BIG DATA DYNAMIC COMPRESSIVE SENSING SYSTEM ARCHITECTURE AND OPTIMIZATION ALGORITHM FOR INTERNET OF THINGS

JIAN-WU XUE* AND XIAO-KUN XU
School of Management
Northwestern Polytechnical University
Xi’an 710072, China

FENG ZHANG
School of Information Engineering
Yulin University
Yulin 719000, China

ABSTRACT. In order to reduce the amount of data collected in the Internet of things, to improve the processing speed of big data. To reduce the collected data from Internet of Things by compressed sensing sampling method is proposed. To overcome high computational complexity of compressed sensing algorithms, the search terms of the gradient projection sparse reconstruction algorithm (GPSR-BB) are improved by using multi-objective optimization particle swarm optimization algorithm; it can effectively improve the reconstruction accuracy of the algorithm. Application results show that the proposed multi-objective particle swarm optimization-Genetic algorithm (MOPSOGA) is than traditional GPSR-BB algorithm iterations decreased 51.6%. The success rate of reconstruction is higher than that of the traditional algorithm of 0.15; it’s with a better reconstruction performance.

1. Introduction. The Internet of things (IoT) is a huge network which is formed by the combination of various information sensing and sensing devices. Therefore, it can be “connected” to the extension and expansion of any goods and items. In the Internet of things, the energy efficient way is a very important problem in the information reliability [4]. It is usually configured with a large number of data acquisition devices, these devices all the time to produce a massive, large heterogeneous data. The existence of mass data acquisition has a serious impact on the performance of the Internet of things, study on the sampling method of dynamic sampling and compression has become an important prerequisite for the application of the Internet of things. Compressed Sensing (CS) [10] is a new sampling method to data acquisition and compression at the same time; it can reduce the amount of the data collected by discarding the redundant information at the same time.

However, in the wireless sensor networks field, due to the uncertainty and diversity of the scene, a lot of sensory information may not meet the high compressed
sensing sparse requirements. So, if you want to make good use of compressed sensing technology advantage, we need to analyze the perceived characteristic under different environmental signals, to find a suitable compressible space, to design efficient compression method, including the corresponding measurement matrix and targeted reconstruction algorithm.

Internet of Things are widely used, such as environmental monitoring, mine safety, forest fire monitoring, traffic monitoring, transportation monitoring, smart home\cite{1}. Large-scale and ultra large scale Internet of Things is a lot of Intelligence of the hundreds of millions of objects connected in dynamic network, the acquisition and interaction of these objects will produce huge amounts of data. For example: in a by IoT support large logistics systems, if go to track the location and status of 2000 million packages or commodity information, the daily production of up to 2GB, the amount of data for one year for 720GB; And a large intelligent transportation or ecological monitoring and other real-time monitoring system, the amount of data that need to be processed every day can reach TB, the annual data will reach PB level. The large amount of information from a large number of intelligent perception system of the flow of information, it’s need to study the new method of data acquisition for dynamic compression processing. Compressed sensing method is a new sampling method, which can reduce the amount of data in the data processing of the Internet of things.

In this paper, the method of data acquisition and processing in the Internet of things is introduced to reduce the amount of data and reduce the burden of communication, but at the same time, it will bring an increase in the amount of computation, which is very unfavorable for the Internet of things. In order to speed up the compressed sensing algorithm, cloud computing technology\cite{7} is a good choice in the environment of things.

2. Compressive sensing theory. Compressive sensing is proposed by Donoho D. Donoho, E. Candes and T. Tao. When the original signal is sparse, the method can be used to reconstruct the original signal accurately. The operation process of the compressed sensing theory is can be shown in Figure 1. Compressed sensing theory that as long as the signal is compressible or on a sparse transform base is sparse, with less information can indicate signal, the signal is then reconstructed by solving optimization problems. In general, according to the sparse representation theory, \( x_{n \times 1} \) can be assumed that the signal has a sparse representation \( \Phi \):

\[
x = \Psi \Phi. \tag{1}
\]

Where, \( \Psi_{n \times n} \) is a sparse transform matrix of \( x \). Then use the observation matrix \( O_{m \times n} \) compression coefficient \( \Phi \):

\[
y = O\Phi = O\Psi^T x. \tag{2}
\]

Matrix \( O_{m \times n} \) must be independent of the \( \Psi \) linear; The compression ratio of the original signal \( x \) and \( y \) between the observed data is \( n : m \).

