Faster Activity and Data Detection in Massive Random Access: A Multi-armed Bandit Approach

Jialin Dong, *Student Member*, *IEEE*, Jun Zhang, *Senior Member*, *IEEE*, Yuanming Shi, *Member*, *IEEE*, and Jessie Hui Wang

***Abstract*—This paper investigates the grant-free random access mechanism for massive Internet of Things (IoT) devices. By embedding the data symbols in the signature sequences, joint device activity detection and data decoding can be achieved, which, however, significantly increases the computational complexity. Coordinate descent algorithms that enjoy a low per-iteration complexity have been employed to solve this detection problem, but previous works typically employ a random coordinate selection policy which leads to slow convergence. In this paper, we develop multi-armed bandit approaches for more efficient detection via coordinate descent, which achieves a delicate trade-off between *exploration* and *exploitation* in coordinate selection. Specifically, we first propose a bandit based strategy, i.e., Bernoulli sampling, to speed up the convergence rate of coordinate descent, by learning which coordinates will result in more aggressive descent of the *nonconvex objective function*. To further improve the convergence rate, an inner multi-armed bandit problem is established to learn the exploration policy of Bernoulli sampling. Both convergence rate analysis and simulation results are provided to show that the proposed bandit based algorithms enjoy faster convergence rates with a lower time complexity compared with the state-of-the-art algorithm. Furthermore, our proposed algorithms are generally applicable to different scenarios, e.g., massive random access with low-precision analog-to-digital converters (ADCs).**

***Index Terms*—Massive connectivity, Internet of Things, coordinate descent, multi-armed bandit, Thompson sampling.**

I. Introduction

The advancements in wireless technologies have enabled ubiquitous connections of sensors, mobile devices, and machines for various mobile applications, leading to an era of Internet-of-Things (IoT) [2]. IoT connectivity involves connecting a massive number of devices, which form the foundation for many applications, e.g., smart home, smart city, healthcare, transportation system, etc. Thus it has been regarded as an indispensable demand for future wireless networks [3]. With a large number of devices to be connected with the base station (BS), in the order of 10^4 to 10^6, massive connectivity brings formidable technical challenges, and has attracted lots of attentions from both the academia and industry [4], [5].

The sporadic traffic is one unique feature in massive IoT connectivity, which means that only a restricted portion of devices are active at any given time instant [6]. This is because IoT devices are often designed to sleep for most of the time to save energy and are activated only when triggered by external events. Therefore, the BS needs to manage the massive random access via detecting the active users before decoding their data. The traditional grant-based random access scheme has been widely applied to allow multiple users to access the network over limited radio resources, e.g., in 4G LTE networks [5]-[7]. Under this scheme, each active device is randomly assigned a pilot sequence from a pre-defined set of preamble sequences to notify the BS of the device's activity state. A connection between an active device and the BS will be established if the pilot sequence of this device is not occupied by any other device. Besides the overhead caused by the pilot sequence, a major drawback of the grant-based random access scheme is the collision issue due to a massive number of requests [6].

To avoid the excessive access latency due to the collision, a grant-free random access scheme has recently been proposed [6]. Under this scheme, the active devices do not need to wait for any grant to access the network, and can directly transmit the payload data following the metadata. At the receiver, following pilot-assisted activity detection and channel estimation, payload data of the active devices can be decoded. The key task of activity detection under the sporadic pattern relies on compressed sensing techniques [8], [9], which have led to a number of effective methods for massive access [10]. Specifically, compared with the grant-based access scheme [5], the grant-free random access paradigm [6] enjoys a much lower access latency. In the scenario where the payload data only contains a few bits, e.g., sending an alarm signal, the efficiency can be further improved by embedding the data symbols in the signature sequences [11], [12]. Some existing works on sparse signal estimation for single measurement vector models [13] and jointly sparse signal estimation [14] focused on estimating a sparse signal from a given measurement. Recently, deep learning based methods have also been proposed to recover jointly sparse signals [15]. Considering the massive devices and massive BS antennas, the high-dimensional detection problems considered in these works bring formidable computational challenges, which motivates our investigation.

A. Related Works

We consider the grant-free massive random access scheme in a network consisting of one multi-antenna BS and a

Part of this work was presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2020 [1]. This work was supported by the General Research Fund (Project No. 15207220) from the Hong Kong Research Grants Council.

J. Dong is with the Department of Electrical and Computer Engineering, University of California, Los Angeles, USA (e-mail: jialind@g.ucla.edu).

J. Zhang is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: eejzhang@ust.hk).

Y. Shi is with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: shiym@shanghaitech.edu.cn).

J. H. Wang is with the Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University, Beijing 100084, China (e-mail: jessiewang@tsinghua.edu.cn). (The corresponding author is J. H. Wang.)

massive number of devices with small payload data, where each message is assigned a unique signature sequence. By exploiting the sparsity structure in both the device activity state and data transmission, joint device activity detection and data decoding can be achieved by leveraging compressed sensing techniques [16], [8]. Recently, a covariance-based method has been proposed to improve the performance of device activity detection [17], where the detection problem is solved by a coordinate descent algorithm with random sampling, i.e., it randomly selects a coordinate-wise iterate to update. This covariance-based method has also been applied for joint detection and data decoding [12]. Furthermore, the phase transition analysis for covariance-based massive random access with massive Multiple-input and Multiple-output (MIMO) has been provided in [18].

Although coordinate descent is an effective algorithm to solve the maximum likelihood estimation problem for joint activity detection and data decoding [12], existing works adopted a random coordinate selection strategy, which yields a slow convergence rate. Besides, a rigorous convergence rate analysis for this strategy has not yet been obtained. In this paper, our principle goal is to develop coordinate descent algorithms with more effective coordinate selection strategies for *faster* activity and data detection in massive random access, supported by rigorous convergence rate analysis.

Coordinate descent algorithms [19] with various coordinate selection strategies have been widely applied to solve optimization problems for which computing the gradient of the objective function is computationally prohibitive. It enjoys a low per-iteration complexity, as only one or a few coordinates are updated in each iteration. In most previous works, e.g., [20], [21], each coordinate is selected uniformly at random at each time step. Recent studies have proposed more effective coordinate selection strategies via exploiting the structure of the data and sampling the coordinates from an appropriate non-uniform distribution, e.g., [22]-[23], which have been shown to outperform the random sampling strategy in the convergence rate.

Specifically, a convex optimization problem that minimizes a strongly convex objective function was considered in [22]. It proposed a GaussSouthwell-Lipschitz rule that gives a faster convergence rate than choosing random coordinates. Subsequently, Perekrestenko *et al.* [24] improved the convergence rate of the basic coordinate descent algorithm by an adaptive scheme on general convex objectives. Additionally, Zhao and Zhang [25] developed an importance sampling rule where the sample distribution depends on the Lipschitz constants of the loss functions. The adaptive sampling strategies in [24], [25] require the full information of all the coordinates, which yield high computation complexity at each step. However, the strategies proposed in [24], [25] are inapplicable to problems with nonconvex objective functions. To reduce the computation complexity, a recent study [23] exploited a bandit algorithm to learn a good approximation of the reward function, which characterizes how much the cost function decreases when the corresponding coordinate is updated. The coordinate descent algorithms proposed in all the works mentioned above are to solve *convex* optimization problems. Different from these works, the covariance-based estimation problem of massive random access is *non-convex*. Hence, efficient algorithms with new reward functions and corresponding theoretical analysis are required, which brings unique challenges. Moreover, state-of-the-art covariance-based methods sample the coordinate uniformly at random, which suffers from slow convergence rate. To improve the convergence rates, we aim to learn the optimal choice of coordinate by using the feedback information during the iterations. Multi-armed bandit problems have become an efficient tool to learn the optimal choice [26], [27], [28], [29]. Moreover, it can be implemented with low computational cost under rigorous theoretical guarantee. Hence, we focus on establishing a multi-armed bandit problem to learn how to choose the coordinate.

B. Contributions

In this paper, we propose coordinate descent algorithms with effective coordinate sampling strategies for faster activity and data detection in massive random access. Specifically, we develop a novel algorithm, i.e., *coordinate descent with Bernoulli sampling*. Inspired by [23], we cast the coordinate selection procedure as a multi-armed bandit (MAB) problem where a reward is received when selecting an arm (i.e., a coordinate), and we aim to maximize the cumulative rewards over iterations. At each iteration, with probability ε the coordinate with the largest reward is selected, and otherwise the coordinate is chosen uniformly at random. We provide the convergence rate analysis on the coordinate descent with both Bernoulli sampling and random sampling in Theorem 1, which theoretically validates the advantages of the proposed algorithm. Different from the prior work [23] which focuses on smooth and strongly convex objective functions, we deal with a nonconvex objective function. To enhance the effectiveness of Bernoulli sampling, we further develop a coordinate descent algorithm with Thompson sampling to improve the convergence rate. Different from the prior work [26] whose theoretical analysis requires integer parameters of the beta distribution in Thompson sampling, we provide theoretical analysis with more general continuous parameters.

Simulation results show that the proposed algorithms enjoy faster convergence rates with lower time complexity than the state-of-the-art algorithm. It is also demonstrated that coordinate descent with Thompson sampling enables to further improve the convergence rate compared to coordinate descent with Bernoulli sampling. Furthermore, we show that the proposed algorithm can be applied to faster activity and data detection in more general scenarios, i.e., with low precision (e.g., 1 – 4 bits) analog-to-digital converters (ADCs).

Some notations used in this paper are introduced in the sequel. The operator $||\cdot||_0$ denotes the ℓ_0 norm, where $\det(\cdot)$ and $\mathrm{Tr}(\cdot)$ are operators that return the determinant and the trace of a matrix, respectively.

II. System model and problem formulation

In this section, we introduce the system model for massive random access, a.k.a. massive connectivity. A covariance-based formulation is then presented for joint device activity detection and data decoding, together with a coordinate descent algorithm with random sampling as a baseline algorithm.

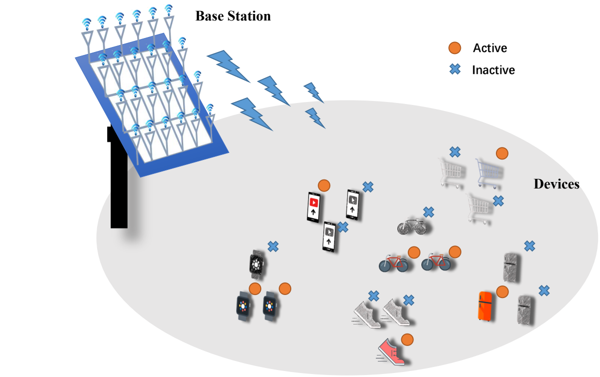

Fig. 1. System model of massive random access.

A. System Model

Consider an IoT network consisting of one BS equipped with M antennas and N single-antenna IoT devices. In the random access scheme, the orthogonal signature sequences are randomly assigned to massive devices. A connection between an active device and the BS will be established if the orthogonal signature sequence assigned to the active device is not occupied by other devices. The channel state vector from device i to the BS is denoted by

$$g_i \boldsymbol{h}_i \in \mathbb{C}^M, \quad i = 1, \ldots, N, \tag{1}$$

where g_i is the pathloss component depending on the device location, and $\boldsymbol{h}_i \in \mathbb{C}^M$ is the Rayleigh fading component over multiple antennas that obeys i.i.d. standard complex Gaussian distribution, i.e., $\boldsymbol{h}_i \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{I})$. Due to the sporadic communications, only a few devices are active out of all devices at a given time instant [30], which is illustrated in Fig. 1. For each active device, J bits of data are transmitted, where J is typically a small number [31][1]. This is the case for many applications, e.g., sending an alarm signal requires only 1 bit. Our goal is to achieve the joint device activity detection and data detection.

