Unsourced Multiple Access Based on Sparse Tanner Graph—Efficient Decoding, Analysis, and Optimization

Jiaai Liu and Xiaodong Wang, *Fellow, IEEE*

***Abstract*— We propose novel sparse-graph-based transmission schemes and receiver algorithms for unsourced multiple access (UMA) in MIMO channels. The channel coherence interval is divided into a number of sub-slots and each active transmitter selects certain sub-slots to repeatedly transmit its codeword according to a sparse Tanner graph. We propose iterative receiver algorithms that at each iteration decode either a single codeword, or two or three codewords jointly, and then subtract the decoded codewords from received signals during all sub-slots. The keys to these decoders are novel blind channel estimation algorithms when the received signal contains one, two, or three codewords. We perform density evolution analysis on the proposed UMA systems to obtain the asymptotic upper bounds on the maximum achievable rates for different decoders under both regular and irregular Tanner graphs. Extensive simulation results are provided to illustrate the performance of the proposed UMA systems, and its advantages over existing compressed-sensing (CS)-based UMA schemes.**

***Index Terms*— Unsourced multiple access (UMA), sparse Tanner graph, MIMO, blind channel estimation, clustering, density evolution.**

I. Introduction

THE anticipated proliferation of IoT devices and associated massive machine-type communications (mMTC) has motivated the development of access protocols that cater to the unique features of mMTC: massive and sporadic connectivity, small data payloads, low power and low latency [1], [2]. Given that protocols for traditional high-throughput few-user communication scenarios are not appropriate for mMTC, new massive multiple access schemes have been developed in recent years, which primarily fall into two categories: one is pilot-based grant-free non-orthogonal transmission with joint device activity detection and channel estimation [3], and the other is pilot-free unsourced multiple access (UMA). In this paper, we focus on UMA and propose transmission protocols based on sparse Tanner graphs and efficient decoding algorithms.

Manuscript received August 10, 2021; revised November 10, 2021; accepted December 16, 2021. Date of publication January 14, 2022; date of current version April 18, 2022. This work was supported in part by the U.S. National Science Foundation (NSF) under Grant CCF 1814803 and Grant SHF 7995357 and in part by the U.S. Office of Naval Research (ONR) under Grant N000142112155. *(Corresponding author: Xiaodong Wang.)*

The authors are with the Electrical Engineering Department, Columbia University, New York, NY 10027 USA (e-mail: wangx@ee.columbia.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2022.3143225.

Digital Object Identifier 10.1109/JSAC.2022.3143225

Early works on UMA include information theoretic studies in [4] and [5], and the T-fold ALOHA scheme [6] that combines compute-and-forward and coding for a binary adder channel. In UMA, all users share a same codebook and at any given time, a small number of active users select codewords from the common codebook to transmit. Hence the decoding can be viewed as a compressed sensing (CS) problem. That is, if each active transmitter wishes to transmit n bits of information, then the total number of codewords is 2^n and hence the common codebook, which in the CS setup, corresponds to the dictionary matrix $\boldsymbol{A}$, has a size of $T \times 2^n$, where T is the size of each transmitted codeword vector. The received signal is $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{n}$ is the noise vector, and $\boldsymbol{x}$ is a K-sparse vector whose non-zero elements indicate the codewords that are transmitted by active users. Since at any time, the number of active users K is small, i.e., $K \ll 2^n$, this is a standard CS problem. However, even though the data packet size is small, e.g., $n = 100$, the codebook size, i.e., 2^n, is still prohibitively large to be amenable to any sparse recovery algorithm. Hence various divide-and-conquer strategies are proposed by either dividing the information bits into sub-blocks or dividing the transmission period into sub-slots to reduce the dimension of the problem.

The existing low-complexity UMA schemes can be classified into two main categories. Methods in the first category divide the information bits into J sub-blocks and the transmission period into J sub-slots. Each encoded sub-block of information bits is transmitted in only one sub-slot. In [7]–[10], the j-th information sub-blocks of K active transmitters are independently encoded using an outer tree encoder and an inner CS encoder and transmitted in the j-th sub-slot. The CS decoder runs on each sub-slot independently and the tree decoder finds the dependence over sub-blocks to obtain the complete estimation of the information bits for each transmitter. In [11] each sub-block of information bits is independently encoded using any channel code and an CS encoder. In addition to CS decoding, the receiver resolves the permutation of sub-blocks based on channel clustering and a single-user decoding is applied to each transmitter to obtain the information bits. Since the CS encoding is applied to sub-blocks of size n/J, the codebook size is reduced to $2^{n/J}$ and hence the decoding complexity can be controlled by choosing proper value of J. On the other hand, the schemes in this category can result in codeword collisions; that is, in the

same transmission sub-slot, if the sub-blocks of more than one transmitters are the same, they would share the same CS codeword and lead to decoding error.

The second category of low-complexity UMA schemes [12]–[14] are closely related to the graph-based slotted ALOHA schemes in [15] and [16]. They divide the transmission period into sub-slots, and each transmitter select several sub-slots to repeatedly transmit its codeword. The decoder performs successive interference cancellation (SIC); that is, the decoded codeword in one sub-slot is subtracted from all other sub-slots that contains it. However, a small segment of the information bits are encoded using a small CS codebook in [12] and [13], which still causes the problem of codeword collision. Reference [14] also employs Tanner graphs and a peeling decoder based on single-tons, but it ignores the channel effect by adopting the unrealistic simple adder channel model. Other related works include [17], which makes use of pilot symbols, and [18], which makes use of tensor decomposition but still needs pilot symbols to resolve ambiguity. (Note that the special case of [18], i.e., matrix decomposition, was treated in [19]).

Note that most of the existing works ignore the channel effect in either CS decoding (first category) or SIC decoding (second category) by assuming all user channels take the same value of 1, which is unrealistic, with the exception of [9], [11] and [20]. In [9] the channels are averaged out by assuming a large number of receive antennas and performing CS decoding on the covariance matrix of the received signal. Reference [11] blindly estimates the channel and resolves the permutation of sub-blocks by clustering the estimated channels. Reference [20] proposes a general sparse recovery algorithm based on Tanner graph and peeling decoding. However, this method is not well suited for UMA application due to the high complexity at the transmitter side. In fact, channel estimation in UMA systems is a fundamental issue given that no pilot symbols are transmitted, which will be addressed in this paper.

In this paper, we propose a novel UMA system that falls into the second category. The transmission is based on a sparse Tanner graph where each transmitter transmits its codeword either a fixed number of times (regular graph) or variable number of times (irregular graph). We propose peeling decoders that at each iteration decode either a single codeword, or two or three codewords simultaneously. The key ingredients of these peeling decoders are the corresponding novel blind channel estimation algorithms. We then perform density evolution analysis on the asymptotic upper bounds on the maximum rates achievable by the proposed UMA systems, for both regular and irregular Tanner graphs. Finally extensive simulation results are provided to demonstrate the performance of the proposed UMA systems, and to compare with existing CS-based UMA schemes.

The remainder of this paper is organized as follows. In Section II we describe the proposed UMA transmission scheme based on sparse Tanner graphs. In Section III we develop decoding algorithms for the proposed UMA systems that perform blind channel estimation, symbol detection and codeword subtraction. In Section IV we analyze the asymptotic performance upper bounds of the proposed UMA systems in terms of the maximum achievable rate, using density evolution. Simulation results are provided in Section V. Finally Section VI concludes the paper.

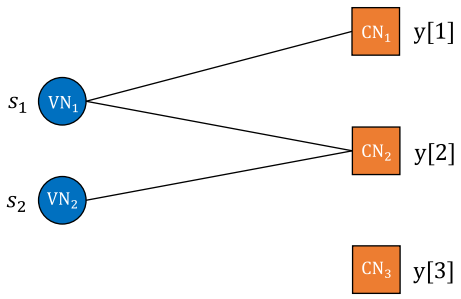

Fig. 1. An example of a (2, 3) Tanner graph.

II. Sparse Graph-Based UMA Transmission Scheme

Assume that there are totally K_{tot} transmitters in the network, and only K ($K \ll K_{tot}$) of them are transmitting data in any channel coherence time interval T using a common codebook. In particular, for a given coherence interval, we index the K active transmitters as $k = 1, 2, \ldots, K$. Each transmitter transmits M bits. Let $\boldsymbol{b}_k \in \{0,1\}^M$ denote the data bit vector of transmitter k, which is mapped to the transmitted signal $\boldsymbol{x}_k \in \mathbb{C}^T$ through a modulation process.

CS-Based UMA Transmission Scheme: For example, in the compressed-sensing (CS) mapping, the total coherence interval T is divided into J sub-intervals of length T/J. The data bit vector $\boldsymbol{b}_k$ is also divided into J blocks, $\boldsymbol{b}_k = [\boldsymbol{b}_k(1)^T, \ldots, \boldsymbol{b}_k(J)^T]^T$, where $\boldsymbol{b}_k(j) \in \{0,1\}^{M/J}$. Each bit block $\boldsymbol{b}_k(j)$ is then mapped to a symbol vector $\boldsymbol{x}_k(j) \in \mathbb{C}^{T/J}$. In particular, using a sensing matrix $\boldsymbol{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N] \in \mathbb{C}^{\frac{T}{J} \times N}$ with $N = 2^{M/J}$, $\boldsymbol{b}_k(j)$ is mapped to the $n_k(j)$-th column of $\boldsymbol{C}$, where $n_k(j)$ is the integer whose binary expansion is given by $\boldsymbol{b}_k(j)$. Since a total of MK data bits are transmitted in T channel uses, the transmission rate is $r = \frac{MK}{T}$.

In this section, we propose a new UMA transmission scheme based on the sparse Tanner graph. We first illustrate the basic idea through a toy example. Then we describe the proposed UMA transmission scheme.

A. An Illustrative Example

The encoding and decoding of the proposed UMA system can be viewed from the angle of a sparse-graph code, which is typically described by a Tanner graph. Recall that a (K, L) Tanner graph consists of K variable nodes (VNs), L check nodes (CNs) and some edges such that each edge connects a VN and a CN. The parity check matrix $\boldsymbol{H}$ of a Tanner graph is an $L \times K$ matrix where $H(j,i) = 1$ if the j-th CN is connected to the i-th VN and $H(j,i) = 0$ otherwise.

To illustrate the idea more clearly, we consider a toy example using a (2, 3) Tanner graph as shown in Figure 1. There are two transmitted bit vectors represented by the two VNs, and the coherence interval is divided into three sub-slots which are represented by the three CNs. If the i-th VN is connected

to the j-th CN, then the i-th bit vector is transmitted during the j-th sub-slot. The binary bit vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are first mapped to transmitted signals $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. Then $\boldsymbol{s}_i$ is repeatedly transmitted in the j-th sub-slots such that $H(j,i)=1, i=1,2$. In this example, bit vector $\boldsymbol{b}_1$ is transmitted in the first and the second sub-slots, and bit vector $\boldsymbol{b}_2$ is transmitted only in the second sub-slot. The received noise-free signals during the three sub-slots are given by

$$\boldsymbol{y}[1] = \boldsymbol{s}_1, \tag{1}$$

$$\boldsymbol{y}[2] = \boldsymbol{s}_1 + \boldsymbol{s}_2, \tag{2}$$

$$\boldsymbol{y}[3] = \boldsymbol{0}, \tag{3}$$

and the received signal during the entire coherence time is denoted as $\boldsymbol{y} = \left[\boldsymbol{y}[1]^T \boldsymbol{y}[2]^T \boldsymbol{y}[3]^T\right]^T$. We call the sub-slot, i.e., the CN, that contains no useful signal a zero-ton, the sub-slot that contains only a single signal a single-ton, and the sub-slot that contains the superposition of multiple signals a multi-ton. For example, in Figure 1, the three CNs are single-ton, multi-ton and zero-ton, respectively.