![Figure 1. The operation process of the compressed sensing theory](image-url)
Reconstructed signal that is solving equation (2), Donoho, Chen and Saunders research that: Under the premise of restricted isometry property (RIP)[8], we can reconstruct the signal optimization problem by solving the 1-norm, that is:

$$\min \|\Phi\|_1 \quad s.t. \quad y = O\Phi = O\Psi^T s.$$  \hspace{1cm} (3)

Formula (3) is a convex optimization problem, this paper studies the use of optimization method to solve the problem.

3. Dynamic compressive sensing. The introduction of compressed sensing algorithm can significantly reduce the amount of data on the network, but the disadvantage is also obvious compressed sensing. The computational complexity of the compressed sensing algorithm is very high, especially the reconstruction algorithm[14]. Compressed sensing method using standard orthogonal basis as the sparse transform matrix, and the standard orthogonal basis does not have the adaptability, not according to the adaptive sampling signal changed. These are the application of compressed sensing methods need to be improved, there are a lot of researches focus on this, but most of them focus on the improvement of the reconstruction algorithm[5], and we propose and implement the parallel compression algorithm performance improvement is also very obvious.

In this section, we will introduce the parallel processing of compressed sensing method in order to reduce the amount of data collected, and a new sampling method is proposed for the introduction of “dynamic compression” in the Internet of things.

3.1. Parallel compressed sensing architecture. The introduction of compressed sensing method in mass data processing in the Internet of things will bring higher computational complexity[12], but the compression method involves a lot of matrix operations, such as signal sparse representation of data observation and signal reconstruction, etc., it’s allows parallel processing can significantly improve the algorithm execution speed. We propose and implement a parallel compressed sensing algorithm as shown in figure 2. Compared with the original compressed

![Parallel Compressed Sensing Architecture](image)

**Figure 2.** The parallel compressed sensing algorithm

sensing method, we proposed the parallel compression sensing algorithm increases the selection of the optimized structure and sparse transformation matrix, a higher degree of computational complexity signal reconstruction partially realized parallel processing. In the next section, we will introduce the parallel compression method in Figure 2.
3.2. **Parallel compressed sensing and its implementation.** Parallel processing is a computing model of a variety of instructions can be executed at the same time [11]. It can take advantage of a variety of computing resources to solve the problem to be solved. The premise of parallel processing is that a large scale problem can be solved by solving a number of small parts; each part can be executed concurrently without mutual influence. The main form of parallel processing is multi-core processing, the general multi-core parallel computing programming has two kinds of CPU multi-core programming and GPU multi nuclear program, And the GPU and CPU hybrid multi-core parallel processing has emerged and has caused the attention of everyone[2]. Cloud computing technology can provide users with computing resources through the network, to provide on-demand services, in order to achieve the desired effect of the user’s satisfaction [15]. In the cloud acceleration scheme, the collected data is usually in the form of compression. When the user requests the reconstructed signal, the computation complexity is not high, the general code will be executed locally, but the computation complexity of the reconstruction algorithm will be automatically migrated to the cloud.

It is also involved in a large number of matrix operations parallel compression sensing algorithm, Such as dynamic structure and selection of the signal of the observation and optimization of reconstruction, etc., These are the basis of improving the performance of parallel algorithms; In the compressed sensing algorithm, the optimal reconfiguration is the most part of the calculation, its high computational complexity and many things are contradictory to the real-time requirements of data processing[6], This makes it necessary to deal with the parallel processing of the compressed sensing algorithm. At present, we have implemented a number of compressed sensing algorithms for multi-core parallel processing, Figure 3 and Figure 4 is a comparison and non-parallel processing parallel computation time consuming when the image is reconstructed.

