

Low Complexity Decoding Algorithm for Uplink SCMA Based on Aerial Spherical Decoding

Xiaohong Ji¹, Junjun Du¹, Guoqing Jia^{1,*} and Weidong Fang^{2,3}

¹Qinghai Nationalities University, Xining, 810007, China

²Science and Technology on Micro-system Laboratory, Shanghai Institute of Micro-System and Information Technology, Chinese Academy of Sciences, Shanghai, 201800, China

³University of Chinese Academy of Sciences, Beijing, 100049, China

*Corresponding Author: Guoqing Jia. Email: guoqing.jia@qhmu.edu.cn

Received: 30 August 2020; Accepted: 13 October 2020

Abstract: As a new non-orthogonal multiple access technology for 5G massive machine type communication scenario, the sparse code multiple access (SCMA) has greatly improved the spectrum efficiency due to the high connection density. SCMA combines QAM (Quadrature Amplitude Modulation) modulation and sparse spreading into a codebook set to obtain forming gain. The user binary bit data is directly mapped into multi-dimensional codewords in the transmitter. The receiver uses the message passing algorithm (MPA) for multi-user detection to achieve efficient decoding. However, MPA is a good solution for SCMA, though its high complexity limits the application in practical systems. In order to reduce the complexity of MPA, this paper proposed a low-complexity decoding algorithm based on the serial spherical decoding message passing algorithm (SSD-MPA), which greatly enhanced the practicability of the SCMA system. The SSD-MPA took into account the distribution characteristics of the Gaussian noise and the reliability of the constellation points, and only updates the trusted constellation point messages of nodes within the spherical radius. At the same time, it used a serial strategy to speed up the convergence rate. The simulation results have shown that when the radius sets properly, the algorithm reduces the complexity by about 2/3 compared with the original MPA, and the performance was almost lossless.

Keywords: 5G; sparse code multiple access; MPA; spherical decoding; serial strategy

1 Introduction

The massive machine type communication (mMTC) is the main application scenario of 5G, and its main feature is the high connection density. The sparse code multiple access (SCMA) is a new type of non-orthogonal multiple access technology in the code domain [1–3], which evolves from the low density spread spectrum (LDS). It combines the quadrature amplitude modulation (QAM) and the sparse spreading into a codebook set, in so a ‘forming gain’ [4,5] is obtained. In a SCMA system, the binary bit



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

data of a user is directly mapped into the multi-dimensional codewords, and the receiver uses the message passing algorithm (MPA) to achieve efficient decoding. In a MPA, the sparseness of the codeword is used to effectively reduce the decoding complexity, and the performance is close to the optimal MPA. Therefore, the MPA is the most commonly used decoding algorithm in a SCMA. The SCMA can increase the spectral efficiency more than 150% and is expected to become a candidate technology for mMTC [6,7].

Although MPA has reduced the decoding complexity, the latter is still high, which limits the application of SCMA in practical communication systems [8–10]. To this end, Mu et al. [11] proposed a partially marginalized MPA (PM-MPA) which reduced the number of update nodes and fixed the complexity of MPA by marginalizing some users. However, the performance of the PM-MPA was significantly reduced, especially in the case of high signal-to-noise ratio (SNR). Du et al. [12] proposed a message passing algorithm (S-MPA) based on a serial strategy. The S-MPA converted the parallel calculations in the original MPA to the serial calculations, thereby, reducing the number of iterations required. Yang et al. [13] used the thresholds to control the iterative process, in order to reduce the complexity, but its performance also decreased. Klimentyev et al. [14] simplified the decoding process and reduced the number of iterations to 2–3 times based on the characteristics of the codeword in the codebook. Wei et al. [15] proposed a low-complexity decoding algorithm based on the list ball decoding (LSD). This algorithm only considered the signals in the hypersphere, and did not need to traverse all possible combinations, and thereby reduced the complexity. Wei et al. [16] introduced two low-complexity algorithms. One is the sign flip algorithm (SFA), which performed hard decisions and flipped the unreliable signals during the resource node updating. The other was a dynamic partial marginalized MPA, which dynamically determined the marginalized users through the reliability defined by SFA. Ma et al. [17] proposed a low-complexity MPA detector based on the dynamic factor graph (DFG-MPA). The DFG-MPA dynamically updated the factor graph by judging the size of the user codeword confidence level, where the larger value of the confidence level did not participate in the current and subsequent updates. And Miao et al. [18] proposed a continuously updated dynamic factor graph algorithm, which could speed up the convergence, and reduce the external information. Shen et al. [19] also introduced a dynamic subgraph MPA (DS-MPA) algorithm, which had a slow convergence and high complexity. Yuan et al. [20] proposed a hybrid message passing algorithm, which limited the user state to a Gaussian state based on the energy minimization approximation. Yang et al. [21] proposed a spherical decoding MPA (SD-MPA), which only updated the constellation point messages within the radius, though the convergence rate was slow.