Assume that the receiver is able to detect single-tons and decode the information bits based on the received signals in the single-tons. Then it first detects $\boldsymbol{y}[1]$ as a single-ton and decodes the information bits of the bit vector $\hat{\boldsymbol{b}}_1 = \boldsymbol{b}_1$. Since the parity check matrix $\boldsymbol{H}$ is known to the decoder, the decoder can then subtract $\hat{\boldsymbol{s}}_1 = \boldsymbol{s}_1$ from all CNs that are connected to the first VN. In particular, the first and the second CNs are updated by $\boldsymbol{y}[1] \leftarrow \boldsymbol{y}[1] - \hat{\boldsymbol{s}}_1 = \boldsymbol{0}$ and $\boldsymbol{y}[2] \leftarrow \boldsymbol{y}[2] - \hat{\boldsymbol{s}}_1 = \boldsymbol{s}_2$ respectively. Note that $\boldsymbol{y}[1]$ becomes a zero-ton and $\boldsymbol{y}[2]$ becomes a new single-ton that can be detected and decoded in the next iteration. This way both transmitted bit vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are decoded.

In this example, the two transmitted signals during the coherence interval are given respectively by

$$\boldsymbol{x}_1 = [\boldsymbol{s}_1^T, \boldsymbol{s}_1^T, \boldsymbol{0}^T]^T \in \mathbb{C}^T, \tag{4}$$

$$\boldsymbol{x}_2 = [\boldsymbol{0}^T, \boldsymbol{s}_2^T, \boldsymbol{0}^T]^T \in \mathbb{C}^T, \tag{5}$$

and the received signal is $\boldsymbol{y} = \boldsymbol{x}_1 + \boldsymbol{x}_2$. The parity check matrix is given by

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}. \tag{6}$$

Hence we have $\boldsymbol{x}_k = \boldsymbol{h}_k \otimes \boldsymbol{s}_k$ where $\boldsymbol{h}_k$ is the k-th column of $\boldsymbol{H}$, and $\otimes$ denotes the Kronecker product. In general, to specify the UMA transmission scheme, we need to design the mapping from the bit vector $\boldsymbol{b}_k$ to the transmitted signal $\boldsymbol{s}_k$, as well as the parity-check matrix $\boldsymbol{H}$, which is discussed next.

B. UMA Transmission Based on Tanner Graph

In the proposed UMA scheme based on Tanner graph, the total coherence interval T is divided into L sub-slots of duration $M+1$, i.e., $T=(M+1)L$. Each VN in the Tanner graph corresponds to a data bit vector $\boldsymbol{b} \in \{0,1\}^M$, and each CN represents one transmission sub-slot. We will specify the column $\boldsymbol{h}(\boldsymbol{b}) \in \{0,1\}^L$ of the parity-check matrix $\boldsymbol{H}$ corresponding to $\boldsymbol{b}$ in this section (see Eq. (9)). Then during the ℓ-th transmission sub-slot, messages in the set $\{\boldsymbol{b} : \boldsymbol{h}(\boldsymbol{b})[\ell]=1\}$ are transmitted simultaneously, $\ell = 1,\ldots,L$.

To transmit the data bit vector $\boldsymbol{b} \in \{0,1\}^M$, we first convert it to the antipodal form and then append an additional "1" to form the transmitted signal

$$\boldsymbol{s}(\boldsymbol{b}) = \sqrt{P}[(2\boldsymbol{b}-\boldsymbol{1})^T, 1]^T \in \{\sqrt{P}, -\sqrt{P}\}^{M+1}, \tag{7}$$

where P is the transmit power. The additional "1" appended is to resolve the sign ambiguity that is inherent to any blind receiver. That is, let $g \in \mathbb{C}$ be a scalar channel coefficient, the receiver is able to tell between $g\boldsymbol{s}$ and $-g(-\boldsymbol{s})$ using the appended bit 1.

In the proposed UMA system, since the receiver only decodes the transmitted messages without knowing the transmitter IDs, the VNs in the Tanner graph correspond to different transmitted messages, rather than different transmitters. We use a d-regular (K,L) Tanner graph where each VN has an edge degree of d, i.e., it connects to d CNs which are selected uniformly. Hence there are d entries being 1 and $(L-d)$ entries being 0 in each column of the parity-check matrix $\boldsymbol{H}$. Given a VN, i.e., a data bit vector $\boldsymbol{b} \in \{0,1\}^M$, we need to specify which d CNs it connects to. To that end, we enumerate all distinct d-selections of indices from $\{1,2,\ldots,L\}$ as $\mathcal{I} = \{\mathcal{I}(i), i=0,1,\ldots,\binom{L}{d}-1\}$. Each element of $\mathcal{I}$ is an instance of the d-selection, e.g., $\mathcal{I}(0)=\{1,2,\ldots,d\}$. Assume that the data bit vectors $\boldsymbol{b}$ are equiprobable, then the elements of $\mathcal{I}$ are uniformly distributed. Define $p \triangleq \binom{L}{d}$ and denote $\mathrm{dec}(\boldsymbol{b})$ as the decimal form of the binary vector $\boldsymbol{b}$, i.e., the integer whose binary expansion is $\boldsymbol{b}$. Then the set of check nodes to which bit vector $\boldsymbol{b}$ is connected, or equivalently, the set of time sub-slots during which $\boldsymbol{b}$ is transmitted, is given by

$$\mathcal{I}\left(\mathrm{dec}(\boldsymbol{b}) \mod p\right). \tag{8}$$

We further define a mapping from the set $\mathcal{I}(i)$ to a binary vector $\boldsymbol{e}(\mathcal{I}(i)) \in \{0,1\}^L$, such that the ℓ-th entry of $\boldsymbol{e}(\mathcal{I}(i))$ is 1 if $\ell \in \mathcal{I}(i)$, and zero otherwise, $\ell = 1,\ldots,L$. Then the column of the parity check matrix $\boldsymbol{H}$ corresponding to the bit vector $\boldsymbol{b}$ is given by

$$\boldsymbol{h}(\boldsymbol{b}) = \boldsymbol{e}\left(\mathcal{I}(\mathrm{dec}(\boldsymbol{b}) \mod p)\right). \tag{9}$$

Then the transmitted signal during the coherence interval corresponding to $\boldsymbol{b}$ is given by

$$\boldsymbol{x}(\boldsymbol{b}) = \boldsymbol{h}(\boldsymbol{b}) \otimes \boldsymbol{s}(\boldsymbol{b}) \in \{-1,0,1\}^T, \tag{10}$$

where $T = L(M+1)$. Since a total of KM bits are transmitted over the coherence interval $T=(M+1)L$, the transmission rate is $r = \frac{KM}{(M+1)L} \approx \frac{K}{L}$.

We consider an uplink system where each transmitter has a single transmit antenna, and the base station employs N_a receive antennas. Denote $\boldsymbol{g}_k \in \mathbb{C}^{N_a}$ as the channel vector between the k-th transmitter and the base station. The signal arriving at the N_a base station receive antennas during the entire coherence interval is given by

$$\boldsymbol{Y} = [\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_L] = \sum_{k=1}^{K} \boldsymbol{g}_k \boldsymbol{x}_k^T + \boldsymbol{Z} \in \mathbb{C}^{N_a \times T}, \tag{11}$$

where $\boldsymbol{Z} \in \mathbb{C}^{N_a \times T}$ is the additive white Gaussian noise (AWGN) consisting of i.i.d. $\mathcal{N}_c(0, \sigma^2)$ entries, and $\boldsymbol{Y}_\ell \in \mathbb{C}^{N_a \times (M+1)}$ is the received signal during the ℓ-th sub-slot.

In the next section, we propose novel algorithms that can recover the transmitted bit vectors $\boldsymbol{b}_k, k = 1, \ldots, K$ of all active transmitters based on the received signal $\boldsymbol{Y}$. Note that as typical in conventional communication systems, each bit vector $\boldsymbol{b}_k$ can be the output of some channel encoder, i.e., $\boldsymbol{b}_k = \text{enc}\{\boldsymbol{d}_k\}$ for some data vector $\boldsymbol{d}_k$. Then for each recovered coded bit vector $\hat{\boldsymbol{b}}_k$, by performing channel decoding, we can obtain the information data $\hat{\boldsymbol{d}}_k$.

III. Peeling UMA Decoders

According to the proposed transmission scheme, $\boldsymbol{Y}_\ell$ is a noisy linear combination of signals transmitted in the ℓ-th sub-slots. Similarly to the example in Section II, the decoder sequentially checks each signal sub-block to see if the information can be decoded. First we need to check if the sub-block is a zero-ton. Note that if the ℓ-th sub-slot is a zero-ton, then $\boldsymbol{Y}_\ell$ contains $(M+1)N_a$ i.i.d. complex Gaussian noise $\mathcal{N}_c(0, \sigma^2)$ samples, i.e., $\boldsymbol{Y}_\ell = \boldsymbol{Z}_\ell$ and hence $\frac{2}{\sigma^2}\|\boldsymbol{Y}_\ell\|_{\mathcal{F}}^2 \sim \chi^2_{2(M+1)N_a}$, i.e., χ^2 distribution with the degree of freedom $2(M+1)N_a$. If it is not a zero-ton, then each element of $\boldsymbol{Y}_\ell$ contains both the noise and signal samples. Hence an energy threshold detector can be employed for zero-ton detection, i.e., if

$$\frac{2}{\sigma^2}\|\boldsymbol{Y}_\ell\|_{\mathcal{F}}^2 \leq \tau, \tag{12}$$

then sub-slot ℓ is declared as a zero-ton. The threshold τ should chosen such that the detection probability is high and the false alarm probability is low. We specify the procedure for choosing τ in Section V-A. If the sub-block is not a zero-ton, we need to decide if it can be decoded as a single-ton, a double-ton, or a triple-ton.