![Figure 3. Comparison between single core and multi-core implementation of reconfiguration time](image)

Figure 3 shows the comparison between the parallel and non parallel execution time of the 15GB image data processed by the OMP reconstruction algorithm [9]. It can be seen from the comparison of the signal reconstruction algorithm of the dual
core with the implementation of the dual core can save about 40% of the program running time, and the use of quad core running time can save 50%-60%. Figure 4 shows the comparison of the parallel and non-parallel execution time of the different sizes of data reconstruction. From the figure, in the processing of data is very small, the performance improvement of the parallel optimization is limited.

Through the above analysis, it can be known that the parallel processing of the compressed sensing algorithm can significantly reduce the running time of the algorithm and improve the real-time performance of data processing, but very small amount of data processing with parallel processing more harm than good, this is because of the realization of the parallel should spend some computing and storage resources.

3.3. The optimization algorithm of compressed sensing in parallel. In the compressed sensing theory, the signal reconstruction is realized by solving numerical optimization problems. According to the method used for reconstruction optimization is different and has a different perception algorithm. For the multi signal source optimization problems, many domestic and foreign scholars have carried out a lot of research, and put forward a variety of processing methods. In practice, there are a lot of multi-objective optimization problems, how to solve these multi-objective optimization problems, how to solve these multi-objective optimization problem is very important.

Multiobjective optimization problem (MOP) is for the conflict between different performance objectives put forward looking to meet constraints and all the objective function of a set of decision variables and corresponding each objective function value set (Pareto optimal).

The multi-objective optimization process is described as: Set the MOP contains a set of \( n \) dimensional vector optimization objective function \( f(x) = (f_1(x), \ldots, f_n(x)) \), where \( f_i(x)(i = 1, 2, \ldots, n) \) is a scalar function, \( f(x) \in Y \subset R^n \); a \( m \) dimensional decision vector set \( x = (x_1, \ldots, x_m), x \in X \subset R^m \); a \( p \) dimensional constraint function vector set \( g(x) = (g_1(x), \ldots, g_p(x)) \), where \( g_i(x)(i = 1, 2, \ldots, p) \) is a scalar function. Function \( f : X \rightarrow Y \) mapped the design vector \( x = (x_1, x_2, \ldots, x_m) \) to the target vector \( Y \) of the target function space \( y = (y_1, y_2, \ldots, y_n), y \in Y, y_i = f_i(x) \).
The MOP mathematical model described as the following:

\[
\begin{aligned}
\text{Maximize} & \quad y = (f_1(x), \ldots, f_n(x))^T \\
\text{s.t.} & \quad g_i(x) \leq 0, \quad i = 1, 2, \ldots, p \\
& \quad h_j(x) = 0, \quad i = 1, 2, \ldots, q
\end{aligned}
\]  

\begin{align}
\text{(4)}
\end{align}

Where, \( g_i(x) \leq 0 \) is the inequality constraint, \( x = (x_1, \ldots, x_m) \in X, X \subseteq \mathbb{R}^m \), \( y = (y_1, y_2, \ldots, y_m) \in Y, Y \subseteq \mathbb{R}^n \). For a multi-objective optimization problem to solve optimization problems often require multiple targets, there is generally no unique global optimal solution, only a Pareto, that is so-called Pareto optimal set.\[3\].

Pareto optimal solution is an important concept in multi-objective optimization. Based on the above formula, some important definitions are given.

**Definition 3.1.** Pareto dominance: Set \( X_A \) is a feasible solution for vector \( a = (a_1, a_2, \ldots, a_k) \), set \( X_B \) is a feasible solution for vector \( b = (b_1, b_2, \ldots, b_k) \), if \( X_A \) is better than \( X_B \), that \( X_A \) is Pareto dominant, if and only if \( \forall i = 1, 2, \ldots, k, f_i(X_A) \leq f_i(X_B) \land \exists j, j = 1, 2, \ldots, k, f_j(X_A) < f_j(X_B) \). Written \( X_A < X_B \), called \( X_A \) dominate \( X_B \), if the \( X_A \) solution is not dominated by other solution, then \( X_A \) is called non-dominated solutions, also known as Pareto solution.

