

Decodificação Iterativa em Árvore de Baixa Complexidade para SCMA

Ana Luiza Scharf, Bartolomeu F. Uchôa-Filho, Bruno Fontana da Silva, Didier Le Ruyet

Resumo—Múltiplo Acesso por Códigos Esparsos (SCMA) é um poderoso candidato para as futuras gerações de comunicações sem fio. Diversos algoritmos de decodificação foram propostos para o SCMA. O conhecido *Message-Passing Algorithm* (MPA) apresenta um bom desempenho, mas com complexidade exponencial. Propomos uma modificação no Log-MPA convencional para reduzir a complexidade da decodificação. Uma estrutura de árvore é associada à equação de decodificação e, evitando-se adequadamente símbolos pouco confiáveis, é obtida uma árvore podada, levando a uma menor complexidade, com um ajuste arbitrário sobre o compromisso entre desempenho e complexidade da detecção conjunta.

Palavras-Chave—SCMA, Log-MPA, Sistema 5G.

Abstract—Sparse code multiple access (SCMA) is a powerful candidate for the future generations of wireless communications. Several decoding algorithms have been proposed for SCMA. The well-known *Message-Passing Algorithm* (MPA) performs well but with exponential complexity. We propose a modification in the conventional Log-MPA to reduce the decoding complexity. A tree structure is associated with the decoding equation and, by properly avoiding symbols with low reliability, a pruned tree is obtained, yielding an arbitrary tradeoff between performance and complexity of the joint detection.

Keywords—SCMA, Log-MPA, 5G system.

I. INTRODUÇÃO

Múltiplo Acesso por Códigos Esparsos (SCMA, do inglês “*Sparse Code Multiple Access*”) é uma técnica de múltiplo acesso não ortogonal proposta para as futuras gerações de sistemas de comunicação sem fio, que proporciona uma alta eficiência e um bom desempenho [1]. Em SCMA, os dados dos usuários são mapeados em palavras-código, que são alocadas de maneira não ortogonal em recursos tais como as subportadoras de uma Multiplexação Ortogonal por Divisão em Frequência (OFDM) [2].

O SCMA é uma generalização do Espalhamento de Baixa de Densidade (LDS, do inglês “*Low Density Spreading*”) [3], que por sua vez é uma versão esparsa do Múltiplo Acesso por Divisão de Tempo (CDMA, do inglês “*Code Division Multiple Access*”) com sequências de espalhamento de baixa densidade [4]. Na técnica LDS, os bits de informação de cada usuário são mapeados em um símbolo complexo de

A. L. Scharf e B. F. Uchôa-Filho fazem parte do GPqCom/LCS/EEL, Universidade Federal de Santa Catarina, Florianópolis 88040-900, Brazil (e-mails: ana.scharf@posgrad.ufsc.br, uchoa@eel.ufsc.br).

B. F. da Silva faz parte do Instituto Federal Sul-rio-grandense (IFSul), Saporanga, Brasil (e-mail: brunosilva@ifsul.edu.br).

D. Le Ruyet faz parte do CEDRIC, Conservatoire National des Arts et Metiers, 75003 Paris, França (e-mail: didier.le_ruyet@cnam.fr).

Este trabalho foi parcialmente financiado pelo CNPq (148251/2018-8, 306780/2017-8) e apoiado pela CAPES através do Programa PrInt, #698503P.

uma constelação de sinais, e este símbolo é repetido em um pequeno número de recursos ou *slots* (subportadoras, no caso do OFDM), enquanto os demais recursos não são utilizados (a potência alocada nesses recursos é nula). A escolha dos recursos a serem utilizados por cada usuário é feita de modo que cada recurso seja utilizado por um pequeno número de usuários. O baixo número de colisões por recurso possibilita uma representação do LDS por um grafo esparso, com base no qual o *Message-Passing Algorithm* (MPA) [3] pode ser utilizado para recuperar os símbolos transmitidos de todos os usuários.

