}(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T} \right\} \\ &= \frac{1}{\sigma^2} \mathbf{\Psi}^T \mathbf{\Phi}^T \mathbf{\Phi} \mathbf{\Psi} + \mathbf{\Gamma}^{-1} \end{aligned}$$

Therefore, in our model, BCRB is represented by $Tr \left\{ \frac{1}{\sigma^2} \mathbf{\Psi}^T \mathbf{\Phi}^T \mathbf{\Phi} \mathbf{\Psi} + \mathbf{\Gamma}^{-1} \right\}$.

4.2. Underwater Energy Consumption Model

Ref. [43] gives an in-depth analysis of the energy consumptions of underwater acoustic networks according to the conceptions of underwater acoustic channel attenuation model, noise model and bandwidth model. The total energy consumption associated to one hop (which includes both the transmit and the receive energy at the two ends of the link), is determined by the power P_r , acoustic electric conversion power $P_t^{el}(l)$ which is the power required to transform acoustic signal to electric signal and single hop transmission delay $t_{hop}(L, l)$. Specifically, the energy consumed by each single hop, E_{hop} is calculated as:

$$E_{hop}(L, l) = t_{hop}(L, l) \times (P_r + P_t^{el}). \quad (17)$$

The single hop transmission delay $t_{hop}(L, l)$ is determined by package length L and the distance l between two transmitting nodes as well as channel effective bandwidth $\alpha B(l)$ where α means channel utilization rate. So the formula becomes

$$E_{hop}(L, l) = t_{hop}(L, l) \times (P_r + P_t^{el}) = \frac{L}{\alpha B(l)} \times (P_r + P_t^{el}) \quad (18)$$

$P_t^{el}(l)$ is calculated as

$$P_t^{el}(l) = \frac{P_t(l) \times 10^{-17.2}}{\eta}. \quad (19)$$

Herein, $10^{-17.2}$ is the conversion factor from acoustic power in $dBre\mu Pa$ to electrical power in Watt, η represents electronic circuit conversion efficiency and $P_t(l)$ is the transmitting power of underwater nodes, which is calculated as

$$P_t(l) = B(l) \times A(l, f_0(l)) \times N(f_0(l)) \times SNR_{tgt}. \quad (20)$$

$N(f(l))$ and $A(l, f(l))$ are noise parameter and attenuation coefficient, respectively. $AN = A(l, f(l))N(f(l))$ is the channel parameter and can be plotted as a curve in terms of f for given l . Therefore, for certain l , we can obtain its best frequency $f_0(l)$ and then find its corresponding AN product, $N(f_0(l)) \times A(l, f_0(l))$ from the AN curve.

SNR_{tgt} denotes the target signal noise rate for the terminal to receive signals correctly. $B(l)$ is the usable bandwidth given SNR_{tgt} , which is $B(l) = b \times l^{-\beta}$. The positive parameters b, β depend on the target SNR as well. Unlike the transmit power, the receive power P_r is independent of distance, and rather depends on the complexity of the receive operations. So it could be set as a constant value. More details can be found in [44].

In conclusion, the energy consumed for each hop in the terms of package length L and distance l is calculated as:

$$\begin{aligned}
E_{hop}(L, l) &= t_{hop}(L, l) \times (P_r + P_t^{el}) \\
&= \frac{L}{\alpha B(l)} \times \left(P_r + \frac{B(l)N(f_0(l))A(l, f_0(l))SNR_{tgt} \times 10^{-17.2}}{\eta} \right)
\end{aligned} \tag{21}$$

Based on such a model, we can calculate the energy consumed by each node which participates the routing.

4.3. PSO for Clustering

First we use PSO to find appropriate nodes to be CHs. Because we divided the network into different segments, the percentage of CHs within each segment is highly related to the sparsity of the segment. In general, we set the CH ratio q to be 0.4, only in some segments which have relative low sparsity, the value of q_i increases slightly in a trend like $q_i = 0.4 + 0.05 \times [s_i - 0.08]$ where s_i is the sparsity of the i th segment.

The particle's dimension is the same as the number of CHs. To initialize the particle, we generate M random integer numbers within $[1, N]$. For example, the i -th particle P_i is M dimensional and $P_{i,m}$ represents the m -th value of it. If $P_{i,m} = 25$, it means that in this particle, we choose node 25 to be the m -th CH. As a node cannot be chosen repeatedly in one particle, if a particle has duplicated numbers, it will be assigned a penalty fitness value -1 to be excluded during later progress.

After initializing all the particles, we calculate the fitness value for each particle. According to their fitness value, we choose the local best and the global best particles, update the particle position which corresponds to the coordinates of the nodes in this particle and velocity to get new fitness values.

In this part, we mainly focus on energy consumption and uniformity of cluster head distribution, hence the fitness function f does not include the measurements for signal recovery qualities in current stage. The fitness function f_1 comprises of three parts as below:

- E_p , represents the energy evaluation factor and E_p is given by

$$E_p = \frac{\sum_{m=1}^M E_{res_m}}{\sum_{i=1}^N E_{init_i}}, \tag{22}$$

which measures the remaining energy of the chosen CHs.

Note that E_{init_i} is the initial energy of the i -th node while E_{res_m} is the residual energy of the m -th node and equals to $E_{ini_m} - E_{c_m}$, and

$$E_{c_m} = \begin{cases} 0 & \text{if node } m \text{ was not involved in the routing} \\ t_{hop}(L, l_{m,r}) \times P_r + t_{hop}(L, l_{m,t}) \times P_t^{el} & \text{if node } m \text{ was involved in the routing,} \end{cases}$$

where $l_{m,r}$ measures the hop length when node m is the receiver and $l_{m,t}$ measures the hop length when node m is the transceiver.