**Definition 3.2.** Pareto optimal solution: For a given multi-objective \( f(x) \), Pareto optimal solution (or non-dominated solutions) is defined as: Let \( Z_f \) is a feasible collection of Multiobjective problems, \( Z^* \) is one of the solutions, \( Z^* \in Z_f \), if \( \neg \exists x \in Z_f : x > Z^* \), then \( Z^* \) is Pareto optimal solution.

**Definition 3.3.** Pareto optimal solution set: For a given multi-objective \( f(x) \), Pareto optimal solution set \( Z_f \) is defined as:

\[
Z_f = \left\{ X \in \mathbb{R}^n \mid \exists X^* \in \mathbb{R}^n, X \neq X^* \text{ make } f(X) < F(X^*) \right\}
\]

**Definition 3.4.** Pareto front end: All the Pareto non-dominated solutions called Pareto front end of the set, denoted as:

\[
P_f = \{ F(x) = (f_1(x)), (f_2(x)), \ldots, (f_n(x)) | x \in Z_f \}.
\]

All located in the Pareto front in all solutions both from the Pareto front solution control. Therefore, the non dominated solutions than other solution with minimal goal conflict, can provide decision makers a better choice space [13]. At the same time, some non dominated solution can weaken at least one other objective function while improving any objective function.

For a multi-objective optimization problem, scholars have done a lot of research; we proposed a variety of treatment methods. There are many on multi-objective optimization problem in practice, how to solve these multi-objective optimization problems is very important.

American electrical engineer Eberhart and social psychologist Kennedy in 1995 by the artificial life research results of illumination of particle swarm optimization (PSO) algorithm for simulating bird group foraging in the process of migration and cluster behaviour. The algorithm can find the global optimal solution of the problem, and has higher computational efficiency.

In the PSO system, each alternative solution is called a particle, a plurality of particles coexistence, cooperation, optimization, each particle based on its own experience optimal solutions to better position flight search in the problem space. The particle flight is shown in Figure 3, Particle swarm optimization process is shown in Figure 6.
For the characteristics of GA and PSO, in this paper, crossover and mutation operator and population segmentation strategy are introduced into PSO algorithm, two kinds of algorithms are mixed to solve the multi-objective optimization problem, a multi-objective particle swarm optimization-genetic algorithm (MOPSOGA) is proposed.

MOPSOGA algorithm flow is shown in Figure 7. MOPSOGA algorithm description code is as follows:

(1) Initialize the particle swarm $p_{list}$, the population size is n, the population is divided into two subgroups, called the nonbranch subset ($NSet$) and the branch subset ($PSet$), the size of the subgroups of $NSet$ and $PSet$ are $n_1, n_2$, and satisfy $n_1 + n_2 = n$. Clearly $\forall x_i \in PSet, \exists x_j \in NSet$, the $x_i$ invariances of $x_j$. Then, initialization of the cross probability $\rho_c$, variation probability $\rho_m$, maximum evolutionary algebra $K$, $c_1, c_2, r_1, r_2, w$.

for(i=1;i<N;i++)
Random initial position;
Particle i velocity;
Initialized to particle i personal best value;
Initialization of global extreme particle i;
end for

(2) Calculate the $fitness(x)$ function value of the population;

(3) Speed and position of each particle $P$ particles inside were updated:

Speed update: $\quad v[i] = \omega \times v[i] + c_1 \times \text{rand}(pbest[i] - p_{list}[i]) + c_2 \times \text{rand}(gbest[i] - p_{list}[i])$

Position update: $\quad p_{list}[i] = p_{list}[i] + v[i]$

(4) (Cross), from $p_{list}$ press roulette wheel selection operator selected $n_2$ individuals, with probability $\rho_c$ pair wise crossover to give the population $p_{list1}$.