Assume the channel coherence block endows with length T_c. The length of the signature sequences L ($L < T_c$) is generally much smaller than the number of devices, i.e., $L \ll N$, due to the massive number of devices and a limited channel coherence block [12], [30]. We first define a unique signature sequence set for the N devices. For each device, we assign a unique sequence to each J-bit message. With $R := 2^J$, this sequence set is known at the BS: $\boldsymbol{Q} = [\boldsymbol{Q}_1 \ \cdots \ \boldsymbol{Q}_N] \in \mathbb{C}^{L \times NR}$, where $\boldsymbol{Q}_i = [\boldsymbol{q}_i^1, \cdots, \boldsymbol{q}_i^R] \in \mathbb{C}^{L \times R}$ with $\boldsymbol{q}_i^r = [q_i^r(1), \cdots, q_i^r(L)]^\top \in \mathbb{C}^L$ for $i = 1, \cdots, N, r = 1, \cdots, R$. We assume that all the signature sequences are generated from i.i.d. standard complex Gaussian distribution. If it is active and aims to send a certain data of J bits, the i-th device will transmit the corresponding sequence from $\boldsymbol{Q}_i$. Specifically, the indicator a_i^r that implies whether the r-th sequence of i-th device is transmitted is defined as follows: $a_i^r = 1$ if the i-th device transmits the r-th sequence; otherwise, $a_i^r = 0$. By detecting which sequences are transmitted based on the received signal, i.e., estimating $\{a_i^r\}$, the BS achieves joint activity detection and data decoding. In this way, the information bits are embedded in the transmitted sequence, and no extra payload data need to be transmitted, which is very efficient for transmitting a small number of bits [11]. Since at most one sequence is transmitted by each device, it holds that $\sum_{r=1}^R a_i^r \in \{0, 1\}$, where $\sum_{r=1}^R a_i^r = 0$ indicates that device i is inactive; otherwise, it is active. Recall that $\boldsymbol{q}_i^r = [q_i^r(1), \cdots, q_i^r(L)]^\top \in \mathbb{C}^L$ for $i = 1, \cdots, N, r = 1, \cdots, R$, and the received signal $\boldsymbol{y}(\ell) \in \mathbb{C}^M$ at the BS is represented as

$$\boldsymbol{y}(\ell) = \sum_{i=1}^{N} \sum_{r=1}^{R} \boldsymbol{h}_i a_i^r q_i^r(\ell) + \boldsymbol{n}(\ell), \tag{2}$$

where $\boldsymbol{n}(\ell) \in \mathbb{C}^M$ is the additive noise such that $\boldsymbol{n}(\ell) \sim \mathcal{CN}(\boldsymbol{0}, \sigma_n^2 \boldsymbol{I})$ for all $\ell = 1, \ldots, L$.

Compact the received signal over M antennas as $\boldsymbol{Y} = [\boldsymbol{y}(1), \ldots, \boldsymbol{y}(L)]^\top \in \mathbb{C}^{L \times M}$, and the additive noise signal over M antennas as $\boldsymbol{N} = [\boldsymbol{n}(1), \ldots, \boldsymbol{n}(L)] \in \mathbb{C}^{L \times M}$. The channel matrix is concatenated as $\boldsymbol{H} = [\boldsymbol{H}_1, \ldots, \boldsymbol{H}_N]^\top \in \mathbb{C}^{NR \times M}$ with $\boldsymbol{H}_i = [\boldsymbol{h}_i, \cdots, \boldsymbol{h}_i]^\top \in \mathbb{C}^{R \times M}$ consisting of repeated rows for $n = 1, \cdots, N$. Recall the signature sequences, and then the model (2) can be reformulated as [12]:

$$\boldsymbol{Y} = \boldsymbol{Q} \boldsymbol{\Gamma}^{\frac{1}{2}} \boldsymbol{H} + \boldsymbol{N}, \tag{3}$$

where the diagonal block matrix is $\boldsymbol{\Gamma}^{\frac{1}{2}} \triangleq \operatorname{diag}(\boldsymbol{D}_1, \ldots, \boldsymbol{D}_N) \in \mathbb{C}^{NR \times NR}$ with $\boldsymbol{D}_i = \operatorname{diag}(a_i^1 g_i, \ldots, a_i^R g_i) \in \mathbb{C}^{R \times R}$ being the diagonal activity matrix with fading components of the i-th device. Let $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1^\top, \cdots, \boldsymbol{\gamma}_N^\top]^\top \in \mathbb{C}^{NR}$ denote the diagonal entries of $\boldsymbol{\Gamma}$, where $\boldsymbol{\gamma}_i = [(a_i^1 g_i)^2, \ldots, (a_i^R g_i)^2]^\top \in \mathbb{C}^R$ for $i = 1, \cdots, N$. Our goal is to detect the values of indicators (i.e., $\{a_i^r\}$) from the received matrix $\boldsymbol{Y}$ with the knowledge of the pre-defined sequence matrix $\boldsymbol{Q}$.

B. Problem Analysis

To achieve this goal, recent works have developed a compressed sensing based approach [16], [32], [33] which recovers $\boldsymbol{\Gamma}^{\frac{1}{2}} \boldsymbol{H}$ from $\boldsymbol{Y}$ via exploiting the group sparsity structure of $\boldsymbol{\Gamma}^{\frac{1}{2}} \boldsymbol{H}$. The indicator a_i^r can then be determined from the rows of $\boldsymbol{\Gamma}^{\frac{1}{2}} \boldsymbol{H}$. However, such an approach suffers an algorithmic complexity that is dominated by M in massive IoT networks, i.e., the high dimension of $\boldsymbol{\Gamma}^{\frac{1}{2}} \boldsymbol{H}$. Furthermore, with messages embedded in the signature sequences, there is no need to estimate the channel state information [12], and thus recent papers [12], [17] have focused on directly detecting activity via estimating $\boldsymbol{\Gamma}$ instead.

Specifically, the estimation of $\boldsymbol{\Gamma}$ can be formulated as a maximum likelihood estimation problem. Given $\boldsymbol{\gamma}$, each column of $\boldsymbol{Y}$, denoted as $\boldsymbol{y}_m \in \mathbb{C}^L$ for $1 \leq m \leq M$, can be termed as an independent sample from a multivariate complex Gaussian distribution such that [12]:

$$\boldsymbol{y}_m \sim \mathcal{CN}\left(\boldsymbol{0}, \boldsymbol{\Sigma}\right), \tag{4}$$

where $\boldsymbol{\Sigma} = \boldsymbol{Q} \boldsymbol{\Gamma} \boldsymbol{Q}^{\mathsf{H}} + \sigma_n^2 \boldsymbol{I}_L$ with the identity matrix $\boldsymbol{I}_L \in \mathbb{R}^{L \times L}$. Based on (4), the likelihood of $\boldsymbol{Y}$ given $\boldsymbol{\gamma}$ is represented as [12]: $P(\boldsymbol{Y}|\boldsymbol{\gamma}) = \prod_{m=1}^{M} \frac{1}{\det(\pi \boldsymbol{\Sigma})} \exp(-\boldsymbol{y}_m^{\mathsf{H}} \boldsymbol{\Sigma}^{-1} \boldsymbol{y}_m) =$

[1] Notice that $J = 0$ is allowed, which corresponds to the case with activity detection only, as considered in [17].

$(\det(\pi\boldsymbol{\Sigma}))^{-M}\exp(-\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\boldsymbol{Y}^{\mathsf{H}}))$. Based on (4), the maximum likelihood estimation problem can be formulated as minimizing $-\log P(\boldsymbol{Y}|\boldsymbol{\gamma})$:

$$\begin{aligned} \underset{\boldsymbol{\gamma}\in\mathbb{R}^{NR}}{\text{minimize}} \quad & \log|\boldsymbol{\Sigma}| + \frac{1}{M}\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\boldsymbol{Y}^{\mathsf{H}}\right) \\ \text{subject to} \quad & \boldsymbol{\gamma} \geq 0, \\ & ||\boldsymbol{\gamma}_i||_0 \leq 1, \quad i = 1,2,\dots,N, \end{aligned} \tag{5}$$

where $\boldsymbol{\gamma} \geq 0$ means that each element of $\boldsymbol{\gamma}$ is greater than or equal to 0. This covariance-based approach was first proposed in [17] for activity detection, and then extended to joint activity and data detection in [12]. Based on the estimated $\hat{\boldsymbol{\gamma}}$ and a pre-defined threshold s_{th}, the indicator can be determined by

$$a_i^r = \begin{cases} 1, & \text{if} \quad \hat{\gamma}_i^r \geq s_{th} \text{ and } \hat{\gamma}_i^r = \max_{j=1}^{R}\{\hat{\gamma}_i^j\}, \\ 0, & \text{else.} \end{cases} \tag{6}$$

From a_i^r that indicates whether the r-th sequence is transmitted by the i-th device, the activity state of the i-th device and the transmitted data can be determined, i.e., achieving joint activity detection and data decoding.

For the ease of algorithm design, an alternative way to solve problem (5) was developed in [12]. By eluding the absolute value constraints, it yields

$$\underset{\boldsymbol{\gamma}\geq 0}{\text{minimize}} \quad F(\boldsymbol{\gamma}) := \log|\boldsymbol{\Sigma}| + \frac{1}{M}\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\boldsymbol{Y}^{\mathsf{H}}\right). \tag{7}$$

The first term in (7) is a concave function that makes the objective nonconvex, thereby bringing a unique challenge. The paper [12] showed that the estimator $\hat{\boldsymbol{\gamma}}$ of problem (7) by coordinate descent is approximately sparse, and thus constraints $||\boldsymbol{\gamma}_i||_0 \leq 1, \forall i$ can be approximately satisfied. Specifically, it demonstrated that as the sample size, i.e., L, increases, the estimator $\hat{\gamma}$ of problem (7) concentrates around the ground truth $\boldsymbol{\gamma}^{\natural}$ and becomes an approximate sparse vector for large M, which implies that constraints $||\boldsymbol{\gamma}_i||_0 \leq 1, \forall i$ are satisfied approximately when M is large. Motivated by its low per-iteration complexity, the papers [12], [17] developed a coordinate descent algorithm to solve the relaxed problem (7), which updates the coordinate of $\boldsymbol{\gamma}$ randomly until convergence (illustrated in Algorithm 1). However, such a simple coordinate update rule yields a less aggressive convergence rate, and lacks rigorous convergence rate analysis. In this paper, we aim to design a novel sampling strategy for coordinate descent to improve its convergence rate.

Algorithm 1: CD-Random

```
1: Input: The sample covariance matrix $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}} = \frac{1}{M}\boldsymbol{Y}\boldsymbol{Y}^{\mathsf{H}}$ of the $L \times M$ matrix $\boldsymbol{Y}$.
2: Initialize: $\boldsymbol{\Sigma} = \sigma_n^2\boldsymbol{I}_L$, $\boldsymbol{\gamma} = \boldsymbol{0}$.
3: for all $t = 1,2,\dots$ do
4:     Select an index $k \in [NR]$ corresponding to the $k$-th component of $\boldsymbol{\gamma}$ randomly.
5:     Let $\boldsymbol{a}_k$ denote the $k$-th column of $\boldsymbol{Q} \in \mathbb{C}^{L\times NR}$, and set $\delta = \max\left\{\frac{\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k - \boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k}{\left(\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k\right)^2}, -\gamma_k\right\}$
6:     Update $\gamma_k \leftarrow \gamma_k + \delta$.
7:     Update $\boldsymbol{\Sigma} \leftarrow \boldsymbol{\Sigma} + \delta(\boldsymbol{a}_k\boldsymbol{a}_k^{\mathsf{H}})$.
8: end for
9: Output: $\boldsymbol{\gamma} = [\gamma_1,\dots,\gamma_{NR}]^{\top}$.
```

There have been lots of efforts in pushing the efficiency of coordinate descent algorithms by developing more sophisticated coordinate update rules. Concerning supervised learning problems, previous works [24], [21] have demonstrated that the coordinate descent algorithm can yield better convergence guarantees when exploiting the structure of the data and sampling the coordinates from an appropriate non-uniform distribution. Furthermore, the paper [23] proposed a multi-armed bandit based coordinate selection method that can be applied to minimize convex objective functions, e.g., Lasso, logistic and ridge regression. Inspired by [23], we shall apply the idea of Bernoulli sampling to solve the estimation problem (7) with a *non-convex* objective function for joint activity and data detection. In the remainder of the paper, we first present a basic coordinate descent algorithm with Bernoulli sampling in Section III, followed by proposing a more efficient algorithm with Thompson sampling in Section IV, both with rigorous analysis. Simulation results are provided in Section VI.

