IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 9, 2020, accepted July 17, 2020, date of publication July 27, 2020, date of current version August 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012007

Low-Complexity Multi-User Detection Based on Gradient Information for Uplink Grant-Free NOMA

FANG JIANG, GUOLIANG ZHENG, YANJUN HU, (Senior Member, IEEE), YI WANG, AND YAOHUA XU

Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei 230601, China

Corresponding author: Fang Jiang (jiangfang@ahu.edu.cn)

This work was supported in part by the Natural Science Research Project of Universities in Anhui Province under Grant KJ2018A0019; and in part by the Dr. Research Initiation Fund of Anhui University.

ABSTRACT Massive machine type communication (mMTC) serves an irreplaceable role in the development process of the Internet of Things (IoT). Because of its characteristics of massive connection and sporadic transmission, compressed sensing (CS) has been applied in joint user activity and data detection in the uplink grant-free non-orthogonal multiple access (NOMA) system. In previous work, greedy iterative-based multi-user detection (MUD) algorithms were developed in mMTC scenarios because of the computational benefit and competitive performance. However, conventional greedy iterative-based MUD algorithms still suffer from high computational complexity due to the process of large-size matrix inversion with the accession of massive devices into the system. In this paper, gradient information is used to address this problem. A low-complexity gradient descent-based gradient pursuit MUD (GDGP-MUD) algorithm is proposed, which uses the gradient information of error function in the process of iteration as a new updating direction, instead of the matrix inversion process. Then, a multi-step quasi-Newton MUD (MSQN-MUD) algorithm is proposed to improve the precision of detection while maintaining low complexity. In the algorithm, high-order information in the process of adjacent iteration is used effectively to update data values more accurately. Moreover, the convergence and complexity analysis of both algorithms are derived. The analysis shows that both proposed algorithms have lower computational consumption than most of the state-of-the-art greedy-based MUD algorithms. It is worth noting that in comparison to most existing CS-based MUD algorithms, the two proposed algorithms do not require the exact user sparsity level and, thus, reduce the dependence on prior knowledge. The numerical experiments demonstrate that the proposed algorithms have better real-time performance than existing greedy-based MUD algorithms with similar symbol error rate performance.

INDEX TERMS Massive machine type communication (mMTC), non-orthogonal multiple access (NOMA), multi-user detection, compressed sensing, gradient method.

I. INTRODUCTION

Massive machine type communication is expected to be a key technology enhancement for 5th-generation wireless networks [1], offering characteristics of massive connectivity, low latency, low power, short-message packets, sporadic communication, etc. To meet the requirements of massive connectivity, non-orthogonal multiple access (NOMA) is considered an essential enabling technology for mMTC [2]. The key idea behind NOMA is to serve multiple devices in the same physical channel, which can support more connectivity as compared to conventional orthogonal multiple access (OMA) techniques [3].

In the existing long-term evolution (LTE) system, the base station (BS) allocates different time-frequency resources to each accessed user in a request-grant manner so that the receiver knows which user is active or inactive. Compared to the LTE system, multiple user signals will be superimposed within the same time-frequency resources, the user activity is unknown to the BS in uplink grant-free NOMA, and the receiver needs to detect user activity and decouple these superimposed signals. In [4], a message passing algorithm

The associate editor coordinating the review of this manuscript and approving it for publication was Zilong Liu.

(MPA) receiver has been employed for different signature-based NOMA (S-NOMA) schemes to implement MUD, which results in zero probability of error in the absence of noise. However, it exhibits the problem that activity information should be known in advance, which offers little obvious advantage in the grant-free scheme.

The grant-free NOMA system is considered a candidate to reduce signaling overhead and transmission delay. The BS does not allocate different channel resources for users. Since the numbers of active devices are far smaller than total devices, the characteristic of sparsity is satisfied. The framework of compressive sensing could be introduced to MUD in the uplink grant-free NOMA system, which has inspired us to design low-complexity MUD algorithms based on CS.

CS is an effective and promising signal processing techniques, which has been widely used in many aspects of communication systems, including joint source-channel-network coding [5], channel estimation [6], [7], spectrum sensing [8] and so on [9], [10]. In addition, CS has been extensively investigated in MUD [11]–[25]. In view of the different streams of the literature, compressive sensing can be divided into three categories: convex optimization algorithms, greedy iterative algorithms and Bayes algorithms. Convex optimization algorithms attribute CS to the sparse LS (least squares) problem, using convex optimization to solve, greedy iterative algorithms via fast iterative computation for the joint detection of user activity and data, and Bayes algorithms relying on a posterior probability to judge active users and data [7]. However, due to the computational benefit and competitive performance, greedy iterative algorithms are widely used in mMTC scenarios [26].

In [27]–[30], greedy iterative algorithms, i.e., orthogonal matching pursuit (OMP) [27], group orthogonal matching pursuit (GOMP) [28], compressive sampling matching pursuit (CoSaMP) [29], and subspace pursuit (SP) [30], have been used to detect user activity and/or data based on CS. However, the user activity and data are independently detected in each time slot, which neglects the correlations of transmitted signals in adjacent time slots. In practice, the signal transmission usually maintains several continuous time slots, which brings out the temporal correlation of active user sets and structured common sparsity in a frame. Therefore, a dynamic compressive sensing based multi-user detection (DCS) [15] algorithm has been proposed by exploiting the temporal correlation. Instead of using an empty set as initial active user set in each time slot, the current active user set has been regarded as the initial set of user activity in the next time slot. Based on the DCS algorithm, a prior-information-aided adaptive compressive sensing (PIA-ASP) algorithm [16] has been put forward to enhance the SER performance.

Apart from temporal correlation, the exploitation of the structured common sparsity of a frame can improve MUD performance. An iterative order recursive least square (IORLS) algorithm [17] and an alternative direction method of multipliers (ADMM) [18] were proposed by exploiting the structured common sparsity. In the two algorithms above, user activity is assumed to remain unchanged within the entire frame. However, in the MTC scenario, user access is random and usually involves transmitting short packets. A more general scenario is that the active user sets maintain certain structured features but can be changed in a frame, which is called the burst-sparsity model [16]. In this scenario, the structured matching pursuit (SMP)-based dynamic MUD (hereafter referred to as SMP) was proposed in [19], and the active user sets have been divided into common user sets and dynamic active user sets. The SMP algorithm achieves better SER gain as a benefit of exploiting structured common sparsity, but the frequent implementation of matrix inversion leads to high computation complexity. Additionally, the user sparsity level supported advance knowledge at the BS, and this assumption is unrealistic. Block-sparsity-based MUD has been used in [20], and the author proposed threshold aided block sparsity adaptive subspace pursuit (TA-BSASP) and cross-validation by statistics and machine learning mechanisms (CVA-BSASP), which do not require user sparsity level as prior information. The former exploits a threshold to terminate iteration, while the latter adopts the statistical and machine learning mechanism cross-validation to determine the stopping condition. These algorithms still suffer from high computational complexity due to the process of large size matrix inversion with the access of massive devices into the system.

To reduce the computational complexity and enhance real-time performance, the gradient information was introduced into MUD. We propose two novel gradient-based algorithms to exploit the gradient information of the error function to avoid matrix inversion. Furthermore, the computational complexity analysis and comparison are provided. Numerical experiments show that the two proposed algorithms have lower computational complexity and higher real-time performance compared with the existing structured CS-based algorithms.

We summarize our major contributions as follows:

To reduce computational complexity, a low-complexity first-order gradient pursuit-based MUD algorithm (GDGP-MUD) is proposed. The MUD is modeled as an optimization problem, and the first-order gradient of the error function is introduced in the iterative process as the updating direction to update the data value. The analysis and experimental results show that the GDGP-MUD algorithm has lower complexity and higher real-time performance than its counterparts.