We tend to choose nodes with higher residual energy to be CHs due to the fact that cluster heads consume more energy than normal nodes.

- E_c indicates the evaluation factor for the intra-cluster compactness and measures the average distance between nodes and their cluster heads.

We calculate

$$f = \max_{m=1,2,\dots,M} \sum_{\exists n_i \in C_m} d_{i,CH_m} / |C_m|,$$

which measures the maximum average Euclidean distance between nodes and their CHs. d_{i,CH_m} measures the distance between node i and CH_m . $|C_m|$ is the number of nodes that belong to cluster C_m . Our aim is to minimize f because the nodes are closer to its cluster heads with smaller f . Therefore, $E_c = 1/f$.

- E_e is the evaluation factor of the uniformity of CH distribution. It measures the uniformity of CH distribution. Firstly we calculate all nodes' distances to each other. $\mathbf{D} = \mathbf{d}_1, \dots, \mathbf{d}_N$ where $\mathbf{d}_j = d_{j,1}, \dots, d_{j,i}, \dots, d_{j,N}, i \neq j$ and $d_{j,i}$ measures the distance between node j and node i . In an unevenly distributed network, the sum of \mathbf{D} must be higher than relatively evenly distributed network. As a consequence, we add this value $E_e = \sum_{j=1}^M d_j / |C_m|$ on our fitness function.

To sum up, the final form of fitness function f_1 of our cluster choosing algorithm is given by:

$$f_1 = w_1 E_p + w_2 E_c + w_3 E_e,$$

$$\text{where } \begin{cases} w_1 + w_2 + w_3 = 1 \\ w_i \geq 0, i \in \{1, 2, 3\} \end{cases}$$

and w_i is the weighting factor for the i -th item. To think about the problem in a balanced way, w_1 and w_2 are randomly chosen from the range $[0.2, 0.4]$ while $w_3 = 1 - w_1 - w_2$.

When a particle contains duplicated nodes, we set a penalty factor to it to exclude it from iterations.

This iteration keeps going on until either the global best fitness value is lower than the threshold we set or the maximal iterative times is reached. The global best particle obtained by the last iteration is the output of the cluster head selection scheme.

4.4. PSO for Choosing Routing Paths

The final step of our algorithm is to find the routing paths, then the measurement matrix. PSO is used in this stage as well and herein each particle contains M sub-particles, where M is the number of CHs. A sub-particle includes the nodeID information while sub-particle position includes priorities corresponding to all the nodes denoted in the sub-particle.

Each sub-particle determines a single rout from each cluster to the base station. Since we have already chosen all the CHs, there are two steps left for each CH to fulfill the corresponding routing path: (1) Each CH chooses the nodes in its cluster which transmit the messages to it in one hop; (2) After collecting messages in its cluster, each CH starts a multi-hop rout to the base station.

In one word, a complete routing path contains two parts, the one-hop inner cluster part and the multi-hop outer part. Therefore we divide the sub-particle into two parts, a and b . Part a determines which nodes from the inner cluster transmit their data to the CH while part b finds the routing path from the CH to the base station. It is difficult to jointly optimize these two part. Instead, we construct inner cluster part and outer cluster part separately.

The dimension of part a of sub-particle i equals to K_i , where K_i is the node numbers of each cluster except the i -th CH. The sub-particle i selects $\epsilon \times K_i$ nodes with highest priorities from part a to transmit data to the i -th CH in one hop. ϵ is the ratio controlling parameter which we would optimize in the simulation section. Figure 5 gives an example on how to choose nodes within one cluster and how to map part a of a sub-particle into real measurement matrix. The measurement matrix with dimension $M \times N$ is initially set to all zero matrix, where M is the number of CHs and N is the total number of nodes. Once a node is chosen into i -th rout, its corresponding position in the i -th row of measurement matrix is set to a randomly distributed coefficient which obeys Gaussian distribution.

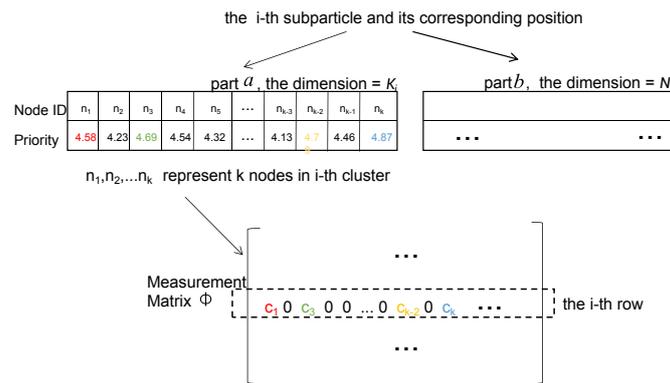


Figure 5. The demo for mapping part a of sub-particle i into Φ , where c_j denotes random measurement coefficient.