In this context, the major contribution of this paper is to propose a serial spherical decoding MPA (SSD-MPA) algorithm, which can further reduce its complexity. Specifically, according to the relationship between the Gaussian noise distribution characteristics and the reliability of the constellation points, this algorithm chooses to update the trusted constellation points in the spherical range, and accelerates the convergence speed through a serial strategy. When the sphere radius sets properly, SSD-MPA reduces the computational complexity by about two-thirds, with almost no loss in the bit error rate (BER) performance. The rest of this paper is organized as follows. Firstly, the system model of SCMA is reviewed and analyzed in Section 2. Then, a serial spherical decoding MPA is proposed in Section 3. Furthermore, the numerical simulation and analysis are discussed in Section 4. Finally, the conclusions are drawn in Section 5.

2 SCMA System Model

2.1 Uplink SCMA System

We assume that, in an uplink SCMA system, J users share K with same resources. The system has L codebooks of dimension $M \times K$, where M represents the number of constellation points of the codebooks,

K represents the number of resources of the codebook. In this system, each user has an exclusive codebook, the system overload rate is defined as $\lambda = J/K$, where $J > K$.

The uplink SCMA system model of $J = 6, L = 6, K = 4$, and $M = 4$ is shown in Fig. 1.

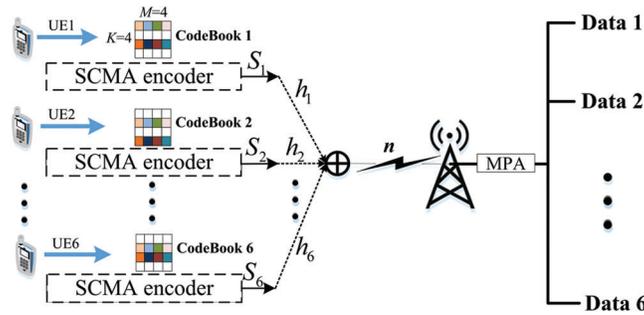


Figure 1: Uplink SCMA system model

At the user's transmitter, each user j uses the SCMA encoder to map a $\text{lb}(M)$ bit data stream into the $N(N < K)$ dimensional codewords according to the codebook allocated by the base station (BS), then N -dimensional codewords are sparsely mapped to the K resources according to the resource mapping relationship of the codebooks, and each user j finally outputs a K -dimensional codeword S_j . In the end, the data of these J users can be superimposed on the same K resources in a non-orthogonal manner, and transmitted to the BS.

The factor graph shows the non-orthogonal superposition of multiple users in Fig. 2. This graph contains two types of nodes: ones are resource nodes (FNs), the others are user nodes (VNs). Among them, the connection represents the payload relationship between the FNs and VNs.

