

Article

Joint Intra/Inter-Slot Code Design for Unsourced Multiple Access in 6G Internet of Things

Yuanjie Li ¹, Kai Niu ^{1,*}, Chao Dong ¹, Shiqiang Suo ² and Jiaru Lin ¹¹ Key Laboratory of Universal Wireless Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China² CICT Mobile Communication Technology Co., Ltd., Beijing 100190, China

* Correspondence: niukai@bupt.edu.cn

Abstract: Unsourced multiple access (UMA) is the technology for massive, low-power, and uncoordinated Internet-of-Things in the 6G wireless system, improving connectivity and energy efficiency on guaranteed reliability. The multi-user coding scheme design is a critical problem for UMA. This paper proposes a UMA coding scheme based on the T-Fold IRSA (irregular repetition slotted Aloha) paradigm by using joint Intra/inter-slot code design and optimization. Our scheme adopts interleave-division multiple access (IDMA) to enhance the intra-slot coding gain and the low-complexity joint intra/inter-slot SIC (successive interference cancellation) decoder structure to recover multi-user payloads. Based on the error event decomposition and density evolution analysis, we build a joint intra/inter-slot coding parameter optimization algorithm to minimize the SNR (signal-to-noise ratio) requirement at an expected system packet loss rate. Numerical results indicate that the proposed scheme achieves energy efficiency gain by balancing the intra/inter-slot coding gain while maintaining relatively low implementation complexity.

Keywords: unsourced multiple access; T-Fold IRSA; IDMA; degree distribution optimization; internet-of-things; 6G



Citation: Li, Y.; Niu, K.; Dong, C.; Suo, S.; Lin, J. Joint Intra/Inter-Slot Code Design for Unsourced Multiple Access in 6G Internet of Things. *Sensors* **2023**, *23*, 242. <https://doi.org/10.3390/s23010242>

Academic Editors: Yang Yue and Fang Yang

Received: 30 September 2022

Revised: 17 December 2022

Accepted: 22 December 2022

Published: 26 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background

Recently, academics and industry have proposed many prospects for the evolution of the IoT (Internet-of-Things) in the next generation [1]. In general, the 6G IoT system mainly faces the following three challenges: The rapid growth of connectivity [2], the guarantee of low latency under specific reliability [3], and the requirement of low power consumption and low implementation complexity [4].

Based on this, Polyanskiy proposed the concept of unsourced multiple access (UMA) in 2017 [5]. The UMA removes the coordination center in the network to reduce the transmission cost and latency of the frequent access of vast short packages, which is the major drawback of traditional schemes OMA (Orthogonal Multiple Access) [6], coordinated NOMA (Non-Orthogonal Multiple Access) [7], and grant-free access [8]. The uplink channel is always available to users, and the network works in the unsourced style, adapting to large-scale frequent connection requests. As a result, the system optimization criteria need to be changed from sum-rate to PUPE (Per-User Probability of Error), that is, to achieve the massive access ability under a specific average packet loss rate. The UMA's achievability bound was given in [5], laying the foundation for research in this field.

1.2. Related Works

After that, many works emerged, exploring how to approach the performance bound through practical coding design under UMA [9,10]. These schemes are based on three basic paradigms: random spreading, T-Fold Aloha [11], and T-Fold IRSA (Irregular Repeated Slotted Aloha) [12].

The random spreading scheme regards the entire transmission frame as the available length of code chips for each user. The users' data packets are superimposed after the symbol-level spreading. Representative works are the sparse IDMA (Interleave-Division Multiple Access) [13] based on LDPC (Low-Density Parity Check Code) and Polar-RS (Random Spreading) [14] and Polar-SS (Sparse Spreading) [15] based on polar codes. As such schemes occupy the entire frame to form a low-rate code, thereby obtaining the highest coding gain, which has a similar performance to that close to the boundary. However, at the same time, their high implementation complexity is also a problem that cannot be ignored.

T-Fold Aloha divides the transmission frame into slots with equal lengths. On each slot, a multi-access code is designed to control the error rate under multi-user superposition, ensuring a decodable threshold up to T . A classical realization is the concatenated code scheme [16], where the outer code guarantees the T-Fold feature, and the inner code treats noise. The compressed sensing (CS) encoder in CCS (Coded Compressed Sensing) [17] and SPARC (SParse Regression Code) [18] schemes plays the above two roles at the same time and has higher coding efficiency. However, the tree code also suffers an error propagation effect, and the CS coding gain is still limited in the region of high users.

T-Fold IRSA paradigm originated from the research on contention resolution Aloha [19] by G. Liva. Its design consists of two main aspects: the T-Fold intra-slot code and the packet-level inter-slot code. Cheng et al. proposed the SC-LDPC+IRSA scheme [20], spreading the SC-LDPC (Spatially Coupled-LDPC) packets by IRSA. Polar-IRSA [21] changed the SC-LDPC to the SCL decoded multi-user polar code with better performance under the short code length and achieved higher performance gain. In addition, [22] also gives a theoretical analysis of the IRSA framework based on the finite block length bound [23] and asymptotic analysis. Such schemes have performance advantages over T-Fold Aloha; under certain conditions, their performance is close to the high-complexity random spreading schemes. However, these schemes are based on the idealized asymptotic assumptions, the exploitation of intra-slot coding gain is limited, and there is still some room compared with the achievability bound.

In addition, some works also discuss the theoretical analysis [24] and coding scheme [25] design transferred from the AWGN channel to the Rayleigh block fading channel. Besides, MIMO (Multi-Input Multi-Output) transmission [26], channel estimation, activity detection [27], and their co-design [28] push the horizon from theory to practical deployment.

1.3. Contributions

Following the works under the T-Fold IRSA paradigm, we continue to investigate the trade-off between the intra-slot coding gain and the inter-slot diversity gain. Besides, with the premise of realization, we attempt to obtain improvement or balance in terms of performance and complexity. The main contributions of the proposed scheme are the following aspects:

1. **Enhanced intra-slot coding structure.** We apply the IDMA scheme with CS header to intra-slot code design. The user payload is split into two parts, encoded by IDMA and CS encoders separately, and combined as the intra-slot codeword. The CS part carries the user-specific interleaver pattern of the IDMA codeword, and the IDMA part is the multi-access code resolving the superposition interference.
2. **Joint intra/inter-slot iterative decoder.** Under the IRSA paradigm, the superimposed spreading pattern of intra-slot code packets is expanded as a compound inter-slot factor graph, recovered by combining the CS pilot decoding results among the slots. The ESE+BP decoder of the intra-slot IDMA code is the embedded operation on the slot nodes. The inter-slot SIC iteration is performed on the graph to eliminate the interference on the slot nodes, making the overload slot nodes decodable.
3. **Joint intra/inter-slot coding parameter optimization.** To minimize the required SNR under the given PUPE standard and a limited number of channel resources, we follow the idea of error event decomposition to build the framework of parameter optimization. The error caused by each coding module is modeled as a function of

its coding parameter. Especially, the inter-slot degree distribution is analyzed by density evolution with finite-length realization and energy cost conditions. Then we integrate these error functions in a global optimization problem and design a heuristic bootstrap search algorithm to jointly optimize all these related parameters, including the intra-slot CS pilot length, the IDMA coding rate, and the inter-slot degree distribution.

1.4. Content Organization

This article organizes its content in the following way. Section 2 gives some basic concepts and universal notations to help clarify the whole framework. Section 3 focuses on the encoder's design and the proposed scheme's decoding algorithm. Next, Section 4 discusses the coding parameter optimization problem based on error analysis by decomposition and the unified joint optimization algorithm. Finally, by numerical simulation, Section 5 evaluates its energy efficiency and computational complexity under the optimized configuration.

2. Definitions and Notations

The core problem of UMA is to construct a coding scheme that allows K_a unsourced active users to transmit length- B data payload for each on a fixed length- N_{tot} frame at a target PUPE and given SNR level. By definition [5], the PUPE can be expressed as:

$$\epsilon = \sum_{k=1}^{K_a} P\{\hat{\mathbf{w}}_k \neq \mathbf{w}_k\}, \quad (1)$$

where k is the index of users and $\hat{\mathbf{w}}_k$ is the decoded version of transmitted payload $\mathbf{w}$. When ϵ , N_{tot} , and B are given, for each K_a , there exists an optimal configuration for a coding scheme that makes the required SNR minimum, which is referred to as the SNR threshold. Thus, the energy efficiency of any UMA scheme can be characterized by the K_a -SNR threshold curve.

Under the T-Fold IRSA paradigm, the entire frame of length N_{tot} is sliced into V slots with length $N = N_{tot}/V$ for each. The degree distribution $\lambda(x)$ determines the distribution of the repetition rate of the intra-slot coded packets for each user. Since the packets are randomly distributed on slots, the number of superimposed packets L_v on slot v is also randomized. The multi-user interference increases with L_v . T-Fold means that the intra-slot code guarantees that at most T_{th} users can be correctly decoded. Thus, T_{th} is the threshold for the T-Fold IRSA system.