The algorithm to find a multi-hop routing path from the i -th CH to the base station from part b is more sophisticated and will be illustrated in detail in the following.

i. Choosing candidate nodes

Regarding that if we take all nodes into consideration for each rout, the dimension of a sub-particle would be too large. Given the fact that when a node try to find its rout, those nodes in the opposite area from sink node may never be included. Therefore, before the initialization, we calculate the distances of all nodes to the sink node Dis_s and the distance of all nodes to the CHs Dis_c . Then choose an appropriate group of candidate nodes for each CH first. $Dis_s = d_{s,1}, \dots, d_{s,N}$ represents nodes' distances to the sink node while $Dis_c = d_{c,1}, \dots, d_{c,N}$ represents nodes' distances to the CH. $dis = \sqrt{(x_s - x_c)^2 + (y_s - y_c)^2 + (z_s - z_c)^2}$ is the distance between the CH and sink node where x_s, y_s, z_s represent the coordinates of the sink node and x_c, y_c, z_c are the coordinates of the CH. Note that these coordinates values could be obtained by range-based or range-free localizations before the data transmission. Only when a node k satisfies $d_{s,k} \leq dis$ and $d_{c,k} \leq dis$, it would be added to the candidate group. Once the candidate groups are chosen, they will not be altered during the whole iteration period. Suppose the size of candidate group for the i -th CH is N'_i , then the dimension of part b of the sub-particle i equals to N'_i .

ii. Initialization

To find part b , we make use of the network's layered structure as we need to take directions into consideration. The nodes that are in the closer layer to the base station ought to have higher priority. This layer-based initialization method can control the routing path's direction towards the sink node as well as avoid trapping into a dead cycle and accelerate iterative process.

The priorities of the nodes in the farthest layer are initialized by randomly generating numbers between 0 to 1. In our layered model, the fourth layer is the farthest layer, so the priorities of the nodes in the fourth layer are set from 0 to 1, while the priorities of the nodes in the third layer are from 1 to 2. In one word, when the number of the layer decreases by 1, the initial priorities of the nodes increase by 1. Hence the priority is not initialized completely randomly but related to the layer the node belongs to.

Figure 6 shows an example of our initialization. In this example, for the CH, there are altogether 12 candidate nodes dispersed in 4 different layers. It can be easily observed that the nodes in the higher layers have lower initialized priorities. That is how we achieve hierarchical initialization and obtain initialized multi-hop routing path from the CH to the sink node.

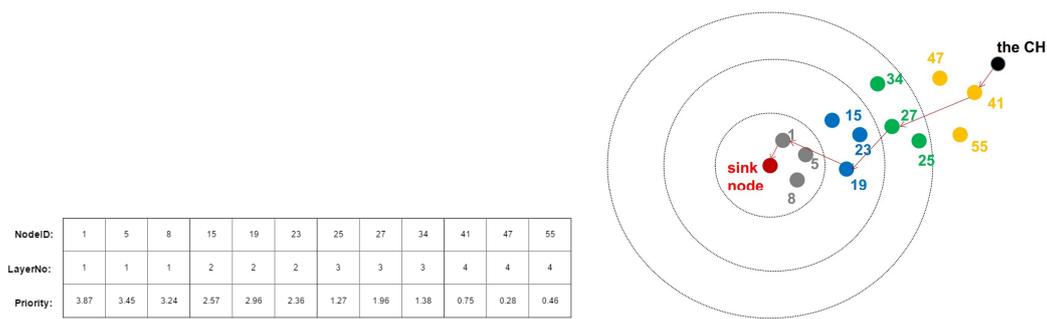


Figure 6. An example of hierarchical priorities initialization for part *b* and its corresponding topological routing path in the cross-section view.

iii. Iteration

After initializing the particles, in each iteration, we are going to calculate particles’ fitness values in terms of measurements matrices and select local best and global best ones. Then the particle positions are updated.

We have already introduced how to map part *a* of the sub-particle *i* to the corresponding row. Here we introduce how we map part *b* of sub-particle *i* to the *i*-th row of the measurement matrix.

Firstly we set the corresponding position of the *i*-th CH in the *i*-th row of Φ to a random Gaussian distributed coefficient. Then we find the next relay *i* node by two steps iteratively until the complete routing path is found.

Step 1: If the previous node’s distance to the sink node is smaller than its transmission range R_c , stop searching since we already obtain a single rout from the *i*-th CH to the sink node. Otherwise, we go to step 2.

Step 2: Find all nodes that are not only in the candidate group but also within the previous node’s transmission range, compare their priorities and find the largest one as the next relay node. Meanwhile, we set the corresponding position of chosen relay node in the *i*-th row of Φ to a random Gaussian distributed coefficient.

After mapping *M* sub-particles to *M* rows, the measurement matrix Φ is determined. The generated Φ is then used to compute the fitness value of corresponding particle. The fitness function includes two main items, the measurement of reconstruction quality, B_r , which is related to BCRB and the measurement of energy consumed, E_s .

As we calculated before, the formula of the BCRB is given by

$$BCRB = Tr\{(\frac{1}{\sigma^2} \Psi^T \Phi^T \Phi \Psi + \hat{\Gamma}^{-1})\}. \tag{23}$$

As BCRB implies a theoretical lower bound of reconstruction’s mean square error, we wish it to be as small as possible. So $B_r = 1/Tr\{(\frac{1}{\sigma^2} \Psi^T \Phi^T \Phi \Psi + \hat{\Gamma}^{-1})\}$.

The energy consumption of the routings we chose is measured by E_s . It consists of three factors like

$$E_s = 1/E_r + E_l + E_b. \tag{24}$$

The first item $E_r = \sum_{k=1}^N \frac{E_{c_k}}{E_{res_k}}$, where E_{c_k} is the energy consumed at this round by node *k* and E_{res_k} is the rest energy of this node. It measures the energy cost ratio of nodes and should be small considering the durability of a network.

Besides E_r , $E_l = \sum_{k=1}^N \frac{E_{res_k}}{E_{init_k}}$, where E_{init_k} is the initial energy of the *k*-th node.

