Hindawi
Journal of Sensors
Volume 2019, Article ID 7039510, 9 pages
https://doi.org/10.1155/2019/7039510

Research Article

WSNs Compressed Sensing Signal Reconstruction Based on Improved Kernel Fuzzy Clustering and Discrete Differential Evolution Algorithm

Zhou-zhou Liu [1,2] and Shi-ning Li [1]

[1] *School of Computer Science, Xi'an Aeronautical University, Xi'an, China*
[2] *School of Computer Science, Northwestern Polytechnical University, Xi'an, China*

Correspondence should be addressed to Zhou-zhou Liu; liuzhouzhou8192@126.com

Received 21 February 2019; Accepted 15 May 2019; Published 16 June 2019

Academic Editor: Stelios M. Potirakis

To reconstruct compressed sensing (CS) signal fast and accurately, this paper proposes an improved discrete differential evolution (IDDE) algorithm based on fuzzy clustering for CS reconstruction. Aiming to overcome the shortcomings of traditional CS reconstruction algorithm, such as heavy dependence on sparsity and low precision of reconstruction, a discrete differential evolution (DDE) algorithm based on improved kernel fuzzy clustering is designed. In this algorithm, fuzzy clustering algorithm is used to analyze the evolutionary population, which improves the pertinence and scientificity of population learning evolution while realizing effective clustering. The differential evolutionary particle coding method and evolutionary mechanism are redefined. And the improved fuzzy clustering discrete differential evolution algorithm is applied to CS reconstruction algorithm, in which signal with unknown sparsity is considered as particle coding. Then the wireless sensor networks (WSNs) sparse signal is accurately reconstructed through the iterative evolution of population. Finally, simulations are carried out in the WSNs data acquisition environment. Results show that compared with traditional reconstruction algorithms such as StOMP, the reconstruction accuracy of the algorithm proposed in this paper is improved by 36.4-51.9%, and the reconstruction time is reduced by 15.1-31.3%.

1. Introduction

The rapid development of wireless sensor networks (WSNs) has binging human society into the era of big data. However, huge amount of data collected from sensor network is also accompanied by the sharp increase of the signal bandwidth [1]. If we still adopt the traditional Nyquist sampling theorem for data acquisition in this case, it will bring unprecedented challenges to the hardware system [2]. The advent of compressive sensing (CS) [3] solved this problem and created a whole new approach for the signal information processing [4, 5]. Through a series of nonlinear optimization algorithm, CS is able to accurately reconstruct compressible signal from a small amount of data and greatly reduced the requirement for the sampling rate [6]. Thus, the theory of compressed sensing is gradually and widely applied to image processing, wireless sensor networks, radio communication, etc.

Sparse representation, observation matrix, and reconstruction algorithm are the core contents in compressed sensing, and the reconstruction algorithm is the most critical part [7]. It is commonly divided into three kinds: convex optimization algorithm, greedy algorithm, and their combination. The key problem in CS reconstruction is how to use a small amount of observed signal to reconstruct the original signal rapidly and accurately. Furthermore, several problems, such as high computational complexity, large amount of data, and preset sparsity, still need to be solved. Reference [7] proposed a SL0 reconstruction algorithm for compressive sensing based on BFGS quasi Newton method, which effectively improved the iterative efficiency of greedy matching pursuit algorithm by taking advantage of the rapid convergence of quasi Newton method. Wang et al. [2] introduced Dice coefficient matching metric into SWOMP reconstruction algorithm to overcome the inherent defect that the inner product matching criterion of SWOMP is sensitive to residual signals. Simulation results show that their algorithm has higher signal reconstruction success rate. In [8], researchers discussed the signal reconstruction algorithm for compressed sensing without prior information of signal sparsity. But the signal reconstruction accuracy of this algorithm needs to be improved.

Currently, most compressed sensing algorithms reconstruct the original signal by minimizing its l_0 or l_1 norm [9]. This is NP-hard problem. Therefore, we propose a WSNs compressed sensing signal reconstruction algorithm based on fuzzy clustering and discrete differential evolution. In this algorithm, we consider signal with unknown sparsity as differential evolution particle coding and reconstruct the signal through population iterative evolution. To improve the efficiency and accuracy of signal reconstruction, fuzzy clustering is utilized while the differential evolution mechanism is redefined. Finally, simulation experiments are carried out to verify its validity.

2. Differential Evolution Algorithm Based on Fuzzy Clustering

2.1. Differential Evolution. Differential evolution (DE) [10] is a kind of intelligent optimization algorithm with random search. It preserves the individuals with better fitness through mutation and competition to achieve the desired end. Due to its excellent global convergence and good robustness, fewer number of control parameters, and easy to be implemented, DE has been widely used in solving complex optimization problems [11].

An iteration of the classical DE algorithm mainly consists of three basic steps: mutation, crossover, and selection. For N-dimensional optimization, there exists a differential evolution population composed of P particles $\boldsymbol{X}_i^t(x_{i1}^t, \cdots, x_{iN}^t)$ $(i \in [1, P])$ for the current generation t. The process of performing mutation on $\boldsymbol{X}_i^t(x_{i1}^t, \cdots, x_{iN}^t)$ can be expressed as

$$\boldsymbol{V}_i^t = \boldsymbol{X}_a^t + \boldsymbol{F} \times \left(\boldsymbol{X}_b^t - \boldsymbol{X}_c^t\right) \tag{1}$$

where $F \in [0.4, 1]$ denotes the scalar number, $\boldsymbol{V}_i^t$ denotes the mutant particle. The indices $a, b, c \in [1, P]$ are mutually exclusive integers randomly chosen from the range $[1, P]$, which are also different from i. In essence, mutation randomly selects three different particles from the current population, multiplies the difference of any two of these three particles with the scalar number F, and adds the scaled difference to the third one to obtain the mutant particle to fuse more individual information.