A. Peeling Decoding Based on Single-Tons

For each CN, i.e., the received signal in each sub-block, we test whether or not it is a single-ton. If it is, then we decode the transmitted signal $\boldsymbol{s}$ and estimate the corresponding channel $\boldsymbol{g}$. Specifically, if the ℓ-th sub-block is a single-ton, the received signal $\boldsymbol{Y}_\ell$ is written as

$$\boldsymbol{Y}_\ell = \boldsymbol{g}\boldsymbol{s}^T + \boldsymbol{Z}_\ell \in \mathbb{C}^{N_a \times (M+1)}, \tag{13}$$

and each entry of $\boldsymbol{Y}_\ell$ is given by

$$\begin{aligned} Y_\ell[n,m] &= g[n]s[m] + Z_\ell[n,m], \\ &\qquad n = 1, \ldots, N_a, \quad m = 1, \ldots, M, \end{aligned} \tag{14}$$

$$Y_\ell[n, M+1] = \sqrt{P}g[n] + Z_\ell[n, M+1], \quad n = 1, \ldots, N_a. \tag{15}$$

Hence we have

$$\begin{aligned} &Y_\ell[n,m] \cdot Y_\ell[n, M+1]^* \\ &\quad = \sqrt{P}|g[n]|^2 s[m] + \sqrt{P}g[n]^* Z_\ell[n,m] \\ &\qquad + g[n]s[m]Z_\ell[n, M+1]^* + Z_\ell[n,m]Z_\ell[n, M+1]^*. \end{aligned} \tag{16}$$

Algorithm 1 UMA Decoder Based on Single-Tons

Input: Received signal $\boldsymbol{Y}$, number of sub-blocks L, d-selection set $\mathcal{I} = \{I(0), \ldots, I(p-1)\}$
Output: Set of estimated bit vectors $\mathcal{B}$.
Initialization: $\mathcal{B} \leftarrow \emptyset$;
repeat
  for $\ell = 1, \ldots, L$ **do**
    if $\boldsymbol{Y}_\ell$ is not a zero-ton **then**
      Compute the estimate $\hat{\boldsymbol{s}}$, $\hat{\boldsymbol{g}}$ and $\hat{\boldsymbol{b}}$ using (17), (18) and (19) respectively;
      if (20) holds (i.e., $\boldsymbol{Y}_{\boldsymbol{\ell}}$ is a single-ton) **then**
        $\mathcal{B} \leftarrow \mathcal{B} \cup \{\hat{\boldsymbol{b}}\}$;
        Update $\boldsymbol{Y}$ using (21);
      end
    end
  end
until no single-ton is detected;

Since $s[m]$ is antipodal, its sign can be estimated as

$$\hat{s}[m] = \text{sign}\left[\sum_{n=1}^{N_a} \Re\Big\{Y_\ell[n,m]Y_\ell[n, M+1]^*\Big\}\right], \quad m = 1, \ldots, M, \tag{17}$$

where $\Re\{\cdot\}$ is the real operator. Once $\hat{\boldsymbol{s}}$ is obtained, we then estimate the effective channel $\sqrt{P}\boldsymbol{g}$ as

$$\hat{\boldsymbol{g}} = \frac{1}{(M+1)}\boldsymbol{Y}_\ell \hat{\boldsymbol{s}} \in \mathbb{C}^{N_a}, \tag{18}$$

and the data bit vector $\boldsymbol{b}$ is estimated by

$$\hat{\boldsymbol{b}} = \frac{1}{2}(\hat{\boldsymbol{s}}[1:M] + \boldsymbol{1}) \in \{0,1\}^M. \tag{19}$$

If $\boldsymbol{Y}_\ell$ is indeed a single-ton, then by subtracting the estimated signals from it, the residual signal becomes a zero-ton. Hence we can determine that the ℓ-th sub-block is a single-ton if

$$\frac{2}{\sigma^2}\|\boldsymbol{Y}_\ell - \hat{\boldsymbol{g}}\hat{\boldsymbol{s}}^T\|_{\mathcal{F}}^2 < \tau. \tag{20}$$

Using (10)-(11) the decoded signal can then be subtracted from the received signal by

$$\boldsymbol{Y} \leftarrow \boldsymbol{Y} - \hat{\boldsymbol{g}}\left(\boldsymbol{e}(\mathcal{I}(\text{dec}(\hat{\boldsymbol{b}}) \mod p)) \otimes \hat{\boldsymbol{s}}\right)^T. \tag{21}$$

And the above single-ton detection and decoding process repeats until no more single-ton is detected in all CNs. The single-ton based UMA decoder is summarized in Alg. 1. Note that the complexity of this decoding algorithm is $\mathcal{O}(KM)$, i.e., linear in terms of both the number of active transmitters K and the data packet size M. Note that in the CS-based transmission scheme, the CS decoding of each sub-block has a complexity of $\mathcal{O}(2^{\frac{M}{J}})$.

Remark: In order for the above UMA decoder to successfully decode all transmitted codewords, the original sparse Tanner graph should be such that after subtracting each decoded codeword, there is at least one single-ton. We next show that this indeed holds with high probability.

Recall that in our transmission scheme each transmitter (VN) is connected to d sub-slots (CN), and each sub-slot is randomly and independently selected by each transmitter with probability d/L. Then the probability of each CN being a single-ton before the decoding process starts is

$$\beta_1 = K(d/L)(1-d/L)^{K-1}. \tag{22}$$

Therefore the probability that there is at least one single-ton at the beginning is $\gamma_1 = 1-(1-\beta_1)^L$. Similarly, after $k-1$ codewords have been decoded and the corresponding VNs are removed, the probability of each CN being a single-ton becomes $\beta_k = (K-k+1)(d/L)(1-d/L)^{K-k}$ and the probability that there is at least one single-ton before decoding the k-th codeword is $\gamma_k = 1-(1-\beta_k)^L, k = 1,\ldots,K$. Figure 2 shows the probability γ_k over k for different combinations of d and K when we fix $L = 40$. It is seen that for a given d, we should choose K properly to ensure that the probability γ_k is always close to 1. For example, when $d = 2$, $K = 80$ leads to $\gamma_k > 0.95$. But when $d = 3$, $K = 80$ leads to low values of γ_k for $k < 30$, which means that Alg. 1 may fail to decode all codewords because it may not be able to find a single-ton for iterations $k < 30$. On the other hand, for small values of K, e.g., $K = 32$, $\gamma_k \approx 1$ for both $d = 2$ and $d = 3$.

B. Peeling Decoding Based on Double-Tons

In Alg. 1 at each iteration we only look for single-tons to decode. For better decoding performance and at an increased complexity, we can look for both single-tons and double-tons to decode. In particular, if the ℓ-th sub-block is a double-ton, the received signal is written as

$$\boldsymbol{Y}_\ell = \boldsymbol{g}_1\boldsymbol{s}_1^T + \boldsymbol{g}_2\boldsymbol{s}_2^T + \boldsymbol{Z}_\ell \in \mathbb{C}^{N_a\times(M+1)}, \tag{23}$$

and each column of $\boldsymbol{Y}_\ell$ is given by

$$\boldsymbol{Y}_\ell[:,m] = \begin{cases} \sqrt{P}(\boldsymbol{g}_1+\boldsymbol{g}_2)+\boldsymbol{Z}_\ell[:,m] \\ \quad \text{if } \mathrm{sign}(s_1[m])=1,\ \mathrm{sign}(s_2[m])=1, \\ \sqrt{P}(\boldsymbol{g}_1-\boldsymbol{g}_2)+\boldsymbol{Z}_\ell[:,m] \\ \quad \text{if } \mathrm{sign}(s_1[m])=1,\ \mathrm{sign}(s_2[m])=-1, \\ \sqrt{P}(-\boldsymbol{g}_1+\boldsymbol{g}_2)+\boldsymbol{Z}_\ell[:,m] \\ \quad \text{if } \mathrm{sign}(s_1[m])=-1,\ \mathrm{sign}(s_2[m])=1, \\ \sqrt{P}(-\boldsymbol{g}_1-\boldsymbol{g}_2)+\boldsymbol{Z}_\ell[:,m] \\ \text{if } \mathrm{sign}(s_1[m])=-1,\ \mathrm{sign}(s_2[m])=-1, \end{cases} \quad m=1,\ldots,M, \tag{24}$$

and

$$\boldsymbol{Y}_\ell[:,M+1] = \sqrt{P}(\boldsymbol{g}_1+\boldsymbol{g}_2)+\boldsymbol{Z}_\ell[:,M+1]. \tag{25}$$

Hence the columns of $\boldsymbol{Y}_\ell$ form four clusters in $\mathbb{C}^{N_a}$ with centers $\{\boldsymbol{c}_1 \triangleq \sqrt{P}(\boldsymbol{g}_1+\boldsymbol{g}_2), \boldsymbol{c}_2 \triangleq \sqrt{P}(\boldsymbol{g}_1-\boldsymbol{g}_2), -\boldsymbol{c}_1, -\boldsymbol{c}_2\}$. If we define

$$\tilde{\boldsymbol{Y}}_\ell = \mathrm{sign}\,(\Re\{\boldsymbol{Y}_\ell\}) \circ \boldsymbol{Y}_\ell, \tag{26}$$

where $\circ$ is the Hadamard product operator, then the columns of $\tilde{\boldsymbol{Y}}_\ell$ form two clusters with centers $\tilde{\boldsymbol{c}}_1 = \mathrm{sign}\,(\Re\{\boldsymbol{c}_1\}) \circ \boldsymbol{c}_1$ and $\tilde{\boldsymbol{c}}_2 = \mathrm{sign}\,(\Re\{\boldsymbol{c}_2\}) \circ \boldsymbol{c}_2$. We can use a clustering algorithm, e.g., K-means, to partition the $M+1$ columns of $\tilde{\boldsymbol{Y}}_\ell$ into

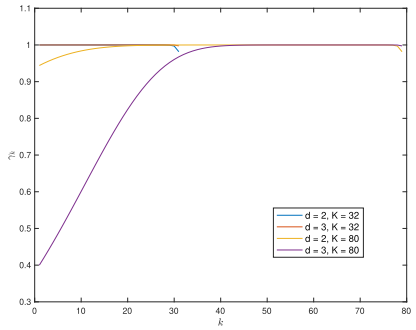

Fig. 2. The probability γ_k that there is at least one single-ton before decoding the k-th codeword in Alg. 1.

two clusters. Let $\mathcal{J}_1$ and $\mathcal{J}_2$ be the index sets of the columns that belong to clusters 1 and 2 respectively. From (25) we can estimate $\mathrm{sign}\,(\Re\{\boldsymbol{c}_1\})$ as $\mathrm{sign}\,(\Re\{\boldsymbol{Y}_\ell[:,M+1]\})$. Then the estimate of $\boldsymbol{c}_1$ is given by

$$\hat{\boldsymbol{c}}_1 = \mathrm{sign}\,(\Re\{\boldsymbol{Y}_\ell[:,M+1]\}) \circ \left(\sum_{j\in\mathcal{J}_1} \tilde{\boldsymbol{Y}}_\ell[:,j]\right) / |\mathcal{J}_1|. \tag{27}$$

Next note that the columns of $\boldsymbol{Y}_\ell$ indexed by $\mathcal{J}_2$, i.e., $\{\boldsymbol{Y}_\ell[:,j], j\in\mathcal{J}_2\}$ form two clusters centered at $\sqrt{P}(\boldsymbol{g}_1-\boldsymbol{g}_2)$ and $\sqrt{P}(\boldsymbol{g}_2-\boldsymbol{g}_1)$. Then by adding $\boldsymbol{c}_1 = \sqrt{P}(\boldsymbol{g}_1+\boldsymbol{g}_2)$ to these columns, the resulting set $\{\boldsymbol{Y}_\ell[:,j]+\boldsymbol{c}_1, j\in\mathcal{J}_2\}$ forms two clusters centered at $2\sqrt{P}\boldsymbol{g}_1$ and $2\sqrt{P}\boldsymbol{g}_2$. Hence by clustering this set the estimates $\hat{\boldsymbol{g}}_1$ and $\hat{\boldsymbol{g}}_2$ of the two effective channels $\sqrt{P}\boldsymbol{g}_1$ and $\sqrt{P}\boldsymbol{g}_2$ can be obtained.