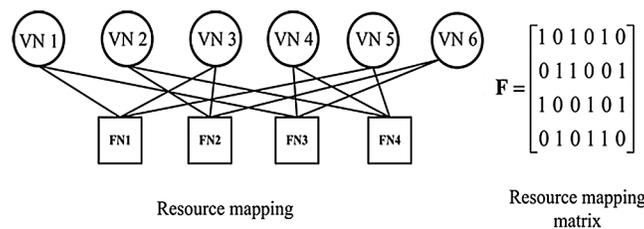


Figure 2: Uplink SCMA system factor graph

Therefore, the received signal of the receiver in a BS is expressed as follows:

$$y = \sum_{j=1}^J (h_j \times S_j) + n \tag{1}$$

where, $S_j = (S_{1,j}, S_{2,j}, \dots, S_{K,j})^T$ represents the codeword of user j on K resources. $h_j = (h_{1,j}, h_{2,j}, \dots, h_{K,j})^T$ is the channel coefficient, and $n \sim CN(0, \sigma^2 I)$ is K -dimensional Gaussian noise.

According to the factor graph, the codewords of d_f -users are superimposed on the same resource. So, the received signal on the k -th resource can be expressed as follows:

$$y_k = \sum_{j \in N(k)} (h_j \times S_j) + n_k \quad (2)$$

where, $N(k) = \{j | S_{k,j} \neq 0\}$ is the user set whose data is superimposed on the k -th resource, and the base is d_j .

2.2 MPA Detection Algorithm

As a typical multi-user detection algorithm, MPA is widely applied in a SCMA. It uses the sparse features of codewords to model the receivers based on the factor graph. The decoding process consists of two parts: iteratively updating node messages and codewords decisions.

1. Iteratively updating node messages

Firstly, an iteratively updating the node messages of resource node k and user node j can be expressed as follows:

$$I_{k \rightarrow j}^t(m_j) = \sum_{\sim m_j} \exp(\Phi(\zeta, k)) \times \prod_{\rho \in N(k) \setminus j} I_{\rho \rightarrow k}^{t-1}(m_\rho) \quad (3)$$

$$I_{j \rightarrow k}^t(m_j) = \text{norm}\left(\prod_{\gamma \in V(j) \setminus k} I_{\gamma \rightarrow j}^t(m_j)\right) \quad (4)$$

where, $V(j) = \{k | S_{k,j} \neq 0\}$ is the set of resources mapped by the j -th user, $\text{norm}(\bullet)$ indicates the normalization, and the modulus value is d_j . $\Phi(\zeta, k)$ represents the Euclidean Distance between the received signal and the constellation point. It is expressed as follows:

$$\Phi(\zeta, k) = \Phi(m_1, m_2, \dots, m_{d_j}, k) = \frac{-1}{\delta^2} \left\| y_k - \sum_{j \in N(k)} h_{k,j} \times S_{k,j}(m_j) \right\|^2 \quad (5)$$

It is worth noting that the message updated by the resource node in Eq. (3), and it needs to be used in Eq. (4) until the next iteration.

2. Codewords decisions

Then, after the iteration is completed, the codeword probability is calculated according to the log-likelihood ratio (LLR). And then the user bit data is judged, it is expressed as follows:

$$Q_j(m_j) = \prod_{k \in V(j)} I_{k \rightarrow j}^{t_{\max}}(m_j) \quad (6)$$

$$LLR(b_j) = \sum_{b_j=0} Q(m_j) / \sum_{b_j=1} Q(m_j) \quad (7)$$

3 Serial Spherical Decoding MPA (SSD-MPA)

3.1 SSD-MPA

Although the MPA can reduce the complexity to the order of M^{d_j} , its complexity increases exponentially with the number of superimposed users d_j and the size of codebook M . The complexity of MPA is mainly concentrated on the process of the iteratively updating node messages and iterative convergence. Considering these two process, in this paper, we propose a low-complexity decoding algorithm based on serial spherical decoding, which called SSD-MPA.

In SSD-MPA, the influence of Gaussian noise on the received signal is considered, that is to say, the Euclidean Distance between the constellation point and the received signal determines the credibility of the constellation point. When the Euclidean Distance is smaller, the credibility of the constellation point is higher, and the correct probability of the codeword corresponding to the constellation point is greater. Therefore, according to the requirements of the actual scenario, setting an appropriate spherical radius (i.e., Euclidean Distance) can achieve a trade-off between the performance and the complexity of BER.