On top of that, the balance of this network also needs to be considered. A well-known fact is that the nodes near the sink node are prone to consume more energy than the nodes far away. Those nodes burden heavier transmission tasks and are easier to convey their data to the sink repeatedly because they are more likely to be relay nodes. So to obtain a balanced network, we introduce a factor E_b to

describe a energy consumption ratio of different layer. $E_b = E_1/E_4 + E_2/E_3$, where E_i represents the remaining energy of the i -th layer. E_b should be as large as possible as the lower layer tends to remain more energy in a balanced network than in an uneven network.

In a conclusion, the final fitness function consists of two parts as below,

$$f_2 = W_1 B_r + W_2 E_s \quad (25)$$

where W_i is a weighting factor and $W_1 + W_2 = 1, W_i \geq 0, i \in \{1, 2\}$. In the simulation, W_1 is chosen randomly from the range $[0.4, 0.6]$ while $W_2 = 1 - W_1$.

5. Simulation Results

We use the real ocean temperature data provided by NASA (<https://www.nasa.gov/specials/ocean-worlds>) to test our algorithm against the two cutting-edge methods. Note that the sensor network they deploy to monitor the environment is huge and we only choose 1000 nodes from this network which are deployed in a 3-D undersea region divided into $10 \times 10 \times 10$ grids. The size of the area is $18.5 \text{ km} \times 14.5 \text{ km} \times 0.8 \text{ km}$. We compare our method with other two cutting edge method which also applies CS into routing. The method introduced in [31] for 3D UWSN is named as “DRMCS” which is the abbreviation of “distributed random multi-hop compressive sensing”. The other method is called “ICCS” following its name in [29] and later on in [30]. We extend it to 3D scenario by changing the clustering and routing methods to 3D protocol through replacing the 2D distance (x, y) to 3D distance (x, y, z) . The parameters we set for the energy consumption model in Section 4.1 are listed in Table 1 according to [43].

Table 1. Energy Consumption Parameters.

Energy Consumption Parameter	Values
η	0.5
α (bps/Hz)	0.5
k	1.5
SNR_{tgt} (dB)	8
P_r (W)	2
b (dB re kHz)	14.39
β (dB re kHz/km)	−0.55
R_c (km)	3.5

Before the whole simulations, we firstly select the appropriate value of parameter ϵ . It is impossible for us to obtain the signal \mathbf{z} before the algorithm is run. To solve this problem, we do some preparatory experiments to decide an appropriate ϵ . In the first five rounds, we set $\epsilon = 0.9$, run the experiments once, and then we obtain the reconstructed signal $\hat{\mathbf{z}}$. The reconstructed signal $\hat{\mathbf{z}}$ is regarded as the approximation of the original signal \mathbf{z} in order to compute the reconstruction error in the following simulations. That is, afterwards we ran the first five rounds with different ϵ ranging from 0.2 to 0.8, respectively. The average consuming energy and the reconstruction error are compared. For example, the results of round 1, 3 and 5 are shown in Table 2. Considering the reconstruction error and energy consumption state integrally, we found that in our algorithm $\epsilon = 0.4$ is the most suitable. As a result, for the rest rounds, ϵ is set to be 0.4.

All experiments were performed on MATLAB 2018b and a computer with intel(R) Core(TM) i7-8700K 2.50 GHz CPU, 16.00 GB RAM and the Windows 10 system.

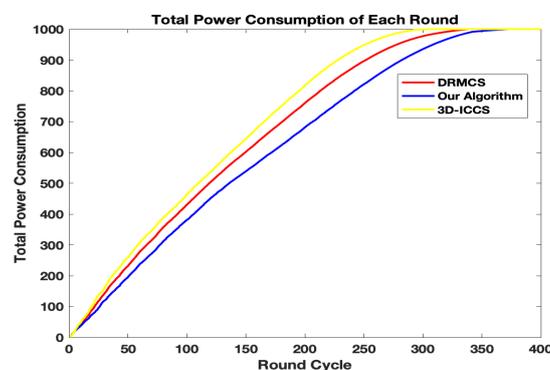
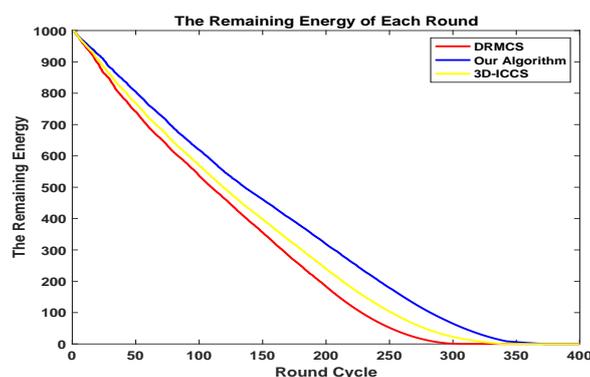
Table 2. Performances by Different ϵ for round 1, 3, 5.

ϵ	Energy Consumption	Reconstruction Error
0.2	3.164	0.01419
0.4	3.643	0.01201
0.6	4.445	0.01249
0.8	5.050	0.01240

ϵ	Energy Consumption	Reconstruction Error
0.2	3.157	0.01421
0.4	3.655	0.01203
0.6	4.426	0.01251
0.8	5.039	0.01242

ϵ	Energy Consumption	Reconstruction Error
0.2	3.168	0.01437
0.4	3.637	0.01208
0.6	4.439	0.01257
0.8	5.012	0.01248

In Figures 7 and 8, we compare the total energy consumption and the remaining energy for each round while the measurement ratio M/N is set to 0.4. Obviously our algorithm survives longer than the other two state-of-the-art methods. The first tier utilizes PSO to find the best cluster heads group while the second tier utilizes PSO to find the best routing matrix so that the whole network has the lowest average consuming energy per cycle while maintaining the sensing error in an acceptable range. Note that in the second tier, the optimization of the routing path can not only reduce the average energy cost per round but also help to make the network more balanced. In other words, besides the fact that our PSO-based routing scheme consumes less energy each round, another important reason for longer network lifetime is that our scheme obtains more balanced energy-consumption condition.