(5) (Variation), press again from $p_{list1}$ roulette wheel selection operator $n_2$ individuals elected to mutation probability $\rho_m$ in turn given to individuals mutation operator, a new population $p_{list2}$.

(6) (Select), from the population $p_{list1} \cup p_{list2}$, the n individual is selected by the elite selection operator, and the next generation population $p_{list}$ is composed, and the global optimal particle gbest is updated.

(7) Calculate $NSet$ the population of non-dominated particles into a non-dominated solution, for each subset of each particle $NSet$ and $PSet$ subset particles one by one to compare and remember $PSet$ subset of particles is $x_1, x_2, ..., x_{n2}$, the particles of the subset is $x_1, x_2, ..., x_{n1}$.
The algorithm description code is as follows:

```c
int tag = 0;
for (i = 1; i < n1; i++)
    for (j = 1; j < n2; j++)
        if (p_list[i] < p_list[j])
            temp = p_list[i]
            p_list[i] = p_list[j]
```

**Figure 6.** Particle swarm optimization process
\[ p_{list}[j] = \text{temp} \]
\[ \text{tag} = 1; \]
\[ \text{end if} \]
\[ \text{end for} \]
\[ \text{if}(\text{tag} == 1) \]
\[ p_{list}[i] \text{ and all } p_{list}[j] \text{ after comparison, if there is no } j, \text{ making it established, then } p_{list}[i] \text{ may also be a non dominated solution, so } p_{list}[i] \text{ also } NSet \text{ subset.} \]
\[ \text{end for} \]

In order to verify with cross on improved particle swarm optimization effect algorithm optimization factors, the following tests by two classic function of the algorithm is tested while MOPSOGA and standard PSO were compared. Multi-objective optimization problem to be described as follows:

\begin{align*}
\text{Minimize } & f_1(x_1, x_2) = x_1^4 - 5x_1^2 + x_1x_2 + x_2^4 - x_1^2x_2^2 \\
\text{Minimize } & f_2(x_1, x_2) = x_2^4 - x_1^2x_2^2 + x_1^4 + x_1x_2 \\
\text{subject to } & \begin{cases} 
-5 \leq x_1 \leq 5 \\
-5 \leq x_2 \leq 5 
\end{cases}
\end{align*}

(6)

Set population N=50,dimension D=6,weight \( \omega = 0.9 \),crossover probability \( \rho_c = 0.7 \), mutation probability \( \rho_m = 0.01 \), maximum evolution generation K =3000, \( c_1 = c_2 = 2 \).

Experimental environment: Intel Core(TM)i5-3337U 1.8GHz,8GB Memory,Windows8.1 Operating system. The results of the multi-objective test questions shown in Figure8.

As can be seen from Figure 8, MOPSOGA approach can effectively test the Pareto optimal front, and the solution is more evenly spread. MOPSOGA arithmetic average distance obtained Pareto front end and the real Pareto front of the evolution is much better than the classic PSO algorithm results obtained. And, distribution MOPSOGA optimization algorithm than non-dominated solutions are also distributed PSO algorithm get a good non-dominated solutions.

3.4. Verification of parallel compressed sensing algorithm. In the compressed sensing algorithm, when the signal is sparse in a certain domain, a high dimensional signal can be projected onto a low dimensional space by using an observation matrix which is not related to the transformation matrix, then, the original signal is reconstructed from the low dimensional data by solving an optimization problem. Based on the sparse representation of signals, the objective function is to reconstruct the compressed sensing convex optimization problem:

\[ f(\omega) = \min_{\omega} \| \omega \|_1 \]
\[ \text{s.t. } y = \Phi x = \Phi \Psi \omega . \]

(7)

In the observation matrix \( \Phi \) meet the RIP, GPSR-BB reconstruction algorithm can achieve the probability of the reconstructed signal. Specific steps are as follows:

(1) Initialization, \( \omega^0, z^0, \omega_{\min}, \omega_{\max}, \omega^0 \in [\omega_{\min}, \omega_{\max}] \), \( \lambda^0 = 1 \) , set \( n = 0 \).
(2) Calculate the search direction \( d^n = \phi^n - z^n \).
(3) line search to determine step \( \lambda^n \), get a new point \( z^{n+1} = z^n + \lambda^n d^n \).
(4) Update \( \omega^n \), calculate \( \gamma^n = (d^n)^T B d^n \), if \( \gamma^n \leq 0 \), then \( \omega^{n+1} = \omega_{\min} \), else \( \omega^{n+1} = \max\{\omega_{\min}, \omega_{\max}, \min\{d^n / \gamma^n \}\} \).
(5) termination of the test, to meet the termination conditions \( \| z^{n+1} - (z^{n+1} - \omega^{n+1}) \| \leq \xi \), algorithm termination, output \( z^{n+1} \).
Figure 7. MOPSOGA algorithm flowchart
In this section, we give the experimental results of the comparison of the performance of the GPSR-BB algorithm and the improved MOPSOGA algorithm. If the reconstructed signal and the original signal are less than $10^{-4}$, the signal can be reconstructed successfully. Using the length of $N=5120$, $K=310$ sparse signal reconstruction, the observations $M$ value is 2048, Observation matrix is the Gauss random matrix of $M \times N$. The parameters in the algorithm are $\omega_{n+1} = 10^{-25}, \omega_{\text{max}} = 10^{25}, \omega^0 = 15, \xi = 10^{-4}$. The MOPSOGA algorithm optimization process population $N=50$, Dimension $D=4$, weight $\omega=0.8$, crossover probability $\rho_c=0.7$, mutation probability $\rho_m=0.01$, the maximum evolution generation $K=20$, $c_1 = c_2 = 0.8$. Use MOPSOGA algorithm for signal reconstruction algorithm operating results is shown in Figure 9.

It can be seen from Figure 9 that the signal is completely reconstructed. The simulation results show that the reconstruction error of the improved GPSR-BB algorithm is $3.216 \times 10^{-3}$, that not improved algorithm reconstruction error is $0.00526$, so the improved algorithm is more accurate and more accurate.

On this basis, we will achieve the above experimental results of 2000 times the average number of iterations of the reconstruction error and the running time is shown in Table 1.

| Algorithm  | Iteration number | Running time /s | Reconstruction error | Objective function value |
|------------|------------------|-----------------|----------------------|-------------------------|
| GPSR-BB    | 100              | 6.21            | 5.986e-003           | 1.653                   |
| MOPSOGA    | 100              | 3.56            | 3.986e-003           | 1.216                   |
| GPSR-BB    | 1000             | 12.33           | 3.921e-003           | 1.486                   |
| MOPSOGA    | 1000             | 7.18            | 1.326e-003           | 0.921                   |
| GPSR-BB    | 2000             | 15.14           | 3.281e-003           | 1.312                   |
| MOPSOGA    | 2000             | 9.32            | 0.986e-003           | 0.528                   |
In Figure 10 and 11, respectively, the relationship between the observation dimension and the reconstruction error and the successful probability of the reconstruction error is expressed in the sparse degree K=200. Obviously, with the increase of the number of channels, the probability of the reconstruction of the signal sparse coefficient is increased, and the reconstruction error is reduced, and in the case of the same observation dimension, the improved algorithm has 0.2 improvement compared with the original GPSR algorithm, the reconstruction error is reduced by 0.06; When the reconstruction probability is 1, the total reconstruction is less than the original algorithm.
4. Conclusions. In this paper, compressive sensing method is introduced Internet of Things data acquisition and processing, design and implementation of a cloud technology to accelerate compressed sensing algorithm architecture. In order to make full use of the existing computing resources in the Internet of things, a parallel compression method for dynamic compression sensing algorithm is proposed. Use of multicore / multi CPU and GPGPU accelerated hybrid approach acceleration theory framework, in order to in using compressed sensing method to reduce the networking data acquisition scale, at the same time, faster speed of implementation of the algorithm. Then combined the GPSR-BB and MOPSOGA algorithm to obtain local optimal solution, and get local optimal solution by MOPSOGA algorithmic search, then local optima GPSR-BB as the initial value of iteration algorithms reduce the number of iterations and Running time. In this paper, the algorithm has low complexity and high accuracy, but it has a certain degree of randomness, and how to find a more stable reconstruction algorithm will be further studied.