To enhance the potential diversity of the population, crossover operation is then performed. It is a process of selecting components from $\boldsymbol{V}_i^t(v_{i1}^t, \cdots, v_{iN}^t)$ and $\boldsymbol{X}_i^t$ according to certain probability to obtain the trial particle, described as

$$\begin{aligned} \boldsymbol{U}_i^t\left(u_{i1}^t, \cdots, u_{iN}^t\right) &\Longleftarrow u_{ij}^t \\ &= \begin{cases} v_{ij}^t & \left(r_{andb}(j) \le C_R \ or \ j = r_{nbr}(i)\right) \\ x_{ij}^t & \left(r_{andb}(j) > C_R \ or \ j \ne r_{nbr}(i)\right) \end{cases} \end{aligned} \tag{2}$$

where $C_R \in (0, 1)$ is called the crossover rate, $r_{andb}(j) \in (0, 1)$ is a random number lying between 0 and 1, and $r_{nbr}(i)$ is an integer randomly selected from range $[1, N]$, which ensures that $\boldsymbol{U}_i^t$ gets at least one component from $\boldsymbol{V}_i^t$.

Then, the next generation particle $\boldsymbol{X}_i^{t+1}$ is selected in the form of roulette to keep the population size constant over subsequent generations. Taking the minimum optimization as an example, the selection operation is described as

$$\boldsymbol{X}_i^{t+1} = \begin{cases} \boldsymbol{U}_i^t, & if \ f\left(\boldsymbol{U}_i^t\right) \le f\left(\boldsymbol{X}_i^t\right) \\ \boldsymbol{X}_i^t, & else \end{cases} \tag{3}$$

where $f(\boldsymbol{X}_i^t)$ is the fitness value of particle $\boldsymbol{X}_i^t$.

We can see that, in each iteration, the population always keeps the better individual with higher fitness, so as to reach the global optimal solution.

2.2. Improved Kernel FCM. Fuzzy C-means clustering algorithm (FCM) is one of the most widely used clustering algorithms. It uses membership to describe the degree to which a data belongs to a certain subclass for data classification [12]. For a S-dimension dataset $\boldsymbol{X} \in R^S$, clustering is to divide n samples $\boldsymbol{x}_k \in \boldsymbol{X}$, $k = 1, 2, \cdots, n$, into C subclasses while making the value of the clustering objective function minimum; that is,

$$\min \quad J(\boldsymbol{U}, \boldsymbol{V}) = \sum_{i=1}^{C} \sum_{k=1}^{n} \mu_{ik}^m \left\| \boldsymbol{x}_k - \boldsymbol{v}_i \right\|^2 \tag{4}$$

where μ_{ik} is the membership of sample $\boldsymbol{x}_k$ to the ith subclass, $\boldsymbol{U} = [\mu_{ik}]_{C\times n}$ is the membership matrix, $\boldsymbol{V} = \{\boldsymbol{v}_i\}$ $(i = 1, 2, \cdots, C)$ is the set of clustering center, and m is the fuzzy weighted index. Let $\partial J/\partial \mu_{ik} = 0$, $\partial J/\partial v_i = 0$; we have

$$\begin{aligned} &\frac{\partial J}{\partial \mu_{ik}} = 0 \Longrightarrow \\ &\mu_{ik} = \frac{\left\| \boldsymbol{x}_k - \boldsymbol{v}_i \right\|^{-2/(m-1)}}{\sum_{j=1}^{C} \left\| \boldsymbol{x}_k - \boldsymbol{v}_j \right\|^{-2/(m-1)}}, \\ &\frac{\partial J}{\partial v_i} = 0 \Longrightarrow \\ &v_i = \frac{\sum_{k=1}^{n} \mu_{ik}^m \boldsymbol{x}_k}{\sum_{k=1}^{n} \mu_{ik}^m} \\ &s.t. \quad \sum_{i=1}^{C} \mu_{ik} = 1, \quad \mu_{ik} \in [0, 1], \\ &\qquad \sum_{k=1}^{n} \mu_{ik} \in (0, 1) \end{aligned} \tag{5}$$

It can be seen that FCM is a kind of local search algorithm and is sensitive to initial values [13]. Since FCM uses Euclidean distance to evaluate the similarity, it is only suitable for processing data with appropriate compactness and good dispersion. Therefore, kernel fuzzy C-means clustering algorithm (KFCM) is used to replace the Euclidean distance with kernel-induced distance, expressed as

$$\min \quad J(\boldsymbol{U},\boldsymbol{V}) = \sum_{i=1}^{C}\sum_{k=1}^{n}\mu_{ik}^{m}\left\|\boldsymbol{\Phi}(\boldsymbol{x}_k)-\boldsymbol{\Phi}(\boldsymbol{v}_i)\right\|^2 \tag{6}$$

$\|\boldsymbol{\Phi}(\boldsymbol{x}_k)-\boldsymbol{\Phi}(\boldsymbol{v}_i)\|^2$ in (6) can be further derived into the following form:

$$\begin{aligned}\left\|\boldsymbol{\Phi}(\boldsymbol{x}_k)-\boldsymbol{\Phi}(\boldsymbol{v}_i)\right\|^2 &= \left(\boldsymbol{\Phi}(\boldsymbol{x}_k)-\boldsymbol{\Phi}(\boldsymbol{v}_i)\right)^T\left(\boldsymbol{\Phi}(\boldsymbol{x}_k)-\boldsymbol{\Phi}(\boldsymbol{v}_i)\right)\\ &= \boldsymbol{\Phi}^T(\boldsymbol{x}_k)\boldsymbol{\Phi}(\boldsymbol{x}_k)-\boldsymbol{\Phi}^T(\boldsymbol{x}_k)\boldsymbol{\Phi}(\boldsymbol{v}_i)\\ &\quad -\boldsymbol{\Phi}(\boldsymbol{x}_k)\boldsymbol{\Phi}^T(\boldsymbol{v}_i)+\boldsymbol{\Phi}^T(\boldsymbol{v}_i)\boldsymbol{\Phi}(\boldsymbol{v}_i)\\ &= K(\boldsymbol{x}_k,\boldsymbol{x}_k)-2K(\boldsymbol{x}_k,\boldsymbol{v}_i)+K(\boldsymbol{v}_i,\boldsymbol{v}_i)\end{aligned} \tag{7}$$

where $K(\boldsymbol{x},\boldsymbol{v}) = \boldsymbol{\Phi}^T(\boldsymbol{x})\boldsymbol{\Phi}(\boldsymbol{v})$ is the inner product of kernel function. In this paper, we select Gaussian function as $K(\boldsymbol{x},\boldsymbol{v})$, which means $K(\boldsymbol{x},\boldsymbol{v}) = \exp(-\|\boldsymbol{x}_k-\boldsymbol{v}_i\|^2/\sigma^2)$, $K(\boldsymbol{x}_k,\boldsymbol{x}_k) = 1$, $K(\boldsymbol{v}_k,\boldsymbol{v}_k) = 1$, and $\boldsymbol{\Phi}^T(\boldsymbol{x}_k)\boldsymbol{\Phi}(\boldsymbol{v}_i) = \boldsymbol{\Phi}(\boldsymbol{x}_k)\boldsymbol{\Phi}^T(\boldsymbol{v}_i)$. Thus, the clustering objective function, membership μ_{ik}, and clustering center $\boldsymbol{v}_i$ can be calculated according to the following equations:

$$\begin{aligned}\min \quad & J(\boldsymbol{U},\boldsymbol{V})\\ &= 2\sum_{i=1}^{C}\sum_{k=1}^{n}\mu_{ik}^{m}\left[1-K(\boldsymbol{x}_k,\boldsymbol{v}_i)+K(\boldsymbol{v}_i,\boldsymbol{v}_i)\right]\end{aligned} \tag{8}$$

$$\begin{aligned}\mu_{ik} &= \frac{\left[1-K(\boldsymbol{x}_k,\boldsymbol{v}_i)\right]^{-1/(m-1)}}{\sum_{j=1}^{C}\left[1-K(\boldsymbol{x}_k,\boldsymbol{v}_j)\right]^{-1/(m-1)}},\\ v_i &= \frac{\sum_{k=1}^{n}\mu_{ik}^{m}K(\boldsymbol{x}_k,\boldsymbol{v}_i)\boldsymbol{x}_k}{\sum_{k=1}^{n}\mu_{ik}^{m}K(\boldsymbol{x}_k,\boldsymbol{v}_i)}\end{aligned} \tag{9}$$

Although KFCM adopts kernel-induced distance to broaden its application range, it still has many problems waiting to be solved. Firstly, the clustering number C should be set in advance, but the preset value will affect the final clustering result directly. Secondly, KFCM usually generates the initial clustering center through random initialization, which reduces the clustering effectiveness. Finally, the value of fuzzy weighted index m, to a great extent, affects the clustering result.

2.3. Implementation of Improved Kernel FCM. In this part, we introduce validity index based on compactness and dispersion to realize the automatic classification of clustering number and initialization of the cluster center with "maximum benefit" to increase the clustering performance.

Validity index an appropriate validity index will well reflect the clustering quality. The number of clusters that makes the validity index reaches the optimum is called the optimum number of clusters. Therefore, we use compactness *Var* and dispersion *Sep* to define validity index $VS(\boldsymbol{U},\boldsymbol{V})$ as

$$VS(\boldsymbol{U},\boldsymbol{V}) = \frac{Var^{\triangle}(\boldsymbol{U},\boldsymbol{V})}{Sep^{\triangle}(\boldsymbol{U},C)} \tag{10}$$

$$\begin{aligned}Var^{\triangle}(\boldsymbol{U},\boldsymbol{V}) &= \frac{Var(\boldsymbol{U},\boldsymbol{V})}{Var_{\max}},\\ & \qquad Var_{\max} = \underbrace{\max}_{C} Var(\boldsymbol{U},\boldsymbol{V})\\ Sep^{\triangle}(\boldsymbol{U},\boldsymbol{V}) &= \frac{Sep(\boldsymbol{U},C)}{Sep_{\max}},\\ & \qquad Sep_{\max} = \underbrace{\max}_{C} Sep(\boldsymbol{U},C)\end{aligned} \tag{11}$$

where $C = 2,3,\cdots,C_{\max}$ and $C_{\max}$ is the maximum number of clusters. The compactness *Var* and dispersion *Sep* are computed in the following way:

$$\begin{aligned}Var(\boldsymbol{U},\boldsymbol{V}) &= \left[\sum_{i-1}^{C}\sum_{j=1}^{n}\frac{\mu_{ik}d^2(\boldsymbol{x}_j,\boldsymbol{v}_i)}{n(i)}\right]\sqrt{\left(\frac{C+1}{C-1}\right)}\\ d(\boldsymbol{x}_j,\boldsymbol{v}_i) &= \sqrt{1-\exp\left(-\beta\left\|\boldsymbol{x}_j-\boldsymbol{v}_i\right\|^2\right)},\\ \beta &= \left(\frac{1}{n}\sum_{j=1}^{n}\left\|\boldsymbol{x}_j-\left(\frac{1}{n}\sum_{j=1}^{n}\boldsymbol{x}_j\right)\right\|^2\right)^{-1}\end{aligned} \tag{12}$$

$$Sep(\boldsymbol{U},C) = 1-\max_{i\neq j}\left(\max_{i\neq j}\left(\min\left(\mu_{ik},\mu_{jk}\right)\right)\right) \tag{13}$$

where $n(i)$ is the sample number of the ith subclass. From the calculation formulas above, it can be seen that a smaller *Var* means better homogeneity within a class, and a larger *Sep* indicates greater separation between classes. In summary, the smaller the value of validity index is, the better the clustering quality.

Initialization of the cluster center with "maximum benefit" to reduce the influence of the random initialization cluster center has on the clustering results, the initialization of the cluster center with "maximum benefit" is designed. In sample set $\boldsymbol{X} = \{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$, randomly select one sample $\boldsymbol{x}_k$, $k = 1,2,\cdots,n$, as the first classification clustering center, written as $\boldsymbol{v}_1 = \boldsymbol{x}_k$. Then, select another sample $\boldsymbol{x}_j$ $(j \neq k)$ with probability $P(\boldsymbol{x}_j \longrightarrow \boldsymbol{v}_2)$ from the remaining samples as the second classification clustering center, expressed as $\boldsymbol{v}_2 = \boldsymbol{x}_j$. Here, the calculation formula for $P(\boldsymbol{x}_j \longrightarrow \boldsymbol{v}_2)$ is

$$\begin{aligned}P(\boldsymbol{x}_j \longrightarrow \boldsymbol{v}_2) &= \frac{\left\|\boldsymbol{x}_j-\boldsymbol{v}_1\right\|^2}{\sum_{i=1}^{n}\left\|\boldsymbol{x}_i-\boldsymbol{v}_1\right\|^2},\\ & \qquad i = 1,2,\cdots,n \wedge i \neq k\end{aligned} \tag{14}$$