**Figure 7.** The comparison results in terms of total energy consumption.**Figure 8.** The comparison results in terms of remaining energy.

The more balanced condition and less energy-consuming phenomenon of our scheme can also be proved in Figure 9, which shows the remaining living nodes of the network in each round. From Figure 9, we can observe that by our algorithm, the first dead node appears much later. The proposed algorithm has its first dead node at round more than 100 while the other algorithms has its first dead node at about round 70. Such a fact indicates that the proposed two-tier protocol consumes less energy each round as well as consumes energy more equally for each node of the whole network. The improvement attributes to the fact that in order to better balance the whole network, we add a factor named “network-balancing” in the particles’ fitness value calculation. Considering that the nodes closer to the sink burden heavier transmission load and hence are easier to die, we put energy consumption ratios of inner and outer layers into consideration. Reducing the ratio can avoid repeatedly choosing inner layer nodes and balance the network energy consumption better.

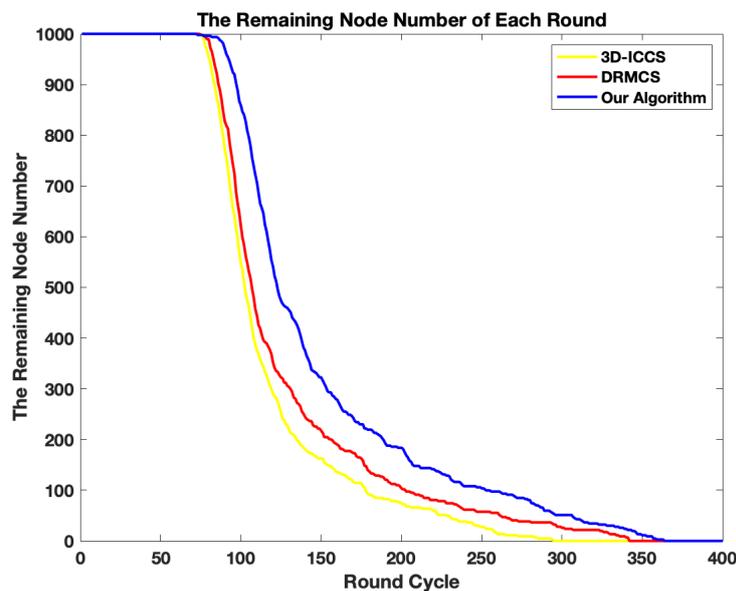


Figure 9. The comparison results in terms of remaining living nodes.

In order to observe the impact of node density on the network performance, we fix the total node number as 1000 and vary the volume of the area where these nodes are deployed. As shown in Figure 10, the average energy consumption in each round decreases with the increase of node density. Since the node number is fixed, the initial energy is fixed. Hence lower average energy consumption in each round means larger round cycles. This is due to the fact that we shrink the volume so that the average distances from CHs to sink node are shortened. From Figure 10, it can also be observed that compared with other two schemes, our algorithm consumes less average energy each round in the whole range of node density. To measure the delay of relevant schemes, we calculate the average hop numbers in the case of two different node densities. Average hop number is obtained by $\sum_{i=1}^M \text{hop}_i / M$, where hop_i measures the hop numbers from the i -th CH to the sink node. Obviously, since the total node number is fixed and the volume is decreased, average hop numbers, i.e., delay is lower with higher node density as shown in Figure 11. On the other hand, our algorithm always presents the lowest delay compared with relevant schemes.

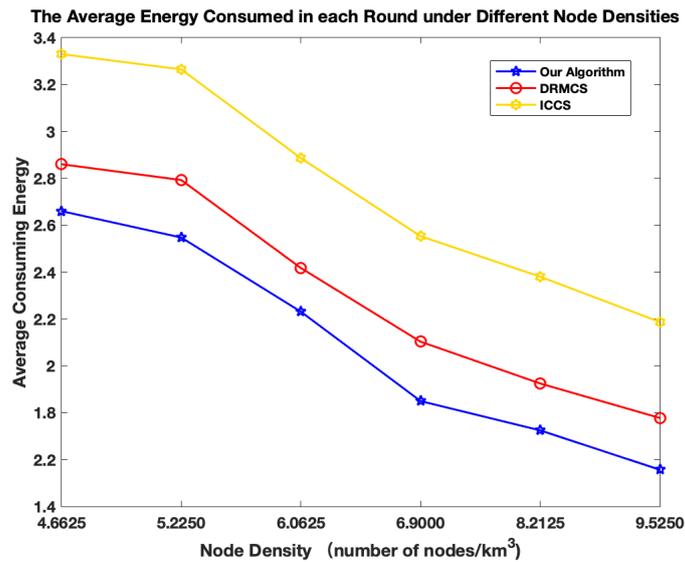


Figure 10. The comparison results in terms of average energy consumption with different node densities.

Another performance we compare is the sensing quality which is measured by MSE between the original signal and reconstructed error. It can be observed from Figure 12 jointly with Figure 8 that our proposed algorithm tends to build a matrix Φ that leads to the smallest sensing error and meanwhile the energy consumed by our algorithm is the least in each round.