At the same time, the SSD-MPA uses a serial strategy to update the nodes message, in order to further reduce the complexity. In this serial update strategy, the updated message of a resource node is directly passed to the user node without waiting for the next iteration, so the user node can use the message of the resource node in time. The SSD-MPA can speed up the convergence speed by serial updating. The steps of SSD-MPA are shown in [Algorithm 1](#):

Algorithm 1: SSD-MPA

Input: y, H, σ, N, R
Initialization:
for all j and $k \in V(j)$ and m do
 $I_{k \rightarrow j}^0(m_j) = 1/M, I_{j \rightarrow k}^0(m_j) = 1/M$
end for
Calculate contingent probability:
for all $k, j \in N(k)$ and $m = 1 : M$ do
 Calculate $\Phi(\zeta, k)$ via (5)
end for
Iteration:
for $t = 1:N$ and all $k, j \in N(k)$ do
 for $m = 1 : M$ do
 If $\text{abs}(\Phi(\zeta, k)) < R$ do
 update $I_{k \rightarrow j}^t(m_j)$ via (3)
 end for
 update $I_{j \rightarrow k}^t(m_j)$ via (4)
 end for
Soft output:
for $j = 1 : J$ and all m do
 Calculate $Q_j(m_j)$ via (6)
 return LLR of bit via (7)
end for

3.2 Complexity Analysis

The complexity of the MPA focuses on the process of the iteratively updating node messages. The original MPA traverses all constellation point sets with each iteration, so the complexity is high. The SSD-MPA limits the update range to a spherical radius, and it does not need to update all node messages. [Tab. 1](#) gives the convergence complexity formula of the SSD-MPA and other algorithms. Among them, N_{MPA} , N_{S-MPA} , N_{SD-MPA} , N_{DS-MPA} , and $N_{SSD-MPA}$ indicate the number of iterations required for the corresponding algorithm to converge, and $|\Phi^*(k)|$ represents the number of nodes on the resource k that fall in a spherical range.

Table 1: Comparison of convergence complexity

Algorithm	Addition	Multiplication	Comparison
MPA	$N_{MPA}kd_fM(M^{d_f-1} - 1)$	$(d_f - 1)N_{MPA}Kd_fM^{d_f}$	0
S-MPA	$N_{S-MPA}kd_fM(M^{d_f-1} - 1)$	$(d_f - 1)N_{S-MPA}Kd_fM^{d_f}$	0
SD-MPA	$N_{SD-MPA}d_f \times$ $\left(\sum_{k=1}^K \Phi^*(k) - KM\right)$	$N_{SD-MPA}(d_f - 1)d_f \sum_{k=1}^K \Phi^*(k) $	$N_{SD-MPA}KM^{d_f}$
DS-MPA	$[Kd_f + (N_{DS-MPA} - 1)K(d_f - 1)]$ $(M^{d_f} - M)$	$(d_f - 1)M^{d_f} \times$ $[Kd_f + (N_{DS-MPA} - 1)K(d_f - 1)]$	$(N_{DS-MPA} - 1)Kd_f$
SSD-MPA	$N_{SSD-MPA}k \sum_{j=1}^{d_f} \sum_{m=1}^M q_j - 1$	$N_{SSD-MPA}d_f \left(\sum_{k=1}^K \Phi^*(k) - KM\right)$	$N_{SSD-MPA}KM^{d_f}$

4 Simulations and Numerical Analysis

In this section, we use the MATLAB simulation platform to numerically simulate the SSD-MPA. The convergence speed, BER, and complexity performance of several algorithms are analyzed. The simulation parameters are shown in [Tab. 2](#).