(1) Set the original number of clusters $C = 2$ and initialize parameters: maximum number of clusters $C_{\max}$, fuzzy weighted index m, maximum number of iterations $T_{\max}$, and the stopping criterion Θ.
(2) Initialize the clustering center $\boldsymbol{V}$ according to the "maximum benefit" clustering center initialization method. Calculate the initial membership matrix according to (9).
For $C = 2 : C_{\max}$
(3) While $(|\boldsymbol{v}_k^t - \boldsymbol{v}_k^{t-1}| \leq \Theta \| t \leq T_{\max})$ do
(4) {Calculate membership matrix $\boldsymbol{U}^t$ and clustering center $\boldsymbol{V}^t$ according to (9).
(5) $t + 1 \longrightarrow t$. }
(6) Calculate validity index $VS_C(\boldsymbol{U}, \boldsymbol{V})$ according to (10)-(13).
End for
(7) Output results: the C corresponding to the minimum $VS_C(\boldsymbol{U}, \boldsymbol{V})$ is the optimal number of clusters, and the corresponding clustering center and membership is the optimal $\boldsymbol{V}$ and $\boldsymbol{U}$ respectively.

Pseudocode 1

The third classification clustering center, $\boldsymbol{v}_3 = \boldsymbol{x}_w$ $(w \neq j \neq k)$, is selected from the remaining samples with the probability $P(\boldsymbol{x}_w \longrightarrow \boldsymbol{v}_3)$:

$$P\left(\boldsymbol{x}_w \longrightarrow \boldsymbol{v}_3\right) = \frac{\left\|\boldsymbol{x}_w - \boldsymbol{v}_1\right\|^2 + \left\|\boldsymbol{x}_w - \boldsymbol{v}_2\right\|^2}{\sum_{i=1}^{n}\left\|\boldsymbol{x}_i - \boldsymbol{v}_1\right\|^2 + \sum_{i=1}^{n}\left\|\boldsymbol{x}_i - \boldsymbol{v}_2\right\|^2}, \quad i = 1, 2, \cdots, n \wedge i \neq k \neq j \tag{15}$$

and so on, until the C clustering centers are all selected. Such procedure ensures enough differences between classes and effectively reduces the probability of the algorithm falling into local optimal.

The improved kernel FCM (IKFCM) can realize the automatic classification of clustering number. Its pseudocode is shown in Pseudocode 1.

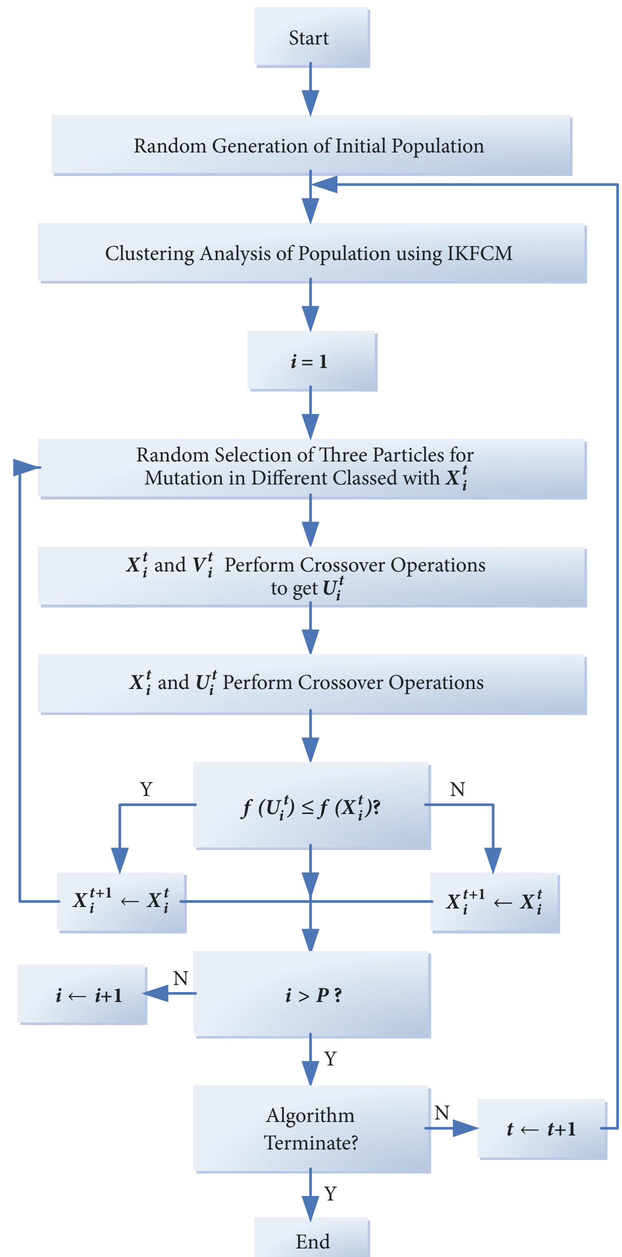

Figure 1: Flow chart of IDDE.

2.4. Implementation of Differential Evolution Based on IKFCM. Like other intelligent optimization algorithms, the population diversity of DE decreases with the increase of iteration times, and the random selection of individuals for mutation may accelerate the population trapped in local optimal [14]. Thus, in order to further improve the global optimization ability and convergence accuracy of DE, we use fuzzy clustering to analyze the population, making it more scientific for selecting particles for mutation in each iteration.

As one of the most widely used clustering algorithms, FCM classifies data through analyzing how closely a data is relative to different classes, making data samples with more similarity be classified into the same class. In this paper, we utilize IKFCM for cluster analysis before DE, and its classification result is considered as the basis of selecting particles for mutation. After a certain number of iterations, the population is divided into C subclasses by IKFCM, and there are more similarities within classes and greater differences between classes. When it comes to mutation, particles $\boldsymbol{X}_a^t$, $\boldsymbol{X}_b^t$, and $\boldsymbol{X}_c^t$ will be selected from different classes, thus effectively expanding the diversity of mutant particle and ensuring that particles with larger differences will be participated in the population's evolution and eventually improves the convergence performance of DE. The specific process of the DE based on fuzzy clustering, named IDDE, is presented in Figure 1.

