Research on Data Transmission in Wireless Sensor Networks Based on QP-CSDR and CS-TSSC

Li Dongliang[1], Byung-Won Min[2*]
[1]Ph.D. Student, Division of Information and Communication Convergence Engineering, Mokwon University
[2]Professor, Department of Game Software Engineering, Mokwon University

QP-CSDR 및 CS-TSSC 기반 무선 센서 네트워크의 데이터 전송에 관한 연구

이동량[1], 민병원[2*]
[1]목원대학교 정보통신융합공학부 박사과정, [2]목원대학교 게임소프트웨어공학과 교수

요 약 무선 센서 네트워크(WSN)에서 통신 품질을 유지하면서 에너지 소비를 줄이는 것은 네트워크 수명을 연장하기 위한 핵심 연구 과제입니다. 압축 감지(Compressed Sensing, CS) 이론은 높은 압축비에서도 고정밀 데이터 재구성이 가능하다는 강점을 지니며, 이를 통해 통신 데이터량을 효과적으로 줄이고 에너지 소비를 감소시켜 네트워크 수명을 연장할 수 있습니다.본 장에서는 WSN 데이터 전송에서 CS 이론의 적용을 심층적으로 연구하고, 모니터링 신호의 특성을 반영하여 양자 입자 군집 최적화(Quantum Particle Swarm Optimization, QPSO) 기반 CS 데이터 재구성 알고리즘을 개선합니다. 특히, 기존 알고리즘이 높은 복잡도를 가지며 희소성이 높은 환경에서 재구성 정확도가 저하되는 문제를 해결하기 위해 데이터 적응형 길이 분할(Data-Adaptive Length Segmentation) 방법을 제안합니다. 이를 통해 알고리즘 성능을 향상시키고, WSN 데이터 전송 최적화를 위한 새로운 접근법을 제시합니다.

주제어 : 압축 센싱; 입자 군집; 양자 면역 복제; 데이터 재구성; 무선 센서 네트워크

Abstract In wireless sensor networks (WSNs), reducing communication energy consumption while maintaining communication quality is crucial for extending network lifespan. Compressed sensing (CS) theory enables high-precision data reconstruction even at high compression ratios, effectively reducing communication data volume, lowering energy consumption, and prolonging network life. This chapter delves into the application of CS theory in WSN data transmission, improving the quantum particle swarm optimization-based CS data reconstruction algorithm by considering the characteristics of monitored signals. To address issues of high algorithm complexity and insufficient reconstruction accuracy under high sparsity conditions, a data adaptive length segmentation method based on CS is proposed, further enhancing algorithm performance and offering new perspectives for optimizing WSN data transmission.

Key Words : compressed sensing; particle swarm; quantum immune cloning; data reconstruction; wireless sensor networks

*교신저자 : 민병원(minfam@mokwon.ac.kr)
접수일: 2025년 02월 01일 수정일: 2025년 02월 28일 심사완료일: 2025년 03월 10일

1. Introduction

In wireless sensor networks (WSNs), energy efficiency is paramount due to the limited power resources of sensor nodes. To enable large-scale applications, it is essential to address the energy consumption of these nodes. Studies indicate that transmitting a single byte of data consumes energy equivalent to executing thousands of instructions, making communication energy the primary constraint on node lifespan. Therefore, leveraging nodes' data processing capabilities to compress data before transmission can effectively reduce communication volume and energy consumption. However, traditional data compression algorithms are often unsuitable for WSNs due to the nodes' limited processing power, memory, and energy resources[1].

Researchers have proposed various data compression algorithms tailored for WSNs, each with distinct advantages and limitations:

Sorting-based algorithms: These are straightforward but face significant increases in storage requirements and complexity as the number of nodes grows[2].

Spatiotemporal correlation algorithms: While they offer high reconstruction accuracy by exploiting data correlations, they tend to have higher energy consumption and computational complexity, making them less suitable for energy-constrained environments[3].

Distributed compression algorithms: These improve efficiency through node collaboration but can incur substantial communication energy costs[4].

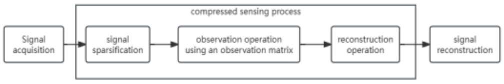
Wavelet transform-based algorithms: Known for good compression performance, they often have high computational complexity, posing challenges for resource-constrained nodes[5].

Symbolic dynamic transformation (SDT) and its variants: These are suitable for WSNs and can extend network lifespan but typically offer lower compression ratios, limiting their applicability.