Na técnica SCMA, os bits de entrada de cada usuário são mapeados em palavras-código (*codewords*) complexas pertencentes a um *codebook* fixo [4]. Tal qual no LDS, em SCMA os símbolos complexos (coordenadas) da palavra-código são espalhados em diferentes recursos, a única diferença sendo que, no SCMA, os símbolos não são meras repetições. Graças a esta característica, SCMA proporciona um melhor desempenho [1]. Também, ao contrário de CDMA, LDS e SCMA atendem à demanda de conectividade maciça dos futuros sistemas de comunicação sem fio, ou seja, permitem um número maior de usuários.

Um dos desafios do SCMA é estabelecer uma boa relação entre desempenho e complexidade de detecção [5]. Em geral, quanto melhor o desempenho, mais complexo se torna o decodificador. Há diversos algoritmos decodificadores, que tentam balancear essa relação. O decodificador de Máxima Verossimilhança (ML, do inglês “*Maximum Likelihood*”) possui um excelente desempenho, contudo, a sua complexidade de detecção é alta. Um decodificador amplamente utilizado é o MPA [3], que perde um pouco de desempenho, mas apresenta uma menor complexidade de detecção em relação ao ML, porém, ainda bastante complexa (exponencial).

No algoritmo MPA convencional, proposto em [3], as equações de decodificação envolvem funções exponenciais, resultando em um grande armazenamento [6]. O problema é reduzido com a utilização do algoritmo Log-MPA [7], cujos cálculos são realizados no domínio logarítmico, à semelhança do que ocorre em códigos turbo [8].

A proposta desse artigo é reduzir a complexidade de detecção do Log-MPA. A ideia é introduzir uma estrutura de árvore para o cálculo deste algoritmo e, ao realizar operações por partes, podem-se remover algumas possibilidades consideradas pouco relevantes segundo um critério aqui proposto. Isso resulta numa redução da complexidade de detecção, com uma perda de desempenho controlável.

Há na literatura trabalhos que propõem modificações do Log-MPA ou MPA. Alguns deles consideram algum código

corretor de erros concatenado ao sistema SCMA, sem o qual a proposta de redução de complexidade não funciona, ou seja, não apresenta desempenho satisfatório. Neste artigo, seguimos a linha dos que não consideram códigos corretores de erros. Uma investigação mais aprofundada do algoritmo proposto, incluindo tais códigos, é deixada para trabalhos futuros. Destaca-se, entretanto, que o desempenho obtido com o algoritmo proposto neste artigo é satisfatório apesar de o sistema ser não codificado.

É também importante mencionar que a maioria dos trabalhos com propostas de redução de complexidade de decodificação do SCMA não fornece informações suficientes para a reprodutibilidade de seus resultados. Por exemplo, se o sistema adota códigos turbo, nada além da taxa do código é especificado e, algumas vezes, nem esta informação é fornecida. Em outros casos, apenas o tamanho do *codebook* é informado. Por esta razão, neste artigo o algoritmo proposto é comparado diretamente com o Log-MPA original.

Este trabalho é organizado da seguinte maneira. Na Seção II, o sistema SCMA é descrito tal qual proposto na literatura. Na Seção III, o detector Log-MPA convencional para SCMA é apresentado resumidamente. O decodificador proposto é apresentado na Seção IV. Na Seção V, resultados de simulação são apresentados.

Notação: Negrilo minúsculo e maiúsculo indicam, respectivamente, vetores e matrizes. $\mathbf{X}_{i,j}$ representa o elemento na i -ésima linha e j -ésima coluna da matriz \mathbf{X} . Já \mathbf{X}_k denota que a matriz está associada ao k -ésimo recurso; o sobrescrito T denota matriz transposta. Já a norma Euclidiana de uma matriz ou um vetor é representada por $\|\cdot\|$; variável maiúscula e caligráfica representa um conjunto. Por fim, $\mathbf{x} \sim \mathcal{CN}(\eta, \mathbf{Z})$ é um vetor aleatório complexo com distribuição Gaussiana com média η e matriz de covariância \mathbf{Z} ; \mathbf{I}_K denota uma matriz de identidade $K \times K$.

II. SISTEMA DE TRANSMISSÃO DO SCMA

O sistema SCMA é composto por J usuários independentes, os quais são multiplexados em K recursos ortogonais. Nesse sistema a quantidade de usuários é maior que o número de recursos, portanto, o fator de carga J/K sempre será maior que 1 ($J > K$), configurando um sistema em sobrecarga.