The MSQN-MUD algorithm is proposed. The main idea of MSQN-MUD is that the weighted average value of two-order derivative gradient information between two successive iterations is introduced to improve SER performance. To avoid the calculation of the two-order derivative and matrix inversion, an approximate positive definite symmetric matrix is introduced to replace the inverse matrix of Hessian matrices (two-order derivative).

In dynamic user information detection, the revised convergence condition has been adopted to allow for faster convergence and the avoidance of requiring prior information (sparsity), as in most state-of-the-art CS-based MUD algorithms. In contrast with algorithms that rely on residual value to determine whether to terminate loops, our algorithms adopt modified residual value, which is deduced via received signal minus revised estimated symbols to break iterations.

The rest of this paper is presented as follows. In Section II, we introduce the system model based on the burst sparsity model. Then, we show two proposed algorithms, GDGP-MUD and MSQN-MUD, in Section III. Convergence and complexity analyses are presented in Section IV. In Section V, we simulate the proposed algorithm and compare it with the respective former literature publications. Finally, we present the conclusions in Section VI.

II. SYSTEM MODEL

We consider an uplink grant-free NOMA scheme for a mMTC scenario, where M users transmit signals to the BS. Due to their size limitation, all user terminals are equipped with a single antenna. We assume that the system uses N subcarriers, where $N < M$. The BS can be configured with single or multiple antennas. If the BS has multiple antennas, then diversity gain can be provided at the receiver. For simplicity, we use the single-antenna receiver model. The transmitted symbol is spread onto a unique spreading sequence s_m with length N, which is known at the BS station. Therefore, the received signal of BS in j slot is formulated as follows:

$$\mathbf{y}_j = \sum_{m=1}^{M} g_{nm} s_{nm} x_m(j) + \mathbf{z}_j, \quad n = 1, \ldots, N \tag{1}$$

where $x_m(j)$ represents the mth user transmit signal in j slot. $\mathbf{z}_j$ is the noise at j slot that follows complex Gaussian distribution with 0 mean and σ^2 variance. s_{nm} is the nth component of the spreading sequence of s_m. g_{nm} is the channel gain in the nth channel of the mth user, and the identically and independently distributed (i.i.d.) complex Gaussian variables with zero mean and unit variance, i.e., Rayleigh fading channel, is considered in this paper. The received signal at j slot can be rewritten as follows:

$$\mathbf{y}_j = \mathbf{A}\mathbf{x}_j + \mathbf{z}_j, \quad j = 1, 2, \ldots, J \tag{2}$$

$\mathbf{A} \in C^{N \times M}$ is the equivalent channel matrix for which the element of the nth row and kth column is $g_{nm} s_{nm}$. $\mathbf{A}$ is formulated as follows:

$$\mathbf{A} = \begin{pmatrix} g_{11}s_{11} & \cdots & g_{1M}s_{1M} \\ \vdots & \ddots & \vdots \\ g_{N1}s_{N1} & \cdots & g_{NM}s_{NM} \end{pmatrix} \tag{3}$$

It is assumed that $\mathbf{A}$ remains unchanged over the entire frame time because the length of a frame is designed to be shorter than the channel coherence time. In the receiver, the equivalent channel matrix $\mathbf{A}$ could be deduced by channel estimation technology.

Because of the characteristics of the sporadic transmitting signal, CS knowledge can be utilized to jointly detect active user location and recover the transmission signal. There are several characteristics in the realistic communication system, including the following: 1) users can randomly connect or leave the system, and 2) there are some correlations in adjacent consecutive time intervals for user transmission data. The burst-sparsity model [16] has been considered in this paper, where only a small percentage of user activity will change.

We refer to users who remain active within a frame as common active users, while users whose activity has changed are called dynamic active users. The union of the dynamic active user set and the common active user set is called the support set. The number of total users M is far larger than the active user number K.

We assume that each user switches their activity and transmits a signal in an identical slot. For an active user, the transmitting signal value after the channel code is taken from a complex constellation set is Λ, while the transmit signal value of an inactive user is equivalent to zero.

III. PROPOSED ALGORITHM

In this section, we propose two low-complexity MUD algorithms to jointly detect user activity and data. In essence, our proposed algorithms are both based on the gradient pursuit framework, but they are different from the classical pursuit framework. More specifically, we derive the common active user set and dynamic active user set of support in the burst-sparsity model so that we can detect common active user information (C-MUD) and dynamic active user information (D-MUD), respectively. Additionally, unlike the classical gradient pursuit framework, which only relies on residuals to determine the termination condition, two methods have been individually adopted into C-MUD and D-MUD to terminate iteration.

The goal of C-MUD is to find all common active users that have fixed locations in a frame and to reconstruct the estimate signal $\hat{\mathbf{x}}_{\Gamma_C, j}$. When the locations and data of all common active users have been detected, the information is used as a priori information to jointly detect dynamic active user location and data in D-MUD. First, we set an iteration termination parameter S, which relies on experience to determine whether to terminate the C-MUD part. It is not necessary for S to precisely match the length of the common active user set. This is because the information detected by C-MUD is not the final output data so that we can approximately detect it in C-MUD. Then, one frame gradient is accumulated for each user to justify the maximum gradient user index update common active support set Γ_C, which is expressed as follows:

$$\Gamma_C = \Gamma_C \cup \arg\max \sum_{j=1}^{J} |\mathbf{G}(m, j)|^2 \tag{4}$$

$G(:,j) = \mathbf{A}^{\mathrm{H}} r_j^{(s-1)}$. In C-MUD, the estimated signal is formulated as follows:

$$\hat{\mathbf{x}}^s_{\Gamma_C,j} = \hat{\mathbf{x}}^{s-1}_{\Gamma_C,j} + a_j^s d_j^s \tag{5}$$

where d and a represent update direction and step size, respectively, and using different methods to solve for d will result in different algorithms. When S iterations are completed, $\hat{\mathbf{x}}^S_{\Gamma_C,j}$, r_j^S, and Γ_C will serve as prior information to detect user activity and data in D-MUD.

The goal of D-MUD is to detect dynamic active user information and revise detected information in the former part. The D-MUD steps are essentially similar to those of the C-MUD, but there are some differences. In this part, we do not require setting an iteration termination parameter but rather rely on a revised residual to determine the break iteration. We stop iterations until the revised residual value is no longer reduced.

The revised residual is defined as follows:

$$r\_revise_j^i = \mathbf{y}_j - \mathbf{A}\tilde{\mathbf{x}}_j^i \tag{6}$$

where $\tilde{x}^i_{m,j} = \begin{cases} \Lambda & if\, m \in \Gamma^i \\ 0 & otherwise \end{cases}$

Furthermore, it is not necessary to accumulate the gradients in one frame, and only the gradients in the individual time slots need to be calculated. Because the locations of dynamically active users are not fixed, we have to search for them by time slot.

A. GDGP-MUD

Conventional greedy multi-user detection algorithms often require the matrix inverse process or pseudo-inverse process to reconstruct the transmitted signal. While matrix inversion requires $\frac{i^3}{3}$ multiplication operations, matrix pseudo-inversion requires $\frac{i^3}{3} + (N+1)i^2 + Ni$ multiplication operations, which we specifically analyse in Section IV. The gradient descent method based on gradient pursuit [31] uses the derivative information of the objective function as the new update direction in each iteration to reconstruct the transmission signal. We can formulate the update direction d_j^i as follows:

$$d_j^i = -g^i_{\Gamma^i,j} \tag{7}$$

The detail of GDGP-MUD is shown in Algorithm 1.