3. Compressed Sensing Signal Reconstruction Based on IDDE

3.1. Compressed Sensing Sparse Signal Reconstruction. In compressed sensing, $\Psi_{N\times N}$ is called sparse matrix if it can be used to linearly describe signal $\boldsymbol{X}_{N\times 1}$ sparsely. We can express it as

$$\boldsymbol{X}_{N\times 1} = \Psi_{N\times N}\boldsymbol{s}_{N\times 1} \tag{16}$$

where $\boldsymbol{s}_{N\times 1}(s_1, \cdots, s_N)$ denotes the linear description of $\boldsymbol{X}_{N\times 1}$ under $\Psi_{N\times N}$ and its sparsity satisfies $K << N$. There exists a measurement matrix $\Phi_{M\times N}$ that allows $\boldsymbol{X}_{N\times 1}$ be described with a small amount of data as

$$\boldsymbol{y}_{M\times 1} = \Phi_{M\times N}\boldsymbol{X}_{N\times 1} \tag{17}$$

where $\boldsymbol{y}_{M\times 1}$ is the measurement vector and satisfies $M << N$. From (6) and (7), we have

$$\boldsymbol{y}_{M\times 1} \overset{A_{M\times N}=\Phi_{M\times N}\Psi_{N\times N}}{=} A_{M\times N}\boldsymbol{s}_{N\times 1} = A\boldsymbol{s} \tag{18}$$

Once $A_{M\times N}$ satisfies RIP condition [12], the original signal can be reconstructed by solving the l_0 norm; that is,

$$\begin{aligned} \min \quad & \|s\|_{l_0} \\ s.t. \quad & y = As = \Phi\Psi s \end{aligned} \tag{19}$$

Since the process of solving (9) is a NP-hard, considering the signal with unknown sparsity as the differential evolution particle coding, we propose a compressive sensing signal reconstruction algorithm based on IDDE, which accurately reconstructed the unknown sparsity signal through population evolution.

3.2. Implementation of Compressed Sensing Signal Reconstruction Based on IDDE. DE is mainly used for continuous optimization, but for discrete optimization, it will generate large numbers of solutions that do not conform to the requirements, severely reducing its convergence efficiency. In this case, taking the CS reconstruction algorithm into account, the differential evolutionary particle coding method and evolutionary mechanism in IDDE are redefined.

Definition 1 (particle coding). In CS reconstruction algorithm optimization, define particle coding as

$$\boldsymbol{X}_i(x_{i1}, \cdots, x_{iN}) \longleftarrow x_{ij} = \begin{cases} 1, & if\ s_j \neq 0 \\ 0, & else \end{cases} \tag{20}$$

It is obvious that $\boldsymbol{X}_i(x_{i1}, \cdots, x_{iN})$ is correspondence with the nonzero elements in the sparse signal $\boldsymbol{s}_{N\times 1}(s_1, \cdots, s_N)$.

Definition 2. Randomly select F' ($1 \leq F' \leq N$) code bits in particle $\boldsymbol{X}_j$, and replace the corresponding code bits in particle $\boldsymbol{X}_i$ ($i \neq j$) with it. The exchange process is denoted as $F'(\boldsymbol{X}_i \longleftarrow \boldsymbol{X}_j)$:

$$F'\left(\boldsymbol{X}_i \longleftarrow \boldsymbol{X}_j\right) = \underbrace{x_{i1} \longleftarrow x_{j1}, \cdots, x_{jk} \longleftarrow x_{jk}}_{F'} \tag{21}$$

Then, the mutation step in IDDE can be refreshed as

$$\begin{aligned} \boldsymbol{V}_{i,1}^t &= F_1'\left(\boldsymbol{X}_i \longleftarrow \boldsymbol{X}_a\right), \\ \boldsymbol{V}_{i,2}^t &= F_2'\left(\boldsymbol{X}_i \longleftarrow \boldsymbol{X}_b\right), \\ \boldsymbol{V}_{i,3}^t &= F_3'\left(\boldsymbol{X}_i \longleftarrow \boldsymbol{X}_c\right) \end{aligned} \tag{22}$$

where $\boldsymbol{X}_a$, $\boldsymbol{X}_b$, and $\boldsymbol{X}_c$ are three particles randomly selected from three different clusters, which are also different from the cluster that contain $\boldsymbol{X}_i$.

As can be seen from (22), by performing crossover on the three mutant particles, respectively, we will obtain three trial particles $\boldsymbol{U}_{i,1}^t$, $\boldsymbol{U}_{i,2}^t$, and $\boldsymbol{U}_{i,3}^t$. Then, a particle with better fitness will be selected between the three trial particles and $\boldsymbol{X}_i$ as

$$\boldsymbol{X}_i^{t+1} = \begin{cases} \boldsymbol{U}_{i,j}^t, & \boldsymbol{U}_{i,j}^t = \min\limits_{f()}\left(\boldsymbol{U}_{i,1}^t, \boldsymbol{U}_{i,2}^t, \boldsymbol{U}_{i,3}^t\right) \\ \boldsymbol{X}_i^t, & else \end{cases} \tag{23}$$

Definition 3 (objective function). In CS reconstruction algorithm optimization, define the objective function as

$$\min \quad f\left(\boldsymbol{X}_i\right) = \left\|\boldsymbol{y} - \Phi\Psi\boldsymbol{X}_i\right\|_2 \tag{24}$$

As the objective function of CS reconstruction algorithm optimization is defined, IDDE would locate the nonzero elements of the sparse signal code after certain times of evolution and obtain the amplitude of the nonzero elements through least square method. Finally, the signal is precisely reconstructed. The performance of the CS reconstruction algorithm is superior due to its utilization of fuzzy clustering and improved discrete differential evolution algorithm. The pseudocode of the CS reconstruction algorithm based on IDDE is shown in Pseudocode 2.