Na transmissão, o índice de informação i do j -ésimo usuário, com $i \in \{1, \dots, M_j\}$, é mapeado na palavra-código $\mathbf{c}_{i,j} \in \mathcal{C}_j \subset \mathbb{C}^N$, em que \mathcal{C}_j é um *codebook* complexo N -dimensional de tamanho M_j .

Os N símbolos complexos da palavra-código são espalhados em K recursos, $N < K$, conforme uma matriz de alocação de recursos, \mathbf{F} , de dimensão $K \times J$, que especifica quais são os recursos usados por cada usuário. Se o elemento $F_{k,j}$ da matriz é igual a 1, então o usuário j faz uso do recurso k e, portanto, um dos símbolos da palavra-código $\mathbf{c}_{i,j}$ é alocado neste recurso. Como exemplo, considere a matriz:

$$\mathbf{F}_{4,6} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad (1)$$

com $K = 4$ (recursos) e $J = 6$ (usuários). Uma característica da matriz de alocação de recursos é a esparsidade, isto é, ela deve possuir poucos elementos não nulos. Defina os dois conjuntos: $\varphi_j = \{k : F_{k,j} \neq 0\}$, que especifica os N recursos utilizados pelo j -ésimo usuário, e o conjunto, $\phi_k = \{j : F_{k,j} \neq 0\}$, que informa quais usuários alocam o k -ésimo recurso. A cardinalidade de φ_j é N , e a de ϕ_k será denotada por d_F . Na matriz de alocação de recursos em (1), temos $d_F = 3$ e $N = 2$.

Com a aplicação de \mathbf{F} , o *codebook* \mathcal{C}_j é redimensionado para uma versão esparsa, $\mathcal{X}_j \subset \mathbb{C}^K$, de acordo com a assinatura \mathbf{f}_j de cada usuário. Na palavra-código redimensionada, $\mathbf{x}_{i,j} \in \mathcal{X}_j$, $K - N$ componentes serão o símbolo zero, e as demais receberão os valores originais dos N símbolos de $\mathbf{c}_{i,j}$.

Assim, o sinal recebido é dado por:

$$\mathbf{y} = \sum_{j=1}^J \text{diag}(\mathbf{h}_j) \mathbf{x}_{i,j} + \mathbf{z}, \quad (2)$$

em que $\mathbf{h}_j = [h_j[1], h_j[k], \dots, h_j[K]]^T$ é o vetor que representa os ganhos de canal do usuário j nos diferentes recursos k , $k = 1, \dots, K$, $\mathbf{x}_{i,j} = [x_{i,j}[1], x_{i,j}[2], \dots, x_{i,j}[K]]^T$ representa a palavra-código associada ao índice de informação i do usuário j , $\mathbf{y} = [y[1], y[k], \dots, y[K]]^T$ denota o vetor dos sinais recebidos nos K recursos, e $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_K)$ indica o ruído complexo aditivo Gaussiano branco.

III. DETECTOR LOG-MPA CONVENCIONAL

A característica de esparsidade do SCMA permite a aplicação do algoritmo de decodificação iterativa Log-MPA [7]. Para isso, constrói-se um grafo bipartido cujos nós representam, de um lado, os nós de usuário (UN, do inglês “*user nodes*”), e do outro, os nós de recurso (RN, do inglês “*resource nodes*”) [9]. A alocação das arestas entre os UNs e RNs é realizada com base nos conjuntos φ_j e ϕ_k , previamente definidos.

A decodificação do Log-MPA para sistema SCMA é realizada pela troca de mensagens entre UNs e RNs. A mensagem do RN k para o UN j é descrita como [7]:

$$M_{k \rightarrow j}^{[t, \text{ext}]} [i_j] = \beta + \widetilde{M}_{k \rightarrow j}^{[t, \text{ext}]} [i_j], \quad (3)$$

em que

$$\beta = -\log \left(\sum_{i=1}^{M_j} \exp \left(\widetilde{M}_{k \rightarrow j}^{[t, \text{ext}]} [i] \right) \right) \quad (4)$$