Given the estimated channels $\hat{\boldsymbol{g}}_1$ and $\hat{\boldsymbol{g}}_2$, we can estimate $s_1[m]$ and $s_2[m]$ as

$$\begin{aligned}&(\hat{s}_1[m],\hat{s}_2[m]) \\ &\quad = \arg\min_{(s_1,s_2)\in\{-1,1\}^2} \|\boldsymbol{Y}_\ell[:,m]-\hat{\boldsymbol{g}}_1 s_1-\hat{\boldsymbol{g}}_2 s_2\|^2, \\ &\qquad\qquad m=1,\ldots,M.\end{aligned} \tag{28}$$

Finally, the ℓ-th sub-block is determined as a double-ton if

$$\frac{2}{\sigma^2}\|\boldsymbol{Y}_\ell-\hat{\boldsymbol{g}}_1\hat{\boldsymbol{s}}_1^T-\hat{\boldsymbol{g}}_2\hat{\boldsymbol{s}}_2^T\|_{\mathcal{F}}^2 < \tau. \tag{29}$$

The double-ton detection and decoding is summarized in Alg. 2(a). The decoded double-ton can be subtracted from the received signal by

$$\begin{aligned}\boldsymbol{Y} \leftarrow \boldsymbol{Y} &- \hat{\boldsymbol{g}}_1\left(\boldsymbol{h}(\mathcal{I}(\mathrm{dec}(\hat{\boldsymbol{b}}_1) \mod p))\otimes\hat{\boldsymbol{s}}_1\right)^T \\ &- \hat{\boldsymbol{g}}_2\left(\boldsymbol{h}(\mathcal{I}(\mathrm{dec}(\hat{\boldsymbol{b}}_2) \mod p))\otimes\hat{\boldsymbol{s}}_2\right)^T,\end{aligned} \tag{30}$$

where the estimates of the bit vectors are obtained by

$$\hat{\boldsymbol{b}}_i = \frac{1}{2}(\hat{\boldsymbol{s}}_i[1:M]+\boldsymbol{1}), \quad i=1,2. \tag{31}$$

C. Peeling Decoding Based on Triple-Tons

For a triple-ton, the received signal is given by

$$\boldsymbol{Y}_\ell = \boldsymbol{g}_1\boldsymbol{s}_1^T + \boldsymbol{g}_2\boldsymbol{s}_2^T + \boldsymbol{g}_3\boldsymbol{s}_3^T + \boldsymbol{Z}_\ell \in \mathbb{C}^{N_a\times(M+1)}. \tag{32}$$

Algorithm 2(a) Detection and Decoding of a Double-Ton

Input: signal sub-block $\boldsymbol{Y}_\ell$
Output: Estimated signals $\hat{\boldsymbol{s}}_1$ and $\hat{\boldsymbol{s}}_2$, channels $\hat{\boldsymbol{g}}_1$ and $\hat{\boldsymbol{g}}_2$, and bit vectors $\hat{\boldsymbol{b}}_1$ and $\hat{\boldsymbol{b}}_2$
Compute $\tilde{\boldsymbol{Y}}_\ell$ using (26) and partition the columns of $\tilde{\boldsymbol{Y}}_\ell$ into two clusters with the corresponding index sets $\mathcal{J}_1$ and $\mathcal{J}_2$;
Compute $\hat{\boldsymbol{c}}_1$ using (27);
Partition the columns of $\{\boldsymbol{Y}_\ell[:,j]+\hat{\boldsymbol{c}}_1, j\in\mathcal{J}_2\}$ into two clusters with centers $2\hat{\boldsymbol{g}}_1$ and $2\hat{\boldsymbol{g}}_2$;
Compute $\hat{\boldsymbol{s}}_1$ and $\hat{\boldsymbol{s}}_2$ using (28);
Compute $\hat{\boldsymbol{b}}_1$ and $\hat{\boldsymbol{b}}_2$ using (31).

The decoding of triple-tons is similar but more complex compared with the decoding of double-tons. The columns of $\boldsymbol{Y}_\ell$ form eight clusters in $\mathbb{C}^{N_a}$ with centers $\{\boldsymbol{c}_1 \triangleq \sqrt{P}(\boldsymbol{g}_1+\boldsymbol{g}_2+\boldsymbol{g}_3), \boldsymbol{c}_2 \triangleq \sqrt{P}(-\boldsymbol{g}_1+\boldsymbol{g}_2+\boldsymbol{g}_3), \boldsymbol{c}_3 \triangleq \sqrt{P}(\boldsymbol{g}_1-\boldsymbol{g}_2+\boldsymbol{g}_3), \boldsymbol{c}_2 \triangleq \sqrt{P}(\boldsymbol{g}_1+\boldsymbol{g}_2-\boldsymbol{g}_3), -\boldsymbol{c}_1, -\boldsymbol{c}_2, -\boldsymbol{c}_3, -\boldsymbol{c}_4\}$. Since each two of the centers are opposite to each other, we can partition the columns of $\tilde{\boldsymbol{Y}}_\ell$ in (26) into four clusters with centers $\{\tilde{\boldsymbol{c}}_i = \text{sign}\,(\Re\{\boldsymbol{c}_i\})\circ\boldsymbol{c}_i, i=1,2,3,4\}$, and the corresponding index sets $\{\mathcal{J}_1,\mathcal{J}_2,\mathcal{J}_3,\mathcal{J}_4\}$. Since $\boldsymbol{Y}_\ell[:,M+1]=\sqrt{P}(\boldsymbol{g}_1+\boldsymbol{g}_2+\boldsymbol{g}_3)+\boldsymbol{Z}_\ell[:,M+1]$, the estimate of $\boldsymbol{c}_1$ is given by

$$\hat{\boldsymbol{c}}_1 = \text{sign}\,(\Re\{\boldsymbol{Y}_\ell[:,M+1]\})\circ\left(\sum_{i\in\mathcal{J}_1}\tilde{\boldsymbol{Y}}_\ell[:,i]\right)/|\mathcal{J}_1|. \quad (33)$$

The columns of $\boldsymbol{Y}_\ell$ indexed by $\mathcal{J}_2$ form two clusters centered at $\sqrt{P}(-\boldsymbol{g}_1+\boldsymbol{g}_2+\boldsymbol{g}_3)$ and $\sqrt{P}(\boldsymbol{g}_1-\boldsymbol{g}_2-\boldsymbol{g}_3)$. Then by adding $\boldsymbol{c}_1$ to these columns, the resulting set $\{\boldsymbol{Y}_\ell[:,j]+\boldsymbol{c}_1, j\in\mathcal{J}_2\}$ forms two clusters centered at $2\sqrt{P}\boldsymbol{g}_1$ and $2\sqrt{P}(\boldsymbol{g}_2+\boldsymbol{g}_3)$, respectively. By clustering it we obtain a pair of estimates $\mathcal{Q}_1=\{\hat{\boldsymbol{g}}_1, \hat{\boldsymbol{g}}_2+\hat{\boldsymbol{g}}_3\}$ of $\sqrt{P}\boldsymbol{g}_1$ and $\sqrt{P}(\boldsymbol{g}_2+\boldsymbol{g}_3)$ although we cannot tell which one is $\hat{\boldsymbol{g}}_1$. Similarly, the set $\{\boldsymbol{Y}_\ell[:,j]+\boldsymbol{c}_1, j\in\mathcal{J}_3\}$ forms two clusters centered at $2\sqrt{P}\boldsymbol{g}_2$ and $2\sqrt{P}(\boldsymbol{g}_1+\boldsymbol{g}_3)$, respectively, and $\{\boldsymbol{Y}_\ell[:,j]+\boldsymbol{c}_1, j\in\mathcal{J}_4\}$ forms two clusters centered at $2\sqrt{P}\boldsymbol{g}_3$ and $2\sqrt{P}(\boldsymbol{g}_1+\boldsymbol{g}_2)$, respectively. By clustering these two sets we obtain two more pairs of estimates as $\mathcal{Q}_2=\{\hat{\boldsymbol{g}}_2, \hat{\boldsymbol{g}}_1+\hat{\boldsymbol{g}}_3\}$ and $\mathcal{Q}_3=\{\hat{\boldsymbol{g}}_3, \hat{\boldsymbol{g}}_1+\hat{\boldsymbol{g}}_2\}$.

To identify $\hat{\boldsymbol{g}}_1$ in $\mathcal{Q}_1$, we note that $\mathcal{Q}_1$ and $\mathcal{Q}_2+\mathcal{Q}_3=\{\hat{\boldsymbol{g}}_2+\hat{\boldsymbol{g}}_3, \hat{\boldsymbol{g}}_1+2\hat{\boldsymbol{g}}_3, \hat{\boldsymbol{g}}_1+2\hat{\boldsymbol{g}}_2, 2\hat{\boldsymbol{g}}_1+\hat{\boldsymbol{g}}_2+\hat{\boldsymbol{g}}_3\}$ share the common element $\hat{\boldsymbol{g}}_2+\hat{\boldsymbol{g}}_3$. Hence we can identify $\hat{\boldsymbol{g}}_2+\hat{\boldsymbol{g}}_3$ as the element in $\mathcal{Q}_1$ that has the minimum distance to the set $\mathcal{Q}_2+\mathcal{Q}_3$, and then $\hat{\boldsymbol{g}}_1$ is the other element of $\mathcal{Q}_1$, i.e.,

$$\hat{\boldsymbol{g}}_1 = \mathcal{Q}_1 - \arg\min_{\boldsymbol{q}\in\mathcal{Q}_1}\min_{\boldsymbol{p}\in\mathcal{Q}_2+\mathcal{Q}_3}\|\boldsymbol{q}-\boldsymbol{p}\|^2. \quad (34)$$

Similarly we can identify $\hat{\boldsymbol{g}}_2$ and $\hat{\boldsymbol{g}}_3$ as

$$\hat{\boldsymbol{g}}_2 = \mathcal{Q}_2 - \arg\min_{\boldsymbol{q}\in\mathcal{Q}_2}\min_{\boldsymbol{p}\in\mathcal{Q}_1+\mathcal{Q}_3}\|\boldsymbol{q}-\boldsymbol{p}\|^2, \quad (35)$$

$$\hat{\boldsymbol{g}}_3 = \mathcal{Q}_3 - \arg\min_{\boldsymbol{q}\in\mathcal{Q}_3}\min_{\boldsymbol{p}\in\mathcal{Q}_1+\mathcal{Q}_2}\|\boldsymbol{q}-\boldsymbol{p}\|^2. \quad (36)$$

Given the estimated channels, the sign of the transmitted signals can then be estimated as

$$(\hat{s}_1[m],\hat{s}_2[m],\hat{s}_3[m]) = \arg\min_{(s_1,s_2,s_3)\in\{-1,1\}^3}\|\boldsymbol{Y}_\ell[:,m]-\hat{\boldsymbol{g}}_1 s_1-\hat{\boldsymbol{g}}_2 s_2-\hat{\boldsymbol{g}}_3 s_3\|^2, \quad m=1,\ldots,M. \quad (37)$$

Algorithm 2(b) Detection and Decoding of a Triple-Ton

Input: signal sub-block $\boldsymbol{Y}_\ell$
Output: Estimated signals $\hat{\boldsymbol{s}}_1$,$\hat{\boldsymbol{s}}_2$ and $\hat{\boldsymbol{s}}_3$, channels $\hat{\boldsymbol{g}}_1$, $\hat{\boldsymbol{g}}_2$ and $\hat{\boldsymbol{g}}_3$, and bit vectors $\hat{\boldsymbol{b}}_1$, $\hat{\boldsymbol{b}}_2$ and $\hat{\boldsymbol{b}}_3$
Compute $\tilde{\boldsymbol{Y}}_\ell$ using (26) and partition the columns of $\tilde{\boldsymbol{Y}}_\ell$ into four clusters with the corresponding index sets $\mathcal{J}_1$, $\mathcal{J}_2$, $\mathcal{J}_3$ and $\mathcal{J}_4$;
Compute $\hat{\boldsymbol{c}}_1$ using (33);
Partition the columns of $\{\boldsymbol{Y}_\ell[:,j]+\hat{\boldsymbol{c}}_1, j\in\mathcal{J}_2\}$, $\{\boldsymbol{Y}_\ell[:,j]+\hat{\boldsymbol{c}}_1, j\in\mathcal{J}_3\}$ and $\{\boldsymbol{Y}_\ell[:,j]+\hat{\boldsymbol{c}}_1, j\in\mathcal{J}_4\}$ into two clusters each to obtain the corresponding sets of centers $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$ respectively;
Compute $\hat{\boldsymbol{g}}_1$, $\hat{\boldsymbol{g}}_2$ and $\hat{\boldsymbol{g}}_3$ using (34)-(36);
Compute $\hat{\boldsymbol{s}}_1$, $\hat{\boldsymbol{s}}_2$ and $\hat{\boldsymbol{s}}_3$ using (37);
Compute $\hat{\boldsymbol{b}}_1$, $\hat{\boldsymbol{b}}_2$ and $\hat{\boldsymbol{b}}_3$ using (40).