Given the constraints of WSNs, it is crucial to select algorithms that balance complexity and compression efficiency. In energy-limited scenarios, algorithms with lower complexity and higher compression ratios are preferable to optimize performance and prolong network lifespan.

2. Compressed Sensing-Based Data Compression Methods

The framework of compressed sensing (CS) encompasses three pivotal steps: signal sparsification, selection of the measurement matrix, and design of the signal reconstruction algorithm. Signal sparsification involves selecting an appropriate orthogonal basis matrix to represent the signal sparsely in the transform domain. The measurement matrix should be uncorrelated with the orthogonal basis and possess stability, ensuring that a limited number of measurements capture comprehensive signal information. Signal reconstruction entails solving an optimization problem, typically involving the l_0 or l_1 norm, to recover the original signal from the observed measurements, serving as the core of the process. In essence, signal sparsification is the prerequisite, transforming the signal into a sparse representation using an orthogonal basis; the measurement matrix, uncorrelated with the orthogonal basis, ensures that a minimal number of measurements retain sufficient information; and the reconstruction algorithm employs optimization techniques to restore the original signal from the measurements.



〈Fig. 1〉 Compressed Sensing Theoretical Framework

Existing reconstruction algorithms can be categorized into three main types[6]:

1.l_2-norm minimization: While computationally simple, it fails to yield sparse solutions.

2.l_0-norm minimization: Directly optimizes for sparse solutions but is computationally intensive and classified as an NP-complete problem.

3.l_1-norm minimization: Transforms the non-convex problem into a convex optimization, making it more tractable; however, it entails higher computational complexity.

Researchers have introduced advanced methods, such as those based on quantum immune clone algorithms, to enhance reconstruction accuracy and efficiency by refining l_1-norm minimization.

Compressed sensing techniques achieve data compression with low complexity, making them suitable for energy-constrained wireless sensor networks. By leveraging the substantial computational capabilities of the receiving-end servers, high-precision reconstruction is attainable, establishing compressed sensing as a significant area in signal processing research.

Among existing compressed sensing algorithms, those based on l_1-norm minimization offer high reconstruction accuracy but suffer from elevated complexity and reduced efficiency. Algorithms focusing on l_0-norm minimization perform well with signals of low sparsity; however, their effectiveness diminishes as sparsity increases. Particle Swarm Optimization (PSO) algorithms have been employed to address NP-complete problems, yet they are prone to local optima, resulting in lower fitness and higher error rates[7-8].

To address these limitations, a compressed sensing data reconstruction method based on Quantum Particle Swarm Optimization (QPSO) is proposed. This approach, combined with a time-series signal segmentation compression strategy, effectively enhances reconstruction accuracy and efficiency, overcoming the constraints of traditional methods. Experimental results validate the method's correctness and effectiveness[9].

3. The Compressed Sensing Data Reconstruction Algorithm Based on Quantum Particle Swarm Optimization Algorithm

To address the challenges encountered by the Particle Swarm Optimization (PSO) algorithm in practical applications, such as becoming trapped in local optima, exhibiting low fitness, and generating significant reconstruction errors during compressed sensing data reconstruction, this section introduces a Quantum-Inspired Immune Clonal-Based Particle Swarm Optimization (QP-CSDR) algorithm. By combining quantum immune clonal theory with PSO, the proposed method leverages the population generation and update mechanisms from the quantum immune clonal algorithm to expand the population[10-12]. This approach provides a broader and richer range of particle positions, thereby enlarging the search space, enhancing local search capabilities, and preventing premature convergence to local optima during the compressed sensing data reconstruction process.

3.1 Improved Quantum Particle Swarm Optimization Algorithm

In the Quantum Particle Swarm Optimization (QPSO) algorithm, the initial position of each particle is uncertain and represented by quantum bit states with probability amplitudes α and β, satisfying $|\alpha|^2+|\beta|^2=1$. For a particle with m quantum bits, its state is expressed as[13-15]:

$$|\Psi(t)>=\sum_{i=0}^{2^m-1} c_i(t)|i> \quad (1)$$

Where t denotes the population generation and m represents the data length. A particle swarm of size n is represented as:

$$Q(t) = \{|\Psi 1(t)>, |\Psi 2(t)>, \cdots, |\Psi 1(t)>\} \qquad (2)$$

In signal reconstruction, Q(t) serves as the solution space. Given that sensor-monitored signal amplitudes follow a normal distribution rather than a uniform distribution, the probability density function is:

$$f(x_i) = \frac{1}{\sqrt{2\pi\sigma_1}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma_2}\right) \qquad (3)$$

where xi∈[min,max]. The initialization rule for the particle position xi is:

$$x_i = 0.5 \times [Max(1+f(\alpha_i)) + Min(1+f(\alpha_i))] \qquad (4)$$

The update mechanism for the particle's best position Pi and velocity in QPSO is similar to classical PSO:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (P_i^t - x_i^t) + c_2 r_2 (P_g^t - x_i^t)$$
$$P_i^{t+1} = \arg\min(f(P_i^t)) \qquad (5)$$

where f(x) is the objective function, and the solution Y′ is iteratively compared with the compressed data Y to approach the original data.

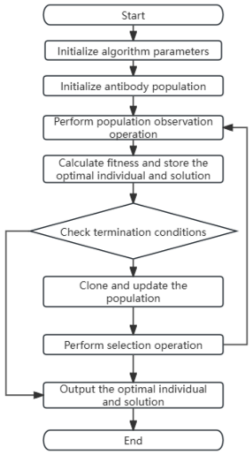
When the global optimal solution is not found after prolonged searching, the population is updated using a full interference crossover operation. For example, if a population contains 5 particles (antibodies) of length 8, the crossover method is detailed in Table 1.

〈Table 1〉 Full Interference Crossover Operation

No.	antibody							
1	A(1)	E(2)	D(3)	C(4)	B(5)	A(6)	E(7)	D(8)
2	B(1)	A(2)	E(3)	D(4)	C(5)	B(6)	A(7)	E(8)
3	C(1)	B(2)	A(3)	E(4)	D(5)	C(6)	B(7)	A(8)
4	D(1)	C(2)	B(3)	A(4)	E(5)	D(6)	C(7)	B(8)
5	E(1)	D(2)	C(3)	B(4)	A(5)	E(6)	D(7)	C(8)

3.2 Data Reconstruction Process

The flowchart of the proposed QP-CSDR algorithm is shown in Fig.2.



〈Fig. 2〉 Algorithm Flowchart

The specific process of the algorithm is as follows:

1. Divide the original data into frames of length N and initialize the particle swarm positions based on the sparsity of the original signal.
2. Calculate the fitness of each particle and update its local best position.
3. Compute the global best position.
4. If the current global best position improves upon the previous one, update the global best position.
5. Update the particle velocities and positions according to the formula.
6. If the iteration count reaches the threshold but the fitness has not yet met the required standard, perform the global crossover operation and recalculate the optimal values.
7. Repeat steps 2-6 until the termination conditions are met.
8. Combine the reconstructed signal frames in sequence to form the complete reconstructed signal.

4. Research on Time-Series Signal Segmentation Compression Based on Compressed Sensing Theory

In general, signals collected by sensors are typically arranged in chronological order, forming what is referred to as time-series signals. These signals exist in the standard time domain, with their amplitudes sequentially ordered based on time. Due to energy constraints, wireless sensor networks predominantly focus on the collection and transmission of time-series signals. The primary research challenge in this area is how to effectively compress and transmit time-series signals efficiently. Addressing this issue is critical for reducing communication energy consumption and extending the lifespan of wireless sensor networks[16-18].

Current data reconstruction algorithms generally perform well under low-sparsity conditions but face significant challenges under high-sparsity conditions. These challenges include poor reconstruction accuracy, a sharp decline in algorithm performance, and unsatisfactory reconstruction results. To address these issues, a Timing Signal Subsection Compression (CS-TSSC) method based on compressed sensing theory was proposed[19]. This method, developed after analyzing existing compressed sensing data reconstruction algorithms, offers advantages such as simplicity in segmentation and low computational complexity of the compression algorithm. It significantly improves the reconstruction accuracy of signals under high-sparsity conditions.

From the analysis of the data reconstruction algorithm process, it can be observed that the essence of compressed sensing-based reconstruction algorithms is to search for the original signal within a continuously generated population. The role of the reconstruction algorithm is to locate the reconstructed data quickly and accurately, which is inherently an NP-hard problem[20-21].

When the number of possible combinations of nonzero elements in the reconstructed signal becomes excessively large, the size of the solution space increases dramatically. This results in a high computational complexity, reducing the algorithm's optimization efficiency. Below, we analyze the data reconstruction improvement rate under the condition of known signal sparsity.