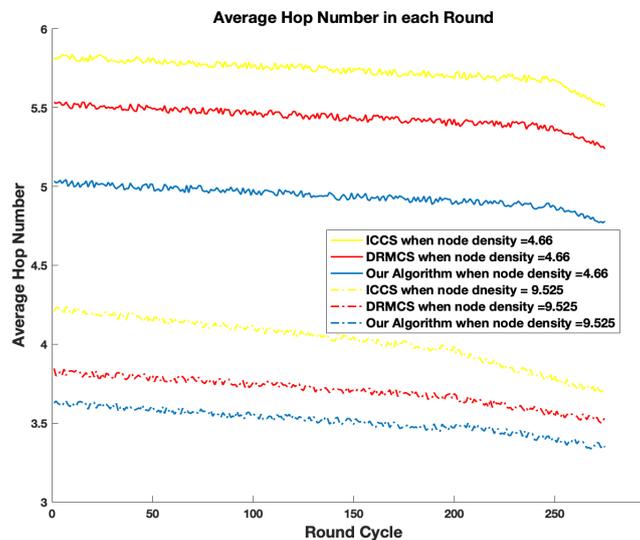


Figure 11. The comparison results in terms of average hop numbers with two different node densities.

Besides the previous experiments, we vary the measurements ratio M/N as well to observe the MSE with varying ratio. We test the average MSE for first 20 rounds. It can be observed that the MSE decreases with the increase of measurement ratio for each algorithm. Figure 13 validates that our algorithm presents significant advantage when the measurement ratio is less than 0.5. The reason is that we add the factor B_r , which considers the reconstruction quality, into the fitness function while performing PSO for routing. Our proposed algorithm works well to reduce the MSE when the measurement ratio is low. While the measurement ratio goes up, the superiority of our proposed algorithm becomes less significant.

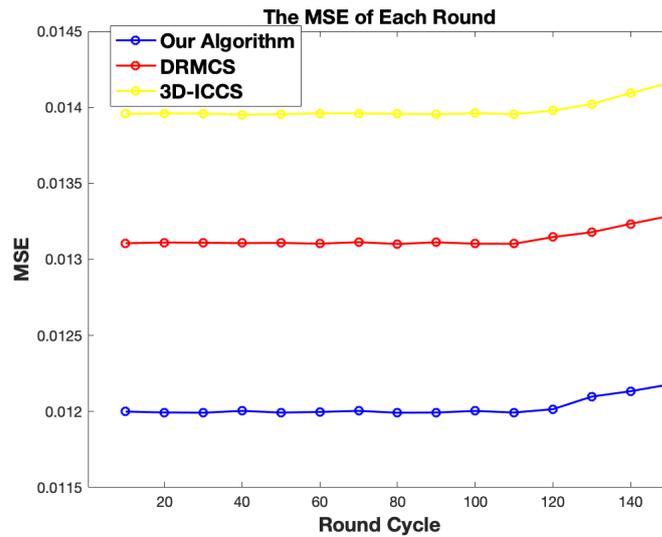


Figure 12. The comparison results in terms of data recovery qualities.

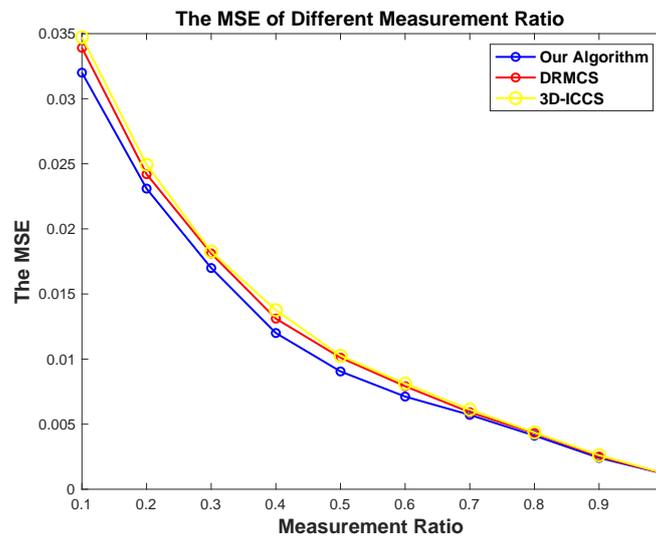


Figure 13. The comparison results in terms of MSE in different measurement ratio.

6. Conclusions and Future Work

In this paper, we proposed a two-tier Particle Swarm Optimization (PSO) algorithm in the clustering and routing stages for distributed multi-hop compressed sensing in 3-D UWSN. We divided the whole network into different segments according to the vertical distances of the nodes. On top of that, we also divided the whole network into different layers according to the horizontal distances from the nodes to the sink and the PSO algorithm we apply in the network is firmly associated with the layers. Dividing the layers help to define the energy consumption in different layers to measure energy-consuming balance and to initialize the particle hierarchically so as to accelerate the particle convergence speed. And a factor to forecast the signal recovery performance is added on the fitness function of the PSO algorithm for routing to control the MSE of the reconstructed signal while maintaining a relatively long lifetime. Our proposed algorithm is compared with recently proposed algorithms for distributed multi-hop compressive sensing routing problem using real marine temperature data rendered by NASA. The results validate that our proposed algorithm has a longer life time with lower measurement error and postpones the round that has the first dead node.

For the future work, we will focus on the synthesized consideration of the clustering and routing stage while doing optimization. In the current study, the two stages are considered separately. On the further study, we will conduct the two steps consistently to improve the performance a step further.