3.3. Application of IDDE in WSNs. In this part, we study the sparse event detection [13] in the context of wireless sensor networks (WSNs). Q wireless sensor nodes are placed within the monitoring scope of WSNs to monitor N event sources. K network events randomly occurred from the N event sources at some time. We use vector $\boldsymbol{s}_{N\times 1} = [s_1, s_2, \cdots, s_N]^T$ to denote the energy signal of the event sources ($s_i = 0$ means that the ith event source did not happen any event). Obviously, if $K \ll N$, $\boldsymbol{s}_{N\times 1}$ is a sparse signal with sparsity of K. Assuming $Q = N$, signals received by N nodes $\boldsymbol{x}_{N\times 1} = [x_1, x_2, \cdots, x_N]^T$ can be written as

(1) Initialize parameters: maximum number of iterations $T_{\max}$, cluster number C and the stopping criterion.
(2) Initialize the population of IDDE, and set $t = 0$.
(3) While (the requirement of IDDE is not satisfied) do
{
(4) While (the stopping criterion of IKFCM is not satisfied) do// cluster analysis the population
{
(5) Calculate $\boldsymbol{H} = [h_{ik}]_{C\times P}$ and $\boldsymbol{O} = \{\boldsymbol{o}_i\}$ according to (3).
}
(6) For $i = 1 : P$// update the particle
(7) Randomly select 3 particles from different clusters which do not contain $\boldsymbol{X}_i^t$. Perform mutation according to (22) to obtain $\boldsymbol{V}_{i,1}^t$, $\boldsymbol{V}_{i,2}^t$ and $\boldsymbol{V}_{i,3}^t$.
(8) Generate $\boldsymbol{U}_{i,1}^t$, $\boldsymbol{U}_{i,2}^t$ and $\boldsymbol{U}_{i,3}^t$ through crossover in the following way:

$$\boldsymbol{U}_i^t\left(u_{i1}^t,\cdots,u_{iN}^t\right) \Longleftarrow u_{ij}^t = \begin{cases} v_{ij}^t & \left(r_{andb}\left(j\right) \le C_R \text{ or } j = r_{nbr}\left(i\right)\right) \\ x_{ij}^t & \left(r_{andb}\left(j\right) > C_R \text{ or } j \ne r_{nbr}\left(i\right)\right) \end{cases}$$

(9) Evaluate the trial particle U_i^t
If $U_{i,j}^t = \min_{f()}(U_{i,1}^t, U_{i,2}^t, U_{i,3}^t)$, then $X_i^{t+1} = U_{i,j}^t$
Else $X_i^{t+1} = X_i^t$
End if
(10) End for
(11) $t + 1 \longrightarrow t$
}
(12) Output results.

Pseudocode 2

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \left|h_{1,1}\right|\left(d_{1,1}\right)^{-\alpha} & \left|h_{1,2}\right|\left(d_{1,2}\right)^{-\alpha} & \cdots & \left|h_{1,N}\right|\left(d_{1,N}\right)^{-\alpha} \\ \left|h_{2,1}\right|\left(d_{2,1}\right)^{-\alpha} & \left|h_{2,2}\right|\left(d_{2,2}\right)^{-\alpha} & \cdots & \left|h_{2,N}\right|\left(d_{2,N}\right)^{-\alpha} \\ \vdots & \vdots & \ddots & \vdots \\ \left|h_{N,1}\right|\left(d_{M,1}\right)^{-\alpha} & \left|h_{N,2}\right|\left(d_{M,2}\right)^{-\alpha} & \cdots & \left|h_{N,N}\right|\left(d_{M,N}\right)^{-\alpha} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix} = \boldsymbol{\Psi}_{N\times N}\boldsymbol{s} \tag{25}$$

where $d_{i,j}$ is the distance the signal transmitted, $h_{i,j}$ is the energy attenuation model. According to CS theory, sparse event can be detected by the data received by M nodes.

$$\boldsymbol{y}_{M\times 1} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots & \\ 0 & 0 & \cdots & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \boldsymbol{\Phi}_{M\times N}\boldsymbol{x}_{N\times 1} \tag{26}$$

By (25) and (26), we have $\boldsymbol{y}_{M\times 1} = \boldsymbol{\Phi}_{M\times N}\boldsymbol{x}_{N\times 1} = \boldsymbol{\Phi}_{M\times N}\boldsymbol{\Psi}_{N\times N}\boldsymbol{s}_{N\times 1} = \boldsymbol{A}_{M\times N}\boldsymbol{s}_{N\times 1}$. It can be shown that when $M \ge O(c(K+1)\ln(N/K))$, $\boldsymbol{A}$ can meet the requirement of RIP. But the CS reconstruction algorithm based on IDDE requires that an accurate measurement matrix should be guaranteed. Thus we transform matrix $\boldsymbol{A}_{M\times N}$ in the following way:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} \tag{27}$$

where $\boldsymbol{\Sigma}_{M\times N}$ is semipositive definite diagonal matrix, $\boldsymbol{U}$, and $\boldsymbol{V}$ is orthogonal matrix. Therefore, (27) can be rewritten as

$$\boldsymbol{y} = \boldsymbol{\Phi}_{M\times N}\boldsymbol{x}_{N\times 1} = \boldsymbol{A}\boldsymbol{s} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{s}$$
$$= \boldsymbol{U}\begin{bmatrix} \overbrace{\begin{matrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_M \end{matrix}}^{M\times M} & \overbrace{\begin{matrix} 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & & 0 \\ 0 & \cdots & 0 \end{matrix}}^{(N-M)\times M} \end{bmatrix}\boldsymbol{V}^T\boldsymbol{s} \tag{28}$$

Equation (28) can be further transformed into

$$\begin{aligned}
&\boldsymbol{\Sigma}_1^{-1}\boldsymbol{U}^T\boldsymbol{y} \\
&\quad = \boldsymbol{\Sigma}_1^{-1}\boldsymbol{U}^T\boldsymbol{U}\left[\overbrace{\boldsymbol{\Sigma}_1}^{M\times M}\ \overbrace{0}^{(N-M)\times M}\right]\left[\overbrace{\boldsymbol{V}_1}^{M\times M}\ \overbrace{\boldsymbol{V}_2}^{(N-M)\times M}\right]\boldsymbol{s} \\
&\quad = \boldsymbol{V}_1{}^T\boldsymbol{s} \Longrightarrow \\
&\boldsymbol{y}' = \boldsymbol{\Sigma}_1^{-1}\boldsymbol{U}^T\boldsymbol{y} = \boldsymbol{V}_1{}^T\boldsymbol{s}
\end{aligned} \tag{29}$$

Equation (29) represents a new measurement system. As seen from the derivation process, the measurement matrix $\boldsymbol{V}_1{}^T$ is row orthogonal matrix; thus RIP condition is satisfied.

In this case, sparse event detection in WSNs is transformed into the problem of using M wireless sensor nodes to detect K event sources. The detailed steps include training parameters with the experimental data in the deployment test stage of WSNs to obtain the optimal parameter configuration; processing received signal by the use of reconstruction algorithm based on IDDE to obtain the sparse signal vector $\boldsymbol{s}_{N\times 1}$ in the detection stage to monitor sparse events effectively and timely.