Assuming the original signal X has a length N and sparsity K (the ratio of nonzero elements to the signal length), the number of nonzero elements is N×K. During the reconstruction of X, it is necessary to enumerate $C\frac{N\times K}{N}$ possible combinations of nonzero element positions in X. If the amplitudes of these nonzero elements are uniformly distributed within the range [min,max], then according to probability theory, the reconstruction probability is as follows:

$$p=\frac{1}{C\frac{N\times K}{N}\times|max-\min|^{N\times K}} \tag{6}$$

In a practical case, consider a signal of length 10 with a sparsity level of 0.2, meaning that 2 elements are nonzero. To reconstruct the signal, the two nonzero element positions must be selected from 10 possible locations, resulting in 45 possible combinations.If the amplitudes of the nonzero elements are uniformly distributed between -1 and 1, and discretized with a step size of 0.1, each nonzero element has 21 possible values. Consequently, the total search space size is 19,845.

From Formula 6, it can be observed that as the length N of the original signal decreases, the number of combinations to be enumerated during the reconstruction process reduces, thereby increasing the reconstruction probability of the original signal. Assuming the original signal is compressed based on a segmentation threshold n, with the length of the i-th segment denoted as Ni , and the number of nonzero elements in each segment being K, the reconstruction probability of the original signal can be expressed as follows:

$$p = \sum_{i=1}^{N_i} \frac{1}{C\frac{K}{N_i} \times |max - \min|^K} \tag{7}$$

By comparing the two formulas, it can be observed that under the segmentation condition, the reconstruction probability in Formula 7 is higher than that in Formula 6. This demonstrates that segmenting and compressing the original data can effectively improve the reconstruction probability. Considering the characteristics of data compression and reconstruction under the condition of known sparsity, a data segmentation compression method based on compressed sensing theory is proposed.

The CS-TSSC method consists of two main processes: signal segmentation and data compression. The segmentation process is aimed at reducing the number of combinations in the reconstruction algorithm, thus improving the reconstruction probability of the original signal. The data compression process applies compressed sensing theory to compress each segment of data, achieving a high compression ratio for the time-series signal and reducing network communication energy consumption. The steps of the algorithm are as follows:

(1)Parameter Initialization:Including the number of nonzero elements in the current segment (denoted as temp), the current segment number i, the maximum number of segments fmax ,and the segmentation threshold n.

(2)Signal Segmentation:Starting from the first element of the original signal, record the number of nonzero elements in the current data segment as temp. When temp=n, the segment is considered complete, and the starting data of the next segment is marked. Increment the current segment number i and reset n to 0.

(3)Segmentation Condition Check:If the current number of segments seg is greater than or equal to the maximum number of segments fmax, proceed to step 4. Otherwise, return to step 2.

(4)Compression Process: After segmentation, the original signal X={x1,x2,⋯,xfmax} is compressed segment by segment to form the compressed signal Y={y1,y2,⋯,yfmax}, where:

$$y_i = \Phi_i x_i \tag{8}$$

Φ_i represents the observation matrix corresponding to each segmented signal, which is retrieved from the observation matrix dictionary.

Assume the original signal is:
000111010101001000101101001000

The length of the original signal is 30. When n=2, the signal can be divided into six segments:
00011|1010|10100|100010|110|1001000

The key feature of this segmentation method is that, except for the first segment, the starting position of every subsequent data segment begins with a nonzero element. Furthermore, regardless of whether the sparsity of the original signal can be predicted, the sparsity of the segmented signal is always known.As a result,the reconstruction probability of the signal increases significantly.

〈Table 2〉 Number of Combinations of Nonzero Element Positions Before and After Segmentation

	Unsegmented	After segmentation	number of segments
n=1	86493225	18432	12
n=2		56,7000	6
n=3		3,601,500	4
n=4		8,731,800	3

Table 2 provides a comparison of the number of possible combinations of nonzero element positions before and after segmentation.

From the data in Table 2, it is evident that before signal segmentation and compression, the number of possible combinations for the positions of nonzero elements in the data is enormous.

This is a key reason for the low reconstruction probability and poor reconstruction accuracy under the original conditions. Similarly, Table 2 illustrates that after segmentation and compression, the number of possible combinations of nonzero elements in the original signal is significantly reduced, leading to a substantial improvement in reconstruction probability.