Here are some notations for matrices and vectors. The upper-bold case $\mathbf{A}$ represents the matrix, and $\mathbf{A}_{i,j}$ represents its element at the i -th row and the j -th column. The lower-bold case $\mathbf{c}$ represents the vector, and $\mathbf{v}_i$ is the i -th element. The vectors are column vectors in default.

3. Joint Intra/Inter-Slot Coding Scheme

The structure of the proposed coding scheme is given in this section. In general, it is an intra/inter-slot nested structure, as Figure 1 shows.

The payload of each user $\mathbf{w}_k$ is encoded by the intra-slot code to produce $\mathbf{v}_k$, and then randomly repeated β_k times to form an inter-slot packet-level spreading pattern. On the receiver side, the slot-by-slot intra-slot decoding is integrated into the packet-level SIC iteration process on the inter-slot factor graph. The intra-slot codeword $\mathbf{v}_k$ contains the IDMA codeword $\mathbf{x}_k$ with a CS pilot $\mathbf{s}_k$ indicating both the intra-slot interleaving pattern and the inter-slot spreading pattern. The inter-slot code is an IRSA structure enabling the SIC. Thus, the proposed scheme is introduced under the above framework.

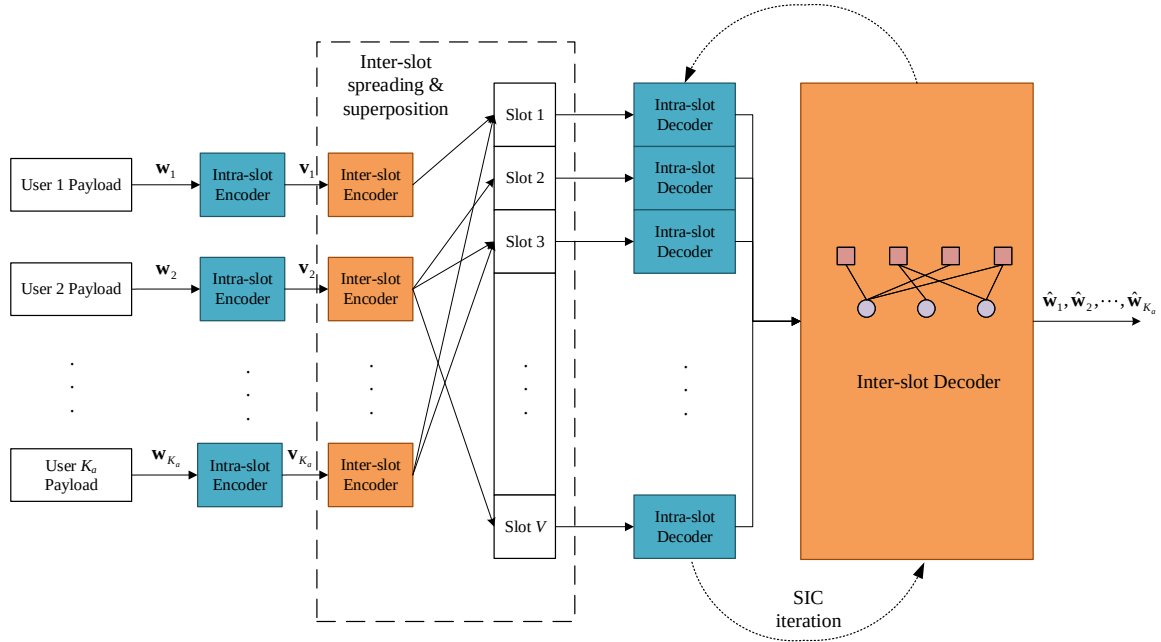


Figure 1. The overall system framework of the proposed scheme.

3.1. Encoder

The intra-slot encoder is concatenated with the inter-slot encoder. The overall scheme of the encoder is depicted in Figure 2a. For each user k , the intra-slot coded packets are the non-zero ‘chips’ of the inter-slot code. The intra-slot codeword consists of the CS pilot and the IDMA-coded part, each carrying part of the payload. Due to the unsourced feature, users must adopt a common codebook. Therefore, the CS pilot not only determines the user-specific configuration of the intra-slot code but also represents the structure of the inter-slot code, which are both randomized to deal with the multi-user interference.

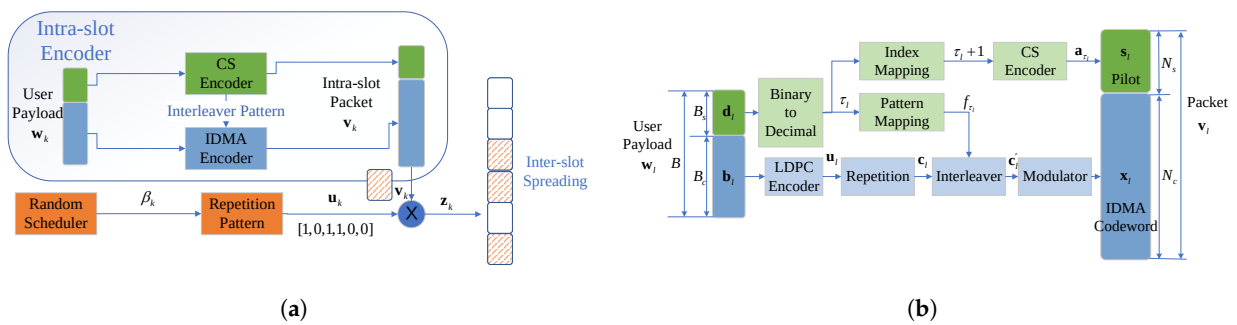


Figure 2. (a) The overall structure of the encoder. (b) The Intra-slot encoder.

3.1.1. Intra-Slot Encoding

The permutation pattern of each user should be known at the receiver side to enable the decoding process, by transmitting it as an encoded pilot attached ahead of the IDMA data coded part. Thus, we slice $\mathbf{w}_l$, the information payload of user l , into two parts: the pilot info sequence $\mathbf{d}_l = [\mathbf{w}_{l,1}, \dots, \mathbf{w}_{l,B_s}]$ of length B_s that carries both the permutation pattern and part of the user info, along with the data info sequence $\mathbf{b}_l = [\mathbf{w}_{l,B_s+1}, \dots, \mathbf{w}_{l,B}]$ of length $B_c = B - B_s$ for IDMA encoder.

The function of the CS encoder is to map $\mathbf{d}_l$ into pilot $\mathbf{s}_l$. The binary sequence $\mathbf{d}_l$ is firstly converted to decimal τ_l . Then τ_l is through a bijective map to the column index of

M_s -by- N_s sensing matrix $\mathbf{A}$, usually configured as a normalized Gaussian random matrix. As the length of $\mathbf{A}$ is B_s , the number of columns of $\mathbf{A}$, $N_s \geq 2^{B_s}$. A natural but effective mapping is to choose the $\tau_l + 1$ -th column of $\mathbf{A}$ as the length- N_s pilot, i.e., $\mathbf{s}_l = \mathbf{a}_{\tau_l+1}$.

The data sequence $\mathbf{b}_l$ is sent to a common rate- R_L LDPC encoder identical among all users, generating codeword bit sequence $\mathbf{u}_l$ with length $B_u = B_c/R_L$. To introduce intra-slot coding diversity, $\mathbf{u}_l$ is repeated R_r times to produce a low rate codeword $\mathbf{c}_l$ with length B_c , where the repetition rate is also the same for all users. The next step is the user-specific interleaver. As mentioned before, the permutation pattern is determined by $\mathbf{d}_l$. The decimal τ_l converted from $\mathbf{d}_l$ is used to choose the τ -th pattern $f_{\tau_l} \in \mathcal{F}$ in the common pattern set $\mathcal{F}$. Through permutation function $\mathbf{c}'_l = f_{\tau_l}(\mathbf{c}_l)$, we get the interleaved bit sequence $\mathbf{c}'_l$. According to constellation $\mathcal{G}$, $\mathbf{c}'_l$ is modulated to IDMA codeword symbol sequence $\mathbf{x}_l$ with length $N_c = B_c/(R_L * R_r * N_m)$, where the size of the constellation is $Q_m = |\mathcal{G}|$, and the modulation order is $N_m = \log_2(Q_m)$. Especially, $N_c = B_c/(R_L * R_r)$ for BPSK (Binary Phase Shift Keying).

Finally, $\mathbf{x}_l$ and $\mathbf{s}_l$ are stitched together to form the whole packet of user l , i.e., $\mathbf{v}_l = [\mathbf{b}_l; \mathbf{d}_l]$. After assembling, the length of $\mathbf{v}_l$ is $N = N_s + N_c$. In general, the intra-slot encoder is common for every user. This not only keeps the simplicity of the encoding process but also ensures the common codebook requirement of unsourced settings. The ability of intra-slot code to distinguish superimposed user packets mainly comes from the compressed sensing pilot code and IDMA's user-specific random interleaver, which are all dependent on the randomness of the pilot info slice $\mathbf{d}_l$. The whole structure of the intra-slot encoder is shown in Figure 2b.