The triple-ton detection and decoding is summarized in Alg. 2(b). The ℓ-th sub-block is determined as a triple-ton if

$$\frac{2}{\sigma^2}\|\boldsymbol{Y}_\ell-\hat{\boldsymbol{g}}_1\hat{\boldsymbol{s}}_1^T-\hat{\boldsymbol{g}}_2\hat{\boldsymbol{s}}_2^T-\hat{\boldsymbol{g}}_3\hat{\boldsymbol{s}}_3^T\|_{\mathcal{F}}^2<\tau. \quad (38)$$

Then the transmitted signal can be subtracted from the received signal by

$$\begin{aligned}\boldsymbol{Y}\leftarrow\boldsymbol{Y}&-\hat{\boldsymbol{g}}_1\left(\boldsymbol{h}(\mathcal{I}(\text{dec}(\hat{\boldsymbol{b}}_1)\mod p))\otimes\hat{\boldsymbol{s}}_1\right)^T\\&-\hat{\boldsymbol{g}}_2\left(\boldsymbol{h}(\mathcal{I}(\text{dec}(\hat{\boldsymbol{b}}_2)\mod p))\otimes\hat{\boldsymbol{s}}_2\right)^T\\&-\hat{\boldsymbol{g}}_3\left(\boldsymbol{h}(\mathcal{I}(\text{dec}(\hat{\boldsymbol{b}}_3)\mod p))\otimes\hat{\boldsymbol{s}}_3\right)^T, \quad (39)\end{aligned}$$

where the estimates of the bit vectors are given by

$$\hat{\boldsymbol{b}}_i=\frac{1}{2}(\hat{\boldsymbol{s}}_i[1:M]+\boldsymbol{1}), \quad i=1,2,3. \quad (40)$$

Finally a general UMA decoder based on detecting and decoding single-tons, double-tons and triple-tons is shown in Alg. 2. Note that in this algorithm, if the detection and decoding of double-tons and triple-tons (lines 9-16) is disabled, then it becomes Alg. 1 that is based on single-tons only; and if the detection and decoding of triple-tons (lines 13-16) is disabled, then the decoder is based on single-tons and double-tons.

IV. Rate Analysis and Optimization

Density evolution is a powerful tool in coding theory that has been widely used in the analysis of LDPC codes by tracking the probability density function (pdf) of the extrinsic messages passed on Tanner graphs [21]. The pdf is updated based on the edge degrees over the decoding iterations to derive thresholds for successful decoding. In this section, following the similar idea of density evolution, we derive the probability of an edge still remains in the Tanner graph after i decoding iterations. We note that for the single-ton-based peeling decoder, our analysis is similar to that in [16] and [20], which are in turn the special case of the general design in [15] where all the component codes used are repetition codes. However here we provide analysis for our proposed new peeling decoders based on double-tons and triple-tons, and

Algorithm 2 UMA Decoder Based on Single-Tons, Double-Tons and Triple-Tons

Input: Received signal $\boldsymbol{Y}$, number of sub-blocks L, d-selection set $\mathcal{I} = \{I(0), \ldots, I(p-1)\}$
Output: Set of estimated bit vectors $\mathcal{B}$.
Initialization: $\mathcal{B} = \emptyset$;
repeat
  for $\ell = 1, \ldots, L$ **do**
    if $\boldsymbol{Y}_\ell$ is not a zero-ton **then**
      Compute the estimate $\hat{\boldsymbol{s}}$, $\hat{\boldsymbol{g}}$ and $\hat{\boldsymbol{b}}$ using (17), (18) and (19);
      if (20) holds (i.e., $\boldsymbol{Y}_\ell$ is a single-ton) **then**
        Update $\boldsymbol{Y}$ using (21); $\mathcal{B} \leftarrow \mathcal{B} \cup \{\hat{\boldsymbol{b}}\}$;
      else
        Run Alg. 2(a);
        if (29) holds (i.e., $\boldsymbol{Y}_\ell$ is a double-ton) **then**
          Update $\boldsymbol{Y}$ using (30); $\mathcal{B} \leftarrow \mathcal{B} \cup \{\hat{\boldsymbol{b}}_1, \hat{\boldsymbol{b}}_2\}$;
        else
          Run Alg. 2(b);
          if (38) holds (i.e., $\boldsymbol{Y}_\ell$ is a triple-ton) **then**
            Update $\boldsymbol{Y}$ using (39);
            $\mathcal{B} \leftarrow \mathcal{B} \cup \{\hat{\boldsymbol{b}}_1, \hat{\boldsymbol{b}}_2, \hat{\boldsymbol{b}}_3\}$;
          end
        end
      end
    end
  end
until no single-ton/double-ton/triple-ton is detected;

show their performance gains over the traditional single-ton based decoder. Recall that in the Tanner graph each edge between a VN and a CN represents the information bit vector that is originated from the transmitter corresponding to the VN and transmitted during the sub-slot corresponding to the CN. During the peeling decoding process, if this bit vector is decoded at certain iteration, then the edge is pruned from the Tanner graph. Denote z_i as the probability that an edge in the Tanner graph is not pruned after the i-th decoding iteration. Through the density evolution analysis, we would like to find conditions on $r = K/L$ and d such that $z_i \to 0$ as $i \to \infty$, i.e., all bit vectors are successfully decoded. The analysis is based on the following two assumptions: 1) The numbers of VNs and CNs in the Tanner graph both approach infinity, i.e., $K \to \infty, L \to \infty$, and the Tanner graph is cycle-free (tree like) [21]; 2) In the peeling decoder, the single-tons, double-tons and triple-tons can be perfectly detected and decoded. We consider two cases. For the case of regular Tanner graph, each VN has a constant degree of d and our goal is to find the maximum transmission rate $r = K/L$ such that $z_i \to 0$. For the case of irregular Tanner graph, the VNs can have different degrees and we aim to find the optimal degree distribution to achieve the maximum rate. Hence the analysis in this section provides asymptotic upper bounds on achievable rates using regular and irregular Tanner graphs.

A. Regular Tanner Graph

For a regular Tanner graph each VN has a degree d, i.e., it is connected to d uniformly selected CNs. That is, each bit vector is transmitted in d out of L sub-slots. Define the right edge degree distribution ρ_j as the proportion of edges connected to CNs that have degree $j, j = 1, \ldots, K$, and we have $\sum_{j=1}^{K} \rho_j = 1$. Define the CN degree distribution Π_j as the fraction of CNs that have degree $j, j = 1, \ldots, K$. We have

$$\rho_j = \frac{\Pi_j L j}{K d} = \frac{j r \Pi_j}{d}. \tag{41}$$

Since each VN uniformly selects d out of L CNs to connect, the CN degree follows the binomial distribution $\mathrm{Binomial}(K, \frac{d}{L})$. It is known that when $n \to \infty$, $\pi \to 0$, $n\pi \to$ a constant λ, then $\mathrm{Binomial}(n, \pi) \to \mathrm{Poisson}(\lambda)$. In our case, we have $K \to \infty$, $\frac{d}{L} \to 0$, $K\frac{d}{L} = dr$. Therefore as $K \to \infty$, $L \to \infty$, the CN degree follows $\mathrm{Poisson}(dr)$, i.e.,

$$\Pi_j = \frac{(dr)^j e^{-dr}}{j!}. \tag{42}$$

Substituting (42) into (41), we have

$$\rho_j = \frac{(dr)^{j-1} e^{-dr}}{(j-1)!}. \tag{43}$$

To derive the probability z_i that an edge in the Tanner graph is not pruned after i decoding iterations, we first consider the peeling decoder that can only detect and decode single-tons. Since we assume that the decoder perfectly detects and decodes single-tons, if at an iteration a CN is detected as a single-ton, the edge that is connected to it is pruned from the graph. Let $z_0 = 1$ because no edge is pruned before the decoding starts. In the i-th iteration, a CN that has degree j is detected as a single-ton if $j-1$ edges that are connected to it have been pruned. Hence the probability that at the i-th iteration an edge is connected to a CN is a single-ton is given by

$$q_i(1) = \sum_{j=1}^{K} \rho_j (1 - z_{i-1})^{j-1}. \tag{44}$$

Next a VN is removed from the graph if at least one of its d edges is connected to a single-ton CN, since by perfectly decoding the single-ton, the bit vector represented by the VN is decoded and removed. An edge originated from a VN is not pruned if all other $d-1$ edges connected to the same VN are not pruned, i.e.,

$$z_i = (1 - q_i(1))^{d-1} = \left(1 - \sum_{j=1}^{K} \rho_j (1 - z_{i-1})^{j-1}\right)^{d-1}, \tag{45}$$

which is the recursive equation of z_i for the ideal peeling decoder based on single-tons.

Similarly, an edge is connected to a double-ton CN that has degree j if among the other $j-1$ edges that are connected to the same CN, $j-1$ have been pruned and one has not.

Therefore the probability that an edge is connected to a double-ton CN in the i-th iteration is given by

$$q_i(2) = \sum_{j=1}^{K} \rho_j \binom{j-1}{1} z_{i-1}(1-z_{i-1})^{j-2}. \tag{46}$$

An edge originated from a VN is not removed in the i-th iteration if all other $d-1$ edges connected to the same VN are not connected to single-ton or double-ton CNs. Hence we obtain the recursive equation of z_i when the peeling decoder is based on both single-tons and double-tons as

$$\begin{aligned} z_i &= (1-q_i(1)-q_i(2))^{d-1} \\ &= \left(1-\sum_{j=1}^{K}\rho_j\left[(1-z_{i-1})^{j-1}+\binom{j-1}{1}z_{i-1}\right.\right. \\ &\qquad \left.\left.\times(1-z_{i-1})^{j-2}\right]\right)^{d-1}. \end{aligned} \tag{47}$$

Moreover, the probability that an edge is connected to a triple-ton CN in the i-th iteration is given by

$$q_i(3) = \sum_{j=1}^{K} \rho_j \binom{j-1}{2} z_{i-1}^2(1-z_{i-1})^{j-3}, \tag{48}$$

and the recursive equation of z_i when the peeling decoder is based on single-ton, double-ton and triple-tons is given by

$$\begin{aligned} z_i = \left(1-\sum_{j=1}^{K}\rho_j\left[(1-z_{i-1})^{j-1}+\binom{j-1}{1}z_{i-1}\right.\right. \\ \left.\left.\times(1-z_{i-1})^{j-2}+\binom{j-1}{2}z_{i-1}^2(1-z_{i-1})^{j-3}\right]\right)^{d-1}. \end{aligned} \tag{49}$$

Typically for a given d, there is a threshold $r_{th}(d)$ such that z_i converges to zero if $r \le r_{th}(d)$ and z_i does not converge to zero if $r > r_{th}(d)$. Given a maximum VN degree d_v, we can find the maximum achievable rate for the regular Tanner graph by searching the threshold $r_{th}(d)$ for each $d = 1, \ldots, d_v$, as follows: We initialize $r = \delta$ and set the rate increment $\delta = 0.01$. We repeatedly evaluate z_i using (45), (47) or (49) until convergence, i.e., $z_i < \epsilon = 10^{-5}$, or the number of iterations reaches $N_0 = 100$. If z_i reaches convergence with rate r, then the rate is increased by δ until we find the maximum convergence rate $r_{th}(d)$. The maximum achievable rate for a specific type of decoder is $r^* = \max_{d\in\{1,\ldots,d_v\}} r_{th}(d)$, and the corresponding VN degree is $d^* = \arg\max_{d\in\{1,\ldots,d_v\}} r_{th}(d)$.