é um fator de normalização das mensagens do RN para o UN, e $\widetilde{M}_{k \rightarrow j}^{[t, \text{ext}]} [i]$ é definido como

$$\begin{aligned} \widetilde{M}_{k \rightarrow j}^{[t, \text{ext}]} [i] = & \max_{\substack{i_l \in \{1, \dots, M_l\} \\ l \in \phi_k \setminus \{j\}}}^* \left(-\frac{1}{2\sigma_z^2} \|y_k - h_j[k] x_{i,j}[k] \right. \\ & \left. - \sum_{l \in \phi_k \setminus \{j\}} h_l[k] x_{i,l}[k] \|^2 + \sum_{l \in \phi_k \setminus \{j\}} M_{l \rightarrow k}^{[t-1, \text{int}]} [i_l] \right). \quad (5) \end{aligned}$$

A função \max^* é o logaritmo jacobiano, cujo uso no algoritmo Log-MPA foi considerado em [8]. Para defini-la,

considere inicialmente dois valores, a e b . Neste caso, a função \max^* retorna:

$$\max^* = \max(a, b) + \log[1 + \exp(-\|a - b\|)]. \quad (6)$$

Note que o resultado não será o valor máximo, uma vez que a parte do log adiciona um ajuste ao valor final. A extensão de \max^* para mais de dois valores, como aparece em (5), é possível considerando-se dois valores por vez, fazendo-se uso da maximização em (6) a cada vez.

Com base em (3) e (5), e considerando $M_j = M$ para todo $j \in \{1, \dots, J\}$, a complexidade do Log-MPA convencional é da ordem de $\mathcal{O}\{M^{d_F}\}$. É importante ressaltar que, para a maximização em (5), o algoritmo precisa considerar todas as palavras-código de todos os usuários que fazem uso do recurso k exceto o usuário j . Para maiores detalhes sobre o algoritmo Log-MPA, vide [7].

IV. DECODIFICAÇÃO LOG-MPA COM ESTRUTURA DE ÁRVORE

Nesta seção, apresentamos uma proposta de redução de complexidade para o algoritmo Log-MPA. A ideia central é considerar uma estrutura de árvore na qual cada nível está associado a um usuário que faz uso do recurso k , e cada ramo deste nível está associado a uma palavra-código deste usuário. A árvore possui d_F níveis.

Para a obtenção da mensagem $\widetilde{M}_{k \rightarrow j}^{[t, \text{ext}]}[i]$ em (5), temos que o nível 0 da árvore está associado ao usuário j . Os $d_F - 1$ níveis inferiores estão associados aos usuários no conjunto $\phi_k \setminus \{j\}$. No algoritmo proposto, com exceção do nível zero, apenas r ramos ($r < M_n$) do nível n , descendentes de um nó do nível $n - 1$, são considerados. Os demais ramos serão podados. Um exemplo de árvore para $d_F = 3$ e $r = 2$ é mostrado na Figura 1, na qual o símbolo \times representa um ramo podado.

O cálculo de $\widetilde{M}_{k \rightarrow j}^{[t, \text{ext}]}[i]$ é quebrado em etapas, executadas à medida que se percorre a árvore de cima para baixo, mas considerando apenas a subárvore que vai permanecendo após as podas. No nível $n = 0$, temos:

$$\widetilde{M}_{k \rightarrow j}^{[t, \text{ext}, 0]}[i] = \left(-\frac{1}{2\sigma_z^2} \|y_k - h_j[k] x_{i,j}[k]\|^2 \right). \quad (7)$$

Antes de avançarmos para o próximo nível, algumas definições se fazem necessárias. Nas derivações que seguem a partir deste ponto, os usuários que utilizam o k -ésimo recurso (exceto o usuário j) são ordenados em ordem decrescente, de acordo com o valor absoluto do ganho de canal. O conjunto dos índices dos usuários ordenados é denotado por: $\gamma_{\setminus j} = \{\gamma_1, \gamma_2, \dots, \gamma_{d_F-1}\}$, conforme:

$$|h_{\gamma_1}| > |h_{\gamma_2}| > \dots > |h_{\gamma_{d_F-1}}|. \quad (8)$$

Assim, no nível $n = 1$, temos:

$$\begin{aligned} \widetilde{M}_{k \rightarrow j}^{[t, \text{ext}, 1]}[i] = \max_{i_{\gamma_1} \in \{1, \dots, M_{\gamma_1}\}} \left(-\frac{1}{2\sigma_z^2} \|y_k - h_j[k] x_{i,j}[k] \right. \\ \left. - h_{\gamma_1}[k] x_{i_{\gamma_1}, \gamma_1}[k]\|^2 + M_{\gamma_1 \rightarrow k}^{[t-1, \text{int}]}[i_{\gamma_1}] \right), \quad (9) \end{aligned}$$