3.1.2. Inter-Slot Irregular Spreading

The inter-slot encoding process is based on the intra-slot code, which adds irregular spreading diversity among different user packets by random scheduling, while the encoder remains the same structure shared by all users. Each user k determines the number of packet repeats β_k based on the local random scheduler, making the distribution of beta $P(\beta_k = i)$ approaches the preset $\lambda(x)$. Under the IRSA scheme, $\lambda(x)$ is the distribution polynomial of β_k , written as:

$$\lambda(x) = \sum_{i=1}^{I_{\max}} \lambda_i x^i = \sum_{i=1}^{I_{\max}} P(\beta_k = i) x^i, \quad (2)$$

where λ_i is the probability of user node of degree- i on the packet superposition factor graph and satisfies the normalized constraint $\lambda(1) = \sum_{i=1}^{I_{\max}} \lambda_i = 1$. Therefore, the random scheduler generates $\beta_k \sim \lambda(x)$ and then creates a vector:

$$\delta'_l = [\underbrace{1, 1, \dots, 1}_{\beta_k}, \underbrace{0, 0, \dots, 0}_{V-\beta_k}], \quad (3)$$

which is then randomly permuted to ensure the 1-elements uniform, producing the β_k -sparse irregular diversity mapping vector δ_k . Let the intra-slot codeword $\mathbf{v}_k$ take Kronecker product with spreading pattern δ_k :

$$\mathbf{z}_k = \mathbf{v}_k \otimes \delta_k. \quad (4)$$

$\mathbf{v}_k$ is sent to the slot where the element is one in δ_k . When $\beta = 1$ the packet is not repeated, and when $\beta > 1$ the packet gains repetition diversity. Each user follows the above structure and superimposes their spread packets $\mathbf{z}_k$ in the AWGN (Additive White Gaussian Noise) channel. The received signal $\bar{\mathbf{y}}$ is:

$$\bar{\mathbf{y}} = \sum_{k=1}^{K_d} \mathbf{z}_k + \mathbf{n}, \quad (5)$$

where $\mathbf{n}$ is the Gaussian noise with variance σ^2 . Through this inter-slot encoding, the repeated user packets on different slots form a sparse spreading structure, providing packet-level diversity gain.

3.2. Joint Intra/Inter-Slot Decoder on Compound Factor Graph

Accordingly, the receiver follows the mirrored structure. The inter/intra-slot coding configurations are first recovered by pilot decoding, and then the intra-slot decoder is embedded as an operation on the slot nodes of the inter-slot SIC procedure. The receiver will be introduced in the following subsections under the above framework.

3.2.1. Intra-Slot Decoder

The iteration begins with the intra-slot packet decoding process. The received packet on each slot v can be sliced from the whole received signal $\bar{\mathbf{y}}$:

$$\mathbf{y}_v = [\bar{\mathbf{y}}_{(v-1)N+1}, \dots, \bar{\mathbf{y}}_{vN}] = \sum_{l=1}^{L_v} \gamma_{v,l} + \mathbf{n}, \quad (6)$$

where L_v is the number of superimposed user on slot v . According to the intra-slot encoding structure, $\mathbf{y}_v$ can be further separated into two parts: the CS header $\mathbf{y}_v^s = [\gamma_{v,1}, \dots, \gamma_{v,N_s}]$ and the IDMA signal $\mathbf{y}_v^c = [\gamma_{v,N_s+1}, \dots, \gamma_{v,N}]$. Since the IDMA decoding requires the interleaver pattern, the CS pilot is decoded first. $\mathbf{y}_v^s$ can be expanded by:

$$\mathbf{y}_v^s = \sum_{l=1}^{L_v} \mathbf{s}_{v,l} + \mathbf{n} = \sum_{l=1}^{L_v} \mathbf{a}_{\tau_l+1} + \mathbf{n} = \sum_{l=1}^{L_v} \mathbf{A} \mathbf{e}_{\tau_l+1} + \mathbf{n} = \mathbf{A} \mathbf{g}_v + \mathbf{n}, \quad (7)$$

where $\mathbf{e}_{\tau_l+1}$ is a 1-sparse vector with all-zeros elements except for position $\tau_l + 1$, and $\mathbf{g}_v = \sum_{l=1}^{L_v} \mathbf{e}_{\tau_l+1}$ is an L -sparse vector, assuming no resource collision among users. By sending $\mathbf{y}_v^s$ to the support recovery algorithm [29], the column index set of $\mathbf{A}$, as we modeled, the support set $\hat{\mathcal{D}}^s$, is searched out. Remap the elements in $\hat{\mathcal{D}}^s$ back to decimals $\hat{\tau}_l$ by aligning the $\hat{\tau}_l - 1$ -th column to $\hat{\tau}_l$. Then convert the decimals $\hat{\tau}_l$ to length- B_s binary sequences $\hat{\mathbf{d}}_l$, which are the pilot info parts of user payloads. Besides, the interleaver patterns $f_{\hat{\tau}_l}$ are recovered by selecting the τ -th pattern $f_{\hat{\tau}_l} \in \mathcal{F}$.

After the recovery of $f_{\hat{\tau}_l}$, the IDMA decoding can be started. Since the intra-slot repetition rates are the same among users, we can utilize a simple linear algorithm, the ESE+BP iterative structure [30], as depicted in Figure 3.

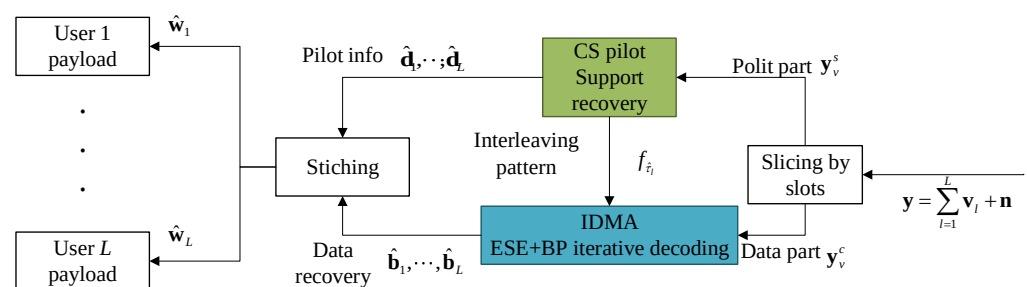


Figure 3. Intra-slot decoder structure at slot v .

The ESE+BP iteration may continue for a specific number of times $I_{\max}^1$, and then the final hard-decision output $\hat{\mathbf{b}}_l$ at the BP decoder of each branch is stitched together with the corresponding pilot info part $\hat{\mathbf{d}}_l$, reconstructing the complete payload decoding result $\hat{\mathbf{w}}_l$.

3.2.2. Inter-Slot Decoder

To recover the superimposed multi-user packets on the overload slots, the inter-slot decoding performs SIC on the compound packet-level factor graph known by the

reconstruction process. After the initial intra-slot decoding on all slots, the CS pilots should be recovered as pilot info $\hat{\mathbf{d}}_v$, while the corresponding IDMA parts $\hat{\mathbf{b}}_v$ are not all successfully decoded, especially for those on the overload slots.

Using the CS pilots as pointers, the repetition relationship can be confirmed by comparing $\hat{\mathbf{d}}_{v,k}$ in one slot with $\hat{\mathbf{d}}_{v_1,k_1}$ on the others. According to Section 3.1.2, the number of unique pilots among all slots is the number of users K_a . For user packet k , the indexes of the slots where a replica of it exists form a set:

$$\mathcal{U}_k = \{v_1 \mid \hat{\mathbf{d}}_{v,k} = \hat{\mathbf{d}}_{v_1,k}, v \neq v_1\}, \quad (8)$$

where $\max |\mathcal{U}_v| = I_{\max}$. We combine those sets $\mathcal{U}_1, \dots, \mathcal{U}_{K_a}$ as adjacent matrix $\mathbf{U}$, where the edge between user node k and slot node v is determined. Thus, the intra-slot packet spreading structure can be recovered as the compound factor graph in Figure 4. And The process of the inter-slot decoding described in Algorithm 1 is performed on this graph structure.

Algorithm 1 Inter-slot SIC decoding on the compound packet-level factor graph

Require: The received signal $\mathbf{y}_v$ on each slot v , compressed sensing matrix $\mathbf{A}$, interleaver set $\mathcal{F}$, constellation $\mathcal{A}$, noise variance σ^2 , threshold T_{th} , maximum SIC iteration J .

Ensure: Decoded payloads of all users $\hat{\mathbf{w}}_k$.