Figures 3 and 4 show the evolution of z_i for different decoders in (45), (47) and (49) over iterations and the threshold behavior, for $d = 2$ and $d = 3$ respectively. The threshold behavior is clear except for the decoder based on single-tons and $d = 2$. For example, in Figure 3, for the decoder based on single-tons and double-tons, when $r = 1.67$ z_i converges to 0 but when $r = 1.68$, z_i converges to 0.58. Hence we have $r_{th}(2) = 1.67$. Moreover, by decoding double-tons and triple-tons, the thresholds increase and hence the system can accommodate more users by employing more sophisticated

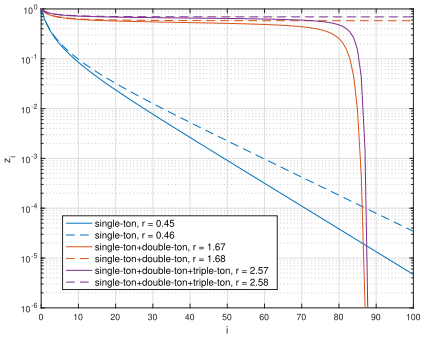

Fig. 3. The evolution of z_i and threshold behavior for different decoders when $d = 2$.

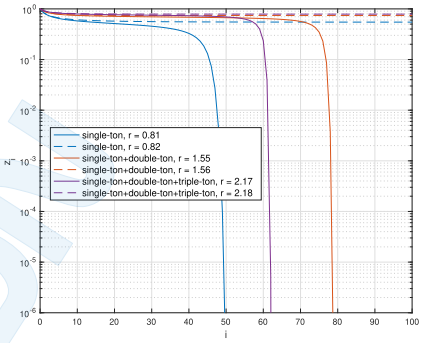

Fig. 4. The evolution of z_i and threshold behavior for different decoders when $d = 3$.

decoding algorithms. With the maximum possible VN degree $d_v = 5$, the maximum achievable rates for the three decoders using regular Tanner graph are given in Table I.

B. Irregular Tanner Graph

For an irregular Tanner graph, the VNs have different degrees. We define the left edge degree distribution λ_j as the proportion of edges connected to VNs with degree $j, j = 1, \ldots, d_v$, where d_v is the maximum degree of all VNs. A VN with degree j is connected to j (out of L) uniformly selected CNs. Denote Γ_j as VN degree distribution, i.e., the proportion of VN that has degree j. The number of VNs with degree j is $K\Gamma_j = \lambda_j E/j$, where E is the total number of edges. Hence we obtain

$$\Gamma_j = \frac{\lambda_j/j}{\sum_{i=1}^{d_v}\lambda_i/i}, \quad j = 1, \ldots, d_v. \tag{50}$$

Define the average VN degree as

$$\bar{d} \triangleq \sum_{j=1}^{d_v} j\Gamma_j = \frac{1}{\sum_{i=1}^{d_v}\lambda_i/i}. \tag{51}$$

For a regular Tanner graph, the probability of each CN being selected by a VN is d/L. Similarly, for an irregular graph with VN degree distribution $\Gamma_j, j = 1, \ldots, d_v$, the probability of each CN being selected by a VN with degree j is j/L. Then the average probability of each CN being selected by a VN is $\sum_{j=1}^{d_v}\Gamma_j j/L = \bar{d}/L$. Hence the right edge degree distribution

of an irregular Tanner graph is given by

$$\rho_j = \frac{(\bar{d}r)^{j-1}e^{-\bar{d}r}}{(j-1)!}. \tag{52}$$

A VN that have degree t is removed from the graph if at least one of its t edges is connected to a single-ton, a double-ton, or a triple-ton. Therefore the recursive equation of z_i for the ideal peeling decoder based on single-tons on an irregular Tanner graph is given by

$$\begin{aligned} z_i &= \sum_{t=1}^{d_v} \lambda_t (1-q_i(1))^{t-1} \\ &= \sum_{t=1}^{d_v} \lambda_t \left(1 - \sum_{j=1}^{K} \rho_j (1-z_{i-1})^{j-1}\right)^{t-1}. \end{aligned} \tag{53}$$

Similarly, if the ideal decoder can decode double-tons, or triple-tons, the recursive equations of z_i are given respectively by

$$\begin{aligned} z_i = \sum_{t=1}^{d_v} \lambda_t \Bigg(1 - \sum_{j=1}^{K} \rho_j \Bigg[(1-z_{i-1})^{j-1} + \binom{j-1}{1} z_{i-1} \\ \times (1-z_{i-1})^{j-2}\Bigg]\Bigg)^{t-1}, \end{aligned} \tag{54}$$

and

$$\begin{aligned} z_i = \sum_{t=1}^{d_v} \lambda_t \Bigg(1 - \sum_{j=1}^{K} \rho_j \Bigg[(1-z_{i-1})^{j-1} + \binom{j-1}{1} z_{i-1} \\ \times (1-z_{i-1})^{j-2} + \binom{j-1}{2} z_{i-1}^2 (1-z_{i-1})^{j-3}\Bigg]\Bigg)^{t-1}. \end{aligned} \tag{55}$$

1) Rate Optimization: Similar to the case of regular Tanner graph, for a given $\{\lambda_j\}$, we can find the threshold $r_{th}(\{\lambda_j\})$ such that z_i converges to zero if $r < r_{th}(\{\lambda_j\})$ and otherwise z_i does not converge to zero. We can then search over the left edge degree distribution $\{\lambda_j\}$ to find the maximum achievable rate r^*, using the differential evolution algorithm, which is a combination of hill-climbing algorithm and a genetic algorithm for multivariate function optimization [22]. This is similar to optimizing the degree profiles of an irregular LDPC code to approach the channel capacity [23]. The procedure for computing the maximum achievable rate for irregular Tanner graph and the corresponding VN degree distribution is as follows: We set the number of iterations as N_t. In each iteration we compute the rate threshold for the current edge distribution $\{\lambda_j\}$ and then the new distribution for the next iteration is given by the differential evolution algorithm. Within each iteration, we gradually increase the rate r by δ and find the maximum of rate $r_{th}(\{\lambda_j\})$ that satisfies $z_{N_0} < \epsilon$ using (53), (54) or (55). The maximum achievable rate is $r^* = \max_{\{\lambda_j\}} r_{th}(\{\lambda_j\})$, and the corresponding optimal degree distribution is $\{\lambda_j\}^* = \arg\max_{\{\lambda_j\}} r_{th}(\{\lambda_j\})$.

We set the maximum VN degree as $d_v = 5$, and let $N_t = 70$, $N_0 = 100$, $\epsilon = 10^{-5}$ and $\delta = 0.01$. Table I compares the maximum achievable rate r^* obtained for regular Tanner graphs and irregular Tanner graphs for different decoders. The corresponding optimal VN degree distributions for irregular Tanner graphs are shown in Table II. Recall that r^* is essentially an upper bound on the achievable rate under ideal decoders in the asymptotic regime of $K, L \to \infty$. We see that even though irregular graphs offer higher rates than regular graph for all three decoders, the gaps are relatively small.

2) Transmission Scheme Based on Irregular Tanner Graph: In Section II we presented the transmission scheme based on the regular Tanner graph with VN degree d. Next we present the transmission scheme based on the irregular Tanner graph with VN degree distribution $\Gamma_j, j = 1, \ldots, d_v$.

We define d_v index set $\mathcal{I}_i$ by enumerating all i-selections of $\{1, 2, \ldots, L\}$, for $i = 1, \ldots, d_v$, then the size of $\mathcal{I}_i$ is $\binom{L}{i}$. The 2^M equiprobable bit vectors $\boldsymbol{b}$ are divided into d_v non-overlapping sets. The bit vector $\boldsymbol{b}$ falls into the α-th set $\mathcal{I}_\alpha$ if $\sum_{i=1}^{\alpha-1} \Gamma_i < (\mathrm{dec}(\boldsymbol{b}) + 1)/2^M \leq \sum_{i=1}^{\alpha} \Gamma_i$, where $\alpha = 2, \ldots, d_v$, and the nonzero entries of $\boldsymbol{h}$ is selected from $\mathcal{I}_\alpha$. We extend the mapping in Section II from the set $\mathcal{I}_\alpha(i)$ to a binary vector $\boldsymbol{e}(\mathcal{I}_\alpha(i)) \in \{0,1\}^L$, such that the ℓ-th entry of $\boldsymbol{e}(\mathcal{I}_\alpha(i))$ is 1 if $\ell \in \mathcal{I}_\alpha(i)$, and zero otherwise, $\alpha = 2, \ldots, d_v, \ell = 1, \ldots, L$. Then the column of the parity check matrix $\boldsymbol{H}$ corresponding to $\boldsymbol{b}$ is given by

$$\boldsymbol{h}(\boldsymbol{b}) = \boldsymbol{e}\left(\mathcal{I}_\alpha\left(\left(\mathrm{dec}(\boldsymbol{b}) - \left\lfloor \Gamma_{\alpha-1} 2^M \right\rfloor\right) \mod \binom{L}{\alpha}\right)\right). \tag{56}$$

The transmitted signal $\boldsymbol{x}(\boldsymbol{b})$ is still given by (10).