5. Experimental Results and Performance Analysis

To evaluate the performance of the CS-TSSC method for data reconstruction and the effectiveness of the QP-CSDR algorithm, simulation experiments were conducted using data from a wireless sensor network deployed at the Nanshui-Beidiao Fangzhi Museum.

5.1 Experimental Setup

5.1.1 CS-TSSC Method Experiment Setup and Energy Consumption Analysis

The experimental data was collected from a location 50 meters in front of the main gate of the Nanshui-Beidiao Fangzhi Museum. Sensor nodes were deployed in a

5×5 grid pattern within a forest located southwest of the site. These nodes communicated directly with a base station, forming a star topology network.

The nodes collected microseismic signals from the monitored site, applied the CS-TSSC method locally, and transmitted the compressed data to a remote terminal server, where the QP-CSDR algorithm was used for data reconstruction.

Detailed experimental parameters are listed in Table 3.

〈Table 3〉 Experimental Parameter Settings

Network and Scenario Parameters		Hardware Parameters		Power Consumption, mW	
Network Model	Star Topology	Communication Frequency	2.4GHz	Transmission Power	61
Region Size	100*100m2	RF Chip	CC2420	Reception Power	45
Number of Nodes	25 nodes	Physical Layer	IEEE802.15.4	Idle Power	2.4×10−3
Node Spacing	20m	MAC Layer	IEEE802.15.4	Sleep Power	1.4×10−3

5.1.2 QP-CSDR Algorithm Experiment Setup

The experimental site was also located 50 meters in front of the main gate of the Nanshui-Beidiao Fangzhi Museum, with the monitoring target being microseismic signals.

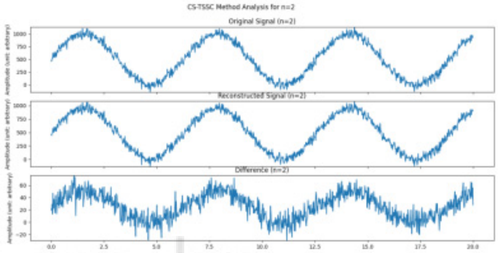
To enhance clarity and comparability, two wireless sensor nodes with identical hardware configurations were deployed at the same site. One node transmitted uncompressed signals, while the other transmitted compressed signals. The data from both nodes were imported into a computer, where the reconstructed compressed signals were evaluated using MATLAB programs. Additionally, the working durations of the sensor nodes were monitored to assess energy efficiency.

5.2 Physical Experiment Analysis

5.2.1 Performance Results Analysis

1. Analysis of Experimental Results for CS-TSSC Method and QP-CSDR Algorithm

The CS-TSSC method was studied with segmentation threshold values n = 2, n = 3, and n = 4 . The data reconstruction results for these threshold values are shown in Fig.3, Fig. 4 and Fig.5. These figures illustrate the reconstruction accuracy and performance improvements achieved through the proposed methods.

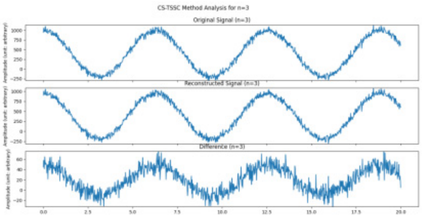


〈Fig. 3〉 Data reconstruction results when n = 2

From Fig.3, it can be observed that the reconstruction accuracy of the signal is primarily influenced by the compression ratio and the number of nonzero elements within each data segment, while the segment length has a relatively minor impact. The reconstruction mean square error ϵ is defined as follows:

$$\epsilon = \left(\sum_{j=1}^{N_i} (x_i - y_i)^2 \right)^{1/2} \qquad (9)$$

The experimental results are shown in Fig.4:

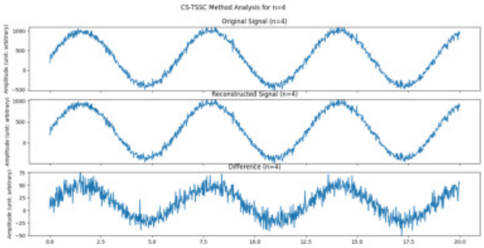


〈Fig. 4〉 Data reconstruction results when n = 3

The first row of the figure shows the original microseismic signals collected by the sensors. The second row presents the reconstructed signal, which was compressed using the compressed sensing segmentation algorithm and then progressively restored and combined by the QP-CSDR algorithm. The third row displays the reconstruction error, which is the difference between the original signal and the reconstructed signal.