Author Contributions: Conceptualization, X.C. and W.X.; methodology, X.C.; software, W.X.; validation, X.C. and W.X.; formal analysis, S.C.; investigation, X.C. and W.X.; resources, X.C.; data curation, W.X.; writing—original draft preparation, W.X.; writing—review and editing, X.C.; visualization, W.X.; supervision, X.C.; project administration, X.C. and S.C.; funding acquisition, X.C. and S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant 61301181, U1530120 and Scientific Research Starting Foundation of Central South University, and the APC was funded by the National Natural Science Foundation of China under No. U1530120.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Y.C.; Chen, G.W. Efficient Data Gathering and Estimation for Metropolitan Air Quality Monitoring by Using Vehicular Sensor Networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 7234–7248. [[CrossRef](#)]
2. Jin J.; Gubbi J.; Marusic S.; Palaniswami M. An Information Framework for Creating A Smart City through Internet of Things. *IEEE Internet Things J.* **2014**, *1*, 112–121. [[CrossRef](#)]
3. Ramírez, C.A.; Barragán, R.C.; García-Torales, G.; Larios, V.M. Low-Power Device for Wireless Sensor Network for Smart Cities. In Proceedings of the Microwave Conference (LAMC) IEEE MTT-S Latin America, Puerto Vallarta, Mexico, 12–14 December 2016.
4. Mahamuni, C.V. A Military Surveillance System Based on Wireless Sensor Networks with Extended Coverage Life. In Proceedings of the 2016 International Conference on Global Trends in Signal Processing Information Computing and Communication (ICGTSPICC), Jalgaon, India, 22–24 December 2016.
5. Goyal, N.; Dave, M.; Verma, A.K. Data Aggregation in Underwater Wireless Sensor Network: Recent Approaches and Issues. *J. King Saud Univ.-Comput. Inf. Sci.* **2019**, *31*, 275–286. [[CrossRef](#)]
6. Chu, Y.; Hsieh, Y.; Wang, C.; Pan, Y.; Chang, R. UPHSM: Ubiquitous Personal Health Surveillance and Management System via WSN Agent on Open Source Smartphone. In Proceedings of the 13th IEEE International Conference on e-Health Networking, Applications and Services, Columbia, MO, USA, 13–15 June 2011.
7. Heidemann, J.; Ye, W.; Wills, J. Research Challenges and Applications for Underwater Sensor Networking. In Proceedings of the 2006 IEEE Wireless Communications and Networking Conference (WCNC), Las Vegas, NV, USA, 3–6 April 2006.
8. Wang, Z.; Han, G.; Qin, H.; Zhang, S.; Sui, Y. An Energy-Aware and Void-Avoidable Routing Protocol for Underwater Sensor Networks. *IEEE Access* **2018**, *6*, 7792–7801. [[CrossRef](#)]
9. Guan, Q.; Ji, F.; Liu, Y.; Yu, H.; Chen, W. Distance-Vector-Based Opportunistic Routing for Underwater Acoustic Sensor Networks. *IEEE Internet Things J.* **2019**, *6*, 3831–3839. [[CrossRef](#)]
10. Khan, W.; Wang, H. A Multi-Layer Cluster Based Energy Efficient Routing Scheme for UWSNs. *IEEE Access* **2019**, *7*, 77398–77410. [[CrossRef](#)]
11. Anuradha, D.; Srivatsa, S.K. Optimal Visiting Tour Scheduling for Mobile Data Gathering in UWSN. In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22–24 March 2017.
12. Donoho, D. Compressed Sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [[CrossRef](#)]
13. Candés, E.J.; Romberg, J.K.; Tao, T. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Commun. Pure Appl. Math.* **2006**, *59*, 1207–1223. [[CrossRef](#)]
14. Chen, X.; Li, F.; Liu, X. Efficient and Robust Distributed Digital Codec Framework for Jointly Sparse Correlated Signals. *IEEE Access* **2019**, *7*, 77374–77386. [[CrossRef](#)]
15. Wu, F.; Yang, K.; Duan, R.; Tian, T. Compressive Sampling and Reconstruction of Acoustic Signal in Underwater Wireless Sensor Networks. *IEEE Sens. J.* **2018**, *18*, 5876–5884. [[CrossRef](#)]