III. Coordinate Descent with Bernoulli Sampling

In this section, a basic algorithm, coordinate descent with Bernoulli sampling, is developed. We begin with introducing a reward function for each coordinate, which quantifies the decrease of the objective function $F(\boldsymbol{\gamma})$ in (7) by updating the corresponding coordinate. Based on the reward function, a coordinate descent algorithm with Bernoulli sampling (CD-Bernoulli) is proposed for joint device activity and data detection. The convergence rate of the proposed algorithm will be provided, and compared with that of coordinate descent with random sampling [17].

A. Reward Function

The coordinate selection strategy depends on the update rule for the decision variable γ_k for $k \in [NR]$. The update rule with respect to the k-th coordinate is denoted as $\mathcal{H}_k$, which is illustrated by Line 5-7 in Algorithm 1. The following lemma quantifies the decrease of updating a coordinate $k \in [NR]$ according to the update rule $\mathcal{H}_k$, which is the reward function in our proposed algorithm and denoted as r_k.

Lemma 1. *Considering problem (7), and in the t-th iteration choosing a coordinate $k \in [NR]$ and updating γ_k^t with the update rule $\mathcal{H}_k$, we have the following bound: $F\left(\boldsymbol{\gamma}^{t+1}\right) \leq F\left(\boldsymbol{\gamma}^t\right) - r_k^t$, where*

$$r_k^t = \frac{\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k}{1+\delta\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k}\delta - \log\left(1+\delta\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k\right). \tag{8}$$

Proof. The proof is based on [17]. Please refer to Appendix A for details. □

Remark 1. *Recall the definition of $\boldsymbol{\Sigma}$ such that $\boldsymbol{\Sigma} = \boldsymbol{Q}\boldsymbol{\Gamma}\boldsymbol{Q}^{\mathsf{H}} + \sigma_n^2\boldsymbol{I}_L$. Here, $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1^{\top},\cdots,\boldsymbol{\gamma}_N^{\top}]^{\top} \in \mathbb{C}^{NR}$ denotes the diagonal entries of $\boldsymbol{\Gamma}$, where $\boldsymbol{\gamma}_i = [(a_i^1 g_i)^2,\dots,(a_i^R g_i)^2]^{\top} \in \mathbb{C}^R$ for $i = 1,\cdots,N$. Since $\{\boldsymbol{\gamma}_i\}$ are stochastic and i.i.d over time, Σ is stochastic and i.i.d over time. With known $\boldsymbol{a}_k$ and $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}} = \frac{1}{M}\boldsymbol{Y}\boldsymbol{Y}^{\mathsf{H}}$, we can conclude that the reward (8) is stochastic and i.i.d over time.*

Algorithm 2: CD-Bernoulli

```
1:  Input: $\varepsilon$ and $B$
2:  Initialize: $\boldsymbol{\Sigma} = \sigma_n^2 \boldsymbol{I}_L$, $\boldsymbol{\gamma} = \boldsymbol{0}$, set $\bar{r}_k^0 = r_k^0$ for all
    $k \in [NR]$.
3:  for $t = 1$ to $T$ do
4:      if $t \mod B == 0$ then
5:          set $\bar{r}_k^t = r_k^t$ for all $k \in [NR]$
6:      end if
7:      Generate $K \sim \text{Bernoulli}(\varepsilon)$
8:      if $K == 1$ then
9:          Select $k_t = \arg\max_{k \in [NR]} \bar{r}_k^t$
10:     else
11:         Select $k_t \in [NR]$ uniformly at random
12:     end if
13:     Update $\gamma_{k_t}^t$ according to the rule $\mathcal{H}_{k_t}$
14:     Compute the reward $r_{k_t}^{t+1}$
15:     Set $\bar{r}_{k_t}^{t+1} = r_{k_t}^{t+1}$ and $\bar{r}_k^{t+1} = \bar{r}_k^t$ for all $k \neq k_t$
16: end for
```

A greedy algorithm based on Lemma 1 is to simply select at time t the coordinate k with the largest r_k^t at time t. However, the cost of computing reward functions for all the $k \in [NR]$ is prohibitively high, especially with a large number of devices. To address this issue, the paper [23] proposed a principled approach using a bandit framework for learning the best r_k^t's, instead of exactly computing all of them. Inspired by this idea, at each step t, we select a single coordinate k and update it according to the rule $\mathcal{H}_k$. The reward function r_k^t is computed and used as a feedback to adapt the coordinate selection strategy with Bernoulli sampling. Thus, only partial information is available for coordinate selection, which reduces the computational complexity of each iteration. Details of the algorithm are provided in the following subsection.

B. Algorithm and Analysis

Consider a multi-armed bandit (MAB) problem where there are NR arms (corresponding to coordinates in our setting) from which a bandit algorithm can select for a reward, i.e., r_k^t as in (8) at time t. The MAB aims to maximize the cumulative reward received over T rounds, i.e., $\sum_{t=1}^{T} r_{k_t}^t$, where k_t is the arm (coordinate) chosen at time t. After the t-th round, the MAB only receives the reward of the selected arm (coordinate) k_t which is used to adjust its arm (coordinate) selection strategy for the next round. For more background on the MAB problem, please refer to [27].

Following the MAB problem introduced above, the CD-Bernoulli algorithm is illustrated in Algorithm 2. To address the computational complexity issue of the greedy algorithm that requires to compute the reward function r_k^t for all $k \in [NR]$ at each round t, Algorithm 2 only computes the reward function r_k^t of all the coordinates $k \in [NR]$ every B rounds (please refer to Line 4-6 in Algorithm 2). In the remaining rounds, $\bar{r}_k$ is estimated based on the most recently observed reward in the MAB. The coordinate selection policy is presented as follows: with probability $(1-\varepsilon)$ a coordinate $k_t \in [NR]$ is determined uniformly at random, while with probability ε the coordinate endowed with the largest $\bar{r}_k^t$ is chosen. It mimics the ϵ-greedy approach for conventional MAB problems [27]. This is to achieve a tradeoff between *exploration* and *exploitation*. That is, whether choosing the coordinate with currently the largest reward or exploring other coordinates. To be specific, Line 9 in Algorithm 2 is the exploitation phase which chooses the coordinate based on the known reward $\bar{r}_k^t$ for $k \in [NR]$. Line 11 in Algorithm 2 is the exploration phase which explores other coordinates instead of referring to the maximal reward. Then the k_t-th coordinate of $\boldsymbol{\gamma}$ is updated according to the update rule $\mathcal{H}_{k_t}$. The k_t-th entry of the estimated reward function is updated as $\bar{r}_{k_t}^{t+1} = r_{k_t}^{t+1}$ with the rest unchanged.

The estimation error is defined as $\epsilon(\boldsymbol{\gamma}) = F(\boldsymbol{\gamma}) - F(\boldsymbol{\gamma}^\star)$, where $\boldsymbol{\gamma}^\star$ is the optimal solution for the non-negative $\boldsymbol{\gamma}$'s such that $\boldsymbol{\gamma}^\star := \operatorname{argmin}_{\boldsymbol{\gamma} \in \mathbb{R}_+^{NR}} F(\boldsymbol{\gamma})$. In contrast to the previous work [23], which considered the objective function consisting of a smooth convex function and a regularized convex function, this paper considers $F(\boldsymbol{\gamma})$ in (7) consists of a concave function and a convex function. Denote the best arm (coordinate) as $j_\star^t = \arg\max_{k \in [NR]} \bar{r}_k^t$ with the estimated reward $\bar{r}_k^t$ in Algorithm 2. The following result shows the convergence rate of coordinate descent for joint activity and data detection with two different coordinate selection strategies, i.e., random sampling and Bernoulli sampling, where the ratio $\max_{k \in [NR]} r_k^t / r_{j_\star^t}^t$ characterizes the level of similarity between the estimated reward $\bar{r}_k^t$ and the true reward r_k^t.

Theorem 1. *Assume that at each iteration t, $\max_{k \in [NR]} r_k^t / r_{j_\star^t}^t \leq c(B, \varepsilon)$ for some constant c that depends on B and ε, then the iterate $\boldsymbol{\gamma}^t$ at the t-th iteration of the CD-Bernoulli algorithm (illustrated in Algorithm 2) for solving problem (7) obeys $\mathbb{E}\left[\epsilon(\boldsymbol{\gamma}^t)\right] \leq \frac{\alpha}{1+t-t_0}$, where $\alpha^{-1} = \frac{1-\varepsilon}{(NR)^2 c_1} + \frac{\varepsilon}{\eta^2 c}$ with some constant $c_1 > 0$, for all $t \geq t_0 = \max\{1, NR \log \frac{NR \cdot \epsilon(\boldsymbol{\gamma}^0)}{\eta^2}\} = \mathcal{O}(NR)$ and where $\eta = \min_{k \in [NR]} \sum_\ell r_\ell^t / r_k^t$ with r_k^t defined in (8). Furthermore, the CD-Random algorithm (illustrated in Algorithm 1) for solving problem (7) yields $\mathbb{E}\left[\epsilon(\boldsymbol{\gamma}^t)\right] \leq \frac{c_2 (NR)^2}{NR+t}$, with some constant $c_2 > 0$.*

Proof. Please refer to Appendix E for details. □

We conclude from Theorem 1 that by choosing proper values of B and ε (we use $B = NR/2$ and $\varepsilon = 0.6$ in the experiments of Section VI) to yield sufficiently small $c(B, \varepsilon)$, the bound with respect to CD-Bernoulli approaches $\epsilon(\boldsymbol{\gamma}^t) = \mathcal{O}(\eta^2/t)$ with $\eta \leq NR$. $\eta = NR$ is satisfied only when r_k are equal for all k's. Thus, the bound with respect to CD-Bernoulli generally outperforms the bound with respect to CD-Random, i.e., $\epsilon(\boldsymbol{\gamma}^t) = \mathcal{O}((NR)^2/t)$. Hence, Theorem 1 demonstrates that for solving covariance-based joint device activity detection and data decoding, CD-Bernoulli yields a faster convergence rate than that with CD-Random.

In Algorithm 2, the value of ε plays a vital role in balancing exploitation and exploration. The larger the value of ε is, the higher profitability of selecting the coordinate endowed with the largest current reward function r_k^t (8) at each iteration t is. However, a larger value of ε leads to insufficient exploration, which may lead to a slow convergence rate. Instead of fixing ε, we prefer developing a more flexible strategy for choosing ε. This motivates an improved algorithm to be presented in the next section.

IV. Coordinate Descent with Thompson sampling

In this section, we improve the convergence rate of the CD-Bernoulli algorithm by incorporating another bandit problem to adaptively choose ε. Specifically, we formulate the choice of the parameter ε as a general Bernoulli bandit problem, and develop a Thompson sampling algorithm for solving this bandit problem. The theoretical analysis is also presented to verify its advantage.

A. A Stochastic MAB Problem for Choosing ε

Based on [26], we first introduce a stochastic q-armed bandit problem for optimizing the parameter ε in Algorithm 2 . We assume that the reward distribution with respect to choosing ε is *Bernoulli*, i.e., the rewards are either 0 or 1. Note that the reward with respect to choosing ε is different from the reward function of selecting the coordinates defined by (8).

An algorithm for the MAB problem needs to decide which arm to play at each time step t, based on the outcomes of the previous $t-1$ plays. Let μ_i denote the (unknown) expected reward for arm i. The means for the q-armed bandit problem, denoted as $\mu_1, \mu_2, \ldots, \mu_q$, are unknown, and are required to be learned by playing the corresponding arms. A general way is to maximize the expected total reward by time T, i.e., $\mathbb{E}[\sum_{t=1}^{T} \mu_{i(t)}]$, where $i(t)$ is the arm played at step t, and the expectation is over the random choices of $i(t)$ made by the algorithm. The expected total *regret* can be also represented as the loss that is generated due to not playing the optimal arm in each step. Let $\mu^* := \max_i \mu_i$, and $d_i := \mu^* - \mu_i$. Also, let $k_i(t)$ denote the number of times arm i has been played up to step $t-1$. Then the expected total regret in time T is given by [26] $\mathbb{E}[\mathcal{R}(T)] = \mathbb{E}\left[\sum_{t=1}^{T} \left(\mu^* - \mu_{i(t)}\right)\right] = \sum_i d_i \cdot \mathbb{E}\left[k_i(T)\right].$

B. Thompson Sampling

We first present some background on the Thompson sampling algorithm for the Bernoulli bandit problem, i.e., when the rewards are either 0 or 1, and for arm i the probability of success (reward =1) is μ_i. More details on Thompson sampling can be found in [34] and [26].