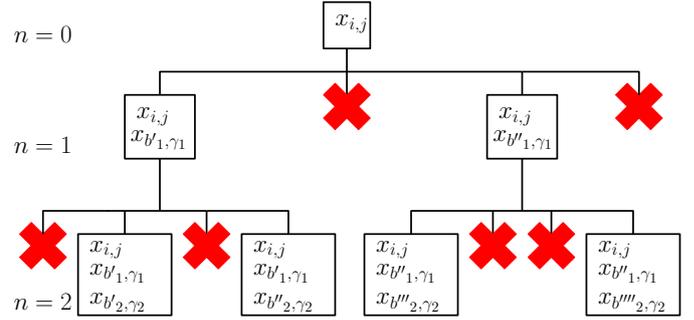


Fig. 1. Possível estrutura de árvore do Log-MPA para $d_F = 3$ e $r = 2$.

em que $\max\text{-}r\{\cdot\}$ generaliza $\max\{\cdot\}$ no sentido de selecionar os r maiores valores que a expressão pode assumir. Os elementos do conjunto $\widetilde{\mathcal{M}}_{k \rightarrow j}^{[t, \text{ext}, 1]}[i]$ em (9) são:

$$\left\{ \widetilde{M}_{k \rightarrow j}^{[t, \text{ext}, 1], [1]}[i], \dots, \widetilde{M}_{k \rightarrow j}^{[t, \text{ext}, 1], [r]}[i] \right\}. \quad (10)$$

Os r índices que maximizam (9) são:

$$\mathcal{B} = \arg \max_{i_{\gamma_1} \in \{1, \dots, M_{\gamma_1}\}} \text{-}r \widetilde{\mathcal{M}}_{k \rightarrow j}^{[t, \text{ext}, 1]}[i]. \quad (11)$$

Generalizando, no nível n , para $n = 1, \dots, d_F - 1$, e para $b_l \in \mathcal{B}_{b_1, \dots, b_{l-1}}$, com $l = 1, \dots, n - 1$, temos

$$\begin{aligned} \widetilde{\mathcal{M}}_{k \rightarrow j}^{[t, \text{ext}, n], [b_1, \dots, b_{n-1}]}[i] = \\ \max_{i_{\gamma_n} \in \{1, \dots, M_{\gamma_n}\}} \text{-}r \left(-\frac{1}{2\sigma_z^2} \|y_k - h_j[k] x_{i,j}[k] \right. \\ \left. - h_{\gamma_n}[k] x_{i_{\gamma_n}, \gamma_n}[k] - \sum_{l=1}^{n-1} h_{\gamma_l}[k] x_{b_l, \gamma_l}[k]\|^2 \right. \\ \left. + M_{\gamma_n \rightarrow k}^{[t-1, \text{int}]}[i_{\gamma_n}] + \sum_{l=1}^{n-1} M_{\gamma_l \rightarrow k}^{[t-1, \text{int}]}[b_l] \right), \quad (12) \end{aligned}$$

em que

$$\mathcal{B}_{b_1, \dots, b_{n-1}} = \arg \max_{i_{\gamma_n} \in \{1, \dots, M_{\gamma_n}\}} \text{-}r \widetilde{\mathcal{M}}_{k \rightarrow j}^{[t, \text{ext}, n], [b_1, \dots, b_{n-1}]}[i]. \quad (13)$$

Finalmente, realiza-se a operação \max^* , agora considerando-se apenas os nós sobreviventes no último nível da árvore. Assim, no algoritmo proposto, a nova mensagem $\widetilde{M}_{k \rightarrow j}^{[t, \text{ext}]}[i]$ é obtida por:

$$\widetilde{M}_{k \rightarrow j}^{[t, \text{ext}]}[i] = \max_{\substack{b_l \in \mathcal{B}_{b_1, \dots, b_{l-1}} \\ l=1, \dots, d_F-1}} \text{-}r \widetilde{\mathcal{M}}_{k \rightarrow j}^{[t, \text{ext}, d_F-1], [b_1, \dots, b_{d_F-2}]}[i]. \quad (14)$$

Em seguida, prossegue-se com os demais passos do algoritmo Log-MPA convencional. Vale ressaltar que o fator de normalização em (4) será aplicado apenas aos nós sobreviventes no último nível, pois, são estes que carregam informação relevante.