B. MSQN-MUD

The Newton method is one of the most effective methods to solve the unconstrained optimization problem, which utilizes the first and second derivatives of the objective function to update the direction.

The update direction is expressed as follows:

$$\nabla^2 f(x^i) d = -\nabla f(x^i) \tag{8}$$

where $\nabla f(x^i)$ and $\nabla^2 f(x^i)$ are represented as the first-order and second-order Hessian matrices of the objective function, respectively. However, as the users transmit signal is increased, the solution of the update direction requires more complex matrix inverse. To reduce the computation complexity, the positive definite symmetric matrix $\mathbf{H}$, which utilizes the gradient information from the previous p steps to establish an extended quasi-Newton equation to approximate the Hessian matrix of the objective function by an interpolation polynomial, is used to replace the inverse of the second-order Hessian matrix $\nabla^2 f(x^i)^{-1}$ in Newton's method [32]. Hence, the update direction in the ith iteration is indicated as follows:

$$d_j^i = -\mathbf{H}_j^i \nabla f(x^i) \tag{9}$$

Algorithm 1 GDGP-MUD

Input:
Received signals: $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_J$
Equivalent channel matrices: $\mathbf{A}$

Output:
Reconstructed sparse signals: $\tilde{\mathbf{x}}_1^{i-1}, \tilde{\mathbf{x}}_2^{i-1}, \ldots \tilde{\mathbf{x}}_J^{i-1}$

Initialize

$\Gamma_C = \emptyset, r_j^0 = \mathbf{y}_j, i = 0, j = 1, 2, \ldots, J$;

C-MUD:

for $s = 1 : S$, make

$\Gamma_C = \Gamma_C \cup \arg\max \sum_{t=1}^{T} |\mathbf{G}(m,j)|^2$, where $\mathbf{G}(:,j) = \mathbf{A}^{\mathrm{H}} r_j^{(s-1)}$

$d_j^s = -g^s_{\Gamma_C,j} = -\mathbf{G}(\Gamma_C, j)$

$c_j^s = \mathbf{A}_{\Gamma_C} d_j^s$

$a_j^s = \frac{\langle r_j^s, c_j^s \rangle}{\|c_j^s\|_2^2}$

$\hat{\mathbf{x}}^s_{\Gamma_C,j} = \hat{\mathbf{x}}^{s-1}_{\Gamma_C,j} + a_j^s d_j^s$

$r_j^s = r_j^{s-1} - a_j^s c_j^s$

end for

D-MUD:

$\Gamma^i = \Gamma_C, i = 1, r_j^0 = r_j^S, \hat{\mathbf{x}}_j^0 = \hat{\mathbf{x}}_j^S$

for $j = 1 : J$, make

while $i = 1$ or $\|r\_revise_j^i\|_2^2 \le \|r\_revise_j^{i-1}\|_2^2$

$i = i + 1$

$\Gamma^i = \Gamma^i \cup \arg\max_m |g_{m,j}|^2$, where $g_j = \mathbf{A}^{\mathrm{H}} r_j^{(i-1)}$

$d_j^i = -g^i_{\Gamma_i,j}$

$c_j^i = \mathbf{A}_{\Gamma^i} d_j^i$

$a_j^i = \frac{\langle r_j^i, c_j^i \rangle}{\|c_j^i\|_2^2}$

$\hat{\mathbf{x}}^i_{\Gamma^i,j} = \hat{\mathbf{x}}^{i-1}_{\Gamma,j} + a_j^i d_j^i$

$r_j^i = r_j^{i-1} - a_j^i c_j^i$

$\tilde{x}^i_{m,j} = \begin{cases} \Lambda & if\, m \in \Gamma^i \\ 0 & otherwise \end{cases}$

$r\_revise_j^i = \mathbf{y}_j - \mathbf{A}\tilde{\mathbf{x}}_j^i$

end while

end for

Moreover, w and t denote estimated signal interpolation and gradient interpolation, respectively.

$$w_j^i = \hat{\mathbf{x}}_j^i - \hat{\mathbf{x}}_j^{i-1} \tag{10}$$

$$t_j^i = \nabla f(x^i) - \nabla f(x^{i-1}) \tag{11}$$

According to the derivation of the interpolation polynomial, the linear combinations of the first p terms $\{w^{i-q}\}_{q=0}^{p-1}$ and $\{t^{i-q}\}_{q=0}^{p-1}$ are represented by u_j^i and v_j^i, respectively.

$$u_j^i = \sum_{q=0}^{p-1} w_j^{i-p} \left\{ \sum_{i=p-q}^{p} \zeta_i'(\tau_p) \right\} \tag{12}$$

$$v_j^i = \sum_{q=0}^{p-1} t_j^{i-p} \left\{ \sum_{i=p-q}^{p} \zeta_i'(\tau_p) \right\} \tag{13}$$

$$\begin{aligned} \zeta_o'(\tau_p) &= (\tau_o - \tau_p)^{-1} \prod_{q=0, q\neq o}^{l-1} \frac{\tau_p - \tau_q}{\tau_o - \tau_q} \\ &= (-1)^{p-o} \frac{p!}{(p-o)!o!(p-o)!} \end{aligned} \tag{14}$$

$$\zeta_p'(\tau_p) = \sum_{q=0}^{p-1} (\tau_p - \tau_q)^{-1} = \sum_{o=1}^{p} \frac{1}{o} \tag{15}$$

where $\zeta_p'(\tau)$ is the first derivative of the ith-order of the pth standard Lagrange polynomial of the set τ_p. The multiple quasi-Newton equation is expressed as follows:

$$\mathbf{H}_j^{i+1} v_j^i = u_j^i \tag{16}$$

In this paper, we consider the two-step quasi-Newtonian. Letting $p = 2$, u_j^i and v_j^i can be updated.

$$u_j^i = \frac{3}{2} w_j^i - \frac{1}{2} w_j^{i-1} \tag{17}$$

$$v_j^i = \frac{3}{2} t_j^i - \frac{1}{2} t_j^{i-1} \tag{18}$$

The update method of the approximate matrix $\mathbf{H}_j^i$ is the same as the following:

$$\mathbf{H}_j^{i+1} = \mathbf{H}_j^i - \frac{\mathbf{H}_j^i v_j^i (v_j^i)^{\mathrm{H}} \mathbf{H}_j^i}{(v_j^i)^{\mathrm{H}} \mathbf{H}_j^i v_j^i} + \frac{u_j^i (u_j^i)^{\mathrm{H}}}{(u_j^i)^{\mathrm{H}} v_j^i} \tag{19}$$

Embedding the idea of the multi-step quasi-Newton method into the framework of gradient pursuit, we propose a new multi-user detection method, called MSQN-MUD. The update direction with the multi-step quasi-Newton method in every iteration is equivalent to Eq. 20.

$$d_j^i = -\mathbf{H}_j^{i-1} g_{\Gamma^i, j}^i \tag{20}$$

The specific steps of the MSQN-MUD algorithm are shown in Algorithm 2.

IV. PERFORMANCE ANALYSIS

In this section, we analyse the convergence and computational complexity of the proposed GDGP-MUD and MSQN-MUD algorithms.