It is convenient to adopt Beta distributions as the Bayesian priors on the Bernoulli means μ_i's. Specifically, the probability density function (pdf) of Beta(α, β), i.e., the beta distribution with parameters $\alpha > 0$, $\beta > 0$, is given by $f(x; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$ with $\Gamma(\cdot)$ being the gamma function. If the prior is a Beta(α, β) distribution, then based on a Bernoulli trial, the posterior distribution can be represented as Beta$(\alpha+1, \beta)$ when the trail leads to a success; otherwise, it is updated as Beta$(\alpha, \beta+1)$.

The previous studies of the Thompson sampling algorithm, e.g., [26], generally assumed that α and β are integers. The algorithm initially assumes that arm i has a prior as Beta$(1,1)$ on μ_i, which is natural because Beta$(1,1)$ is the uniform distribution on the interval $(0,1)$. At time t, having observed $S_i(t)$ successes (reward = 1) and $F_i(t)$ failures (reward = 0) in $k_i(t) = S_i(t) + F_i(t)$ plays of arm i, the algorithm updates the distribution on μ_i as Beta$(S_i(t)+1, F_i(t)+1)$. The algorithm then samples from these posterior distributions of the μ_i's, and plays an arm according to the probability of its mean being the largest.

Algorithm 3: CD-Thompson

```
Input: $E$.
Initialize: $\boldsymbol{\Sigma} = \sigma_n^2 \boldsymbol{I}_L$, $\boldsymbol{\gamma} = \boldsymbol{0}$,
  set $\bar{r}_k^0 = r_k^0$ for all $k \in [NR]$,
  the TS parameters $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_q]$ and with
  $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_q]$ some integer $q$.
for $t = 1$ to $T$ do
    if $t \mod E == 0$ then
        set $\bar{r}_k^t = r_k^t$ for all $k \in [NR]$
    end if
    For each arm $i = 1, \cdots, q$, sample $\nu_i^t \sim \mathrm{Beta}(\alpha_i, \beta_i)$
    $j_t = \arg\max_i(\nu_i^t)$
    Generate $K \sim \mathrm{Bernoulli}(\nu_{j_t}^t)$
    if $K == 1$ then
        Select $k_t = \arg\max_{k \in [NR]} \bar{r}_k^t$
        Compute $\kappa_{k_t}^t = r_{k_t}^t / F(\boldsymbol{\gamma}^t)$ based on (8).
        Update $\alpha_{j_t} = \alpha_{j_t} + \nu_{j_t}^t \cdot \kappa_{k_t}^t$
    else
        Select $k_t \in [NR]$ uniformly at random
        Compute $\kappa_{k_t}^t = r_{k_t}^t / F(\boldsymbol{\gamma}^t)$ based on (8).
        Update $\beta_{j_t} = \beta_{j_t} + (1 - \nu_{j_t}^t)\kappa_{k_t}^t$
    end if
    Update $\gamma_{k_t}^t$ according to the rule $\mathcal{H}_{k_t}$
    Set $\bar{r}_{k_t}^{t+1} = r_{k_t}^{t+1}$ and $\bar{r}_k^{t+1} = \bar{r}_k^t$ for all $k \neq k_t$
end for
```

Different from previous methods, in this paper, we consider a more general way to update the parameters α and β by evaluating the reward function r_k^t, to be presented in the following subsection.

C. CD-Thompson

The common approach for solving MAB, e.g., the upper confidence bound (UCB) [35], tends to be fairly conservative compared to Thompson Sampling [26] and Epsilon Greedy (e.g., $\epsilon = 0.1$). To be specific, the upper-confidence bound action selection leverages uncertainty in the action-value estimates for dealing with the trade-off between exploration and exploitation. Thompson Sampling builds up a probability model from the obtained rewards, and then samples from it to choose an action, which is more complex compared to Epsilon Greedy. Even though Thompson Sampling yields an increasingly accurate estimate, the sample complexity and computational cost can be prohibitively large when dealing with random access in IoT where the number of arms of the corresponding MAB problem is NR. Here, N can be extremely large due to massive devices. To control the overall complexity, we propose to utilize ϵ-Greedy, a low-complexity approach, to deal with coordination descent where the MAB problem has a large number of arms. To enhance the performance, we further use Thompson Sampling to learn the optimal choice of ϵ where the MAB problem has a relatively small number of arms.

To be specific, the coordinate descent algorithm via Thompson sampling (CD-Thompson) is illustrated in Algorithm 3. In this algorithm, a stochastic MAB problem for learning the best ν_i^t for arms $i = 1, \cdots, q$ at the t-th iteration is established, and a Thompson sampling algorithm is developed to solve this bandit problem. Each arm of this MAB problem corresponds to the choice of parameter ν_i^t for $i \in [q]$. Here, each ν_i^t is generated from $\nu_i^t \sim \mathrm{Beta}(\alpha_i, \beta_i)$ with parameter pair (α_i, β_i) for $i \in q$. By selecting the optimal arm, i.e., $j_t = \arg\max_i(\nu_i^t)$, we generate $K \sim \mathrm{Bernoulli}(\nu_{j_t}^t)$. In Algorithm 3, the reward r_k^t for selecting the k-th coordinate at time step t is taken into consideration to update the parameters $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_q], \boldsymbol{\beta} = [\beta_1, \cdots, \beta_q]$, thereby choosing ν_i^t based on $\nu_i^t \sim \mathrm{Beta}(\alpha_i, \beta_i)$. To be specific, for the index $j_t = \arg\max_i(\nu_i^t)$ and the Bernoulli variable $K \sim \mathrm{Bernoulli}(\nu_{j_t}^t)$, if $K = 1$, we update

$$\alpha_{j_t} = \alpha_{j_t} + \nu_{j_t}^t \cdot r_{k_t}^t / F(\boldsymbol{\gamma}^t); \tag{9}$$

otherwise, we update

$$\beta_{j_t} = \beta_{j_t} + (1 - \nu_{j_t}^t) r_{k_t}^t / F(\boldsymbol{\gamma}^t), \tag{10}$$

where $r_{k_t}^t$ is defined in (8) and $F(\boldsymbol{\gamma}^t)$ is defined in (7). For illustration, the main processes of CD-Bernoulli and CD-Thompson are illustrate in Fig. 2.

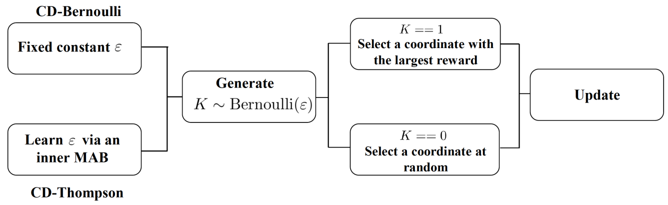
CD-Bernoulli
Fixed constant ε
Learn ε via an inner MAB
CD-Thompson
Generate $K \sim \mathrm{Bernoulli}(\varepsilon)$
$K == 1$ Select a coordinate with the largest reward
$K == 0$ Select a coordinate at random
Update

Fig. 2. The main processes of CD-Bernoulli and CD-Thompson.

Recall that μ_i denotes the (unknown) expected reward for arm i. At time t, if arm i has been played a sufficient number of times, ν_i^t is tightly concentrated around μ_i with high probability. In the following analysis, we assume that the first arm is the unique optimal arm, i.e., $\mu_1 = \arg\max_{i \neq 1} \mu_i$. The convergence analysis of CD-Thompson is the same as the one of CD-Bernoulli except that a stochastic MAB problem is established to learn a better parameter ε (recall ε given in Theorem 1). The expected regret for the stochastic MAB problem in Algorithm 3 is presented as follows.

Theorem 2. *The q-armed stochastic bandit problem for choosing ν_i^T for $i = 1, \cdots, q$ in Algorithm 3 has an expected regret as $\mathbb{E}[\mathcal{R}(T)] \leq \mathcal{O}\left(\left(\sum_{b=2}^{q} \frac{1}{d_b^2}\right)^2 \ln T\right)$ in time T, where $d_i = \mu_1 - \mu_i$.*

Proof. Please refer to Appendix F for a brief summary of the proof. □

Remark 2. *Algorithm 2 adopts a fixed constant $\varepsilon > 0$ as the probability of updating coordinate k with the largest reward function (i.e., coordinate-wise descent value) r_k^t (8) at time step t, which lacks flexibility for better exploration-exploitation trade-off. In contrast, Algorithm 3 improves the strategy of choosing the parameter ε in Algorithm 2. This is achieved by establishing a stochastic q-armed bandit problem for choosing the corresponding probability. This multi-armed stochastic bandit problem achieves an exploitation/exploration trade-off by sequentially designing ν_i^T for $i = 1, \cdots, q$ at time step t. During the sequential decision, Algorithm 3 is able to approximate the optimal value of the probability. Theoretically, Theorem 2 demonstrates that Algorithm 3 enjoys a logarithmic expected regret for the stochastic q-armed bandit problem, which typically is the best to expect. Thus, by taking the posterior sample as ϵ such that $K \sim \mathrm{Bernoulli}(\epsilon)$ (Line 9 in Algorithm 3), the optimal convergence rate of the ϵ-based algorithm can be achieved with high probability. Furthermore, the exploitation/exploration trade-off in Algorithm 3 eludes the situation where the large value of ν_i^t in Algorithm 3 is maintained in many time steps, and thus avoids high computational cost for computing r_k^t for all $k \in [NR]$ at time step t.*

Remark 3. *Different from the previous MAB based coordinate descent algorithm [23] that solves* convex *optimization problems, our proposed algorithm solves a covariance-based estimation problem that is* non-convex. *Beta distribution, i.e., Beta(α, β), is a powerful tool to learn the priors for Bernoulli rewards. Specifically, we consider a more general way to update the parameters α and β based on the reward function r_k^t. Compared to previous work by Auer et al. [35] which aims to measure the likelihood that one variant/arm is truly more effective than another, our proposed algorithm tends to optimize long-term overall regret. Moreover, the algorithm introduced by Auer et al. [35] suffers from larger noise due to the random sampling step in the algorithm and is also fairly conservative. This may lead to a less attractive convergence rate. Our proposed algorithm turns out to enjoy a faster convergence rate with modest computational time complexity.*

V. Application to Massive Connectivity with Low-precision ADCs

While the formulation in Section II presents a basic massive connectivity system, the proposed algorithms, i.e., CD-Bernoulli and CD-Thompson, can also be applied to the massive connectivity problem in more general scenarios. In this section, we introduce massive connectivity with low-precision analog-to-digital converters (ADCs) as a practical application scenario, which enjoys low cost and power consumption while guaranteeing the accuracy of recovery [36], [37], [31]. In the following, we illustrate how the proposed algorithms can be applied to this new scenario.

At each of the receive antennas, the A/D converter samples the received signal and utilizes a finite number of bits to represent corresponding samples. Each entry, i.e., Y_{ij}, of $\boldsymbol{Y}$ (3) for $1 \leq i \leq L, 1 \leq j \leq M$ is quantized into a finite set of pre-defined values by a b-bit quantizer Q_c. The quantized received signal is thus represented by [31]

$$\boldsymbol{Y}_{\mathrm{q}} = \mathrm{Q}_c(\boldsymbol{Y}) = \mathrm{Q}_c(\boldsymbol{Q}\boldsymbol{\Gamma}^{\frac{1}{2}}\boldsymbol{H} + \boldsymbol{N}), \tag{11}$$

where the complex-valued quantizer $\mathrm{Q}_c(\cdot)$ is defined as $X_{\mathrm{q}} = \mathrm{Q}_c(X) \triangleq \mathrm{Q}(\mathrm{Re}\{X\}) + \mathrm{i}\mathrm{Q}(\mathrm{Im}\{X\})$, i.e., the real and imaginary parts are quantized separately. The real valued quantizer Q maps a real-valued input to one of the 2^b bins, which are characterized by the set of $2^b - 1$ thresholds $[r_1, r_2, \ldots, r_{2^b-1}]$, such that $-\infty < r_1 < r_2 < r_2 < \cdots < \infty$. For $z = 1, \ldots, 2^b - 1$, an element of the output $\boldsymbol{Y}_{\mathrm{q}}$ is assigned a value in $(r_{z-1}, r_z]$ when the quantizer entry of the input $\boldsymbol{Y}$ falls in the z-th bin, i.e., the interval $(r_{z-1}, r_z]$.