Table 2: Simulation parameters settings

Parameter	Value
Channel model	AWGN
User number J	6
Number of resources K	4
Codebook number L	6
Constellation point size M	4
Overlay users d_f	3
Overlay resources d_v	2

In [Fig. 3](#), the convergence performance of several algorithms is shown at SNR = 12 dB. According to the simulation results of [Fig. 3](#), it can be known that the DS-MPA has the slowest convergence speed, and it requires approximately 7 iterations. While the MPA requires approximately 5–6 iterations. In contrast, the S-MPA, SD-MPA and SSD-MPA converge faster, requiring only 2–3 iterations to converge. It should be

noted that although SD-MPA with a radius of δ has converged in 3 iterations, however the performance is poor. However, 5 iterations are required for SD-MPA with a radius of 2δ . From the analysis in Fig. 3, it can be concluded that SSD-MPA has the fastest convergence speed.

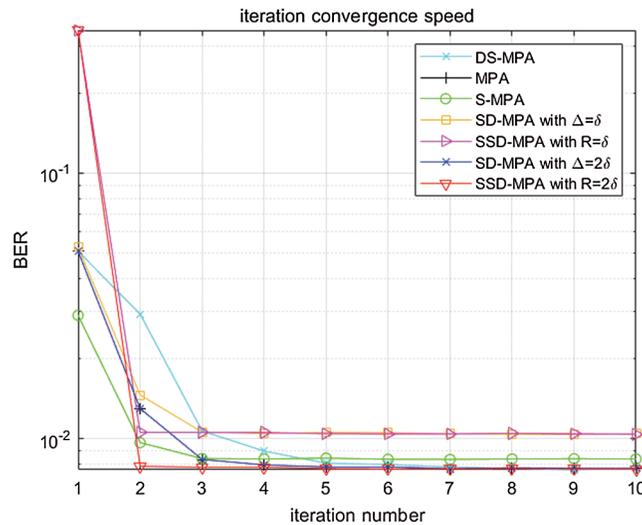


Figure 3: Comparison of convergence speed

According to the simulation results obtained in Fig. 3, the SSD-MPA only needs 2 iterations to converge, and the number of iterations that is required by the algorithm does not exceed 10 times. Therefore, the BER performance simulations at $N = 2$ and $N = 10$ are given in Figs. 4 and 5.

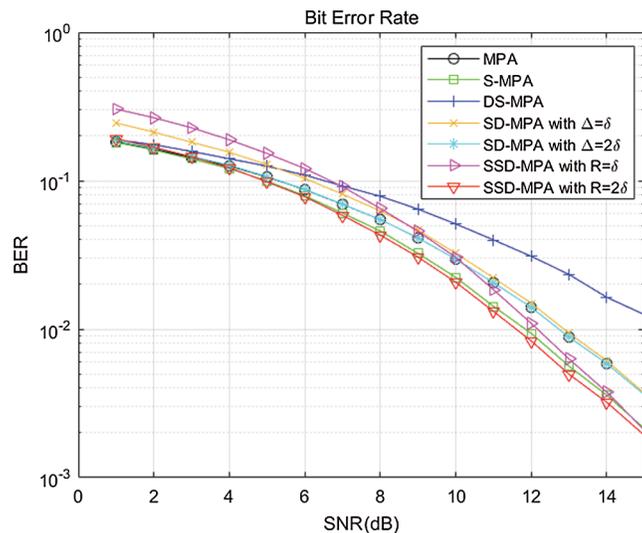


Figure 4: BER performance comparison graph for $N = 2$

In Fig. 4, the BER performance comparison of various algorithms is given, when N is 2. It can be seen from Fig. 4 that when the SNR is low, the performance of SD-MPA and SSD-MPA with a radius of δ is the lowest, and the performance of S-MPA and SSD-MPA with a radius of 2δ is the best. With the increase of

SNR, the performance of several algorithms is gradually approaching, but the performance of DS-MPA becomes the worst, and the performance of SSD-MPA with radius 2δ is always optimal.