Algorithm 2 MSQN-MUD

Input:
Received signals: $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_J$
Equivalent channel matrices: $\mathbf{A}$

Output: Reconstructed sparse signals: $\tilde{\mathbf{x}}_1^{i-1}, \tilde{\mathbf{x}}_2^{i-1}, ...\tilde{\mathbf{x}}_J^{i-1}$

Initialize

$\Gamma_C = \emptyset, r_j^0 = \mathbf{y}_j, i = 0, \mathbf{H}_j^0 = 1$

C-MUD:

for $s = 1 : S$ do

$\Gamma_C = \Gamma_C \cup \arg\max \sum_{j=1}^{J} |\mathbf{G}(m, j)|^2$, where $\mathbf{G}(:, j) = \mathbf{A}^{\mathrm{H}} r_j^{(s-1)}$

$d_j^s = -\mathbf{H}_j^{s-1} g_{\Gamma_C, j}^s = -\mathbf{H}_j^{s-1} \mathbf{G}(\Gamma_C, j)$

$c_j^s = \mathbf{A}_{\Gamma_C} d_j^s$

$a_j^s = \frac{\langle r_j^s, c_j^s \rangle}{\|c_j^s\|_2^2}$

$w_j^s = \hat{\mathbf{x}}_j^s - \hat{\mathbf{x}}_j^{s-1} = a_j^s d_j^s, t_j^s = \mathbf{A}_{\Gamma_C}^{\mathrm{H}} \mathbf{A}_{\Gamma_C} w_j^s$

$u_j^s = \frac{3}{2} w_j^s - \frac{1}{2} w_j^{s-1}, v_j^s = \frac{3}{2} t_j^s - \frac{1}{2} t_j^{s-1}$

using Equation (19), update $\mathbf{H}_j^s$

$\hat{\mathbf{x}}_{\Gamma_C, j}^s = \hat{\mathbf{x}}_{\Gamma_C, j}^{s-1} + a_j^s d_j^s$

$r_j^s = r_j^{s-1} - a_j^s c_j^s$

end for

D-MUD:

$\Gamma = \Gamma_C, i = 1, r_j^0 = r_j^S, \hat{\mathbf{x}}_j^0 = \hat{\mathbf{x}}_j^S, \mathbf{H}_j^i = \mathbf{H}_j^S$

for $j = 1 : J$ do

while $i = 1$ or $\|r\_revise_j^i\|_2^2 \leq \|r\_revise_j^{i-1}\|_2^2$

$i = i + 1$

$\Gamma^i = \Gamma^{i-1} \cup \arg\max_m |g_{m,j}|^2$, where $g_j = \mathbf{A}^{\mathbf{H}} r_j^{(i-1)}$

$d_j^i = -\mathbf{H}_j^{i-1} g_{\Gamma^i, j}^i$

$c_j^i = \mathbf{A}_{\Gamma^i, j} d_j^i$

$a_j^i = \frac{\langle r_j^i, c_j^i \rangle}{\|c_j^i\|_2^2}$

$w_j^i = \hat{\mathbf{x}}_j^i - \hat{\mathbf{x}}_j^{i-1} = a_j^i d_j^i, t_j^i = \mathbf{A}_{\Gamma^i, j}^{\mathrm{H}} \mathbf{A}_{\Gamma^i, j} w_j^i$

$u_j^i = \frac{3}{2} w_j^i - \frac{1}{2} w_j^{i-1}, v_j^i = \frac{3}{2} t_j^i - \frac{1}{2} t_j^{i-1}$

Compute $\mathbf{H}_j^i$ using Equation (19);

$\hat{\mathbf{x}}_{\Gamma^i, j}^i = \hat{\mathbf{x}}_{\Gamma^i, j}^{i-1} + a_j^i d_j^i$

$r_j^i = r_j^{i-1} - a_j^i c_j^i$

$\tilde{x}_{m,j}^i = \begin{cases} \Lambda & if\ m \in \Gamma^i \\ 0 & otherwise \end{cases}$

$r\_revise_j^i = \mathbf{y}_j - \mathbf{A}\tilde{\mathbf{x}}_j^i$

end while

end for

A. CONVERGENCE ANALYSIS

The GDGP-MUD and MSQN-MUD algorithms, similar to the GP algorithm, should have global convergence to achieve user activity and data detection. Since the revised residual value has been applied to break iteration, we propose theorem 1 to prove the optimality of the algorithms.

Theorem 1: for both algorithms, there exists a constant $c < 1$, which only depends on $\mathbf{A}$, such that the revised residual value decays as

$$\|r\_revise_j^i\|_2^2 \le c\|r\_revise_j^{i-1}\|_2^2 \tag{21}$$

Proof: the revised residual can be formulated as $r\_revise_j^i = \mathbf{y}_j - \mathbf{A}\tilde{\mathbf{x}}_j^i$, and we can obtain

$$r\_revise_j^i - r\_revise_j^{i-1} = \mathbf{A}(\tilde{\mathbf{x}}_j^{i-1} - \tilde{\mathbf{x}}_j^i) \tag{22}$$

After phase shifting, Eq. 22 can be rewritten as

$$r\_revise_j^i = r\_revise_j^{i-1} - \mathbf{A}(\tilde{\mathbf{x}}_j^i - \tilde{\mathbf{x}}_j^{i-1}) \tag{23}$$

We can determine that

$$\begin{aligned}
\|r_revise_j^i\|_2^2 &= \|r_revise_j^{i-1} - \mathbf{A}(\tilde{\mathbf{x}}_j^i - \tilde{\mathbf{x}}_j^{i-1})\|_2^2 \\
&\le \|r_revise_j^{i-1} - \mathbf{A}(\hat{\mathbf{x}}_j^i - \hat{\mathbf{x}}_j^{i-1})\|_2^2 \\
&= \|r_revise_j^{i-1} - \mathbf{A}_{\Gamma^i}(\hat{\mathbf{x}}_{\Gamma^i,j}^i - \hat{\mathbf{x}}_{\Gamma^i,j}^{i-1})\|_2^2 \\
&= \|r_revise_j^{i-1} - a_j^i \mathbf{A}_{\Gamma^i} d_j^i\|_2^2 \\
&= \|r_revise_j^{i-1}\|_2^2 - \frac{\left\langle r_j^{i-1}, \mathbf{A}_{\Gamma^i} d_j^i \right\rangle^2}{\|\mathbf{A}_{\Gamma^i} d_j^i\|_2^2}
\end{aligned} \tag{24}$$

For the GDGP-MUD algorithm, $d_j^i = -g_{\Gamma^i,j}^i = -{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}$, and we can formulate

$$\frac{\left\langle r_j^{i-1}, \mathbf{A}_{\Gamma^i} d_j^i \right\rangle^2}{\|\mathbf{A}_{\Gamma^i} d_j^i\|_2^2} = \frac{\|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^4}{\|\mathbf{A}_{\Gamma^i} {\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^2} \tag{25}$$

For the MSQN-MUD algorithm, $d_j^i = -\mathbf{H}_j^{i-1} g_{\Gamma^i,j}^i = -\mathbf{H}_j^{i-1} {\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}$, and we also can formulate that

$$\begin{aligned}
\frac{\left\langle r_j^{i-1}, \mathbf{A}_{\Gamma^i} d_j^i \right\rangle^2}{\|\mathbf{A}_{\Gamma^i} d_j^i\|_2^2} &= \frac{\|\mathbf{H}_j^{i-1}\|_2^2 \|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^4}{\|\mathbf{H}_j^{i-1}\|_2^2 \|\mathbf{A}_{\Gamma^i} {\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^2} \\
&= \frac{\|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^4}{\|\mathbf{A}_{\Gamma^i} {\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^2}
\end{aligned} \tag{26}$$

Thus, we can have that

$$\begin{aligned}
\frac{\|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^4}{\|\mathbf{A}_{\Gamma^i} {\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^2} &\ge \frac{\|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^4}{\|\mathbf{A}_{\Gamma^i}\|_2^2 \|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^2} \\
&\ge \frac{\|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^2}{\|\mathbf{A}_{\Gamma^i}\|_2^2}
\end{aligned} \tag{27}$$