Para o decodificador proposto neste trabalho, e considerando $M_j = M$ para todo $j \in \{1, \dots, J\}$, a ordem de complexidade é

$$\mathcal{O}_{\text{prop}} = M + M^2 \sum_{n=0}^{d_F-2} r^n \quad (15)$$

ou, em forma fechada,

$$\mathcal{O}_{\text{prop}} = \begin{cases} M \left(1 + \frac{M(r^{d_F-1} - 1)}{r-1} \right), & \text{se } r = 2, \dots, M-1 \\ M(1 + M(d_F - 1)), & \text{se } r = 1 \end{cases} \quad (16)$$

Uma simples análise da equação (16) mostra que a complexidade de decodificação do algoritmo proposto cresce com M desde M^2 (se r for constante) a até M^{d_F} (se $r = \alpha M$, para $0 < \alpha < 1$). Ainda assim, neste último caso, há uma redução significativa da complexidade em relação ao Log-MPA. A ordem de complexidade do algoritmo proposto é comparada à do Log-MPA convencional na Tabela I para diversos cenários.

TABELA I
COMPARAÇÃO DAS ORDENS DE COMPLEXIDADE PARA DIFERENTES VALORES DE r

M	d_F	r	\mathcal{O} prop.	\mathcal{O} conv.	% (prop./conv)
4	3	1	36	64	56,25
		2	52	64	81,250
4	4	1	52	256	20,31
		2	116	256	45,31
8	3	1	136	512	26,56
		2	200	512	39,06
		3	264	512	51,56
		4	328	512	64,06
		5	392	512	76,56
		6	456	512	89,06
8	4	1	200	4096	4,88
		2	456	4096	11,13
		3	840	4096	20,50
		4	1352	4096	33,00
		5	1992	4096	48,63
		6	2760	4096	67,38

V. RESULTADOS DE SIMULAÇÃO

Nesta seção, resultados de simulação Monte Carlo para a taxa de erro de bit (BER), em função da relação energia média de bit por densidade espectral potência do ruído (E_b/N_0), são apresentados. Para $E_b/N_0 < 10$ dB, o critério de parada foi de 500 erros de símbolos, e para valores maiores de E_b/N_0 , consideramos 250 erros.

Nos dois cenários simulados, os parâmetros $N = 2$, $K = 4$, $J = 6$ e $d_F = 3$, com a matriz de alocação de recursos $F_{4,6}$ dada em (1), foram considerados. Em cada cenário, o mesmo *codebook* é adotado para todos os usuários. No modelo de canal da equação (2), o desvanecimento Rayleigh foi adotado.

Em todas as simulações, os decodificadores Log-MPA [7] e o proposto realizaram cinco iterações. Para o decodificador proposto, mostramos resultados para todos os valores possíveis da variável r . O caso $r = M$ corresponde ao próprio Log-MPA convencional, adotado como referência.

No cenário de simulação da Figura 2, adotou-se para todos os usuários o *codebook* para SCMA projetado em [10], que possui $M = 4$ palavras-código. Como se pode observar, à medida que o valor de r aumenta, o desempenho do Log-MPA com estrutura de árvore proposto se aproxima do desempenho do Log-MPA convencional. O decodificador apresenta desempenho muito semelhante para $r = 2$.