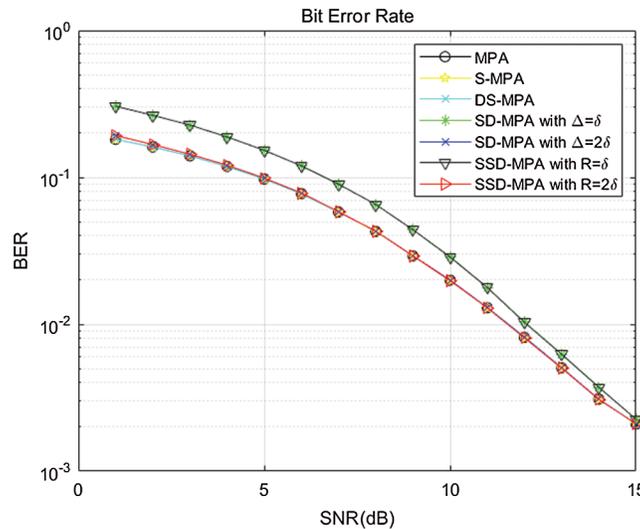


Figure 5: BER performance comparison graph for $N = 10$

In Fig. 5 the BER performance comparison of various algorithms is given, when N is 10, in which all the algorithms are converged. The performance of the SD-MPA and SSD-MPA with a radius of 2δ is similar to the performance of the S-MPA and MPA, while the performance of the SD-MPA and SSD-MPA with a δ radius is consistent. Therefore, it can be concluded from the analysis in Fig. 5 that the performance of the SSD-MPA with a radius of 2δ has no loss.

Although the complexity formulas of several algorithms have been given in Tab. 1, as the number of constellation points falling in a spherical range is unknown, so the actual calculation amount of the SD-MPA and SSD-MPA cannot give specific formulas. Therefore, in Fig. 6 the actual operation amount of several algorithms is plotted in the simulation process.

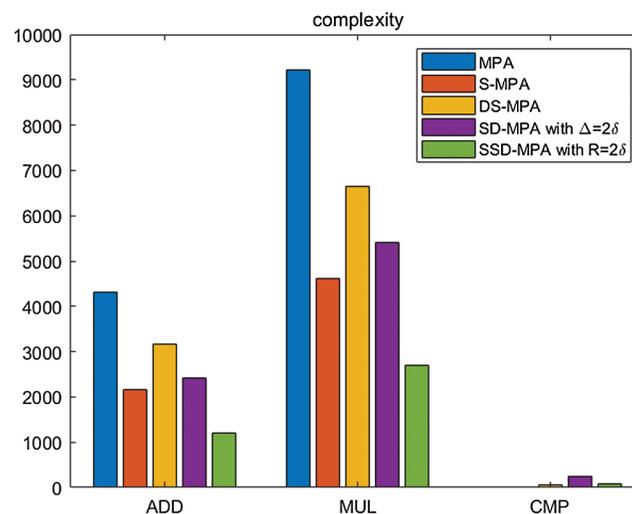


Figure 6: Comparison of complexity

In Fig. 6, the calculations of addition, multiplication, and comparison of several algorithms are respectively counted. Although the MPA does not have a comparison operation, the addition and multiplication operations are very large. In contrast, the amount of the SSD-MPA operation is only about 1/3 of the MPA.

5 Conclusion

As a new non-orthogonal multiple access technology for 5G, the SCMA has greatly improved spectrum efficiency. However, the SCMA's complexity is high, which limits its application in practical communication systems. This paper proposed a low complexity decoding algorithm based on serial spherical decoding, and verified the BER performance and complexity of the proposed algorithm through MATLAB simulation platform. The results show that, when the sphere radius sets properly, the SSD-MPA can reduce the complexity of MPA by about 2/3, and the performance is almost lossless. The practicability of the MPA is greatly enhanced. The relationship between sphere radius and performance will be researched in the further work.