Generally, the quantization operation is nonlinear. For the ease of applying coordinate descent algorithms to solve quantized model, we linearize the quantizer. Based on Bussgang's theorem, the quantizer output $\boldsymbol{Y}_{\mathrm{q}}$ can be decomposed into a signal component plus a distortion $\boldsymbol{W}_{\mathrm{q}} \in \mathbb{C}^{L\times M}$ that is uncorrelated with the signal component $\boldsymbol{Y}$ [36], [38], i.e., $\boldsymbol{Y}_{\mathrm{q}} = (\boldsymbol{I}_M - \boldsymbol{\rho})\,\boldsymbol{Y} + \boldsymbol{W}_{\mathrm{q}}$, where $\boldsymbol{\rho}$ is the real-valued diagonal matrix containing the M distortion factors:

$$\boldsymbol{\rho} = \begin{bmatrix} \rho_1 & & \\ & \ddots & \\ & & \rho_M \end{bmatrix} \approx \begin{bmatrix} 2^{-2b_1} & & \\ & \ddots & \\ & & 2^{-2b_M} \end{bmatrix}, \quad (12)$$

with b_j for $j = 1, \cdots, M$ denoting the bit resolution of the scalar quantizer with respect to each antenna.

Since $\boldsymbol{W}_{\mathrm{q}}$ is uncorrelated with the signal component $\boldsymbol{Y}$, the covariance matrix of the quantizer can be represented as

$$\boldsymbol{\Sigma}_{\mathrm{q}} = \mathrm{E}\left[\boldsymbol{Y}_{\mathrm{q}}\boldsymbol{Y}_{\mathrm{q}}^{\mathsf{H}}\right] = \boldsymbol{\rho}\boldsymbol{\Sigma}\boldsymbol{\rho} + \boldsymbol{\rho}\left(\boldsymbol{I}_M - \boldsymbol{\rho}\right)\mathrm{diag}\left(\boldsymbol{\Sigma}\right), \quad (13)$$

where $\boldsymbol{\Sigma}$ is defined in (4). Hence, the joint device activity detection and data decoding with low-precision ADCs can be formulated as

$$\underset{\boldsymbol{\gamma}\geq 0}{\mathrm{minimize}} \quad F(\boldsymbol{\gamma}) := \log|\boldsymbol{\Sigma}_{\mathrm{q}}| + \frac{1}{M}\mathrm{Tr}\left(\boldsymbol{\Sigma}_{\mathrm{q}}{}^{-1}\boldsymbol{Y}_{\mathrm{q}}\boldsymbol{Y}_{\mathrm{q}}^{\mathsf{H}}\right). \quad (14)$$

Problem (14) can be efficiently solved by the proposed algorithms, i.e., Algorithm 2 and Algorithm 3. Simulations will be presented in the next section.

VI. Simulation Results

In this section, we provide simulation results to demonstrate that the proposed algorithms enjoy faster convergence rates than coordinate descent with random sampling for joint device activity detection and data decoding. Furthermore, we apply our proposed algorithms to massive connectivity with low-precision ADCs.

A. Simulation Settings and Performance Metric

The simulation settings are given as follows:

- The signature matrix $\boldsymbol{Q} \in \mathbb{C}^{L\times NR}$ with $R = 2^J$ is generated from i.i.d. standard complex Gaussian distribution, followed by normalization, i.e., $\boldsymbol{Q} \sim \mathcal{N}(\boldsymbol{0}, \frac{1}{2L}\boldsymbol{I}_L) + \mathrm{i}\mathcal{N}(\boldsymbol{0}, \frac{1}{2L}\boldsymbol{I}_L)$.
- The channel matrix $\boldsymbol{H} \in \mathbb{C}^{NR\times M}$ consists of Rayleigh fading components that follow i.i.d. standard complex Gaussian distribution, i.e., $\boldsymbol{H} \sim \mathcal{N}(\boldsymbol{0}, \frac{1}{2}\boldsymbol{I}_{NR}) + \mathrm{i}\mathcal{N}(\boldsymbol{0}, \frac{1}{2}\boldsymbol{I}_{NR})$.Meanwhile, the fading component g_i in (1) for device i with $i = 1, \cdots, N$ is given as $g_i = -128.1 - 37.6\log_{10}(d_i)$ in dB.
- The additive noise matrix $\boldsymbol{N} \in \mathbb{C}^{L\times M}$ is generated from i.i.d. complex Gaussian distribution, i.e., $\boldsymbol{N} \sim \mathcal{N}(\boldsymbol{0}, \frac{1}{2\sigma_n^2}\boldsymbol{I}_L) + \mathrm{i}\mathcal{N}(\boldsymbol{0}, \frac{1}{2\sigma_n^2}\boldsymbol{I}_L)$, where the variance σ_n^2 is the background noise power normalized by the device transmit power. In the simulations, the background noise power is set as -99 dBm.
- The adopted performance metric is defined in the following. The missed detection occurs when a device is active but is detected to be inactive, or a device is active and is detected to be active but the data decoding is incorrect. The event of false alarm is the circumstance that a device is inactive but detected as active. Different probabilities of missed detection and false alarm can be obtained by adjusting the value of the threshold s_{th} in (6). In the simulations, we choose the threshold s_{th} in (6) to achieve a point where the probability of false alarm and probability of missed detection are equal, which is represented as the *probability of error* [12].

The following three algorithms are compared:

- **Proposed coordinate descent with Bernoulli sampling (CD-Bernoulli)**: Problem (7) is solved by Algorithm 2 with the setting of $B = NR/2$ and $\varepsilon = 0.6$. Note that the computational time will increase as the value of ε increases. The convergence rate of CD-Bernoulli will decrease as the value of ε decreases. We thus pick a modest value to illustrate the performance of CD-Bernoulli.
- **Proposed coordinate descent with Thompson sampling (CD-Thompson)**: Problem (7) is solved by Algorithm 3 with the setting of $B = NR/2$ and $q = 10$.
- **Coordinate descent with random sampling (CD-Random)**: Problem (7) is solved by Algorithm 1 with uniformly randomly choosing a coordinate to update.

All the algorithms stop when the relative change of the objective function $F(\boldsymbol{\gamma}^t)$ is lower than a certain level, i.e., $\frac{|F(\boldsymbol{\gamma}^{t+1}) - F(\boldsymbol{\gamma}^t)|}{|F(\boldsymbol{\gamma}^t)|} \leq 10^{-6}$ or the number of iterations exceeds 1500.

B. Convergence Rate

Consider a single cell of radius 1000m containing $N = 1500$ devices, among which $K = 50$ devices are active. In the simulations, the number of antenna is $M = 16$, and each device transmits a message of $J = 0, 1, 2$ bits. The transmit power of each device is set as 40 dBm, and the distances are set as $d_i = 1000$ for all $\forall i \in [N]$. The convergence rates of different algorithms are illustrated in Fig. 3 with $J = 0$ and the length of the signature sequences being $L = 45$. Fig. 4 further illustrates the convergence rates of different algorithms with $J = 1, 2$ and $L = 300$. We validate the convergence rate analysis in Theorem 1 by comparing CD-Bernoulli (i.e., Algorithm 2) with CD-Random (i.e., Algorithm 1). Furthermore, Fig. 3 and Fig. 4 show that CD-Thompson with a more sophisticated strategy for choosing the probability of updating the coordinate has better performance than Algorithm 2. As illustrated in Fig. 3 and Fig. 4, as well as demonstrated in Theorem 1, a larger value of J yields a large value of NR, which leads to a slower convergence rate. In summary, this simulation shows that the proposed algorithms yield faster convergence rates than the state-of-the-art algorithm [12].

C. Probability of Error

Consider a network containing $N = 5000$ devices, among which $K = 1000$ devices are active. The transmit power of each device is set as 20 dBm, and the distances d_i for $\forall i \in [N]$ are chosen randomly from $(0, 1000]$. Under the setting of $L = 4000, J = 1, M = 256$, the computational time of three algorithms are illustrated in Fig. 5. It shows that the proposed algorithms achieve the same level of detection accuracy with much less computational time than the algorithm in [12]. The reason is that the coordinate selection policy with Bernoulli sampling or Thompson sampling is able to choose the coordinate that yields a larger descent in the objective value.

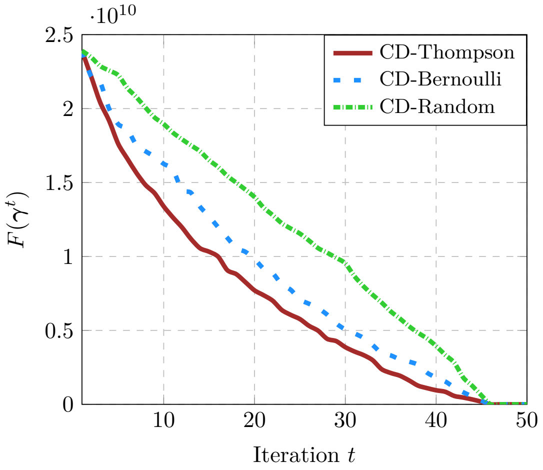

Fig. 3. Convergence rates of coordinate descent with respect to three coordinate selection strategies.

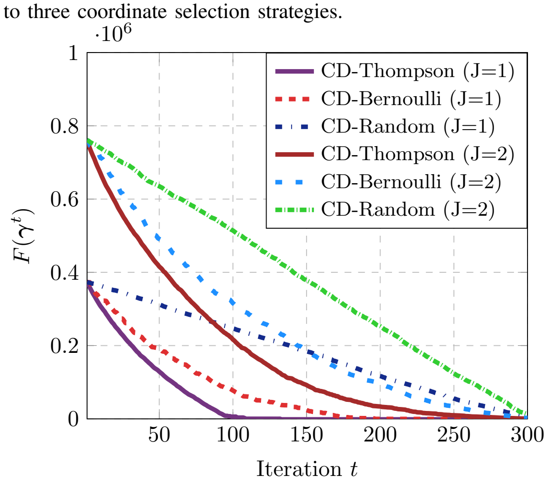

Fig. 4. Convergence rates of coordinate descent with respect to three coordinate selection strategies.

Additionally, Fig. 5 also shows that Algorithm 3 can further reduce the computational time, compared to Algorithm 2. This is achieved by a better exploitation/exploration trade-off by Algorithm 3.

D. Applications in Low-precision ADCs

Consider a single cell of radius 1000m containing $N = 1500$ devices, among which $K = 50$ devices are active. In this part, we test the proposed algorithms with low-precision ADCs. For the quantization procedure, we use the typical uniform quantizer with the quantization step-size $s_{\mathrm{q}} = 0.5$. For b-bit quantization, the threshold of this uniform quantizer is given by

$$r_z = (-2^{b-1} + z)s_{\mathrm{q}}, \quad \text{for } z = 1, \ldots, 2^b - 1, \tag{15}$$

and the element of the quantization output $\boldsymbol{Y}_{\mathrm{q}}$ is assigned the value $r_z - \frac{s_{\mathrm{q}}}{2}$ when the input falls in the z-th bin, i.e., $(r_{z-1}, r_z]$.

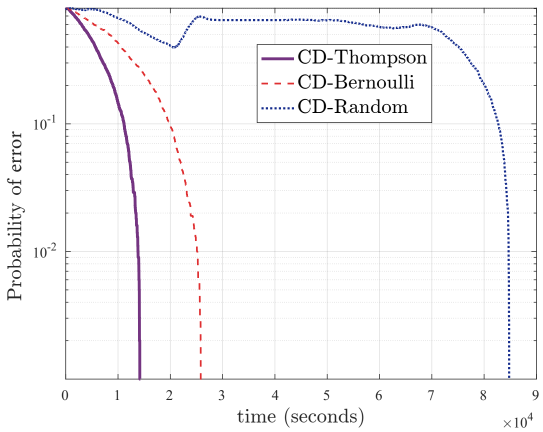

Fig. 5. Probability of error vs. computational time.