The residual value can be formulated as $r_j^i = r_j^{i-1} - a_j^i c_j^i = y_j - \mathbf{A}_{\Gamma^i} \hat{x}_{\Gamma^i,j}^i$ so that we can have that $\|r_j^i\|_2^2 \ge \|r\_revise\|_2^2$. Eq. 27 can be formulated as

$$\begin{aligned}
\frac{\|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^4}{\|\mathbf{A}_{\Gamma^i} {\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^2} &\ge \frac{\|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_j^{(i-1)}\|_2^2}{\|\mathbf{A}_{\Gamma^i}\|_2^2} \\
&\ge \frac{\|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r_revise_j^{(i-1)}\|_2^2}{\|\mathbf{A}_{\Gamma^i}\|_2^2}
\end{aligned}$$

$$\ge \frac{\|{\mathbf{A}_{\Gamma^i}}^{\mathrm{H}} r\_revise_j^{(i-1)}\|_\infty^2}{\|\mathbf{A}_{\Gamma^i}\|_2^2} \tag{28}$$

According to the nature of the 2 norm and the infinite norm, for any x, there exists $\omega > 0$, which makes $\|\mathbf{A}^{\mathrm{H}} x\|_\infty^2 \ge \omega \|x\|_\infty^2$. Due to the selection procedure, $\|\mathbf{A}^{\mathrm{H}} x\|_\infty^2 = \|\mathbf{A}_{\Gamma^i}^{\mathrm{H}} x\|_\infty^2$. We see that theorem 1 holds for $c = (1 - (\omega / \|\mathbf{A}\|_2^2))$.

B. COMPUTATIONAL COMPLEXITY ANALYSIS

In this subsection, we analyse the computational complexity of the ADMM algorithm proposed in [20], the SMP algorithm proposed in [21], and our two proposed algorithms. In analysing the complexity, we count floating point operations (Flops) such as multiplication [23].

To elaborate on the complexity analysis process, we use the SMP algorithm as an example for analysis. Since it individually detects common active user information and dynamic active user information, we divide the two parts to count SMP Flops. In the signal detection for common active users, the pseudo-inverse of $\mathbf{A}_{\Gamma_C}$ is required when updating the residual, where $\mathbf{A}_{\Gamma_C}^\dagger = (\mathbf{A}_{\Gamma_C}^{\mathrm{H}} \mathbf{A}_{\Gamma_C})^{-1} \mathbf{A}_{\Gamma_C}^{\mathrm{H}}$. Therefore, the Flops of updating the residual is

$$\begin{aligned}
C_{residual} &= \sum_{s=1}^{S} [\frac{s^3}{3} + (N+1)s^2 + 2Ns]J \\
&\approx (\frac{S^4}{12} + \frac{NS^3}{3})J
\end{aligned} \tag{29}$$

Apart from updating the residual, the correlation between the equivalent channel matrix and the residual is calculated to update common support Γ_C in the signal detection for common active users. Thus, the Flops of this step is

$$\begin{aligned}
C_{\Gamma_C} &= \sum_{s=1}^{S} (MN + M)J \\
&\approx MNSJ
\end{aligned} \tag{30}$$

In the signal detection for dynamic active users, SMP Flops is

$$\begin{aligned}
&C_{SMP,d} \\
&= \sum_{i=1}^{\bar{I}} [(\frac{2S^3}{3} + 2(N+1)S^3 + 3NS + M(N+2) + 2N] \\
&\approx (\frac{2S^3 \bar{I}}{3} + 2NS^2 \bar{I} + MN\bar{I})J
\end{aligned} \tag{31}$$

$\bar{I}$ is the average iteration number per slot. Combining (29), (30) and (31), we formulate SMP Flops as

$$C_{SMP} = [\frac{S^4}{12} + \frac{(N + 2\bar{I})S^3}{3} + 2NS^2\bar{I} + MN(S + \bar{I})]J \tag{32}$$

As mentioned in Section 3, as in the SMP, our proposed algorithms can also be divided into two parts. However, the difference is that SMP relies on the exact active user number to terminate the common support update, and our

algorithms rely on iteration termination parameter S, where it is not necessary to be precise to terminate the updating common support update. Additionally, the matrix or vector dimension of the SMP algorithm does not extend again in the dynamic support detection, while the matrix or vector dimension in our algorithms is expanded with iteration number increase.

We now analyse the computational complexity of our proposed algorithms. In C-MUD, the Flops of GDGP-MUD algorithm and MSQN-MUD algorithm are

$$\begin{aligned} C_{GDGP-MUD,c} &= \sum_{s=1}^{S}(MN+M+3N+Ns+s)J \\ &\approx (\frac{NS^2}{2}+MNS)J \end{aligned} \tag{33}$$

$$\begin{aligned} C_{MSQN-MUD,c} &= \sum_{s=1}^{S}[MN+(N+7)s^2+(N+4)s+3N+M]J \\ &\approx (\frac{NS^3}{3}+\frac{NS^2}{2}+MNS)J \end{aligned} \tag{34}$$

Both algorithms require updating the support set, reconstructing the sparse signal, and computing the step size and residual: those operations consume $MN + M$, s and $Ns + 2N$and s, respectively. The difference between the GDGP-MUD algorithm and MSQN-MUD algorithm is that the former obtains a new update direction via the gradient descent method, while the latter obtains the update direction via a multi-step quasi-Newton method. For the GDGP-MUD algorithm, it is not necessary to count new update direction Flops in each iteration since the new update direction is equal to the gradient information we compute in the former. Apart from all required Flops, the MSQN-MUD algorithm requires an additional $(N + 7)s^2 + (N + 3)s$ Flops in C-MUD.

Since the D-MUD part is similar to the C-MUD part, we will not repeat the analysis. In D-MUD, the GDGP-MUD algorithm Flops is

$$\begin{aligned} C_{GDGP-MUD,d} &= \sum_{i=1}^{\bar{I}}[2MN+M+5N+(N+1)(S+i)]J \\ &\approx (\frac{N\bar{I}^2}{2}+NS\bar{I}+2MN\bar{I})J \end{aligned} \tag{35}$$

Moreover, the MSQN-MUD algorithm Flops in D-MUD is

$$\begin{aligned} C_{MSQN-MUD,d} &= \sum_{i=1}^{\bar{I}}[2MN+(N+7)(S+i)^2 \\ &\quad +(N+4)(S+i)+5N+M]J \\ &\approx [2MN\bar{I}+\frac{N(S+\bar{I})^3}{3}+\frac{N(S+\bar{I})^2}{2}-\frac{NS^3}{3} \\ &\quad -\frac{NS^2}{2}]J \end{aligned} \tag{36}$$

TABLE 1. Computational complexity comparison of different algorithms.

Algorithm	Computation Complexity
ADMM	$(\frac{M^3\bar{I}}{3}+M^2N\bar{I})J$
SMP	$[\frac{S^4}{12}+\frac{(N+2\bar{I})S^3}{3}+2NS^2\bar{I}+MN(S+\bar{I})]J$
GDGP-MUD	$[\frac{N(S+\bar{I})^2}{2}+MN(S+2\bar{I})]J$
MSQN-MUD	$[\frac{N(S+\bar{I})^3}{3}+\frac{N(S+\bar{I})^2}{2}+MN(S+2\bar{I})]J$

Combining (33) and (35), we can formulate the Flops of GDGP-MUD as

$$C_{GDGP-MUD} \approx [\frac{N(S+\bar{I})^2}{2}+MN(S+2\bar{I})]J \tag{37}$$

Combining (34) and (36), we can formulate the Flops of MSQN-MUD as

$$C_{MSQN-MUD} \approx [\frac{N(S+\bar{I})^3}{3}+\frac{N(S+\bar{I})^2}{2}+MN(S+2\bar{I})]J \tag{38}$$

For the ADMM algorithm, the Flops of all slots are

$$C_{ADMM} \approx (\frac{M^3\bar{I}}{3}+M^2N\bar{I})J \tag{39}$$

Table 1 specifically shows the computational complexity of different algorithms. We observe that algorithm computational complexity depends on several parameters such as M, N, J, S and $\bar{I}$. When M, N, and J remain constant, the computation complexity of the algorithm is only related to S and the number of average iterations $\bar{I}$.