Funding Statement: This work is supported by the National Natural Science Foundation of Qinghai Province, China (No. 2020-ZJ-724).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. Agiwal, A. Roy and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.
- [2] W. Zhang, W. Fang, Q. Zhao, X. Ji and G. Jia, "Energy efficiency in Internet of Things: An overview," *Computers Materials & Continua*, vol. 63, no. 2, pp. 787–811, 2020.
- [3] H. Nikopour and H. Baligh, "Sparse code multiple access," in *Proc. PIMRC*, London, UK, pp. 332–336, 2013.
- [4] W. Fang, N. Cui, W. Chen, W. Zhang and Y. Chen, "A trust-based security system for data collecting in smart city," *IEEE Transactions on Industrial Informatics*. (Early Access).
- [5] R. Hoshyar, F. P. Wathan and R. Tafazolli, "Novel low-density signature for synchronous CDMA systems over AWGN channel," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1616–1626, 2008.
- [6] F. Bi, X. Fu, W. Chen, W. Fang, X. Miao *et al.*, "Fire detection method based on improved fruit fly optimization-based SVM," *Computers, Materials & Continua*, vol. 62, no. 1, pp. 199–216, 2020.
- [7] W. Fang, W. Zhang, Q. Zhao, X. Ji, W. Chen *et al.*, "Comprehensive analysis of secure data aggregation scheme for industrial wireless sensor network," *Computers, Materials & Continua*, vol. 61, no. 2, pp. 583–599, 2019.
- [8] X. F. Li, Y. B. Zhuang and S. X. Yang, "Cloud computing for big data processing," *Intelligent Automation & Soft Computing*, vol. 23, no. 4, pp. 545–546, 2017.
- [9] J. Peng, M. Tang, M. Li and Z. Zha, "A load balancing method for massive data processing under cloud computing environment," *Intelligent Automation & Soft Computing*, vol. 23, no. 4, pp. 547–553, 2017.
- [10] W. Wu, Y. Chen and D. Seng, "Implementation of web mining algorithm based on cloud computing," *Intelligent Automation & Soft Computing*, vol. 23, no. 4, pp. 599–604, 2017.
- [11] H. Mu, Z. Ma, M. Alhaji, P. Fan and D. Chen, "A fixed low complexity message pass algorithm detector for uplink SCMA system," *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp. 585–588, 2015.
- [12] Y. Du, B. Dong, Z. Chen, J. Fang and L. Yang, "Shuffled multiuser detection schemes for uplink sparse code multiple access systems," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1231–1234, 2016.
- [13] L. Yang, Y. Liu and Y. Siu, "Low complexity message passing algorithm for SCMA system," *IEEE Communications Letters*, vol. 20, no. 12, pp. 2466–2469, 2016.

- [14] V. P. Klimentyev and A. B. Sergienko, "A low-complexity SCMA detector for AWGN channel based on solving overdetermined systems of linear equations," in *Proc. REDUNDANCY*, St. Petersburg, pp. 61–65, 2016.
- [15] F. Wei and W. Chen, "Low complexity iterative receiver design for sparse code multiple access," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 621–634, 2017.
- [16] L. Wei, B. Huang and J. Zheng, "Low-complexity detectors for uplink SCMA: Symbol flipping and dynamic partial marginalization-based MPA," in *Proc. VTC Spring*, Porto, pp. 1–5, 2018.
- [17] X. Ma, L. Yang, Z. Chen and Y. Siu, "Low complexity detection based on dynamic factor graph for SCMA systems," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2666–2669, 2017.
- [18] J. Miao, X. Hu and Z. Zhao, "A low complexity multiuser detection scheme with dynamic factor graph for uplink SCMA systems," in *Proc. ICC*, Changchun, China, pp. 846–851, 2019.
- [19] M. Shen, J. Li and Y. He, "Dynamic subgraph detection algorithm for uplink SCMA system," *Telecommunication Engineering*, vol. 59, no. 7, pp. 749–754, 2019.
- [20] W. Yuan, N. Wu, A. Zhang, X. Huang, Y. Li *et al.*, "Iterative receiver design for FTN signaling aided sparse code multiple access," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 915–928, 2020.
- [21] L. Yang, X. Ma and Y. Siu, "Low complexity MPA detector based on sphere decoding for SCMA," *IEEE Communications Letters*, vol. 21, no. 8, pp. 1855–1858, 2017.