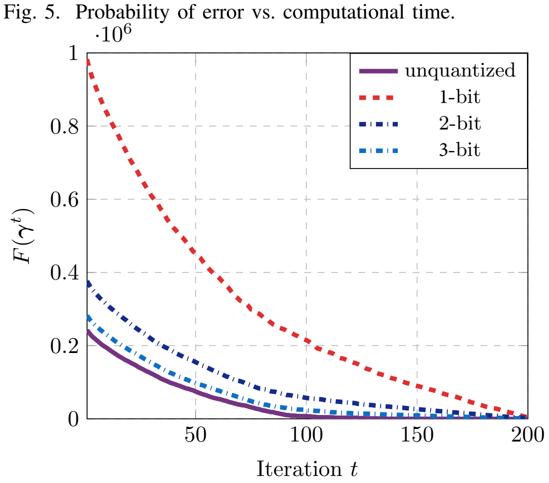

Fig. 6. Convergence rates of coordinate descent with Thompson sampling for massive connectivity with low-precision ADCs.

Under the same setting as Section VI-C, Fig. 6 shows the unquantization case, and the quantization case with different quantization levels, i.e., $b = \{1, 2, 3\}$. Different from Fig. 5, Fig. 6 primarily illustrates that a low quantization level, e.g., $b = 3$, can achieve a similar level of accuracy of recovery as the unquantized case and simultaneously reducing the power consumption and computational cost. To further illustrate the computational cost of the proposed algorithm applied to the low-precision ADCs, Fig. 7 shows the probability of missed detection with respect to computational time. These results demonstrate that 3-bit quantization is sufficient to achieve similar convergence rate and accuracy as the unquantization scenario.

VII. Conclusions

In this paper, we developed efficient algorithms based on multi-armed bandit to solve the joint device activity detection and data decoding problem in massive random access. Specifically, we exploited a multi-armed bandit algorithm to learn to update the coordinate, thereby resulting in more aggressive descent of the objective function. To further improve the convergence rate, an inner multi-armed bandit problem was established to improve the exploration policy. The performance gains in the convergence rate and time complexity of the proposed algorithms over the start-of-the-art algorithm were demonstrated both theoretically and empirically. Furthermore, our proposed algorithms can be applied to a more general scenario, i.e., activity and data detection in the low-precision

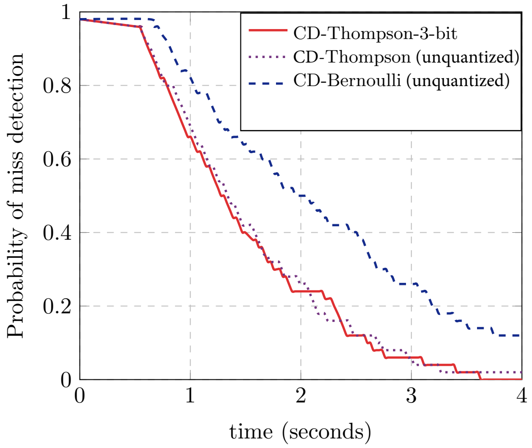

Fig. 7. Probability of missed detection.

analog-to-digital converters (ADCs), thereby saving energy and reducing the power consumption.

Our proposed algorithm only updates a single coordinate at each time step t. It is interesting to further investigate the effect of choosing multiple coordinates from a budget at each time step. At a high level, the proposed approach can be regarded as an instance of "learning to optimize", i.e., applying machine learning to solve optimization problems. Specifically, it belongs to optimization policy learning [39], which learns a specific policy for some optimization algorithms. One related work is [40], which learns the pruning policy of the branch-and-bound algorithm. It is interesting to apply such an approach to other optimization algorithms to improve the computational efficiency for massive connectivity.

Appendix A
Computation of the Reward Function

In this section, we derive the reward function for the multiple-armed bandit problem for coordinate descent. Define $k \in [N]$ as the index of the selected coordinate and define $F_k(d) = F(\boldsymbol{\gamma} + d\boldsymbol{e}_k)$ where $\boldsymbol{e}_k$ denotes the k-th canonical basis with a single 1 at its k-th coordinate and zeros elsewhere. We can simplify $F_k(d)$ as follows

$$\begin{aligned} F_k(d) = & \log\left|\boldsymbol{\Sigma}\right| + \frac{1}{M}\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{Y}\boldsymbol{Y}^{\mathsf{H}}\right) + \log(1 + d\,\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k) \\ & - \frac{\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k}{1 + d\,\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k} d. \end{aligned} \tag{16}$$

According to [17], the global minimum of $F_k(d)$ in $(-\frac{1}{\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k}, +\infty)$ is $\delta = \frac{\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k - \boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k}{(\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k)^2}$, so the descent value of the cost function $F(\boldsymbol{\gamma})$ is:

$$\begin{aligned} F(\boldsymbol{\gamma}) - F_k(\delta) &= F(\boldsymbol{\gamma}) - F(\boldsymbol{\gamma} + \delta) \\ &= \frac{\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{y}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k}{1 + \delta\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k}\delta - \log\left(1 + \delta\boldsymbol{a}_k^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\boldsymbol{a}_k\right). \end{aligned} \tag{17}$$

Hence, the reward function r_k is defined as $r_k = F(\boldsymbol{\gamma}) - F_k(\delta)$.

Appendix B
Primary theorems for the Proof of Theorem 1

Several theorems are needed to pave the way for the proof of Theorem 1.

Theorem 3. *Recall the reward function r_k defined in* (8). *Under the assumptions of Lemma 1, if we choose the coordinate k with the largest r_k^t at the t-th iteration, it yields the following linear convergence guarantee:*

$$\epsilon(\boldsymbol{\gamma}^t) \leq \epsilon(\boldsymbol{\gamma}^0)\prod_{j=1}^{t}\left(1 - \max_{k\in[d]}\frac{r_k^t}{\sum_\ell r_\ell^t}\right), \tag{18}$$

for all $t > 0$, where $\epsilon(\boldsymbol{\gamma}^0)$ is the sub-optimality gap at $t = 0$.

Proof. Please refer to Appendix C for details. □

Theorem 4. *Under the assumptions of Lemma 1, we have the following convergence guarantee:*

$$\epsilon(\boldsymbol{\gamma}^t) \leq \frac{\eta^2}{NR + t - t_0} \tag{19}$$

for all $t \geq t_0$, where $t_0 = \max\{1, NR\log\frac{NR\epsilon(\boldsymbol{\gamma}^0)}{\eta^2}\}$, $\epsilon(\boldsymbol{\gamma}^0)$ is the sub-optimality gap at $t = 0$ and $\eta = O(NR)$ is an upper bound on $\min_{k\in[NR]}\frac{\sum_\ell r_\ell^t}{r_k^t}$ for all iterations $j \in [t]$.

Appendix C
Proof of Theorem 3

The selection strategy concerned in this proof is to choose the coordinated k with the largest reward function r_k^t defined in (8), which is denoted by $k^\star$. Hence, based on the fact $\sum_\ell r_\ell^t \geq \epsilon(\boldsymbol{\gamma}^t)$ it yields that

$$\begin{aligned} \epsilon(\boldsymbol{\gamma}^{t+1}) - \epsilon(\boldsymbol{\gamma}^t) = F(\boldsymbol{\gamma}^{t+1}) - F(\boldsymbol{\gamma}^t) &\leq -r_{k^\star}^t \\ &= -\sum_\ell r_\ell^t \max_{k\in[NR]}\frac{r_k^t}{\sum_\ell r_\ell^t} \\ &\leq -\epsilon(\boldsymbol{\gamma}^t)\max_{k\in[NR]}\frac{r_k^t}{\sum_\ell r_\ell^t}, \end{aligned} \tag{20}$$

that induces $\epsilon(\boldsymbol{\gamma}^{t+1}) \leq \epsilon(\boldsymbol{\gamma}^t) - \epsilon(\boldsymbol{\gamma}^t)\max_{k\in[NR]}\frac{r_k^t}{\sum_\ell r_\ell^t}$, which leads to

$$\epsilon(\boldsymbol{\gamma}^{t+1}) \leq \epsilon(\boldsymbol{\gamma}^t)\left(1 - \max_{k\in[NR]}\frac{r_k^t}{\sum_\ell r_\ell^t}\right). \tag{21}$$

Appendix D
Proof of Theorem 4

According to $F(\boldsymbol{\gamma}^{t+1}) - F(\boldsymbol{\gamma}^t) = \epsilon(\boldsymbol{\gamma}^{t+1}) - \epsilon(\boldsymbol{\gamma}^t)$, we get $\epsilon(\boldsymbol{\gamma}^{t+1}) - \epsilon(\boldsymbol{\gamma}^t) \leq -r_{k^\star}^t$.

As $k^\star$ is the coordinate with the largest r_k^t, we have

$$\epsilon(\boldsymbol{\gamma}^{t+1}) - \epsilon(\boldsymbol{\gamma}^t) \leq -r_{k^\star}(\boldsymbol{\gamma}^t) \leq -\frac{\sum_\ell r_\ell^t}{NR}. \tag{22}$$

Recall the definition of $\epsilon(\boldsymbol{\gamma}^t)$ and the coordinate-wise descent value of objective function (8). The sum of the descent value generated based on the gradient of each coordinate (i.e., $\sum_l r_l^t$) is no less than the descent value generated with respect to a certain direction which is from the current iterate to the optimal solution (i.e., $F(\boldsymbol{\gamma}^t) - F(\boldsymbol{\gamma}^\star)$). Thus, we have

$\epsilon(\boldsymbol{\gamma}^t) \leq \sum_{\ell=1}^{NR} r_\ell^t$. Plugging the inequality $\epsilon(\boldsymbol{\gamma}^t) \leq \sum_{\ell=1}^{NR} r_\ell^t$ in (22) yields

$$\epsilon(\boldsymbol{\gamma}^{t+1}) - \epsilon(\boldsymbol{\gamma}^t) \leq -\frac{\sum_\ell r_\ell^t}{NR} \leq -\frac{\epsilon(\boldsymbol{\gamma}^t)}{NR}, \tag{23}$$

thus, it arrives

$$\epsilon(\boldsymbol{\gamma}^{t+1}) \leq \epsilon(\boldsymbol{\gamma}^t) \cdot \left(1 - \frac{1}{NR}\right). \tag{24}$$

Furthermore, the inductive step at time $j+1$ is justified by plugging (19) in (24):

$$\epsilon(\boldsymbol{\gamma}^{t+1}) \leq \frac{\eta^2}{NR + t - t_0}\left(1 - \frac{1}{NR}\right) \leq \frac{\eta^2}{NR + t + 1 - t_0}. \tag{25}$$

To complete the proof, the induction base case for $t = t_0$ needs to be justified, i.e., we need to show that

$$\epsilon(\boldsymbol{\gamma}^{t_0}) \leq \frac{\eta^2}{NR}. \tag{26}$$

The proof based on the contradiction is used to identify the induction base, that is, assuming $\epsilon(\boldsymbol{\gamma}^{t_0}) > \frac{\eta^2}{NR}$ leads to a contradiction. If $\epsilon(\boldsymbol{\gamma}^{t_0}) > \frac{\eta^2}{NR}$, then

$$\frac{1}{NR} < \frac{\epsilon(\boldsymbol{\gamma}^{t_0})}{\eta^2}. \tag{27}$$

Based on (24), there is

$$\epsilon(\boldsymbol{\gamma}^{t_0}) \leq \epsilon(\boldsymbol{\gamma}^0)\left(1 - \frac{1}{NR}\right)^{t_0}. \tag{28}$$

Based on the inequality such that $1 + x < \exp(x)$ for $x < 1$, we have

$$\epsilon(\boldsymbol{\gamma}^{t_0}) \leq \epsilon(\boldsymbol{\gamma}^0)\exp(-\frac{t_0}{NR}). \tag{29}$$

Recall $t_0 = \max\{1, NR\log\frac{NR\cdot\epsilon(\boldsymbol{\gamma}^0)}{\eta^2}\}$ defined in Theorem 1, and we arrive

$$\begin{aligned}\epsilon(\boldsymbol{\gamma}^{t_0}) &\leq \epsilon(\boldsymbol{\gamma}^0)\exp(-\log\frac{NR\cdot\epsilon(\boldsymbol{\gamma}^0)}{\eta^2}) \\ &= \epsilon(\boldsymbol{\gamma}^0)\frac{\eta^2}{NR\cdot\epsilon(\boldsymbol{\gamma}^0)} = \frac{\eta^2}{NR},\end{aligned}$$

which yields a contradiction with respect to the assumption $\epsilon(\boldsymbol{\gamma}^{t_0}) > \frac{\eta^2}{NR}$. It thus shows that the induction base holds and completes the proof.