V. Simulation Results

In this section we present simulation results to illustrate the performance of the proposed sparse graph-based UMA systems. For simplicity we number the decoders based on single-tons, single-tons + double-tons, and single-tons + double-tons + triple-tons as 1, 2 and 3, respectively. The number of bits transmitted by each transmitter in each coherence interval is $M = 70$. The number of receive antennas at the BS is $N_a = 4$ and the channel vector of each transmitter is generated as $\boldsymbol{g} \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{I}_{N_a})$. The signal-to-noise ratio (SNR) per codeword is defined as $\mathrm{SNR} = \frac{\mathbb{E}\{\|\boldsymbol{g}\boldsymbol{s}^T\|^2\}d}{N_a T \sigma^2} = \frac{Pd}{\sigma^2 L}$ for a regular Tanner graph with VN degree d, and $\mathrm{SNR} = \frac{P\bar{d}}{\sigma^2 L}$ for an irregular Tanner graph with average VN degree $\bar{d}$. In each coherence interval, K bit vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K$ are transmitted. For each decoded bit vector $\hat{\boldsymbol{b}}$ and the corresponding channel estimate $\hat{\boldsymbol{g}}$, if $\hat{\boldsymbol{b}} \notin \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K\}$, then $\hat{\boldsymbol{b}}$ corresponds to a decoding error, and the decoding performance metric – the frame error rate (FER) is the ratio between the total number of decoding errors and the total number of transmitted bit vectors. On the other hand, for each channel estimate $\hat{\boldsymbol{g}}$, since we do not know which real channel in $\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K\}$ it corresponds to, we choose the one that is the closest to $\hat{\boldsymbol{g}}$ and calculate the normalized squared error as $\mathrm{NSE}(\hat{\boldsymbol{g}}) = \|\sqrt{P}\boldsymbol{g} - \hat{\boldsymbol{g}}\|_{\mathcal{F}}^2 / \|\sqrt{P}\boldsymbol{g}\|_{\mathcal{F}}^2$ where $\boldsymbol{g} = \arg\min_{\boldsymbol{g}' \in \{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K\}} \|\sqrt{P}\boldsymbol{g}' - \hat{\boldsymbol{g}}\|_{\mathcal{F}}^2$. Then by averaging over all channel estimates, we obtain the channel estimation

TABLE I

MAXIMUM ACHIEVABLE RATE r^* OF REGULAR AND IRREGULAR TANNER GRAPHS WITH $d_v = 5$

Decoder	single-ton	single-ton + double-ton	single-ton + double-ton + triple-ton
Regular	0.81 ($d = 3$)	1.67 ($d = 2$)	2.57 ($d = 2$)
Irregular	0.87	1.77	2.59

TABLE II

OPTIMAL Γ_j^* FOR DIFFERENT IDEAL DECODERS OBTAINED BY DIFFERENTIAL EVOLUTION

j	single-ton	single-ton + double-ton	single-ton + double-ton + triple-ton
1	0	0	0
2	0.4360	0.8244	0.9091
3	0.2032	0	0
4	0	0	0
5	0.3608	0.1756	0.0909

performance metric – the normalized mean-squared error (NMSE).

A. Performance of Proposed UMA Systems

We first consider the performance of the zero-ton detector in (12) and specify the procedure of choosing the threshold τ. Denote the decision statistic $v \triangleq \frac{2}{\sigma^2}\|\boldsymbol{Y}_\ell\|_{\mathcal{F}}^2$. Recall that when $\boldsymbol{Y}_\ell$ is a zero-ton, $v \sim \chi^2_{2(M+1)N_a}$. It is known that when k is large, the χ^2_k distribution can be approximated by a Gaussian distribution $\mathcal{N}(k, 2k)$. Therefore the detection probability can be approximated as

$$P_D = P(v \le \tau \mid \boldsymbol{Y}_\ell \text{ is a zero-ton}) \approx \Phi\left(\frac{\tau - 2(M+1)N_a}{2\sqrt{(M+1)N_a}}\right), \quad (57)$$

where $\Phi(\cdot)$ is the CDF of a standard Gaussian $\mathcal{N}(0,1)$ variable. Next we consider the false alarm probability. First if $\boldsymbol{Y}_\ell$ is a single-ton, then $\mathbb{E}\{Y_\ell[n,m]^2\} = P + \sigma^2$, we have $\frac{2}{P+\sigma^2}\|\boldsymbol{Y}_\ell\|^2 = \frac{1}{1+P/\sigma^2}v \sim \chi^2_{2(M+1)N_a}$. Hence the distribution of the decision statistic v in this case is approximated as $\mathcal{N}(2(M+1)N_a(1+P/\sigma^2), 4(M+1)N_a(1+P/\sigma^2)^2)$. Since $P/\sigma^2 = \text{SNR}\frac{L}{d}$, an upper bound on the false alarm probability is given by

$$P_{FA} \le P(v \le \tau \mid \boldsymbol{Y}_\ell \text{ is a single-ton}) \approx \Phi\left(\frac{\tau - 2(M+1)N_a(1+\text{SNR}\frac{L}{d})}{2(1+\text{SNR}\frac{L}{d})\sqrt{(M+1)N_a}}\right), \quad (58)$$

since when $\boldsymbol{Y}_\ell$ contains more signals, its power increases and the probability of $v \le \tau$ decreases. Letting $L = 40$ and $d = 3$, in Figure 5 we plot P_D in (57) and P_{FA} in (58) versus τ for SNR $= -5$ dB and 0 dB. We see that the zero-ton detector in (12) has a near-ideal performance in the sense that across a wide range of τ, e.g., $\tau \in [1000, 2000]$ for SNR $= -5$ dB, and $\tau \in [1000, 6000]$ for SNR $= 0$ dB, it achieves $P_D \approx 1$ and $P_{FA} \approx 0$. In subsequent simulations, we set the detection threshold in (12), (20), (29) and (38) as $\tau = 4(M+1)N_a = 1136$.

We fix the number of sub-slots $L = 40$, hence the total coherence interval is $T = (M+1)L = 2840$. Figures 6 and 7 show the decoding and channel estimation performance versus the number of active transmitters K when SNR $= 0$ dB. We plot the results of the three decoders using regular Tanner graphs with $d = 2$ and $d = 3$ as well as the irregular Tanner graph given in Table II. It is seen that more sophisticated decoder can sustain more active transmitters, under the same decoding and channel estimation performance, i.e., decoder 3 > decoder 2 > decoder 1. For example, for a target FER value of 0.01, decoder 1 can support $K = 20$ (with $d = 3$), decoder 2 can support $K = 40$ (with $d = 3$) and decoder 3 can support $K = 56$ (with $d = 3$). Moreover, the actual decoding performance with finite K, L and practical decoders may not be in line with the asymptotic rate upper bound obtained by density evolution in Section IV. For example, for decoder 2, density evolution analysis yields $r_{th}(2) > r_{th}(3)$ whereas in practice $d = 3$ outperforms $d = 2$. Further, the irregular code in practice does not outperform regular code. Nevertheless, the performance upper bounds provided by the density evolution analysis motivate future work on developing more powerful decoding schemes to reduce the gap between practically achievable performance and these bounds.

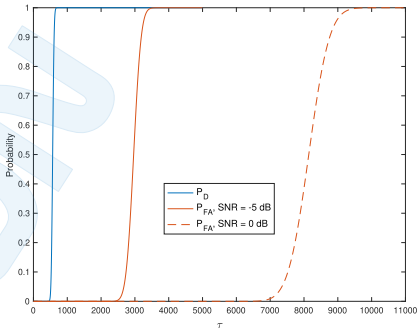

Fig. 5. P_D and P_{FA} versus τ of the zero-ton detector for different SNRs.

Figures 8 and 9 show the decoding and channel estimation performance of the three decoders versus SNR per codeword for fixed rate and $d = 2, 3$. We see that for decoder 1, the performance for $d = 3$ is significantly better than that for $d = 2$; whereas for decoder 2 and 3, the performances for $d = 2$ is better than that for $d = 3$ when SNR is low, and the

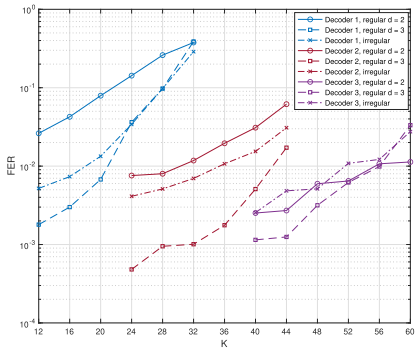

Fig. 6. Decoding performance versus number of active transmitters for the three proposed decoders with different VN degrees. $L = 40$ and SNR $=$ 0 dB.

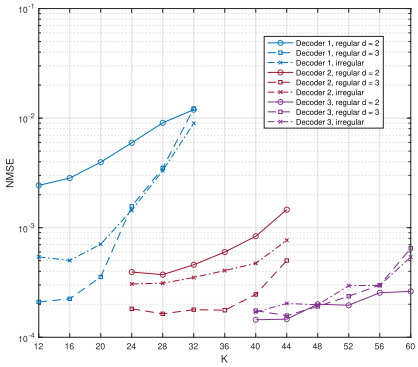

Fig. 7. Channel estimation performance versus number of active transmitters for the three decoders with different VN degrees. $L = 40$ and SNR $= 0$ dB.

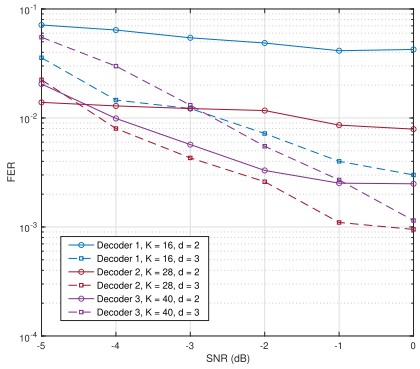

Fig. 8. Decoding performance versus SNR per codeword for the three decoders with different number of active transmitters and VN degrees. $L = 40$.

other way when SNR is high. For example, the decoder 3 with $d = 2$ has lower FER than $d = 3$ when SNR ≤ -1 dB and has higher FER when SNR > -1 dB. The reason is that when the channel condition is poor, the estimates of signal and channel are less accurate. Consequently, more edges in the graph would lead to more pruning and therefore more errors. Moreover, for each of the three decoders, the slope of $d = 3$ is steeper than that of $d = 2$.

B. Comparison With CS-Based UMA

We now compare the CS-based schemes in [7], [9], and [11] with our proposed sparse graph-based UMA with different decoders. In the UMA system, each transmitter transmits $M = 70$ bits during the coherent interval T. In the CS-based schemes, the $M = 70$ bits are divided into $J = 11$ blocks of sizes $m_1 = 10$ and $m_j = 6, j = 2, \ldots, 11$. For the j-th block, ℓ_j parity bits are appended to the m_j information bits such that all blocks have equal length, where $\ell_1 = 0$ and $\ell_j = 4, j = 2, \ldots, 11$. Denote the information bits and parity check bits in the j-th block as $\boldsymbol{b}(j) \in \{0,1\}^{m_j}$ and $\boldsymbol{p}(j) \in \{0,1\}^{l_j}, j = 1, \ldots, J$. Specifically, the ℓ_j parity check bits in the j-th block are generated by the mod-2 multiplication of all the information bits in the preceding blocks and a Rademacher matrix, i.e., $\boldsymbol{p}(j) = [\boldsymbol{b}(1)^T, \ldots, \boldsymbol{b}(j-1)^T]^T \boldsymbol{G}, j = 2, \ldots, J$ where the entries of $\boldsymbol{G} \in \{0,1\}^{\left(\sum_{j'=1}^{j-1} m_{j'}\right) \times l_j}$ are uniform Bernoulli trials. Then according to the CS-based UMA transmission scheme described in Section II, each 10 bits of coded data $[\boldsymbol{b}(j)^T, \boldsymbol{p}(j)^T]^T$ in one block is mapped to a M_s-dimensional signal using a sensing matrix $\boldsymbol{C} \in \mathbb{C}^{M_s \times 1024}$ that contains $\mathcal{CN}(0, P)$ elements. We consider two coherent intervals with $M_s = 130$ and $M_s = 260$, corresponding to $T = M_s \times J = 1430$ and 2860 respectively. Note that even though the channel model in [7] is a simple noisy superposition of all received codewords, the method in [7] can be easily modified to account for the channel effects. In particular, instead of using the non-negative least squares (NNLS) algorithm as in [7], the modified decoder first performs the Simultaneous Orthogonal Matching Pursuit (S-OMP) [24] for each sub-block of the received signal to obtain the K channel estimates and K decoded bit sequences. Then the tree decoder in [7] is applied over J sub-blocks to decode K blocks of information bits. On the other hand, [9] essentially averages out the channel effect by forming the covariance matrix of the received signal and assuming a large number of receive antennas, i.e., $N_a \to \infty$, and the CS decoding is based on the outer product model $\Sigma_y^{(j)} = \sum_{r=1}^{2^{M_b}} \gamma_r^{(j)} \boldsymbol{c}_r \boldsymbol{c}_r^H$ where $\Sigma_y^{(j)}$ is the covariance matrix of the received signal in the j-th sub-slot and $\boldsymbol{\gamma}^{(j)}$ is the activity vector in [9]. We simulated the algorithm in [9] for both the ideal case, i.e., $N_a \to \infty$ and the case of $N_a = 4$. Moreover, the blind receiver method Alg. 1 in [11] is also simulated. The SNR per codeword for the CS-based scheme is given by SNR $= \frac{J\mathbb{E}\{\|\boldsymbol{g}\boldsymbol{c}^T\|^2\}}{N_a T \sigma^2} = \frac{P}{\sigma^2}$. For our proposed scheme, we set $L = 20$ and 40 corresponding to $T = L(M+1) = 1420$ and 2840. All the three proposed decoders uses regular Tanner graph with $d = 3$.