V. SIMULATION

In this section, we verify that our algorithms exhibit performance strength in uplink grant-free NOMA systems. We use pseudo-random noise as the spreading matrix so that restricted isometry property (RIP) can be satisfied with high probability. Additionally, quaternary phase shift keying (QPSK) is considered. As analysed in Section IV, when M, N, and S are determined, the algorithm Flops is only related to the number of iterations. Since ADMM [18] calculates all user information (including inactive users), i.e., consumes $(\frac{M^3\bar{I}}{3}+M^2N\bar{I})J$ multiplication operations, where $M^3 >> NS^2$, the Flops of ADMM is far greater than those of other algorithms. Fig. 1 shows the number of iterations for three algorithms to complete 1,000 experiments in different SNR conditions, where $M = 200$ and $N = 100$, and there are 15 common active users and 5 dynamic active users. For our proposed algorithm, we select S = 15 as the iteration termination parameter in the C-MUD part to terminate common support set searching. Note that the selection of S exerts a minimal effect on performance, which we analyse in later simulations.

In these conditions, we calculated the average iteration number per time slot for SMP [19], GDGP-MUD and MSQN-MUD. Specific values are shown in Table 2. All parameters are applicable for simulations of Fig. 2, Fig. 3 and Fig. 4. Fig. 1 and Table 2 indicate that our proposed algorithms have faster convergence speed.

Fig. 2 investigates the Flops of different algorithms. From Fig. 2, we observe that the GDGP-MUD algorithm has a minimum Flops in comparison with the other three algorithms. The MSQN-MUD algorithm Flops is obviously lower than the SMP [19] algorithm while slightly greater than GDGP-MUD. This is because the MSQN-MUD algorithm sacrifices some complexity for performance improvements.

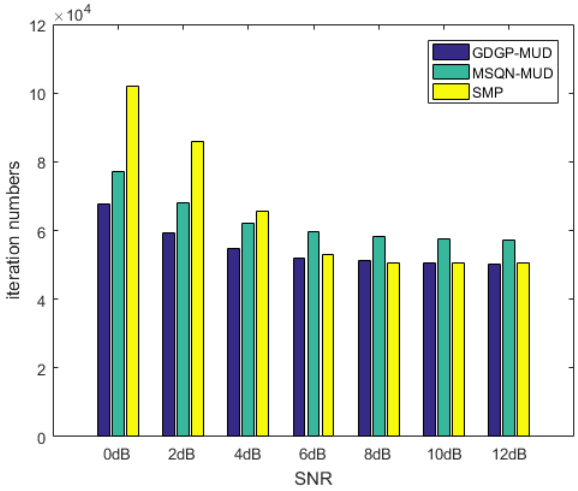


FIGURE 1. Iteration number for different algorithms to complete 1,000 experiments.

TABLE 2. Average iteration number per time slot.

Algorithm \ SNR(dB)	0	2	4	6	8	10	12
GDGP-MUD	10	9	8	7	7	7	7
MSQN-MUD	11	10	9	9	8	8	8
SMP	15	12	9	8	7	7	7

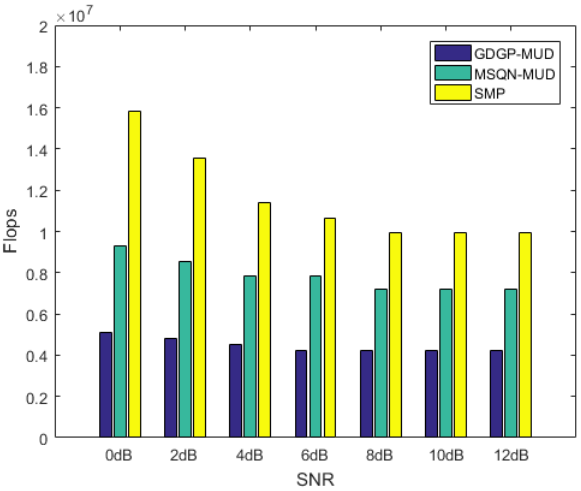


FIGURE 2. Flops comparison.

Fig. 3 depicts the time consumption of different algorithms in completing 1,000 experiments. We observe that the GDGP-MUD algorithm completes 1,000 experiments with minimal time consumption, while the ADMM [18] algorithm contributes to maximum time consumption. This phenomenon also indicates that one of our proposed algorithms, GDGP-MUD, has a low computation complexity. For the MSQN-MUD algorithm, the time consumption is greater than the GDGP-MUD algorithm. Fortunately, this gap remains within an acceptable range.

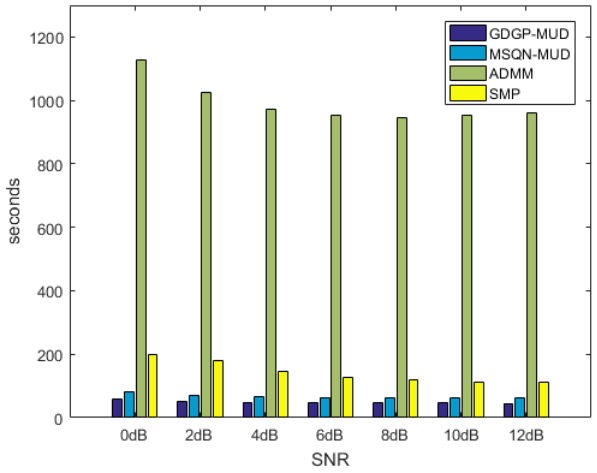


FIGURE 3. Time consumption comparison for 1,000 experiments.

Fig. 4 compares the SER of two proposed algorithms and three algorithms proposed in the former literature. We observe that the detection performance of our algorithm not only outperforms ADMM [18] and OMP [27] but also approaches the detection accuracy of SMP [19]. Combining Fig. 2 and Fig. 3, we conclude that the GDGP-MUD algorithm and MSQN-MUD could detect data with low computational complexity. Especially in $SNR < 6dB$ conditions, the GDGP-MUD performance outperforms the dynamic MUD algorithm SMP, but SMP exhibits superior detection in the $SNR > 8dB$ condition. From the perspective of the MSQN-MUD algorithm, it maintains a detection performance advantage under small SNR conditions in comparison to SMP and approaches SMP detection performance under large SNR conditions.

Fig. 5 shows the SER comparison with only sparsity changed. Sparsity is defined as $q = K/M$. We set constants $M = 200$, $N = 100$, $J = 7$, and $SNR = 8dB$, and the active user number remains 5. From Fig. 5 we observe that three dynamic algorithms always exhibit strength in detection as compared with OMP and ADMM. We also observe that SMP performs better with low sparsity, but the detection accuracy of our proposed algorithm approach to SMP even exceeds SMP as the active user numbers increase.