Appendix E
Proof of Theorem 1

We first consider the iterate $\boldsymbol{\gamma}^t$ at the t-th iteration of the coordinate descent with Bernoulli sampling (illustrated in Algorithm 2). Suppose that (19) holds for some $t \geq t_0$. We shall verify it for $t+1$. We start the analysis by computing the expected marginal decrease for ε in Algorithm 2,

$$\mathbb{E}\left[r_k^t|\boldsymbol{\gamma}^t\right] \geq (1-\varepsilon)\frac{1}{c_1\cdot NR}r_k^t + \varepsilon\frac{r_{k^\star}^t}{c}, \tag{30}$$

where $c_1 > 0$ is some finite constant and c is a finite constant defined in Theorem 1 and $k^\star = \arg\max_{k\in[NR]} r_k^t$. The expectation is with respect to the random choice of the algorithm which is characterized by the random variable ϵ.

For all $k \in [NR]$, it holds

$$\begin{aligned}\mathbb{E}\left[r_k^t|\boldsymbol{\gamma}^t\right] &\geq (1-\varepsilon)\frac{1}{c_1\cdot NR}\left(\sum_{\ell=1}^{NR}\frac{(r_\ell^t)^2}{NR}\right) + \varepsilon\frac{(r_{k^\star}^t)^2}{c} \\ &\geq (1-\varepsilon)\frac{\left(\sum_{\ell=1}^{NR} r_\ell^t\right)^2}{(NR)^2 c_1} + \varepsilon\frac{\left(\sum_{\ell=1}^{NR} r_\ell^t\right)^2}{\eta^2 c},\end{aligned} \tag{31}$$

where (31) follows from the assumption $\sum_\ell r_\ell^t \leq \eta r_{k^\star}^t$ in Theorem 1. Plugging the inequality $\epsilon(\boldsymbol{\gamma}^t) < \sum_\ell r_\ell^t$ in (31), it yields

$$\mathbb{E}\left[r_k^t|\boldsymbol{\gamma}^t\right] \geq \epsilon^2(\boldsymbol{\gamma}^t)\left(\frac{1-\varepsilon}{(NR)^2 c_1} + \frac{\varepsilon}{\eta^2 c}\right) = \frac{\epsilon^2(\boldsymbol{\gamma}^t)}{\alpha}. \tag{32}$$

Then, based on (32), the induction hypothesis is scrutinized by

$$\begin{aligned}\mathbb{E}[\epsilon(\boldsymbol{\gamma}^{t+1})] - \mathbb{E}[\epsilon(\boldsymbol{\gamma}^t)] \leq & -\mathbb{E}\left[r_k^t|\boldsymbol{\gamma}^t\right] \leq -\mathbb{E}\left[\frac{\epsilon^2(\boldsymbol{\gamma}^t)}{\alpha}\right] \\ & \leq -\frac{\mathbb{E}[\epsilon(\boldsymbol{\gamma}^t)]^2}{\alpha},\end{aligned} \tag{33}$$

where the last inequality is based on the Jensen's inequality (i.e., $\mathbb{E}[\epsilon(\boldsymbol{\gamma}^t)]^2 \leq \mathbb{E}[\epsilon^2(\boldsymbol{\gamma}^t)]$). By reformulating the terms in (33) we get

$$\mathbb{E}[\epsilon(\boldsymbol{\gamma}^{t+1})] \leq \mathbb{E}\left[\epsilon(\boldsymbol{\gamma}^t)\right]\left(1 - \frac{\mathbb{E}\left[\epsilon(\boldsymbol{\gamma}^t)\right]}{\alpha}\right). \tag{34}$$

Let $f(x) = x\left(1 - \frac{x}{\alpha}\right)$, as $f'(x) > 0$ for $x < \frac{\alpha}{2}$, and plugging $\mathbb{E}\left[\epsilon(\boldsymbol{\gamma}^t)\right] \leq \frac{\alpha}{1+t-t_0}$ in (34), it leads to the inductive step at time $t+1$:

$$\begin{aligned}\mathbb{E}[\epsilon(\boldsymbol{\gamma}^{t+1})] &\leq \mathbb{E}\left[\epsilon(\boldsymbol{\gamma}^t)\right]\left(1 - \frac{\mathbb{E}\left[\epsilon(\boldsymbol{\gamma}^t)\right]}{\alpha}\right) \\ &\leq \frac{\alpha}{1+t-t_0}\cdot\left(1 - \frac{1}{1+t-t_0}\right) \leq \frac{\alpha}{1+t+1-t_0}.\end{aligned} \tag{35}$$

We are left to show that the induction basis is satisfied. By using the inequality (33) for $t = 1, \ldots, t_0$ we get

$$\mathbb{E}[\epsilon(\boldsymbol{\gamma}^{t_0})] \leq \epsilon(\boldsymbol{\gamma}^0) - \sum_{t=0}^{t_0-1}\frac{\mathbb{E}[\epsilon(\boldsymbol{\gamma}^t)]^2}{\alpha}. \tag{36}$$

Since at each iteration the cost function decreases, we have $\epsilon(\boldsymbol{\gamma}^{t+1}) \leq \epsilon(\boldsymbol{\gamma}^t)$ for all $t \geq 0$. Hence, if $\mathbb{E}[\epsilon(\boldsymbol{\gamma}^t)] \leq \frac{\alpha}{2}$ for each $0 \leq t \leq t_0$, it concludes that $\mathbb{E}[\epsilon(\boldsymbol{\gamma}^{t_0})] \leq \frac{\alpha}{2}$. The induction hypothesis is justified via showing that $\mathbb{E}[\epsilon(\boldsymbol{\gamma}^{t_0})] > \frac{\alpha}{2}$ results in a contradiction. Under this assumption, (36) is reformulated as

$$\mathbb{E}[\epsilon(\boldsymbol{\gamma}^{t_0})] \leq \epsilon(\boldsymbol{\gamma}^0) - t_0\frac{\alpha}{2} = \epsilon(\boldsymbol{\gamma}^0)\left(1 - t_0\frac{\alpha}{2\epsilon(\boldsymbol{\gamma}^0)}\right). \tag{37}$$

Furthermore, based on the inequality $1 + x \leq \exp(x)$ with (37), we get

$$E[\epsilon(\boldsymbol{\gamma}^{t_0})] \leq \epsilon(\boldsymbol{\gamma}^0)\exp\left(-t_0\frac{\alpha}{2\epsilon(\boldsymbol{\gamma}^0)}\right). \tag{38}$$

We plug $t_0 = \frac{2\epsilon(\boldsymbol{\gamma}^0)}{\alpha}\log(\frac{\epsilon(\boldsymbol{\gamma}^0)}{\alpha})$ in (38) to get $E[\epsilon(\boldsymbol{\gamma}^{t_0})] \leq \alpha$, which completes the proof.

Then, we focus on the analysis of the iterate $\boldsymbol{\gamma}^t$ at the t-th iteration of the coordinate descent with random sampling (illustrated in Algorithm 1). Suppose that (19) holds for some $t \geq t_0$. We want to verify it for $t{+}1$. The analysis is begin with computing the expected marginal decrease for ε in Algorithm 1. For some constant $c_2 > 0$, there is $\mathbb{E}\left[r_k^t|\boldsymbol{\gamma}^t\right] \geq \frac{1}{c_2 \cdot NR} r_k^t$. For all $k \in [NR]$, it has

$$\mathbb{E}\left[r_k^t|\boldsymbol{\gamma}^t\right] \geq \frac{1}{c_2 \cdot NR}\left(\sum_{\ell=1}^{NR}\frac{(r_\ell^t)^2}{NR}\right) \geq \frac{\left(\sum_{\ell=1}^{NR} r_\ell^t\right)^2}{(NR)^2 c_2}. \tag{39}$$

We plug the inequality $\epsilon(\boldsymbol{\gamma}^t) < \sum_\ell r_\ell^t$ in (39), and get

$$\mathbb{E}\left[r_k^t|\boldsymbol{\gamma}^t\right] \geq \frac{\epsilon^2(\boldsymbol{\gamma}^t)}{(NR)^2 c_2}. \tag{40}$$

Based on (40) and Jensen's inequality to check the induction hypothesis

$$\mathbb{E}[\epsilon(\boldsymbol{\gamma}^{t+1})] - \mathbb{E}[\epsilon(\boldsymbol{\gamma}^t)] \leq \ \mathbb{E}\left[r_k^t|\boldsymbol{\gamma}^t\right] \leq -\frac{\mathbb{E}[\epsilon(\boldsymbol{\gamma}^t)]^2}{(NR)^2 c_2}. \tag{41}$$

The following proof for the convergence analysis for Algorithm 1 is similar to the proof for Algorithm 2 as discussed above. Hence, we omit the details here.

Appendix F
Proof of Theorem 2

In this section, we prove Theorem 2 which demonstrates the expected regret for the N-armed bandit problem in Algorithm 3. Recall that all arms are assumed to have Bernoulli distributed rewards, and that the first arm is the unique optimal arm.

Main technical arguments. Thompson sampling performs exploration by selecting the arm with the best sampled mean to play. Therein, sampled means are generated from beta distributions around the empirical means. As the number of plays of an arm increases, the beta distribution converges to the corresponding empirical mean. The main technical issue needed to be addressed in the analysis is that if the number of previous plays of the first arm is small, then the probability of playing the second arm will be as large as a constant even if it has already been played a large number of times. To address this, we introduce two types of arms, i.e., saturated and unsaturated arms, and bound the regret caused by each arm separately. Different from the previous analysis of Thompson sampling where the parameters of the beta distribution are required to be integral, i.e., [26], [41], our analysis applies to the beta distribution of which the parameters are not required to be integral, and the more general and natural forms are represented in (9) and (10).

Notaions. We take the inner 2-armed bandit problem in Algorithm 3 as an example to illustrate corresponding notations in our paper. We denote j_0 as the number of plays of the first arm until T_p plays of the second arm. Denote t_j as the time step where the j-th play of the first arm occurs (note that $t_0 = 0$). Furthermore, $Y_j = t_{j+1} - t_j - 1$ is defined to characterize the number of time steps between the j-th and $(j+1)$-th plays of the first arm. The random variable s_j is represented the number of successes in the first j plays of the first arm.

The random variable $X(j, s, y)$ is defined to characterize the expectation of Y_j. To begin with, considering perform an experiment until it succeeds: examine if a Beta$(s{+}R^j, j{-}s{+}R^j)$ distributed random variable surpasses a threshold y. Here, $R^j = r_{k_{t_j}}^{t_j} / F(\boldsymbol{\gamma}^{t_j})$ with $F(\boldsymbol{\gamma}^t)$ defined in (7) and $r_{k_{t_j}}^{t_j}$ defined in (8) is the reward obtained in Algorithm 3 when the first arm of the inner MAB is played. For each experiment, the beta-distributed random variables are generated independently of the previous ones. We define $X(j, s, y)$ as the number of trials before the experiment succeeds. Thus, $X(j, s, y)$ is a random variable with parameter (success probability) $1 - F_{s+R^j, j-s+R^j}^{beta}(y)$. Here $F_{\alpha,\beta}^{beta}$ denotes the cumulative distribution function (cdf) of the beta distribution with parameters α, β. Also, let $F_{n,p}^B$ denote the cdf of the *binomial* distribution with parameters (n, p).