4. Simulation

Simulations are performed in MATLAB to demonstrate the effectiveness of our algorithm. We deploy $Q = 500$ sensor nodes to monitor the temperature of $N = 120$ places. The event source sequence corresponded to event energy signal $\boldsymbol{s}_{N\times 1} = [s_1, s_2, \cdots, s_N]^T$. Reconstruction success rate RS, reconstruction time T, and reconstruction relative error RE are selected to evaluate the reconstruction performance. The reconstruction relative error RE is calculated as follows:

$$RE = \sqrt{\frac{\left[\sum_{i=1}^{N}\left(s_i - \widehat{s}_i\right)\right]}{\left[\sum_{i=1}^{N} s_i{}^2\right]}} \times 100\% \tag{30}$$

where $\boldsymbol{s}_{N\times 1}(s_1, \cdots, s_N)$ denotes the original signal and $\widehat{\boldsymbol{s}_{N\times 1}}(\widehat{s_1}, \cdots, \widehat{s_N})$ stands for the reconstructed signal. The reconstruction success rate RS refers to the probability that RE is less than the threshold δ.

4.1. Reconstruction Performance. Selecting typical discrete signal as the original signal and the reconstruction performance of the reconstruction algorithm based on IDDE under the assumption that K is unknown is analyzed. The performance of our algorithm is also compared to the blind sparsity reconstruction algorithm in [8], SWOMP in [2], and the classical StOMP. Run each algorithm independently for 50 times, calculate the average reconstruction success rate $\overline{RS}$, the average reconstruction time $\overline{T}$, and the average reconstruction relative error $\overline{RE}$ for performance analysis, and set different δ for different signal. The reconstruction result of the CS reconstruction algorithm based on IDDE is shown in Figure 2. The evaluation index results of the 4 reconstruction algorithms are compared in Table 1.

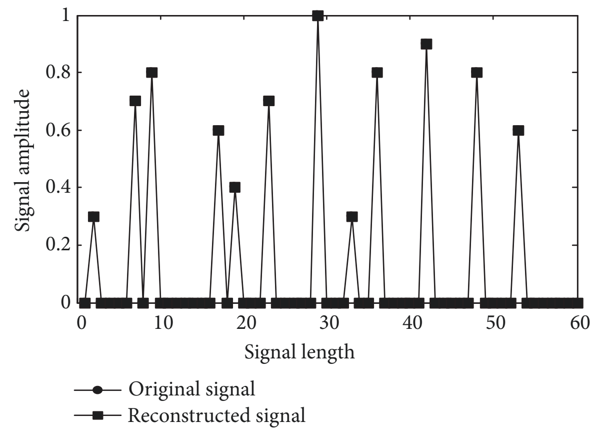

Figure 2: Reconstruction result of the IDDE reconstruction algorithm.

It can be seen from Figure 2 that the CS reconstruction algorithm based on IDDE can obtain enough recovered data points from the compressed sensing data and reconstruct to the original signal, indicating that the algorithm we proposed in this paper is capable of reconstructing signal with unknown sparsity. In Table 1, the reconstruction success rates of the algorithm we propose reached 100% for different original signals, the algorithm in [8] reached above 97%, and the algorithm in [2] was only about 50% and StOMP could hardly reconstruct the original signal. In terms of reconstruction time, our algorithm is significantly faster than the other 3 algorithms, with $\overline{T}$ reduced by 15.1-31.3%. As for reconstruction error, compared with the algorithm in [2], which has relatively better reconstruction success rate, the RE of our algorithm decreased by about 36.4-51.9%. In summary, compared with previous reconstruction algorithms such as StOMP, the CS reconstruction algorithm based on IDDE proposed in this paper performs better, much more suitable for reconstructing signal with unknown sparsity. All these advantages on one hand could be accredited to the utilization of IDDE. Through IDDE, the signal sparsity is transformed into particle coding, which makes the differential evolution algorithm be able to find the global optimum solution while obtaining the location of nonzero elements. On the other hand, fuzzy clustering is used to analyze the population, making the learning target of the population evolution more rational and scientific, which ultimately improves the reconstruction performance of our algorithm.

4.2. Effects of Parameters on the Reconstruction Algorithm Performance. Sparsity K and observation time M are critical to the reconstruction accuracy. Simulations were carried out to examine the reconstruction success rate of the CS reconstruction algorithm based on IDDE under different K and M, respectively, the blind sparsity reconstruction algorithm in [8], SWOMP in [2], and StOMP are also simulated as a comparison. Each algorithm runs 50 times. Comparison results are shown in Figures 3 and 4.

Figure 3 shows that, when the signal length and M are fixed, all the four reconstruction success rate curves decrease with the increase of K. Compared with the other 3 algorithms, our algorithm maintains a reconstruction success rate of over

Table 1: Evaluation index comparisons of the four reconstruction algorithms.

Signal	Index	Algorithm			
		This paper	Literature[8]	Literature[2]	StOMP
Signal1	$\overline{RS}$	100%	97.3%	50.2%	17.3%
	$\overline{T}$/s	0.17	0.38	0.28	0.37
	$\overline{RE}$	0.007	0.012	1.845	3.330
Signal2	$\overline{RS}$	100%	98.2%	49.2%	15.3%
	$\overline{T}$/s	0.25	0.44	0.36	0.82
	$\overline{RE}$	0.014	0.018	1.038	2.143
Signal3	$\overline{RS}$	100%	97.6%	55.7%	10.8%
	$\overline{T}$/s	0.56	0.92	0.66	1.27
	$\overline{RE}$	0.013	0.017	1.836	3.045

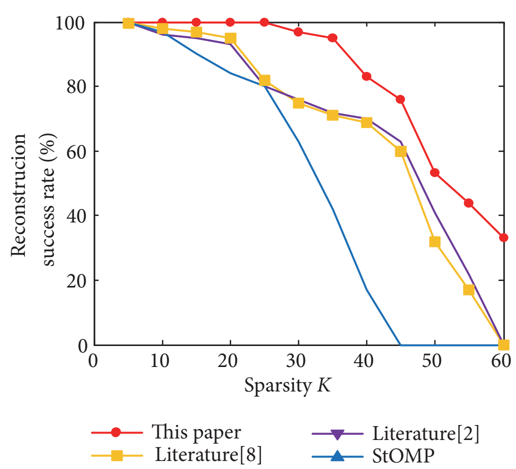

Figure 3: Sparsity K versus reconstruction success rate ($M = 100$).