Fig. 6 and Fig. 7 show the detection performance with different iteration termination parameter S for two proposed algorithms in different SNR conditions. We clearly observe that when S differs, our proposed algorithms exhibit similar detection performance. Especially in a low SNR condition ($SNR < 6dB$), multiple performance curves are coupled into

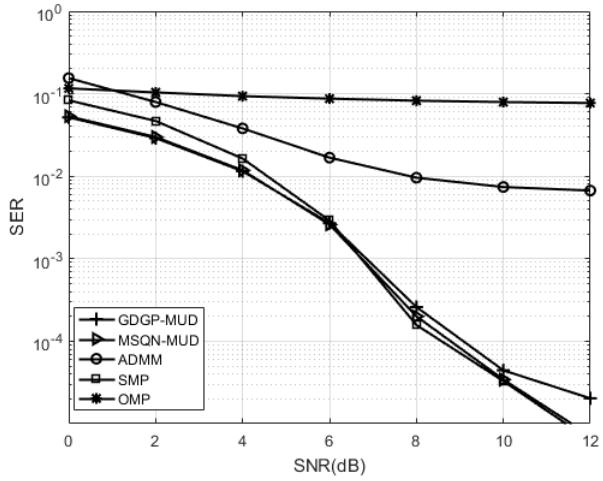


FIGURE 4. SER comparison of different SNR conditions.

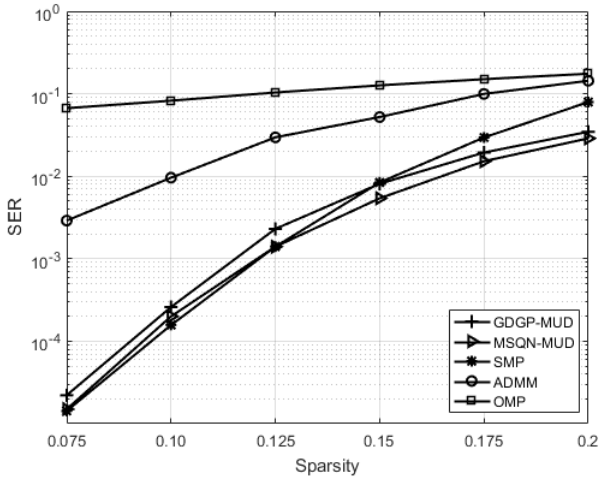


FIGURE 5. SER comparison with only sparsity changed.

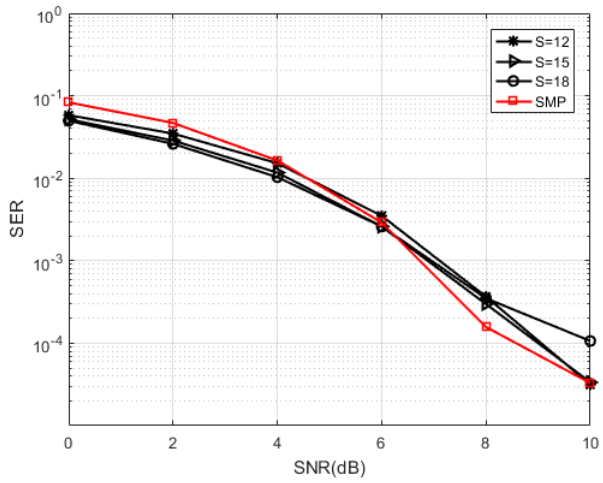


FIGURE 6. SER comparison of SMP algorithm and GDGP-MUD with different iteration termination parameter S.

one curve. When $SNR > 8dB$, using different S caused a slight deviation in detection performance, although the slight fluctuation is within an acceptable range.

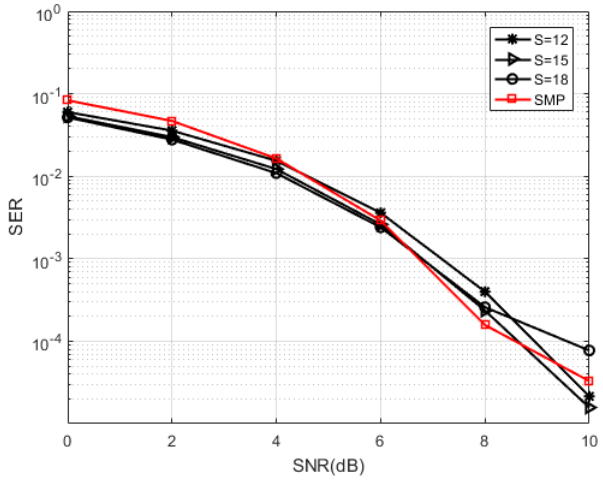
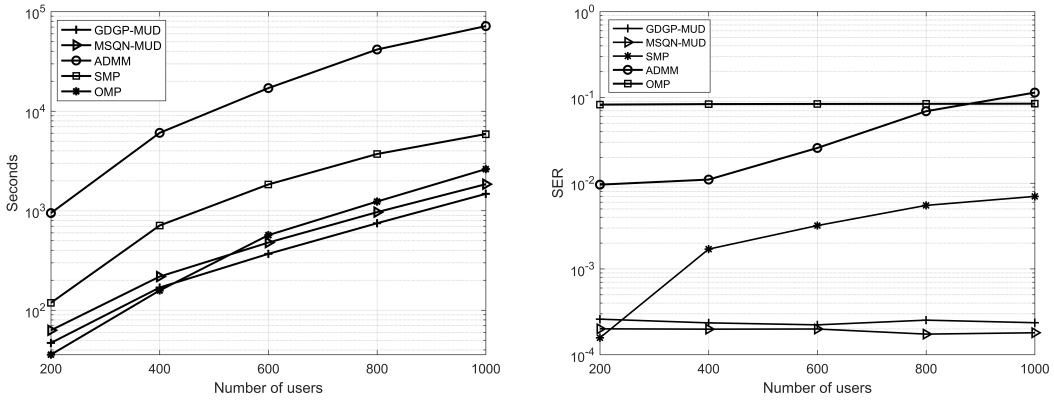


FIGURE 7. SER comparison of SMP algorithm and MSQN-MUD with different iteration termination parameter S.

Fig. 8 shows the performance comparisons for each algorithm with the total number of users increased when $SNR = 8dB$, the number of experiments of $1,000$, and the overload factor and sparsity remaining unchanged. Fig. 8(a) and Fig. 8(b), respectively, reflect the real-time performance and SER performance comparison of each algorithm with the total number of users increased. From Fig. 8(a), we can see that when the sparsity and overload rate are fixed, the ADMM [18] algorithm consumes much more time to complete $1,000$ experiments than do the greedy iterative algorithms, and as the total number of users increases, the gap between them exhibits a gradually increasing trend. From Fig. 8(b), we can find that the SER of the ADMM [18] and SMP [19] will increase with increasing total user number, and the two algorithms we proposed can always maintain a stable state, even when the total number of users is changed, which also shows that our proposed algorithms can be applied to the massive communication device scenario.

VI. CONCLUSION

In this paper, we first propose a low-complexity algorithm, the GDGP-MUD algorithm, which is based on a gradient pursuit framework. The process of matrix inversion in conventional greedy MUD algorithms is replaced by gradient projection, where the GDGP-MUD algorithm jointly detects user activity and data via the gradient information of the objective function to update the optimization direction in each iteration. To speed up convergence, we adopt a modified residual value to determine whether to terminate the loop. Additionally, we propose another low-complexity MUD algorithm based on the GDGP-MUD algorithm, the MSQN-MUD algorithm, which uses a multi-step quasi-Newton method to update the optimization direction. The simulation results show us that both proposed algorithms possess a lower computational complexity compared to the conventional greedy MUD algorithms. With the SNR increase, our



(a) Real-time comparison of different user numbers (b) SER comparison of different user numbers

FIGURE 8. Performance comparisons of different user numbers.

proposed algorithms outperform conventional greedy MUD algorithms with respect to the real-time characteristic. The MSQN-MUD algorithm sacrifices a small amount of complexity for an increase in detection accuracy.