The performance comparisons are shown in Figure 10. It is seen that even decoder 1 that is based on single-tons outperforms the three CS-based decoders – recall that the former has a linear complexity in M whereas the latter has an exponential complexity in M. Moreover, at the increased complexity, decoder 2 and decoder 3 offer substantially better performances. Moreover, when the coherence interval T is doubled, the performance of the CS-based decoders improves only marginally, whereas that of the proposed three decoders significantly improves. This is because by increasing L, the proposed decoders are based on more sparse Tanner graphs with more CNs and therefore more active users can be

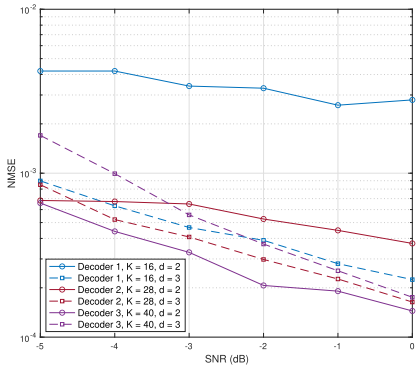

Fig. 9. Channel estimation performance versus SNR per codeword for the three decoders with different number of active transmitters and VN degrees. $L = 40$.

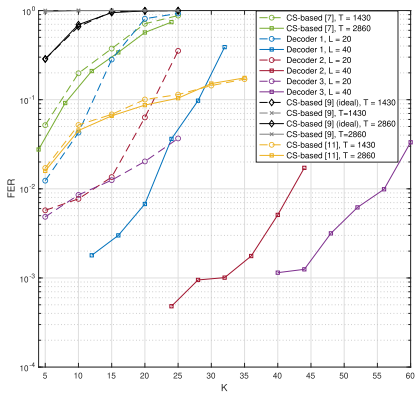

Fig. 10. Decoding error performance comparison between proposed decoders and the CS-based decoder. SNR $= 0$ dB.

accommodated. On the other hand, increasing M_s can decrease the error of the sparse recovery in the CS-based decoder, but the problem of codeword collisions still remains and is the main cause of decoding errors.

C. Throughput Analysis

Next following [15] we present the throughput comparison between the theoretical analysis and simulation results under several assumptions. The system throughput is defined as the average number of information packets successfully decoded per sub-slot. We consider the following four cases:

- The ideal case: $K, L \to \infty$ with fixed rate $r = K/L$ and perfect peeling decoding. Since this is the assumption of the density evolution analysis, the throughput is computed by $T(r) = r(1 - z_{N_0}(r))$ where $z_{N_0}(r)$ is computed using (45), (47) and (49) for different peeling decoders.
- The finite-size case: We fix $L = 40$, and $K = rL$. We randomly generate regular (K, L) Tanner graphs with VN degree d and perform perfect peeling decoding on each of them. Let $\tilde{K}$ be the average number of information packets decoded by a specific decoder, the throughput is given by $T(r) = \tilde{K}(r)/L$.
- The practical case: We fix $L = 40$, and $K = rL$. We set SNR $= 0$ dB and use the same simulation setup as described in Section V-A and measure the FER(r) for rate r. The throughput is computed as $T(r) = r(1-\text{FER}(r))$.
- The known channel case: This is the same as the practical case except that the channel vectors are assumed perfectly known to the receiver. The throughput is also computed as $T(r) = r(1 - \text{FER}(r))$.

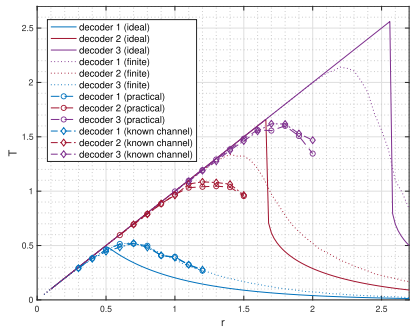

Fig. 11. Throughput comparison of proposed decoders under different assumptions. $d = 2$.

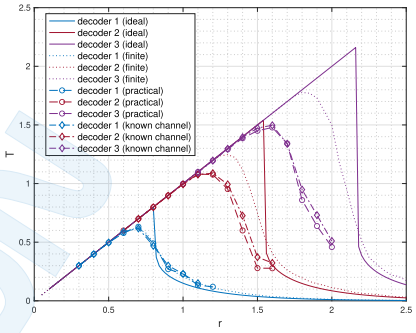

Fig. 12. Throughput comparison of proposed decoders under different assumptions. $d = 3$.

Fig. 11 and Fig. 12 show the throughput T versus transmission rate r of the three decoders for the four cases using regular Tanner graphs with $d = 2$ and $d = 3$, respectively. Consistently, we see that the proposed double-ton-based and triple-ton-based decoders offer significantly higher throughput than the traditional single-ton-based decoder, under each scenario, namely, the ideal case, the finite-size case and the practical case. When SNR $= 0$ dB the practical case of decoder 1 can achieve the performance of the finite case. However, there are gaps between the practical case and the finite case of decoder 2 and decoder 3, because double-ton and triple-ton based decoders are more sensitive to decoding errors. For the practical decoders, the decoders with blind channel estimation perform similarly to those with perfect channel estimation. Moreover, it is seen that the asymptotic analysis provides a good performance approximation for the proposed practical decoders in low-rate regions.

VI. Conclusion

In this paper, we have proposed a new UMA transmission scheme based on the sparse Tanner graph and corresponding receiver algorithms in MIMO channels. During the transmission, the channel coherence interval is split into several

sub-slots and each active transmitter selects a few sub-slots to transmit its data repeatedly according to the given sparse Tanner graph, which can be either regular or irregular. Three iterative receiver algorithms are proposed each detecting and decoding different number of codewords in each iteration. The key ingredient of these decoders are the corresponding clustering-based blind channel estimators. We also present the density evolution analysis for both regular and irregular Tanner graphs to obtain the asymptotic upper bound on the maximum achievable rate. Simulation results show that among the three proposed decoders, the ones that perform joint decoding of two or three codewords offer better performance. Compared with the existing CS-based UMA systems, the proposed sparse graph-based UMA transmission and the corresponding receiver algorithms offer better performance and lower receiver complexity. Finally, the density evolution analysis provides accurate throughput predictions for practical decoders in the low-rate regime.

REFERENCES

[1] C. Bockelmann *et al.*, "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, Sep. 2016.

[2] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, and R. Schober, "Massive access for 5G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 615–637, Mar. 2021.

[3] L. Liu and W. Yu, "Massive connectivity with massive MIMO—Part I: Device activity detection and channel estimation," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2933–2946, Jun. 2018.

[4] Y. Polyanskiy, "A perspective on massive random-access," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2523–2527.

[5] X. Chen, T.-Y. Chen, and D. Guo, "Capacity of Gaussian many-access channels," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3516–3539, Jun. 2017.

[6] O. Ordentlich and Y. Polyanskiy, "Low complexity schemes for the random access Gaussian channel," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2528–2532.

[7] V. K. Amalladinne, A. Vem, D. K. Soma, K. R. Narayanan, and J. Chamberland, "A coupled compressive sensing scheme for unsourced multiple access," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Oct. 2018, pp. 6628–6632.

[8] A. Fengler, P. Jung, and G. Caire, "SPARCs and AMP for unsourced random access," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 2843–2847.

[9] A. Fengler, S. Haghighatshoar, P. Jung, and G. Caire, "Grant-free massive random access with a massive MIMO receiver," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 23–30.

[10] R. Calderbank and A. Thompson, "CHIRRUP: A practical algorithm for unsourced multiple access," 2018, *arXiv:1811.00879*.

[11] J. Liu and X. Wang, "Sparsity-exploiting blind receiver algorithms for unsourced multiple access in MIMO and massive MIMO channels," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8055–8067, Dec. 2021.

[12] A. Vem, K. R. Narayanan, J.-F. Chamberland, and J. Cheng, "A user-independent successive interference cancellation based coding scheme for the unsourced random access Gaussian channel," *IEEE Trans. Commun.*, vol. 67, no. 12, pp. 8258–8272, Dec. 2019.

[13] A. Pradhan, V. Amalladinne, A. Vem, K. R. Narayanan, and J.-F. Chamberland, "A joint graph based coding scheme for the unsourced random access Gaussian channel," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Feb. 2019, pp. 1–6.

[14] J. Dommel, Z. Utkovski, L. Thiele, and S. StaÅ, "Sparse code-domain non-orthogonal random access with peeling decoder," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, 2019, pp. 984–988.

[15] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6815–6832, Dec. 2015.

[16] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, Feb. 2011.

[17] A. Fengler, P. Jung, and G. Caire, "Pilot-based unsourced random access with a massive MIMO receiver in the quasi-static fading regime," Tech. Rep., 2021.

[18] A. Decurninge, I. Land, and M. Guillaud, "Tensor-based modulation for unsourced massive random access," *IEEE Wireless Commun. Lett.*, vol. 10, no. 3, pp. 552–556, Mar. 2021.

[19] S. Liang, X. Wang, and L. Ping, "Semi-blind detection in hybrid massive MIMO systems via low-rank matrix completion," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5242–5254, Nov. 2019.

[20] X. Li, D. Yin, S. Pawar, R. Pedarsani, and K. Ramchandran, "Sub-linear time support recovery for compressed sensing using sparse-graph codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6580–6619, Oct. 2019.

[21] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[22] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, pp. 341–359, Oct. 1997.

[23] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[24] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Process.*, vol. 86, no. 3, pp. 572–588, 2006.

Jiaai Liu received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2018, and the M.S. degree in electrical and computer engineering from Duke University, Durham, NC, USA, in 2020. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Columbia University, New York, NY, USA. Her research interests include channel coding, massage passing algorithms, and machine learning for communications.

Xiaodong Wang (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Princeton University. He is a Professor of electrical engineering with Columbia University, New York. His research interests fall in the general areas of computing, signal processing, and communications, and has published extensively in these areas. Among his publications is a book *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Prentice Hall, 2003). His current research interests include wireless communications, statistical signal processing, and genomic signal processing. He is listed as an ISI highly-cited author. He received the 1999 NSF CAREER Award, the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award, and the 2011 IEEE Communication Society Award for Outstanding Paper on New Communication Topics. He has served as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON INFORMATION THEORY.