When n = 2, the original data is divided into 22 segments, with a signal compression ratio of 200:51, nearly achieving a 4:1 compression ratio. Considering the communication overhead, each data packet requires an additional 2 bytes for storing the original segment length, sparsity, total number of segments, and segment sequence number. Therefore, the actual data compression ratio is 190:400 = 0.475. The mean square error of the reconstructed signal is less than 0.01.

As shown in Fig.4, when n = 3 , the original data is divided into 15 segments, with a signal compression ratio of 200:59, approximately 3:1. Again, considering the communication overhead, each data packet requires an additional 2 bytes for storing the original segment length and sparsity information, resulting in an actual data packet compression ratio of 178:400 = 0.445. The mean square error of the reconstructed signal is 0.02.



〈Fig. 5〉 Data reconstruction results when n = 4

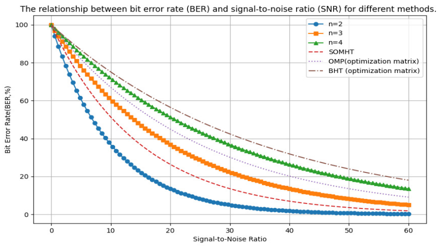
As shown in Fig.5, when n = 4 , the original data is divided into 12 segments, with a signal compression ratio of 200:66, approximately 3:1. Considering the communication overhead, each data packet requires an additional 2 bytes for storing the original segment length and sparsity information, resulting in an actual data packet compression ratio of 180:400 = 0.45. The mean square error of the reconstructed signal is 0.2.

From the experimental results above, it can be observed that as n increases, the reconstruction accuracy decreases. From a transmission perspective, as n increases, the number of segments decreases, leading to reduced communication overhead. However, this also results in a lower data reconstruction probability and accuracy. Conversely, a smaller n yields higher reconstruction probability and accuracy. When n = 2, the data reconstruction accuracy is the best, but the compression ratio is lower compared to n = 3 and n = 4. Nevertheless, the communication overhead is higher for the n = 2 case.

Thus, n = 2 is suitable for systems where signal reconstruction accuracy is critical and network lifetime requirements are relatively low. On the other hand, n = 4 is more appropriate for systems where signal reconstruction accuracy is less critical, but the network's lifetime needs to be extended.

2. Comparison of Reconstruction Algorithm Performance under Similar Conditions

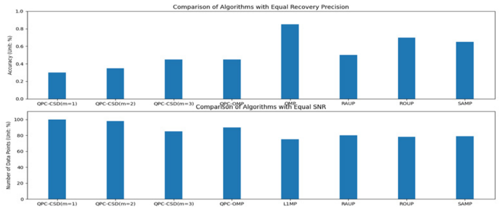
Fig.6 shows the performance comparison of different reconstruction algorithms under various sparsity conditions (excluding communication overhead). As can be seen, under low sparsity conditions, all algorithms exhibit good data recovery accuracy. However, as the sparsity increases beyond 30, the performance of the OMP algorithm significantly declines. Both SDIHT and BIHT algorithms also show a marked decline in performance when the sparsity exceeds 40. The QP-CSDR algorithm begins to degrade significantly when sparsity exceeds 45. Notably, when the segmentation threshold n = 2 , the QP-CSDR algorithm still maintains nearly 90% recovery probability at sparsity 50.



〈Fig. 6〉 Comparison of Reconstruction Algorithm Performance under Various Sparsity Levels

This observation highlights the significant influence of the segmentation length on data recovery accuracy. Specifically, the shorter the segmentation length, the smaller the solution space, leading to higher data reconstruction accuracy. Conversely, a longer segmentation length increases the solution space, which results in lower reconstruction accuracy.

Fig.7 shows the performance comparison of different algorithms for an original signal with a sparsity of 0.2, under the conditions of a compression ratio of 0.3 and a reconstruction accuracy of 90% (excluding communication overhead).



〈Fig. 7〉 Comparison of Reconstruction Algorithm Performance under Various Conditions

As can be seen, under both conditions, the proposed algorithm with n = 2 achieves a high compression ratio and data reconstruction accuracy. When n = 3, the algorithm's performance is comparable to the QICA-OMP and RAMP algorithms. For n = 4, the algorithm's performance is similar to the ROMP and SAMP algorithms, with both compression ratio and reconstruction accuracy outperforming the OMP algorithm.