REFERENCES

[1] W. V. S. Wong, R. Schobe, D. W. K. Ng, and C. Li, *Key Technologies for 5G Wireless Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[2] L. Dai, B. Wang, Y. Yuan, S. Han, C.-L. I, and Z. Wang, "Non-orthogonal multiple access for 5G: Solutions, challenges, opportunities, and future research trends," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 74–81, Sep. 2015.

[3] Z. Wei, L. Yang, D. W. K. Ng, J. Yuan, and L. Hanzo, "On the performance gain of NOMA over OMA in uplink communication systems," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 536–568, Jan. 2020.

[4] M. Mohammadkarimi, M. A. Raza, and O. A. Dobre, "Signature-based nonorthogonal massive multiple access for future wireless networks: Uplink massive connectivity for machine-type communications," *IEEE Veh. Technol. Mag.*, vol. 13, no. 4, pp. 40–50, Dec. 2018.

[5] S. Feizi and M. Medard, "A power efficient sensing/communication scheme: Joint source-channel-network coding by using compressive sensing," in *Proc. 49th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2011, pp. 1048–1054.

[6] M.-H. Ran, J.-G. Huang, and H.-J. Fu, "Sparse channel estimation based on compressive sensing for OFDM underwater acoustic communication," *Syst. Eng. Electron.*, vol. 33, no. 5, pp. 1157–1161, 2011.

[7] Z. Chen, F. Sohrabi, and W. Yu, "Sparse activity detection for massive connectivity," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1890–1904, Apr. 2018.

[8] V. Havary-Nassab, S. Hassan, and S. Valaee, "Compressive detection for wide-band spectrum sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2010, pp. 3094–3097.

[9] S. Sharma, A. Gupta, and V. Bhatia, "A new sparse signal-matched measurement matrix for compressive sensing in UWB communication," *IEEE Access*, vol. 4, pp. 5327–5342, 2016.

[10] S. Liu, F. Yang, C. Zhang, and J. Song, "Compressive sensing based narrowband interference cancellation for power line communication systems," in *Proc. Globecom*, Dec. 2014, pp. 2989–2994.

[11] F. Monsees, M. Woltering, C. Bockelmann, and A. Dekorsy, "Compressive sensing multi-user detection for multicarrier systems in sporadic machine type communication," in *Proc. IEEE Veh. Technol. Conf.*, May 2015, pp. 1–5.

[12] C. Bockelmann, H. F. Schepker, and A. Dekorsy, "Compressive sensing based multi-user detection for machine-to-machine communication," *Trans. Emerg. Telecommun. Technol.*, vol. 24, no. 4, pp. 389–400, Jun. 2013.

[13] Z. Gao, L. Dai, Z. Wang, S. Chen, and L. Hanzo, "Compressive-sensing-based multiuser detector for the large-scale SM-MIMO uplink," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8725–8730, Oct. 2016.

[14] Y. Ji, C. Bockelmann, and A. Dekorsy, "Numerical analysis for joint PHY and MAC perspective of compressive sensing multi-user detection with coded random access," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2017, pp. 1018–1023.

[15] B. Wang, L. Dai, Y. Zhang, T. Mir, and J. Li, "Dynamic compressive sensing-based multi-user detection for uplink grant-free NOMA," *IEEE Commun. Lett.*, vol. 20, no. 11, pp. 2320–2323, Nov. 2016.

[16] Y. Du, B. Dong, Z. Chen, X. Wang, Z. Liu, P. Gao, and S. Li, "Efficient multi-user detection for uplink grant-free NOMA: Prior-information aided adaptive compressive sensing perspective," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 12, pp. 2812–2828, Dec. 2017.

[17] A. T. Abebe and C. G. Kang, "Iterative order recursive least square estimation for exploiting frame-wise sparsity in compressive sensing-based MTC," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1018–1021, May 2016.

[18] A. C. Cirik, N. Mysore Balasubramanya, and L. Lampe, "Multi-user detection using ADMM-based compressive sensing for uplink grant-free NOMA," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 46–49, Feb. 2018.

[19] B. Wang, T. Mir, R. Jiao, and L. Dai, "Dynamic multi-user detection based on structured compressive sensing for IoT-oriented 5G systems," in *Proc. URSI Asia–Pacific Radio Sci. Conf. (URSI AP-RASC)*, Aug. 2016, pp. 431–434.

[20] Y. Du, C. Cheng, B. Dong, Z. Chen, X. Wang, J. Fang, and S. Li, "Block-sparsity-based multiuser detection for uplink grant-free NOMA," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 7894–7909, Dec. 2018.

[21] B. K. Jeong, B. Shim, and K. B. Lee, "MAP-based active user and data detection for massive machine-type communications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8481–8494, Sep. 2018.

[22] D. L. Donoho and J. Tanner, "Sparse nonnegative solution of underdetermined linear equations by linear programming," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 27, pp. 9446–9451, Jul. 2005.

[23] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math., J. Issued Courant Inst. Math. Sci.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[24] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[25] B. Wang, L. Dai, T. Mir, and Z. Wang, "Joint user activity and data detection based on structured compressive sensing for NOMA," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1473–1476, Jul. 2016.

[26] J. W. Choi, B. Shim, Y. Ding, B. Rao, and D. I. Kim, "Compressed sensing for wireless communications: Useful tips and tricks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1527–1550, 3rd Quart., 2017.

[27] B. Shim and B. Song, "Multiuser detection via compressive sensing," *IEEE Commun. Lett.*, vol. 16, no. 7, pp. 972–974, Jul. 2012.

[28] J. Wang, S. Kwon, and B. Shim, "Generalized orthogonal matching pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6202–6216, Dec. 2012.

[29] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, May 2009.

[30] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.
[31] T. Blumensath and M. E. Davies, "Gradient pursuits," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2370–2382, Jun. 2008.
[32] Y. Hu, L. Cheng, F. Jiang, and R. Wang, "A gradient pursuit algorithm based on multi-step quasi-Newton method," in *Proc. IEEE 4th Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2019, pp. 559–565.

FANG JIANG received the Ph.D. degree in electronic science and technology from Anhui University, in 2016. She is currently a Lecturer with the School of Internet, Anhui University. Her research interests include wireless communications, compressive sensing, and wireless sensor networks.

GUOLIANG ZHENG received the B.Eng. degree in electronic information engineering from the Taiyuan Institute of Technology, in 2018. He is currently pursuing the MA.Eng. degree with the School of Electronics and Information Engineering, Anhui University. His research interests include compressive sensing and multi-user detection.

YANJUN HU (Senior Member, IEEE) received the B.S. degree in electronic engineering and the M.S. degree in circuit and system from Anhui University, in 1989 and 1992, respectively, and the Ph.D. degree in communication and information system from the University of Science and Technology of China (USTC), in June 2001. Since 1992, she has been teaching with the Department of Electrical Engineering and Information Science, Anhui University. From 2002 to 2003, she was a Postdoctoral Fellow with the School of Information Technology, USTC. From February 2005 to August 2006, and from September to December 2013, she was working as a Visiting Research Scientist with the PARADISE Research Laboratory, School of Information Technology and Engineering, University of Ottawa, Canada. She is currently a Full Professor with the School of Electronic Information Engineering, Anhui University. Her research interests include wireless and mobile networking, wireless sensors networks, and wireless multimedia.

YI WANG has long been engaged in research and teaching in the fields of mobile communication and wireless sensor networks. He has published more than ten papers on heterogeneous networks, edge computing, collaborative transmission, and cognitive radio. He has applied for multiple invention patents. His research interests include mobile communications and wireless sensor networks.

YAOHUA XU received the MA.Eng. degree in circuit and system from Anhui University, in 2004. He is currently an Associate Professor with the School of Electronic Information Engineering, Anhui University. His research interests include wireless communication signal processing and the industrial Internet of Things.