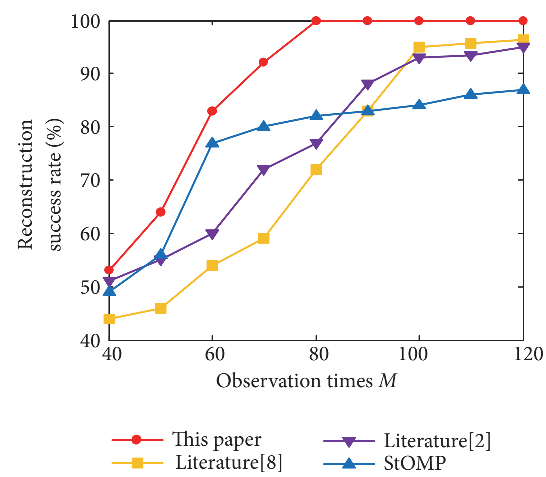

Figure 4: Observation time M versus reconstruction success rate ($K = 20$).

94% when K is within the range of (25, 35), while that of the other 3 algorithms decrease significantly.

As shown in Figure 4, when the signal length and K are unchanged, all the 4 curves rise with the increase of M, and the $\overline{RS}$ of the algorithm we proposed is much better than the other 3 algorithms with the same M. Especially when $M = 80$, the $\overline{RS}$ of our algorithm can reach 100%.

However, large number of sensor nodes means large network energy consumption, which is unfavorable to prolong the survival time of WSNs. Thus, in the following part, we analyze the robustness and antinoise performance of our algorithm.

4.3. Antinoise Performance. Add white noise into the discrete signal to be detected and select the reconstruction algorithm in [8] and StOMP as a comparison. We investigate their antinoise performances under *SNR* ranging from 20dB to 45dB.

From Figure 5, we can see that if the noise is not very serious, the reconstruction success rates of the three algorithms change a little as the noise level changes. When the noise is at a high level, the reconstruction success rates change significantly with the noise level. The $\overline{RS}$ of the algorithm we propose will still remain around 93% if the SNR decrease to 15dB while the other two algorithms decrease obviously, indicating that our algorithm is good at resisting noise.

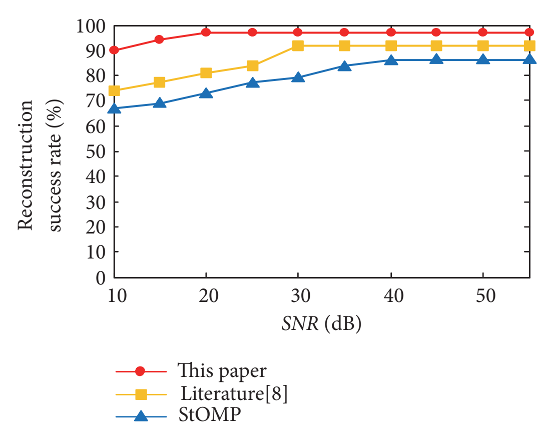

Figure 5: SNR versus reconstruction success rate ($K = 30$, $M = 100$, $N = 120$).

5. Conclusions

In this paper, we propose a compressed sensing signal reconstruction algorithm based on IDDE. The algorithm employs intelligent optimization and fuzzy clustering together, transforming signal reconstruction into the global optimization of differential algorithm, to reconstruct signal with unknown sparsity through fuzzy cluster analysis, particle coding, and differential evolution. Its application in WSNs sparse event detection is also discussed. Simulation results testify that the algorithm we propose performs excellent in signal reconstruction and is robust to noise and interference.

Data Availability

The research data used to support the findings of this study were supplied by government agencies such as laboratories under license and so cannot be made freely available. Requests for access to these data should be made to Zhouzhou Liu via liuzhouzhou8192@126.com.

Conflicts of Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article

Acknowledgments

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported, in part, by the National Natural Science Foundation of China under Grant 61871313, China Postdoctoral Science Foundation funded project, China, under Grant 2018M633573, and Shaanxi Postdoctoral Science Foundation funded project, under Grant 2018BSHQYXMZZ11.

References

[1] T. Wang, Y. Liang, W. Jia, M. Arif, A. Liu, and M. Xie, "Coupling resource management based on fog computing in smart city systems," *Journal of Network and Computer Applications*, vol. 135, pp. 11–19, 2019.

[2] D. Wang, Y. Wu, W. Cao, and J. Xu, "Improved reconstruction algorithm for compressed sensing," *Journal of Northwestern Polytechnical University*, vol. 35, no. 5, pp. 774–779, 2017 (Chinese).

[3] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[5] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?" *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[6] J. Zeng, T. Wang, W. Jia, S. Peng, and G. Wang, "A survey on sensor-cloud," *Computer Research and Development*, vol. 54, no. 5, pp. 925–939, 2017.

[7] N. Sun, J. Liu, and D. Xiao, "SL0 reconstruction algorithm for compressive sensing based on BFGS quasi Newton method," *Journal of Communications*, vol. 40, no. 10, pp. 2408–2414, 2018 (Chinese).

[8] Z.-N. Zhang, R.-T. Huang, and J.-W. Yan, "A blind sparsity reconstruction algorithm for compressed sensing signal," *Acta Electronica Sinica*, vol. 39, no. 1, pp. 18–22, 2011 (Chinese).

[9] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed l^0 norm," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 289–301, 2009.

[10] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[11] H. Feng, L. Luo, Y. Wang, and M. Ye, "Multi-objective data collecting strategies for wireless sensor network based on the time variable multi-salesman problem and genetic algorithm," *Journal on Communication*, vol. 38, no. 3, pp. 112–123, 2017.

[12] F. Zhou, X. Y. Zeng, and L. H. Yang, "A regularized sparse representation method," *Acta Mathematica Sinica*, vol. 58, no. 4, pp. 649–660, 2015 (Chinese).

[13] B. Sun, X. Shan, K. Wu, and Y. Xiao, "Anomaly detection based secure in-network aggregation for wireless sensor networks," *IEEE Systems Journal*, vol. 7, no. 1, pp. 13–25, 2013.

[14] Q. Wu, C. Zhang, and L. Gao, "Differential evolution algorithm based on hybrid crossover," *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, vol. 46, no. 5, pp. 78–105, 2018 (Chinese).