- 1: Initialization: Perform the intra-slot decoding in Section 3.2.1 on each slot, and get the pilot info $\hat{\mathbf{d}}_{v,k}$, the IDMA decoding result $\hat{\mathbf{b}}_{v,k}$, and the superimposed number of users $L_v^1 = |\hat{\mathcal{D}}_v^s|$ on each slot;
 - 2: Factor graph reconstruction: Compare all the CS pilots $\hat{\mathbf{d}}_{v,k}$, and combine the repetition relationship sets $\mathcal{U}_l$ into adjacent matrix $\mathbf{U}$ by (8);
 - 3: **while** $j \leq J$ **do**
 - 4: Update the slot nodes subsets $\mathcal{V}_j^+$ and $\mathcal{V}_j^-$ by (9), and the user node subsets $\mathcal{K}_j^+, \mathcal{K}_j^-$ and $\mathcal{K}_{v'}^+$ accordingly;
 - 5: **for all** $v' \in \mathcal{V}^-$ **do**
 - 6: **if** $L_{v'}^j - \mathcal{K}_{v'}^+ \leq T_{th}$ **then**
 - 7: Forward message Passing (from user node l to slot node v');
 - 8: Remap the user messages in the effective edge set $\hat{\mathbf{b}}_k, k \in \mathcal{K}_{v'}^+$ to IDMA packets $\mathbf{x}_k^{\text{SIC}}$;
 - 9: Peel off the known interference $\mathbf{x}_k^{\text{SIC}}$ as (11);
 - 10: Backward Message Passing (from slot node v' to user node k);
 - 11: Intra-slot decoding: Perform the IDMA decoding part on slot v' , get the recovered user information $\hat{\mathbf{b}}_{v',k}$;
 - 12: User node update: Add/Subtract the newly recovered user on slot v' in $\mathcal{K}_{j+1}^+ / \mathcal{K}_{j+1}^-$;
 - 13: Update the slot counter $L_{v'}^{j+1} = L_{v'}^j - |\mathcal{K}_{v'}^+|$.
 - 14: **else**
 - 15: SIC not started, reserve the slot counter $L_{v'}^{j+1} = L_{v'}^j$.
 - 16: **end if**
 - 17: **end for**
 - 18: **end while**
-

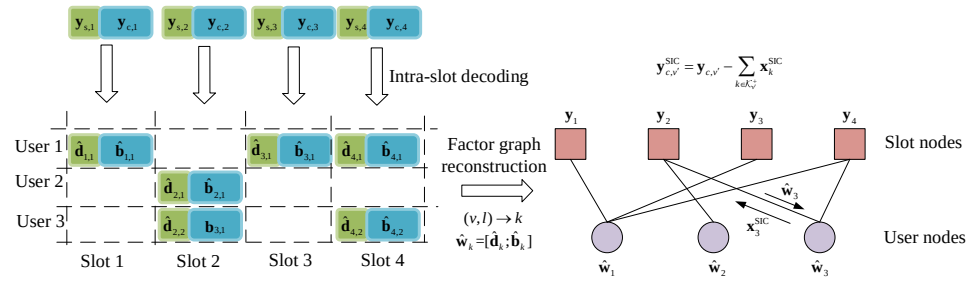


Figure 4. An example for inter-slot factor graph reconstruction and SIC process, where $K_a = 3$ and $V = 4$.

At the j -th SIC iteration, according to the T-Fold IRSA model, the slot nodes set can be separated into two subsets by the given threshold T_{th}

$$\begin{aligned} v &\in \{v \mid L_v^j \leq T_{th}\} = \mathcal{V}^+ \\ v' &\in \{v' \mid L_{v'}^j > T_{th}\} = \mathcal{V}^- \end{aligned} \quad (9)$$

where the initial slot node degree $L_v^1 = |\hat{\mathcal{D}}_v^s|$. Also, the user nodes set has two subsets, the successfully decoded set $\mathcal{K}_j^+$ and the undecoded set $\mathcal{K}_j^-$. For each underload slot node $v \in \mathcal{V}^+$ that satisfies the decodable condition, add its adjacent user node k into $\mathcal{K}_j^+$. Based on this graph, the message passed from the user node to the slot node is the remapped decoded IDMA packet $\mathbf{x}_k^{\text{SIC}}$. After the SIC peeling session, if the degree of an overload slot node can be reduced under the threshold T_{th} , this slot will be decodable, where its output message is the reliable decoded IDMA info parts $\hat{\mathbf{w}}_k$. Define the effective edge set of slot v' as $\mathcal{K}_{v'}^+ = \{k \mid \mathbf{U}_{k,v} = 1, k \in \mathcal{K}^+\}$, then the decodable condition of overload slot v' after the j -th SIC is:

$$L_{v'}^j - |\mathcal{K}_{v'}^+| \leq T_{th}. \quad (10)$$

If this condition is satisfied, peel the known interference off on this slot:

$$\mathbf{y}_{v'}^{\text{SIC}} = \mathbf{y}_{v'} - \sum_{k \in \mathcal{K}_{v'}^+} \mathbf{x}_k^{\text{SIC}}. \quad (11)$$

Then send $\mathbf{y}_{v'}^{\text{SIC}}$ to the intra-slot decoder in Section 3.2.1, where the reliable decoded message $\hat{\mathbf{w}}_{v',k}$ can be recovered. Consequently, the user node subsets $\mathcal{K}_j^+$ and $\mathcal{K}_j^-$ can be updated by the backward messages. Perform this process on each overload slot $v' \in \mathcal{V}^-$. As a result, $|\mathcal{K}_j^-|$ will decrease after each iteration, for the underload slots are always helping the overload ones by message propagation on the factor graph. Under appropriate conditions, the SIC can converge to the required level.

4. Performance Analysis and Parameter Optimization

The goal of the coding parameter optimization problem of the proposed scheme is to minimize the SNR threshold under a given PUPE (in Section 2). However, due to the complicated intra/inter-slot encoding structure, the relationship between the coding parameters and the performance indicator is not explicit. Therefore, this section addresses the problem by error event decomposition [16]. By breaking the system-level PUPE down to module-level error rates, the error contribution of each module can be analyzed and correlated with their parameters and eventually form a system-level parameter optimization problem.

4.1. Error Rate Analysis by Decomposition

According to the decoder structure described in Section 3.2, the PUPE in (1) can be decomposed into four parts, as depicted in Figure 5.

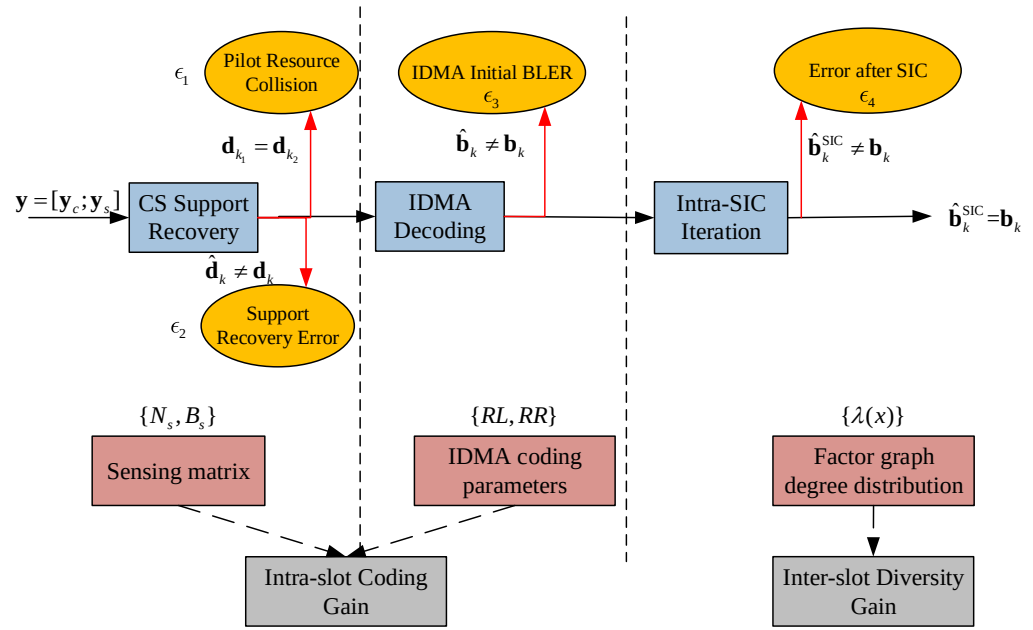


Figure 5. The system-level packet loss broken down into module-level error events, with the corresponding coding parameters and coding gains.

Although, as described in Algorithm 1, the decoding structure consists of the compound intra/inter-slot iteration, it can nonetheless be regarded as a three-stage process when analyzing errors due to its successive style. The first stage is the CS pilot recovery, where the pilot resource collision $\mathbf{d}_{k_1} = \mathbf{d}_{k_2}, k_1 \neq k_2$ occurs at the transmitter side and the support recovery error $\hat{\mathbf{d}}_k \neq \mathbf{d}_k$ at the receiver side. The pilot plays three significant roles: the first part of the user payload, the user-specific IDMA interleaver, and the pointer used to reconstruct the packet-level factor graph. Thus, the pilot error will cause not only packet loss of its user but also the chain effect spreading to the correlated packets of other users. The IDMA decoding error $\hat{\mathbf{b}}_k \neq \mathbf{b}_k$ and the remaining error after the intra-slot SIC process $\hat{\mathbf{b}}_k^{\text{SIC}} \neq \mathbf{b}_k$ are both conditioned on the successfully recovered pilots. Then the modular errors are expressed as the functions of their corresponding coding parameters in (12).

$$\begin{aligned}
 P(\mathbf{w}_l \neq \hat{\mathbf{w}}_l) &= P\{\mathbf{d}_{k_1} = \mathbf{d}_{k_2}\} + P\{\hat{\mathbf{d}}_l \neq \mathbf{d}_l\} \\
 &+ P\{\hat{\mathbf{b}}_k \neq \mathbf{b}_k \mid \hat{\mathbf{d}}_l = \mathbf{d}_l\} + P\{\hat{\mathbf{b}}_k^{\text{SIC}} \neq \mathbf{b}_k \mid \hat{\mathbf{d}}_k = \mathbf{d}_k\} \\
 &= \epsilon_1(N_s) + \epsilon_2(N_s, B_s) + \epsilon_3(R_L, R_R) + \epsilon_4(\lambda(x))
 \end{aligned} \quad (12)$$

The CS pilot parameters N_s and B_s are restricted by the collision and detection conditions. ϵ_1 is the resource collision avoiding condition in [16]. The length of the separated pilot info part from user payload B_s should be large enough to provide non-collision patterns. As for ϵ_2 , the dimension of the sensing matrix is bounded by the Restricted Isometry Property (RIP) [31] condition. In other words, when T_{th} is fixed, large row dimension N_s can reduce the ϵ_2 . Thus, we can conclude that $N_s \propto T_{th}$ and $B_s \propto T_{th}$.