Acknowledgments. This work is partially supported by Natural Science and Technology Project Plan in Yulin University of China (No.2014xyy-09), Funding Project for Department of Education of Shaanxi Province of China (No.14JK1864), Thanks for the help.

REFERENCES

[1] H. Ammari, J. Garnier and V. Jugnon, Detection, reconstruction, and characterization algorithms from noisy data in multistatic wave imaging, *Discrete and Continuous Dynamical Systems - Series S*, 8 (2015), 389–417.

[2] S. J. Birkinshaw and G. Parkin, A hybrid neural networks and numerical models approach for predicting groundwater abstraction impacts, *Journal of Hydroinformatics*, 10 (2013), 127–137.

[3] B. Bonnard, T. Combot and L. Jassionnesse, Integrability methods in the time minimal coherence transfer for Ising chains of three spins, *Discrete and Continuous Dynamical Systems - Series A*, 35 (2015), 4095–4114.
[4] M. Costantiti, A. Farina and F. Zirilli, The fusion of different resolution SAR images, in Proceedings of the IEEE. Vol. 85, IEEE, 1997, 139–146.

[5] Z. Du, X. Chen and Z. Feng, Multiple positive periodic solutions to a predator-prey model with Leslie-Gower Holling-type II functional response and harvesting terms, Discrete and Continuous Dynamical Systems - Series S, 7 (2014), 1203–1214.

[6] F. Zhang, H.-F. Xue and D.-S. Xu, Big data cleaning algorithms in cloud computing, International Journal of Online Engineering, 9 (2013), 77–81.

[7] K. Zhang, H. Huang and H. Yang, A transformer fault diagnosis method integrating improved genetic algorithm with least square support vector machine, Power System Technology, 34 (2010), 164–168.

[8] J. Li, H. Liu and Q. Wang, Fast imaging of electromagnetic scatterers by a two-stage multilevel sampling method, Discrete and Continuous Dynamical Systems - Series S, 8 (2015), 547–561.

[9] M. Padilla, A. Perera, I. Montoliu, A. Chaudry, K. Persaud and S. Marco, Drift compensation of gas sensor array data by orthogonal signal correction, Chemometrics and Intelligent Laboratory Systems, 100 (2010), 28–35.

[10] C. Pohl and J. L. Van Genderen, Image fusion in remote sensing: Concepts, methods and applications, International Journal of Remote Sensing, 19 (1999), 823–854.

[11] V. M. Quiroga and I. Popescu, Cloud and cluster computing in uncertainty analysis of integrated flood models, Journal of Hydroinformatics, 15 (2013), 55–70.

[12] R. Hwang Ryol and M. Fred Huber, A particle filter approach for multi-target tracking intelligent robots and systems, Intelligent Robots and Systems, 11 (2007), 2753–2760.

[13] H. Schätter and U. Ledzewicz, Fields of extremals and sensitivity analysis for multi-input bilinear optimal control problems, Discrete and Continuous Dynamical Systems - Series A, 35 (2015), 4611–4638.

[14] D. Tsujinishi and S. Abe, Fuzzy least squares support vector machines for multiclass problems, Neural Networks, 16 (2003), 785–792.

[15] B. Ustün and W. J. Melssen, Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization, Analytical Chimica Acta (S0003-2670), 544 (2005), 292–305.

Received June 2015; revised August 2015.

E-mail address: xuejw@nwpu.edu.cn
E-mail address: xxk328@mail.nwpu.edu.cn
E-mail address: tfnew21@sina.com