Proof. At any step t, we divide the set of suboptimal arms into two subsets: *saturated* and *unsaturated*. The saturated arm i is the arm which have been played an enough large number ($L_i = c_L(\ln T)/\Delta_i^2$) for some large constant $c_L > 0$ of times. The set of saturated arms at time t is denoted as $C(t)$. Note that, for the set $C(t)$, with high probability, $\nu_i(t)$ is concentrated around μ_i. To bound the regret, we begin with estimating the number of steps between two consecutive plays of the first arm. After the j-th play of the first arm, the $(j{+}1)$-th play of the first arm will happen at the earliest time t where $\nu_1(t) > \nu_i(t), \forall i \neq 1$. The number of steps before $\nu_1(t)$ is larger than $\nu_i(t)$ of each saturated arm $a \in C(t)$, and can be tightly approximated via a geometric random variable with the parameter being $\Pr(\nu_1 \geq \max_{a \in C(t)} \mu_i)$. We justify that the expected number of steps until the $(j{+}1)$-th play can be upper bounded by the product of the expected value of a geometric random variable $X(j, s_j, \max_i \mu_i)$, if j plays of the first arm with s_j have succeeded. Additionally, the expected number of interruptions by the unsaturated arms is bounded by $\sum_{u=2}^N L_u$, since an arm u becomes saturated after L_u plays.

Based on the above discussion, the expected regret of the inner q-armed stochastic bandit problem in Algorithm 3 can be bounded by the regrets due to unsaturated arms at saturated arms, given by $\mathbb{E}[\mathcal{R}(T)] \leq \mathbb{E}[\mathcal{R}_{\text{uns}}(T)] + \mathbb{E}[\mathcal{R}_{\text{s}}(T)]$. Since an unsaturated arm u becomes saturated after L_u plays, the regret generated by unsaturated arms is bounded by

$$\mathbb{E}[\mathcal{R}_{\text{uns}}(T)] \leq \sum_{u=2}^N L_u d_u = c_L(\ln T)\left(\sum_{u=2}^N \frac{1}{d_u}\right), \tag{42}$$

for some large constant $c_L > 0$. Prior to bounding $\mathbb{E}[\mathcal{R}_{\text{s}}(T)]$, we introduce some notations. Denote θ_j as the total number of plays of unsaturated arms in the interval between (and excluding) the j^{th} and $(j+1)^{th}$ plays of the first arm. Recalling the definition of $X(j, s, y)$ and $d_i = \mu_1 - \mu_i$ defined Theorem 2, we then focus on the regret generated by playing saturated arms until $\sum_{i=2}^q L_i$ plays of the first arm. After sufficient plays, the first arm is sufficiently concentrated such that the probability of playing any saturated arm will be very small, which yields small regret. Thus, the regret due to the play of the saturated arm can be represented by

$$
\begin{aligned}
&\mathbb{E}[\mathcal{R}_{\mathrm{s}}(T)]\\
\leq& C\cdot\mathbb{E}\left[\sum_{j=0}^{\sum_i L_i}\mathbb{E}\left[(\theta_j+1)\,|s_j\right]\sum_a d_a\mathbb{E}\left[\min\left\{X\left(j,s_j,y_a\right),T\right\}|s_j\right]\right]\\
\leq& C\cdot\mathbb{E}\Bigg[\left(\sum_{j=0}^{\sum_i L_i}\mathbb{E}\left[(\theta_j+1)\,|s_j\right]\right)\cdot\\
&\left(\sum_{j=0}^{\sum_i L_i}\sum_a d_a\mathbb{E}\left[\min\left\{X\left(j,s_j,y_a\right),T\right\}|s_j\right]\right)\Bigg]\\
\leq& C\cdot\left(\sum_{i=2}^{q}L_i\right)\cdot\left[\left(\sum_{j=0}^{\sum_i L_i}\sum_i d_i\mathbb{E}\left[\min\left\{X\left(j,s_j,y_i\right),T\right\}|s_j\right]\right)\right],
\end{aligned}
$$

for some constant $C > 0$.

To complete the proof, the term $\mathbb{E}\left[\min\left\{X\left(j,s_j,y_i\right),T\right\}|s_j\right] = \frac{1}{1-F^{beta}_{s+R^j,j-s+R^j}(y)} - 1$ is required to be bounded. Our proof is inspired by the paper [26]. However, different from the previous analysis of Thompson sampling where the parameters of the beta distribution are required to be integral, i.e. [26], [41], our analysis applies to the beta distribution of which the parameters the beta distribution are in more general and natural form, represented in (9) and (10). Hence, it yields that $\mathbb{E}[\mathcal{R}_{\mathrm{s}}(T)] \leq C\left(\left(\sum_i L_i\right)^2\right) = C\cdot\left(\sum_i \frac{\log T}{d_i^2}\right)^2$. Hence, we conclude that

$$
\begin{aligned}
\mathbb{E}[\mathcal{R}(T)] \leq& \mathbb{E}[\mathcal{R}_{\mathrm{uns}}(T)] + \mathbb{E}[\mathcal{R}_{\mathrm{s}}(T)]\\
\leq& c_L(\ln T)\left(\sum_{u=2}^{N}\frac{1}{d_u}\right) + C\cdot\left(\sum_i\frac{\log T}{d_i^2}\right)^2\\
=& O\left(\left(\sum_{b=2}^{q}\frac{1}{d_b^2}\right)^2\ln T\right).
\end{aligned}
\tag{43}
$$

References

[1] J. Dong, J. Zhang, and Y. Shi, "Bandit sampling for faster activity and data detection in massive random access," in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, pp. 8319–8323, 2020.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, pp. 22–32, Feb. 2014.

[3] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Communications Magazine*, vol. 57, pp. 84–90, Aug. 2019.

[4] L. Liu, E. G. Larsson, W. Yu, P. Popovski, C. Stefanovic, and E. de Carvalho, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the Internet of Things," *IEEE Signal Process. Mag.*, vol. 35, pp. 88–99, Sep. 2018.

[5] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-advanced networks: Issues and approaches," *IEEE Commun. Mag.*, vol. 51, pp. 86–93, Jun. 2013.

[6] L. Liu, E. G. Larsson, W. Yu, P. Popovski, C. Stefanovic, and E. De Carvalho, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the Internet of Things," *IEEE Signal Process. Mag.*, vol. 35, pp. 88–99, Sep. 2018.

[7] A. Ghosh, J. Zhang, J. G. Andrews, and R. Muhamed, "Fundamentals of LTE," *Englewood Cliffs, NJ, USA: Prentice-Hall*, 2010.

[8] Z. Chen, F. Sohrabi, and W. Yu, "Sparse activity detection for massive connectivity," *IEEE Trans. Signal Process.*, vol. 66, pp. 1890–1904, Jan. 2018.

[9] Y. Shi, J. Dong, and J. Zhang, *Low-overhead Communications in IoT Networks — Structured Signal Processing Approaches*. Springer, 2020.

[10] Y. Wu, X. Gao, S. Zhou, W. Yang, Y. Polyanskiy, and G. Caire, "Massive access for future wireless communication systems," *IEEE Wirel. Commun.*, vol. 27, no. 4, pp. 148–156, 2020.

[11] K. Senel and E. G. Larsson, "Grant-free massive MTC-enabled massive MIMO: A compressive sensing approach," *IEEE Trans. Commun.*, vol. 66, pp. 6164–6175, Aug. 2018.

[12] Z. Chen, F. Sohrabi, Y. Liu, and W. Yu, "Covariance based joint activity and data detection for massive random access with massive MIMO," in *IEEE Int. Conf. Commun. (ICC)*, pp. 1–6, May 2019.

[13] H. Han, Y. Li, W. Zhai, and L. Qian, "A grant-free random access scheme for m2m communication in massive mimo systems," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3602–3613, 2020.

[14] U. K. Ganesan, E. Björnson, and E. G. Larsson, "Clustering based activity detection algorithms for grant-free random access in cell-free massive mimo," *IEEE Transactions on Communications*, 2021.

[15] Y. Cui, S. Li, and W. Zhang, "Jointly sparse signal recovery and support recovery via deep learning with applications in mimo-based grant-free random access," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 3, pp. 788–803, 2020.

[16] L. Liu and W. Yu, "Massive connectivity with massive MIMO part I: Device activity detection and channel estimation," *IEEE Trans. on Signal Process.*, vol. 66, pp. 2933–2946, Mar. 2018.

[17] S. Haghighatshoar, P. Jung, and G. Caire, "Improved scaling law for activity detection in massive MIMO systems," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 381–385, IEEE, 2018.

[18] Z. Chen and W. Yu, "Phase transition analysis for covariance based massive random access with massive MIMO," in *Asilomar Conf. Signals Syst. Comput.*, 2019.

[19] S. J. Wright, "Coordinate descent algorithms," *Math. Program.*, vol. 151, no. 1, pp. 3–34, 2015.

[20] S. Shalev-Shwartz and T. Zhang, "Accelerated mini-batch stochastic dual coordinate ascent," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, pp. 378–385, 2013.

[21] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," *J. Mach. Learn. Res.*, vol. 14, no. Feb, pp. 567–599, 2013.

[22] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke, "Coordinate descent converges faster with the Gauss-Southwell rule than random selection," in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 1632–1641, 2015.

[23] F. Salehi, P. Thiran, and E. Celis, "Coordinate descent with bandit sampling," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, pp. 9247–9257, 2018.

[24] D. Perekrestenko, V. Cevher, and M. Jaggi, "Faster coordinate descent via adaptive importance sampling," in *Proc. Int. Conf. Articial Intelligence and Statistics (AISTATS)*, pp. 869–877, 2017.

[25] P. Zhao and T. Zhang, "Stochastic optimization with importance sampling for regularized loss minimization," in *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 1–9, 2015.

[26] S. Agrawal and N. Goyal, "Analysis of Thompson sampling for the multi-armed bandit problem," in *Proc. Conference On Learning Theory (COLT)*, vol. 23, pp. 39.1–39.16, 2012.

[27] S. Bubeck, N. Cesa-Bianchi, *et al.*, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, pp. 1–122, Dec. 2012.

[28] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, "Learning-based context-aware resource allocation for edge-computing-empowered industrial iot," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4260–4277, 2020.

[29] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7108–7123, 2020.

[30] G. Wunder, H. Boche, T. Strohmer, and P. Jung, "Sparse signal processing concepts for efficient 5G system design," *IEEE Access*, vol. 3, pp. 195–208, Feb. 2015.

[31] C.-K. Wen, C.-J. Wang, S. Jin, K.-K. Wong, and P. Ting, "Bayes-optimal joint channel-and-data estimation for massive MIMO with low-precision ADCs," *IEEE Trans. Signal Process.*, vol. 64, pp. 2541–2556, Dec. 2015.

[32] T. Jiang, Y. Shi, J. Zhang, and K. B. Letaief, "Joint activity detection and channel estimation for IoT networks: Phase transition and computation-estimation tradeoff," *IEEE Internet Things J.*, vol. 6, pp. 6212–6225, Aug. 2019.

[33] L. Liu and W. Yu, "Massive connectivity with massive MIMO part II: Achievable rate characterization," *IEEE Trans. on Signal Process.*, vol. 66, pp. 2947–2959, Mar. 2018.

[34] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*, pp. 2249–2257, 2011.

[35] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine learning*, vol. 47, no. 2, pp. 235–256, 2002.

[36] J. Mo, P. Schniter, and R. W. Heath, “Channel estimation in broadband millimeter wave MIMO systems with few-bit ADCs,” *IEEE Trans. on Signal Process.*, vol. 66, pp. 1141–1154, Dec. 2017.

[37] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, “Throughput analysis of massive MIMO uplink with low-resolution ADCs,” *IEEE Wireless Commun.*, vol. 16, pp. 4038–4051, Jun. 2017.

[38] A. Fengler, S. Haghighatshoar, P. Jung, and G. Caire, “Non-bayesian activity detection, large-scale fading coefficient estimation, and unsourced random access with a massive mimo receiver,” *IEEE Trans. Inf. Theory*, vol. 67, no. 5, pp. 2925–2951, 2021.

[39] Y. Bengio, A. Lodi, and A. Prouvost, “Machine learning for combinatorial optimization: a methodological tour d’horizon,” *Eur. J. Oper. Res.*, 2020.

[40] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, “LORM: Learning to optimize for resource management in wireless networks with few training samples,” *IEEE Trans. Wireless Commun.*, vol. 19, pp. 665–679, Jan. 2020.

[41] S. L. Scott, “A modern Bayesian look at the multi-armed bandit,” *Appl. Stoch. Model Bus.*, vol. 26, no. 6, pp. 639–658, 2010.