The IDMA block error rate ϵ_3 is the function $\epsilon_3 = P(B_c, R_L, R_r, L_v, \text{SNR})$. According to the classical analysis of IDMA, ϵ_3 is based on the single-user performance of the inner rate- R_L FEC code and degrades with the gradually severe interference caused by the increase of the number of superimposed users L_v . The SNR cost of ϵ_3 at a required level can be extracted on the performance curve of a given threshold T_{th} by simulation.

4.2. Inter-Slot Degree Distribution Analysis

As the SNR cost of the IDMA system increases with T_{th} , the SIC decoding on the compound factor graph of the inter-slot code can further reduce the threshold T_{th} to

reach the same level of PUPE, resulting in the reduction of SNR threshold. Thus, degree distribution optimization aims to minimize T_{th} .

First, we start with an idealized asymptotic investigation. Previously the T-Fold IRSA was modeled as a factor graph, on which the user node degree distribution $\lambda(x)$ determines the slot node degree distribution $\rho(x)$:

$$\rho(x) = \sum_{i=r}^{R_{\max}} \rho_r x^r = \sum_{r=1}^{R_{\max}} P(L_v = r) x^r \quad (13)$$

$$\rho_r = P(\text{Bino}(\lambda'(1)K_a, 1/V) = r)$$

where $\text{Bino}(\cdot)$ denotes the binomial distribution. The probability of high-degree slot nodes increases with the average packet repetition rate $\lambda'(1)$ and decreases with more slots V , representing intensified superposition. Figure 6a gives an example of this rule.

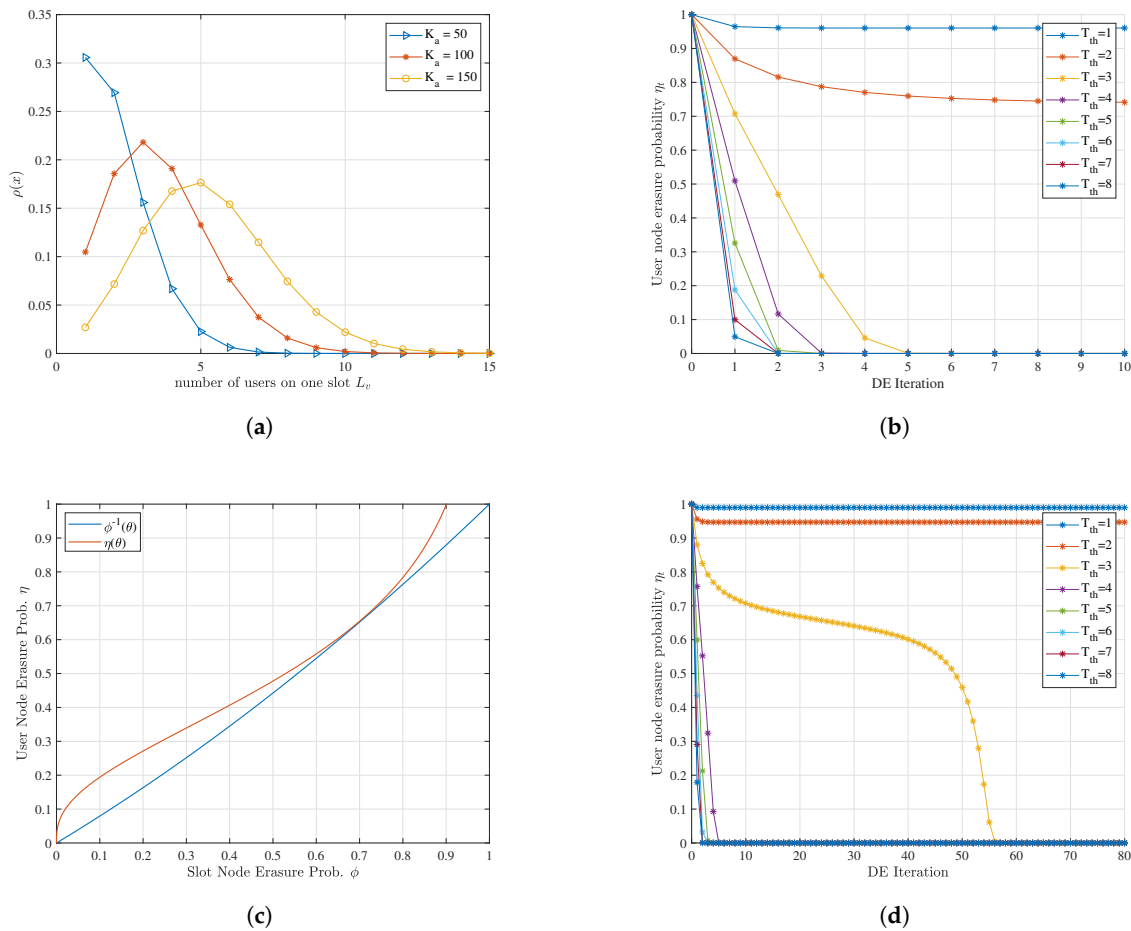


Figure 6. Density evolution analysis of inter-slot code: (a) The degree distribution of slot nodes $\rho(x)$ at $\lambda(x) = 0.5x^2 + 0.5x$ and different users. (b) The density evolution procedure at $\lambda(x) = 0.5x^2 + 0.5x$, $K_a = 100$ and $V = 28$, when $T_{th} < 3$ it does not converge. (c) The density evolution EXIT chart at $\lambda(x) = 0.23x^2 + 0.77x$, $K_a = 150$, $V = 28$ and $T_{th} = 3$. The tunnel is about to close, representing the boundary condition. (d) Corresponding density evolution iteration procedure at the same configuration with (c) but varying T_{th} , where $T_{th} = 3$ is just able to converge.

Moreover, the convergence behavior of the SIC on the factor graph can be characterized by density evolution (DE), which is similar to the LDPC message passing procedure under

the erasure channel [32]. At the t -th iteration, the erasure probability of the slot nodes is ϕ_t and η_t for the user nodes:

$$\begin{aligned}\phi_t(\eta_t) &= 1 - \left(\sum_{r=1}^{T_{th}} \rho_r + \sum_{r>T} \rho_r \sum_{i=0}^{T_{th}-1} C_{r-1}^i (1 - \eta_t)^{r-1-i} \eta_t^i \right) \\ \eta_{t+1}(\phi_t) &= \lambda(\phi_t).\end{aligned}\quad (14)$$

where $x_0 = 1$ at the beginning without a priori knowledge of user nodes, and $\rho(x)$ can be derived from λ as in (13). Under appropriate $\lambda(x)$, the erasure probability of slot nodes ϕ_t gradually descends with iteration. As $\lambda(\phi_t)$ is a positive-coefficient polynomial function, $\eta_{t+1}(\phi_t)$ is monotonically increasing, indicating that η_t also converges to 0 with descending ϕ_t . Meanwhile, the higher the threshold T_{th} is, the higher the decodable probability of slot nodes $\sum_{r=1}^{T_{th}} \rho_r$ is. Notice that the two-probability expression corresponds to the two message-passing procedures in Algorithm 1, respectively. Therefore, the SIC iteration can make as many overload user nodes decodable as possible. An example of the SIC procedure simulated by density evolution is depicted in Figure 6b. When V and $\lambda(x)$ are fixed, the cost of carrying more users K_a is the increase of threshold T_{th} to guarantee convergence under limited iterations, i.e., the rise of IDMA's SNR requirement.