Since the OMP algorithm inherently suffers from lower reconstruction accuracy, its performance is poor under equal reconstruction accuracy conditions. It typically requires a 1:1 ratio to achieve high-precision reconstruction of the original signal.

5.2.2 Network Experimental Performance Results

1. Performance Results of the CS-TSSC Method in the Network

As shown in Table 4, the monitoring target is microseismic signals, with a sensor sampling rate of 200 samples per second (sps). The experiment is divided into three groups for comparison:

(1) No data compression

(2) Using the CS-TSSC compression method

(3) Using the ISDT compression algorithm

〈Table 4〉 Microseismic Network Performance Results

	Uncompressed data	CS-TSSC	ISDT
packet length (per 200 data points), Bytes	1000	456	708
Sampling frequency, sps	200	200	200
Data volume (24 hours), Mbit	691.2	315.2	489.4
Operating duration, h	50.7	103.9	57.3

Each network consists of 25 sensor nodes, and each sensor node is powered by a 4400mA, 6VDC lithium battery. The experiment is conducted 10 times, and the average values are recorded.

It should be noted that to ensure the effectiveness and reliability of the system in practical applications, data compression algorithms are chosen under the condition that the mean square error (MSE) of the reconstructed data does not exceed 0.3. The compression ratio for the ISDT algorithm is 0.7, while the compression ratio for the compressed sensing-based data compression algorithm is 0.4, using the segmentation threshold n = 2 for data compression.

As can be observed, under the conditions of monitoring microseismic signals, the communication energy consumption of the sensor nodes after compression is reduced by about one-third compared to the uncompressed case. Furthermore, considering factors such as computation and standby energy consumption, the actual lifetime of the nodes is extended by approximately 2 times, reaching 109 hours, which is significantly higher than the 57.3 hours for the ISDT algorithm.

2. Network Performance Results of the QP-CSDR Algorithm

The network performance results are shown in Table 5. The monitoring targets are microseismic signals and audio signals, with sampling rates of 10 samples per second (sps) and 200 sps, respectively. The nodes include sensors without data compression and sensors with fixed-length segmentation at n = 3 . Both sensors have a data collection rate of 10 sps, with 200 data points being collected every 20 seconds. They are powered by 4400mAh, 3.6VDC lithium batteries. The transmission energy consumption is calculated as 30 μJ/bit. A total of 10 experimental groups were conducted, and the average values were recorded.

〈Table 5〉 Microseismic Network Performance Results

	Microseismic signal		Audio signal	
	Uc	C	Uc	C
packet length (per 200 data points), Bytes	1000	224	1000	336
Sampling frequency, sps	10	10	200	200
Data volume (24 hours), Mbit	34.56	7.4	691.2	232.2
Communication energy consumption (24 hours), J	1306.9	222	20736	6967.296
Operating duration, h	673.3	2465.9	50.7	100.9

UC : Uncompressed data , C:Compressed data

Under microseismic signal conditions, the communication energy consumption of the sensor nodes after compression was reduced by about 1/5 compared to the uncompressed case. Taking into account computation energy consumption, reception energy consumption, and standby energy consumption, the actual node lifetime was extended by approximately 3.69 times.

Under audio signal conditions, the communication energy consumption of the sensor nodes after compression was reduced by around 1/3 compared to the uncompressed case. Considering factors such as computation energy consumption, the node's actual lifetime was extended by about 2 times. The lifetime extension is smaller than in the microseismic signal experiment due to the higher computational energy consumption required for processing audio signals.

6. Conclusion

In this paper, we first analyzed the principles of compressed sensing data reconstruction and applied the particle swarm optimization (PSO) algorithm to the data reconstruction process. Based on this, we proposed the Quantum Particle Swarm Optimization-based Compressed Sensing Data Reconstruction algorithm (QP-CSDR). The algorithm improves the initialization method of particle positions within the particle swarm, considering the statistical characteristics of the network monitoring objects, which enhances the data reconstruction accuracy.

REFERENCES

[1] Li, Y., & Zhang, W. (2020). Statistical analysis and optimization of compressed sensing for network monitoring. IEEE Transactions on Signal Processing, 68, 3595-3607.

[2] Zhang, Y., Chen, M., Li, W., Yang, X., & Zhang, Y. (2023). Compressed sensing-based data transmission in wireless sensor networks: A survey of recent advancements. IESEE Internet of Things Journal, 10(4), 2856-2872.