16. Liu, B.; Zhang, Z. Quantized Compressive Sensing for Low-Power Data Compression and Wireless Telemonitoring. *IEEE Sens. J.* **2016**, *16*, 8206–8213. [[CrossRef](#)]
17. Haupt, J.; Bajwa, W.; Rabbat, M.; Nowak, R. Compressed Sensing for Networked Data. *IEEE Signal Process. Mag.* **2008**, *25*, 92–101 [[CrossRef](#)]
18. Quer, G.; Masiero, R.; Pillonetto, G.; Rossi, M.; Zorzi, M. Sensing, Compression, and Recovery for Wsns: Sparse Signal Modeling and Monitoring Framework. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 3447–3461. [[CrossRef](#)]
19. Davenport, M.; Laska, J.; Treichler, J.; Baraniuk, R. The Pros and Cons of Compressive Sensing for Wideband Signal Acquisition: Noise Folding Versus Dynamic Range. *IEEE Trans. Signal Process* **2012**, *60*, 4628–4642. [[CrossRef](#)]
20. Hwang, S.; Ran, R.; Yang, J.; Kim, D.K. Multivariate Bayesian Compressive Sensing in Wireless Sensor Networks. *IEEE Sens. J.* **2016**, *16*, 2196–2206. [[CrossRef](#)]
21. Luo, C.; Wu, F.; Sun, J.; Chen, C.W. Efficient Measurement Generation and Pervasive Sparsity for Compressive Data Gathering. *IEEE Trans. Wirel. Commun.* **2010**, *9*, 3728–3738. [[CrossRef](#)]
22. Xie, R.; Jia, X. Transmission-Efficient Clustering Method for Wireless Sensor Networks Using Compressive Sensing. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 806–815. [[CrossRef](#)]
23. Razzaque, M.A.; Dobson, S. Energy-Efficient Sensing in Wireless Sensor Networks Using Compressed Sensing. *Sensors* **2014**, *14*, 2822–2859.
24. Dengiz, O.; Konak, A.; Smith, A.E. Connectivity Management in Mobile Ad Hoc Networks using Particle Swarm Optimization. *Ad Hoc Netw.* **2011**, *9*, 1312–1326. [[CrossRef](#)]
25. Shen, M.; Zhan, Z.; Chen, W.; Gong, Y.; Zhang, J.; Li, Y. Bi-Velocity Discrete Particle Swarm Optimization and Its Application to Multicast Routing Problem in Communication Networks. *IEEE Trans. Ind. Electron.* **2014**, *61*, 7141–7151. [[CrossRef](#)]
26. Tukisi, T.W.; Mathaba, T.N.D. Multi-hop PSO Based Routing Protocol for Wireless Sensor Networks with Energy Harvesting. In Proceedings of the 2019 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, 6–8 March 2019. [[CrossRef](#)]
27. RejinaParvin, J.; Vasanthanayaki, C. Particle Swarm Optimization-Based Clustering by Preventing Residual Nodes in Wireless Sensor Networks. *IEEE Sens. J.* **2015**, *15*, 4264–4274. [[CrossRef](#)]
28. Nguye, M.T.; Rahnavard, N. Cluster-Based Energy-Efficient Data Collection in Wireless Sensor Networks Utilizing Compressive Sensing. In Proceedings of the Military Communications Conference, MILCOM 2013-2013 IEEE, San Diego, CA, USA, 18–20 November 2013.
29. Nguyen, M.T.; Teague, K.A.; Rahnavard, N. Inter-Cluster Multi-hop Routing in Wireless Sensor Networks employing Compressive Sensing. In Proceedings of the 2014 IEEE Military Communications Conference, Baltimore, MD, USA, 6–8 October 2014. [[CrossRef](#)]
30. Nguye, M.T.; Teague, K.A.; Rahnavard, N. CCS: Energy-Efficient Data Collection in Clustered Wireless Sensor Networks Utilizing Block-Wise Compressive Sensing. *Comput. Netw.* **2016**, *106*, 171–185. [[CrossRef](#)]
31. Liu, G.; Kang, W. Distributed Random Multi-Hop Compressed Sensing in 3-D Underwater Sensor Networks. *Inf. Technol. J.* **2014**, *13*, 941–947. [[CrossRef](#)]
32. Chen, S.; Donoho, D.; Saunders, M. Atomic Decomposition by Basis Pursuit. *SIAM Rev.* **2001**, *43*, 129–159.
33. Fosson, S.M.; Matamoros, J.; Antón-Haro, C. Distributed Recovery of Jointly Sparse Signals under Communication Constraints. *IEEE Trans. Signal Process.* **2016**, *64*, 3470–3482. [[CrossRef](#)]
34. Matamoros, J.; Fosson, S.M.; Magli, E.; Antón-Haro, C. Distributed ADMM for In-Network Reconstruction of Sparse Signals with Innovations. *IEEE Trans. Signal Inf. Process. Netw.* **2015**, *1*, 225–234.
35. Tropp, J.; Gilbert, A. Signal Recovery from Random Measurements via Orthogonal Matching Pursuit. *IEEE Trans. Inf. Theory* **2007**, *53*, 4655–4666.
36. Dai, W.; Milenkovic, O. Subspace Pursuit for Compressive Sensing Signal Reconstruction. *IEEE Trans. Inf. Theory* **2009**, *55*, 2230–2249.
37. Needell, D.; Tropp, J.A. CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples. *Appl. Comput. Harmon. Anal.* **2009**, *26*, 301–321.
38. Preisig, J.C. Performance Analysis of Adaptive Equalization for Coherent Acoustic Communications in the Time-Varying Ocean Environment. *J. Acoust. Soc. AM* **2005**, *118*, 263–278. [[CrossRef](#)]

39. Iglesias, I.; Song, A.; Garcia-Frias, J.; Badiey, M.; Arce, G.R. Image Transmission Over the Underwater Acoustic Channel via Compressive Sensing. In Proceedings of the 2011 45th Annual Conference on Information Sciences and Systems, Baltimore, MD, USA, 23–25 March 2011. [[CrossRef](#)]
40. Hassanin, M.; Garcia-Frias, J. Analog Mappings for Non-linear Channels with Applications to Underwater Channels. *IEEE Trans. Commun.* **2020**, *68*, 445–455. [[CrossRef](#)]
41. Ji, S.; Xue, Y.; Carin, L. Bayesian Compressive Sensing. *IEEE Trans. Signal Process.* **2008**, *56*, 2346–2356. [[CrossRef](#)]
42. Van, T.; Harry, L. *Detection, Estimation and Modulation Theory*; Wiley: New York, NY, USA, 2004. [[CrossRef](#)]
43. Zorzi, M.; Casari, P.; Baldo, N.; Harris, A.F. Energy-Efficient Routing Schemes for Underwater Acoustic Networks. *IEEE J. Sel. Areas Commun.* **2008**, *26*, 1754–1766. [[CrossRef](#)]
44. Stojanovic, M. On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel. In Proceedings of the International Workshop on UnderWater Networks (WUWNet), Los Angeles, CA, USA, 25 September 2006. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).