The tool to determine the convergence condition of density evolution iteration is its EXIT (EXtrinsic Information Transfer) chart [33]. Under given K_a , V and T_{th} , plot the erasure probability curves $\phi^{-1}(\theta)$ and $\eta(\theta)$ in (14) on one chart, and find a trajectory between those two curves starting from $\eta(1) = 1$. If the trajectory reaches $\eta(0) = 0$, this $\lambda(x)$ enables the SIC iteration to converge, conversely not. Figure 6c,d shows a boundary condition case where the iteration tunnel is about to close for T_{th} . Since $\lambda(x)''(x) > 0$, $\eta(\theta)$ is always concave. Moreover, this effect becomes stronger when the weights of higher-order coefficients increase. However, it is not easy to explicitly express the boundary condition. We adopt the numerical method, differential evolution [34], to solve the implicit equation and search the range of suitable $\lambda(x)$:

$$\begin{aligned}\lambda^*(x) &\in \{\lambda(x) \mid \phi^{-1}(\theta) - \eta(\theta) > 0, \theta \in (0, 1)\} \\ \text{s.t.} \quad &\phi(\theta) = 1 - \left(\sum_{r=1}^T \rho_r + \sum_{r>T} \rho_r \sum_{i=0}^{T-1} C_{r-1}^i (1 - \theta)^{r-1-i} \theta^i \right) \\ &\eta(\theta) = \lambda(\theta), \quad \lambda_i \geq 0, \quad T \in \mathcal{N}^+ \\ &\rho_r = P(\text{Bino}(\lambda'(1)K_a, 1/V) = r)\end{aligned}\quad (15)$$

Then we move further to practical consideration. An important preassumption for density evolution analysis is that $K_a, V \rightarrow \infty$ ensures the isotropy of distribution, while in practice, they are limited. It is challenging for the user-independent random schedulers to guarantee $\beta_k \sim \lambda(x)$ when K_a is relatively small, especially for the repetition times with low probability (high-order degrees). On the other hand, the finite length effect causes short cycles and trapping loops in the randomly formed factor graph. In some worse cases, the SIC iteration cannot start or converge. Although the effect of ϵ_3 can be ignored when considering boundary conditions, the remaining errors on the underload slot nodes may cause error propagation. Nonetheless, Λ^* gives a basic scope, which just needs to be narrowed.

We use Monte-Carlo simulation to practically examine the effectiveness of candidates in Λ^* . Under finite K_a , V , and ϵ_3 , the post-iteration packet loss rate of user nodes obtained by simulation is the actual PUPE. Besides, the cost of the packet diversity gain is the additional energy spent on repeated packets. The SNR threshold after inter-slot encoding is:

$$E_b/N_0 = (E_b/N_0)_{T_{th}} \lambda'(1) \quad (16)$$

where $(E_b/N_0)_{T_{th}}$ is the SNR when ϵ_3 reaches an effective level under threshold T_{th} . After that, the trade-off of intra-slot encoding should also be considered.

4.3. Joint Parameter Optimization Algorithm

Integrating the analysis on the above modules, the complete parameter optimization procedure is proposed as Algorithm 2.

Algorithm 2 Joint optimization of coding parameters

Require: the length of user payload B , number of active users K_a , frame length N_{tot} , and target PUPE ϵ .

Ensure: the length of CS pilot info part B_s , length of CS pilot N_s , LDPC code rate R_L , intra-slot repetition rate R_r , and inter-slot packet spreading distribution $\lambda(x)$.

- 1: Initialization: Determine N_s and B_s by the CS decoding conditions, randomly chose rate configuration $\{R_L, R_c\}$, and calculate V ;
- 2: **while** $R > R_{\min}$ **do**
- 3: **for** T_{th} from $T_{\min}$ to $T_{\max}$ **do**
- 4: Obtain ϵ_3 and $(E_b/N_0)_{T_{th}}$ by IDMA simulation;
- 5: Search the convergable range Λ^* using density evolution by (15);
- 6: **for** $\lambda(x)$ from $\min \lambda'(1)$ to $\max \lambda'(1)$ **do**
- 7: Perform Monte-Carlo simulation to validate $\lambda(x)$ under the residual error ϵ_3 ;
- 8: Output the post-SIC-iteration error rate ϵ_4 ;
- 9: **if** $\epsilon_4 \leq \epsilon$ **then**
- 10: Reserve $\lambda(x)$ and break;
- 11: **end if**
- 12: **end for**
- 13: **end for**
- 14: **if** SNR threshold (16) increases **then**
- 15: Choose the $\{R_L, R_c\}$ pair with higher total rate R , adjust V accordingly;
- 16: **else**
- 17: Lower the total rate R , adjust V accordingly.
- 18: **end if**
- 19: **end while**
- 20: Calculate the SNR threshold by (16) and output the optimal coding parameters.

Overall, we use a heuristic bootstrap method to jointly optimize the coding parameters analyzed above. The CS pilot configurations control ϵ_1 and ϵ_2 , basically determined by K_a . To tackle the contradiction between intra-slot coding gain and inter-slot diversity gain, the IDMA rate R increases with iterations, while the inter-slot diversity decreases in each iteration. The initial rate R is randomly chosen, and then V can be determined. $(E_b/N_0)_{T_{th}}$ at ϵ_3 is extracted on simulation curves. The convergence condition is ensured by (15). Then the effectiveness of $\lambda(x)$ with increasing energy cost is checked by simulation until it satisfies the post-SIC-iteration PUPE requirement. If the SNR threshold raises compared with the last iteration after optimization, R should be increased to enhance the inter-slot code. If not, lower R to reduce the multi-user interference.

5. Numerical Results

In this section, we evaluate the proposed scheme using numerical indexes of two aspects: the K_a -SNR threshold curve representing energy efficiency and the FLOPf (Floating-point Operations Per frame) comparison representing computational complexity.

5.1. Energy Efficiency Analysis

The SNR threshold, by definition, is the minimum required SNR that achieves a target PUPE under specific configurations. As described in Section 2, the K_a -SNR curve is acquired under some fundamental constraints. Thus, we give the primary scenario configurations in Table 1.

Table 1. Primary scenario and configurations in numerical simulations.

Parameters	Configurations
Total frame length N_{tot}	30,000
User payload length B	100
LDPC encoder	5G NR BG2 [35]
Number of active users K_a	50 to 300
IDMA modulation	BPSK
CS pilot sensing matrix	Gaussian random matrix
PUPE requirement ϵ	0.1

These parameters are shared in the following simulations. To get the SNR threshold for each K_a , we optimize the coding parameters by Algorithm 2 point-by-point. The results are displayed in Table 2. Under all the configurations, $\epsilon_1 < 10^{-3}$ and $\epsilon < 10^{-4}$, so that the pilot error would not affect the subsequent decoders. Through adjustment, the intra-slot coding gain and inter-slot diversity gain are balanced, the sum of which reaches the optimal point.

Table 2. Optimized coding parameters of the proposed scheme.

Parameters	Optimized Results					
Number of active users K_a	50	100	150	200	250	300
Pilot info length B_s	11	11	11	11	12	12
IDMA data info length B_c	89	89	89	89	88	88
CS pilot length N_s	100	100	100	100	125	125
LDPC rate R_L	1/5	1/5	1/3	1/3	2/5	1/4
Intra-slot repetition rate R_r	1/2					
IDMA rate R	0.1	0.1	0.167	0.167	0.2	0.25
Number of slots V	28	28	43	43	48	57
threshold T_{th}	2	3	3	3	4	5
Optimized degree distribution	$0.11x^2 + 0.89x$	$0.12x^2 + 0.88x$	$0.13x^2 + 0.87x$	$0.15x^2 + 0.85x$	$0.18x^2 + 0.82x$	$0.2x^2 + 0.8x$
Average packet repetition rate	1.11	1.12	1.13	1.15	1.18	1.20
SNR threshold	2.75	2.89	3.23	3.91	4.11	4.79

It can be observed that the proportion of the two types of gains varies with K_a . In the low K_a region, the intra-slot coding gain is more effective against multi-user interference, and the cost of reducing V is affordable. However, when it comes to the high K_a region, the main problem is to tackle the rise of threshold T_{th} by the inter-slot SIC. Meanwhile, the severe superposition of packets requires V to increase, so the inter-slot diversity gain dominates.

Next, we compare the proposed scheme with several existing representatives in Figure 7a. Based on the T-Fold Aloha scheme, the CCS-AMP [36] uses CS as the intra-slot encoder to resolve superposition and enhance the AMP algorithm. As introduced in Section 1, the CCS-AMP scheme is based on the T-Fold Aloha scheme, while SC-LDPC+SIC [20] and Polar-IRSA [21] are T-Fold IRSA. Sparse IDMA [13] and Polar-sparse spreading (Polar-SS) [15] are in the random spreading paradigm. Polar-IRSA achieves the best performance in the IRSA category with a better intra-slot code design. The excellent performance of polar codes in short length makes polar code base schemes stand out in low K_a conditions where the intra-slot coding gain is critical. Our scheme occupies the middle position among the three for its repetition diversity in intra-slot IDMA code provides more coding gain than pure LDPC. The random spread spectrum schemes have the best performance of all paradigms, which can be decomposed into a cascaded structure on one frame-length long time slot. The sparse outer code can eliminate multi-user interference and provide a particular coding gain in the low K_a region.

As K_a increases, our scheme maintains the lowest slope and eventually achieves a performance advantage in the high K_a region. This is mainly because we effectively control

the increase of the threshold T_{th} by utilizing fine optimization of the intra-slot gain and the inter-slot gain. The outer code of the sparse spreading schemes tends to be rateless, and its sparsity and gain gradually diminish. Meanwhile, other IRSA schemes only exploiting packet-level diversity suffer the same problem, let alone the bit-level spreading outperforms the packet-level one.