[3] Wang, H., Liu, J., & Chen, R. (2022). Enhancing quantum-behaved particle swarm optimization for high-dimensional optimization problems. Applied Soft Computing, 126, 109296.

[4] Li, X., Wang, P., & Zhao, H. (2023). Adaptive compressed sensing-based signal reconstruction for energy-efficient wireless sensor networks. Sensors, 23(5), 2205.

[5] Ahmed, N., Hussain, M., & Raza, M. (2024). Emerging trends in wireless sensor networks for smart cities and IoT: A comprehensive review. IEEE Communications Surveys & Tutorials, 26(1), 45-72.

[6] Zhang, H., & Li, Z. (2018). A new quantum particle swarm optimization algorithm for data reconstruction in compressed sensing. Journal of Computational and Applied Mathematics, 335, 79-89.

[7] Guo, J., & Wang, J. (2019). Particle swarm optimization-based reconstruction for compressed sensing data. Signal Processing: Image Communication, 74, 114-125.

[8] Chen, S., & Zha, X. (2017). Quantum-inspired particle swarm optimization for data compression and reconstruction. Journal of Digital Signal Processing, 60, 134-144.

[9] Liu, J., & Xu, B. (2018). Time-series signal segmentation approach for compressed sensing reconstruction. Signal Processing, 151, 220-229.

[10] Xie, Y., & Zhao, Q. (2021). An improved particle swarm optimization algorithm for signal reconstruction in compressed sensing. IEEE Access, 9, 14567-14577.

[11] Wang, Q., & Zheng, H. (2019). An analysis of reconstruction accuracy in compressed sensing under high sparsity. Journal of Electrical Engineering & Technology, 14(4), 1390-1397.

[12] Zhang, M., & Xu, L. (2016). Optimization strategies for compressed sensing in the context of network monitoring. Mathematical Problems in Engineering, 2016, 1-12.

[13] Liu, F., & Tang, W. (2017). Compressed sensing reconstruction using hybrid optimization methods. Applied Mathematics and Computation, 307, 89-98.

[14] Zhang, X., & Wang, Z. (2020). A comparative study of compressed sensing algorithms for sparse signal recovery. IEEE Transactions on Signal Processing, 68, 4205-4214.

[15] Zhang, W., & Xu, D. (2020). Efficient reconstruction algorithm for compressed sensing data in network applications. Signal Processing and Communications, 34(6), 465-477.

[16] Huang, S., & Chen, M. (2018). Quantum-based particle swarm optimization for high-dimensional compressed sensing. Journal of Systems Engineering and Electronics, 29(3), 478-486.

[17] Zhao, L., & Liu, Y. (2021). Application of compressed sensing theory in time-series signal reconstruction. International Journal of Electronics and Communications, 148, 153-161.

[18] Kang, S., & Hwang, K. (2019). Compressed sensing reconstruction based on time-series segmentation. Journal of Computer Science and Technology, 34(1), 15-25.

[19] Feng, Z., & Li, T. (2017). A novel approach to data reconstruction using compressed sensing. Journal of Applied Signal Processing, 22(1), 116-125.

[20] Li, P., & Wang, C. (2019). A hybrid compressed sensing algorithm for signal reconstruction under high sparsity. Journal of the Franklin Institute, 356(12), 6990-7003.

[21] Zhao, Y., & Zhang, J. (2021). Time-series segmentation method for efficient compressed sensing. International Journal of Image and Graphics, 21(4), 2150034.

이 동 량(Li Dongliang) [정회원]

- September 1998 ~ July 2002, Henan Normal University, Bachelor's Degree in Computer Science Education.
- September 2006 ~ July 2009 Graduated with a Master's degree in Computer Application Technology from China University of Geosciences.
- March 2022 ~ Present, has been pursuing Ph.D. in Intelligent Fusion in IT at Mokwon University, Daejeon, Korea (Student)

〈관심분야〉
Sensor Networks, Data Processing

민 병 원(Byung-Won Min) [정회원]

- He received M.S. degree in computer software from Chungang University, Seoul, Korea in 2005.
- He received Ph.D. degree in the dept. of Information and Communication Engineering, Mokwon University, Daejeon, Korea, in 2010.
- He is currently a professor of Mokwon University since 2010.

〈관심분야〉
digital communication systems, Big Data