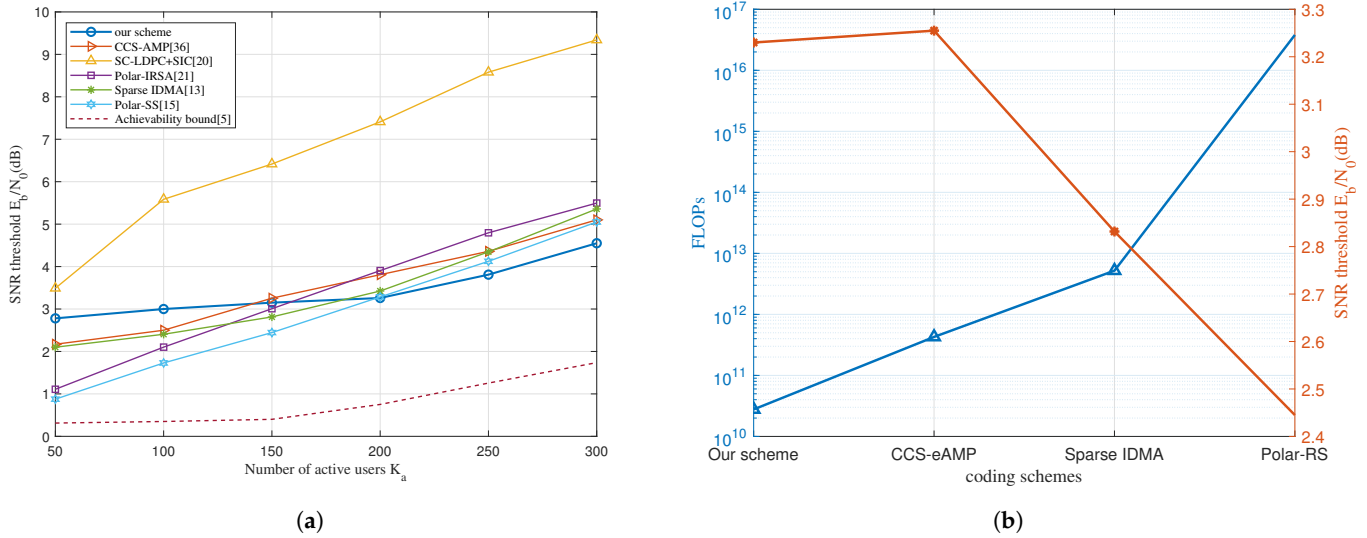


Figure 7. (a) Comparison of K_a -SNR threshold curves of different coding schemes. $N_{tot} = 30,000$, $B = 100$, PUPE = 0.1. (b) Performance and computational complexity comparison at $N_{tot} = 30,000$, $B = 100$, $K_a = 150$, PUPE = 0.1.

5.2. Complexity Analysis

Based on the analytical framework proposed in [10], the FLOP can represent the complexity of the decoder. For each encoding scheme, FLOP can be expressed as a function of its coding parameters. We compare the proposed scheme with Polar-RS, Polar-IRSA, and CCS-eAMP, the expressions of which are listed in Appendix A. When comparing these schemes, it is insightful to investigate the complexity cost of the corresponding energy efficiency gain. Figure 7b shows the SNR level and the required FLOP of each scheme when the number of users K_a reaches 150.

Although our scheme's and CCS-AMP's performance are the same, the latter costs about ten times more FLOP than the former. Meanwhile, sparse IDMA spends 100 times more complexity for 0.4dB energy gain, while for the polar-RS scheme, it is 10^6 for 0.8 dB. The intra-slot decoder is a linear iteration and converges after 3 to 5 iterations due to the high coding gain. Besides, the inter-slot iteration is restricted to 10 times by the joint degree distribution optimization. Therefore, our scheme achieves a better energy-complexity trade-off.

6. Conclusions

The proposed coding scheme achieves performance gain in the high K_a region, benefiting from intra/inter-slot gain balance under jointly optimized parameters. Although it is not as good as existing solutions in the low K_a region, it achieves a better trade-off between performance and complexity. The joint design of intra/inter-slot code exploits the potential of the T-Fold IRSA scheme more thoroughly. Thus, our scheme would be a prospective candidate for UMA design in next-generation IoT.

For future research, we can consider the fading scenario. Our scheme can easily be promoted to the block fading channel because of two inherent advantages: 1. Inserting the pilot at each slot means the channel response of each user at each slot can be independently estimated; 2. The intra-slot IDMA is a universal code. After optimizing it under the AWGN channel, its configuration can be directly applied to the fading scenario.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L. and C.D.; software, Y.L.; validation, Y.L.; formal analysis, Y.L. and K.N.; investigation, Y.L. and C.D.; resources, S.S. and J.L.; data curation, Y.L., C.D. and S.S.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L., C.D. and K.N.; visualization, Y.L.; supervision, C.D., K.N. and J.L.; project administration, S.S. and K.N.; funding acquisition, S.S. and K.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Key Program of the National Natural Science Foundation of China (No. 92067202), the National Natural Science Foundation of China (No. 62071058), CICT Mobile Communication Technology Co., Ltd., and Beijing University of Posts and Telecommunications—China Mobile Research Institute Joint Innovation Center.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UMA	Unsourced Multiple Access
IoT	Internet-of-Things
IRSA	Irregular Repeated Slotted Aloha
OMA	Orthogonal Multiple Access
NOMA	Non-Orthogonal Multiple Access
RACH	Random Access CHannel
PUPE	Per-User Probability of Error
IDMA	Interleave-Division Multiple Access
LDPC	Low Density Parity Check Code
RS	Random Spreading
SS	Sparse Spreading
SCL	Successive Cancellation List
SIC	Successive Interference Cancellation
BCH	Bose Ray-Chaudhuri and Hocquenghem
CS	Compressed Sensing
CCS	Coded Compressed Sensing
SPARC	SParse Rgression Code
AMP	Approximate Message Passing
SC-LDPC	Spatially Coupled-LDPC
BP	Belief Propagation
MIMO	Multi-Input Multi-Output
ESE	Elementary Symbol Estimator
SNR	Signal-to-Noise Ratio
OMP	Orthogonal Match Pursuit
AWGN	Additive White Gaussian Noise
LLR	Log-Likelihood Ratio
MUD	Multi-User Detection
EXIT	EXtrinsic Information Transfer
FLOP _f	FLoating-point Operations Per frame

Appendix A. The Expressions of FLOP_f

The expressions of computational complexity and their corresponding coding parameters are in Table A1. The detailed optimized parameters of the compared schemes can be found in their respective papers.

Table A1. FLOPf expressions of different coding schemes.

Coding Scheme	Related Parameters	FLOPf
Our scheme	N_s CS pilot length; B_s pilot info length; T_{th} threshold; N_c length of IDMA codeword; Z_c LDPC lifting size; I_1 maximum iteration of LDPC SPA decoder; n_v number of variable nodes; n_c number of parity check nodes; V number of slots; I_2 maximum iteration of ESE+SPA IDMA decoder; K_a number of users; V number of slots	$T_{th}(2N_sB_s + B_s^2N_s + N_s^3) + 2VT_{th}[I_2(12N_c + 3Z_c(n_c + n_v)^2I_1)] + K_aN_c$
CCS-AMP [37]	N_s row size of CS (Compressed Sensing) matrix; 2^G column size of CS matrix; K_a number of users; J number of sub-blocks; I_{max} maximum iteration of CS AMP decoder	$4JI_{max}K_aN_s(2^G + 1) + K_a\log_2(K_a)$
Sparse IDMA [13]	N_p row size of CS matrix for pilot coding; 2^G column size of CS matrix for pilot coding; K_a number of users; $RR = \lambda'(1)$ average repetition rate; RL basic LDPC code rate; $N_c = n - N_p$ IDMA code length; $N_c = k - B_p$ IDMA information bit length; I_1 BP user layer iteration; I_2 BP channel layer iteration	$4N_sK_a(2^G + 1) + I_2[I_1K_aN_c^2((RR + 1)/RL - 1)^2 + (K_a * RR * N_c/RL + n)^2] * 3$
Polar-RS [14]	2^{B_s} total number of spreading sequences; N_s length of spreading sequences; N_c length of information bits for polar coding; r length of CRC (Cyclic Redundancy Check) parity bits; B_c polar code length; g length of segment in energy detector; I_{max} maximum iteration of polar list decoder	$2n2^{g+B_s}B_c/g + 2N_s2^{B_s} + K_a(N_s2^{B_s} + 2^{(3B_s)} + 2N_s2^{B_s} + I_{max}B_c\log_2(B_c) + B_cr)$

References

1. You, X.; Wang, C.X.; Huang, J.; Gao, X.; Zhang, Z.; Wang, M.; Huang, Y.; Zhang, C.; Jiang, Y.; Wang, J.; et al. Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts. *Sci. China Inf. Sci.* **2021**, *64*, 110301. [\[CrossRef\]](#)
2. Giordani, M.; Polese, M.; Mezzavilla, M.; Rangan, S.; Zorzi, M. Toward 6G Networks: Use Cases and Technologies. *IEEE Commun. Mag.* **2020**, *58*, 55–61. [\[CrossRef\]](#)
3. Chen, S.; Hu, J.; Shi, Y.; Zhao, L.; Li, W. A Vision of C-V2X: Technologies, Field Testing, and Challenges With Chinese Development. *IEEE Internet Things J.* **2020**, *7*, 3872–3881. [\[CrossRef\]](#)
4. Sarkar, S.; Debnath, A. Green IoT: Design Goals, Challenges and Energy Solutions. In Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 8–10 July 2021; pp. 637–642. [\[CrossRef\]](#)
5. Polyanskiy, Y. A perspective on massive random-access. In Proceedings of the IEEE International Symposium on Information Theory, Aachen, Germany, 25–30 June 2017; pp. 2523–2527. [\[CrossRef\]](#)
6. Li, H.; Ru, G.; Kim, Y.; Liu, H. OFDMA capacity analysis in MIMO channels. *IEEE Trans. Inf. Theory* **2010**, *56*, 4438–4446. [\[CrossRef\]](#)
7. Razavi, R.; Dianati, M.; Imran, M.A. Non-Orthogonal Multiple Access (NOMA) for future radio access. In *5G Mobile Communications*; Springer: Cham, Switzerland, 2016; pp. 135–163. [\[CrossRef\]](#)
8. Yuan, Y.; Wang, S.; Wu, Y.; Poor, H.V.; Ding, Z.; You, X.; Hanzo, L. NOMA for Next-Generation Massive IoT: Performance Potential and Technology Directions. *IEEE Commun. Mag.* **2021**, *59*, 115–121. [\[CrossRef\]](#)
9. Wu, Y.; Gao, X.; Zhou, S.; Yang, W.; Polyanskiy, Y.; Caire, G. Massive Access for Future Wireless Communication Systems. *IEEE Wirel. Commun.* **2020**, *27*, 148–156.
10. Li, Y.; Dai, J.; Si, Z.; Niu, K.; Dong, C.; Lin, J.; Wang, S.; Yuan, Y. Unsourced multiple access for 6G massive machine type communications. *China Commun.* **2022**, *19*, 70–87. [\[CrossRef\]](#)
11. Polonelli, T.; Brunelli, D.; Marzocchi, A.; Benini, L. Slotted aloha on lorawan-design, analysis, and deployment. *Sensors* **2019**, *19*, 838. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Liva, G. A slotted ALOHA scheme based on bipartite graph optimization. In Proceedings of the 2010 International ITG Conference on Source and Channel Coding, SCC 2010, Siegen, Germany, 18–21 January 2010; pp. 1–6.

13. Pradhan, A.K.; Amalladinne, V.K.; Vem, A.; Narayanan, K.R.; Chamberland, J. A Joint Graph Based Coding Scheme for the Unsourced Random Access Gaussian Channel. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019.
14. Pradhan, A.K.; Amalladinne, V.K.; Narayanan, K.R.; Chamberland, J. Polar Coding and Random Spreading for Unsourced Multiple Access. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Virtual, 7–11 June 2020; pp. 1–6.
15. Zheng, M.; Wu, Y.; Zhang, W. Polar Coding and Sparse Spreading for Massive Unsourced Random Access. In Proceedings of the IEEE Vehicular Technology Conference, Virtual, 18 November–16 December 2020; pp. 1–4.
16. Ordentlich, O.; Polyanskiy, Y. Low complexity schemes for the random access Gaussian channel. In Proceedings of the IEEE International Symposium on Information Theory, Aachen, Germany, 25–30 June 2017; pp. 2528–2532. [\[CrossRef\]](#)
17. Amalladinne, V.K.; Member, G.S.; Chamberland, J.F.; Member, S.; Narayanan, K.R. A Coded Compressed Sensing Scheme for Unsourced Multiple Access. *IEEE Trans. Inf. Theory* **2020**, *66*, 6509–6533. [\[CrossRef\]](#)
18. Fengler, A.; Jung, P.; Caire, G. SPARCs and AMP for Unsourced Random Access. In Proceedings of the IEEE International Symposium on Information Theory, Paris, France, 7–12 July 2019; pp. 2843–2847.
19. Paolini, E.; Stefanović, Č.; Liva, G.; Popovski, P. Coded random access: Applying codes on graphs to design random access protocols. *IEEE Commun. Mag.* **2015**, *53*, 144–150. [\[CrossRef\]](#)
20. Vem, A.; Narayanan, K.R.; Chamberland, J.F.; Cheng, J. A User-Independent Successive Interference Cancellation Based Coding Scheme for the Unsourced Random Access Gaussian Channel. *IEEE Trans. Commun.* **2019**, *67*, 8258–8272. [\[CrossRef\]](#)
21. Marshakov, E.; Balitskiy, G.; Andreev, K.; Frolov, A. A polar code based unsourced random access for the gaussian MAC. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5. [\[CrossRef\]](#)
22. Glebov, A.; Matveev, N.; Andreev, K.; Frolov, A.; Turlikov, A. Achievability Bounds for T-Fold Irregular Repetition Slotted ALOHA Scheme in the Gaussian MAC. In Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC, Marrakesh, Morocco, 15–18 April 2019; pp. 12–17.
23. Polyanskiy, Y.; Poor, H.V.; Verdú, S. Channel coding rate in the finite blocklength regime. *IEEE Trans. Inf. Theory* **2010**, *56*, 2307–2359. [\[CrossRef\]](#)
24. Kowshik, S.S.; Polyanskiy, Y. Fundamental Limits of Many-User MAC With Finite Payloads and Fading. *IEEE Trans. Inf. Theory* **2021**, *67*, 5853–5884. [\[CrossRef\]](#)
25. Kowshik, S.S.; Andreev, K.; Frolov, A.; Polyanskiy, Y. Energy efficient random access for the quasi-static fading MAC. In Proceedings of the IEEE International Symposium on Information Theory, Paris, France, 7–12 July 2019; pp. 2768–2772. [\[CrossRef\]](#)
26. Li, T.; Wu, Y.; Zheng, M.; Wang, D.; Zhang, W. SPARC-LDPC Coding for MIMO Massive Unsourced Random Access. In Proceedings of the IEEE Globecom Workshops, Taipei, Taiwan, 7–11 December 2020; pp. 8–13.
27. Wang, Q.; Liu, L.; Zhang, S.; Lau, F.C. On Massive IoT Connectivity with Temporally-Correlated User Activity. In Proceedings of the IEEE International Symposium on Information Theory, Melbourne, Australia, 12–20 July 2021; pp. 3020–3025.
28. Fengler, A.; Haghighatshoar, S.; Jung, P.; Caire, G. Non-Bayesian Activity Detection, Large-Scale Fading Coefficient Estimation, and Unsourced Random Access with a Massive MIMO Receiver. *IEEE Trans. Inf. Theory* **2021**, *67*, 2925–2951.
29. Koep, N.; Behboodi, A.; Mathar, R. An Introduction to Compressed Sensing. In *Applied and Numerical Harmonic Analysis*; Birkhäuser: Cham, Switzerland, 2019. [\[CrossRef\]](#)
30. Xiong, X.; Hu, J.; Tian, L. A fast converging multi-user detection for IDMA based on time-reversal. In Proceedings of the 2007 6th International Conference on Information, Communications and Signal Processing, ICICS, Singapore, 10–13 December 2007. [\[CrossRef\]](#)
31. Leinonen, M.; Codreanu, M.; Giannakis, G.B. *Compressed Sensing with Applications in Wireless Networks*; Now Foundations and Trends: Boston, MA, USA, 2019. [\[CrossRef\]](#)
32. Richardson, T.; Urbanke, R. *Modern Coding Theory*; Cambridge University Press: Cambridge, UK, 2008. [\[CrossRef\]](#)
33. Hagenauer, J. The exit chart—Introduction to extrinsic information transfer in iterative processing. In Proceedings of the 2004 12th European Signal Processing Conference, Vienna, Austria, 6–10 September 2004; pp. 1541–1548.
34. Price, K.V. Differential evolution. In *Handbook of Optimization*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 187–214.
35. 3GPP. TS 138 212-V15. 2.0-5G; NR; Multiplexing and Channel Coding (3GPP TS 38.212 Version 15.2. 0 Release 15). Retrieved 26 June 2022. Available online: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.02.00_60/ts_138212v150200p.pdf (accessed on 29 September 2022).
36. Amalladinne, V.K.; Kumar Pradhan, A.; Rush, C.; Chamberland, J.F.; Narayanan, K.R. On Approximate Message Passing for Unsourced Access with Coded Compressed Sensing. In Proceedings of the 2020 IEEE International Symposium on Information Theory (ISIT), Los Angeles, CA, USA, 21–26 June 2020; pp. 2995–3000. [\[CrossRef\]](#)
37. Calderbank, R.; Thompson, A. CHIRUP: A practical algorithm for unsourced multiple access. *Inf. Inference* **2020**, *9*, 875–897. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.