Com o propósito de comprovar, no algoritmo proposto, a importância do ordenamento dos usuários em $\phi_k \setminus \{j\}$ segundo

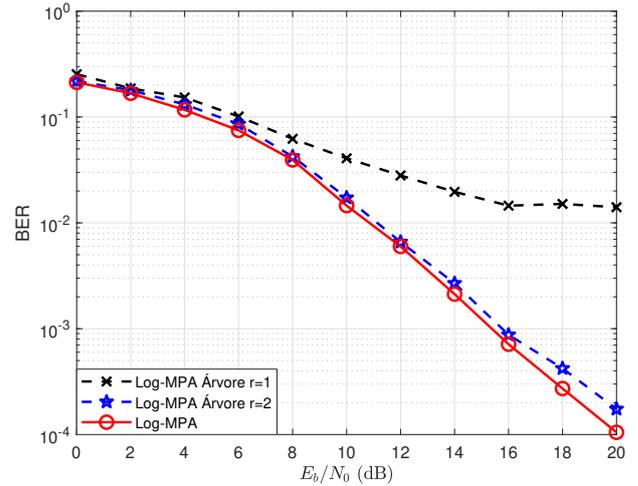


Fig. 2. BER do *codebook* para SCMA projetado em [10].

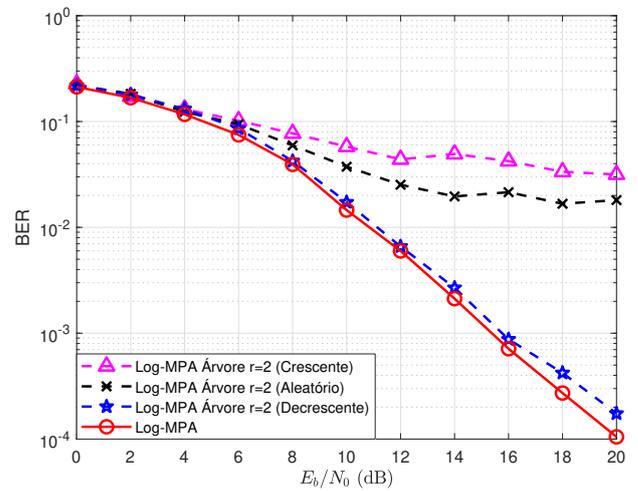


Fig. 3. BER do *codebook* para SCMA projetado em [10], em que γ_j é ordenado de forma: crescente, aleatória e decrescente.

a ordem decrescente dos ganhos de canal, apresentada em (8), simulamos o mesmo cenário da Figura 2, porém considerando os ordenamentos aleatório, crescente e decrescente (proposto). O resultado é mostrado na Figura 3. Observa-se que a ordem decrescente é a que produz o melhor desempenho, o que era esperado visto que, a cada nível da árvore, a contribuição da interferência associada ao canal com maior relevância na composição do sinal recebido é removida primeiro.

No cenário de simulação da Figura 4, adotou-se para todos os usuários o *codebook* para LDS com modulação digital 8-QAM, ou seja, $M = 8$ palavras-código. Para este caso, os valores de $r = 1, \dots, 6$ foram considerados. O resultado de desempenho é consistente com o da Figura 2, como se previa. Ou seja, o desempenho do algoritmo proposto se aproxima do desempenho do Log-MPA convencional à medida que r cresce.

É importante comentar que, nos resultados de simulação acima, o desempenho apresentado não parece ser muito satisfatório quando valores de r pequenos são escolhidos. En-

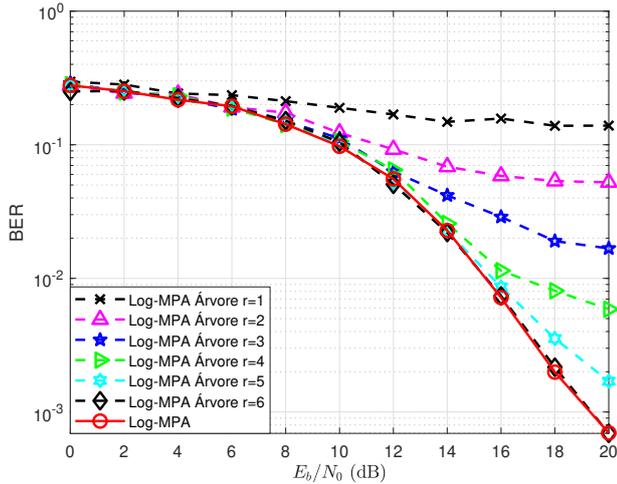


Fig. 4. BER do codebook LDS-QAM ($M = 8$).

tretanto, cabe mencionar que, na literatura, o sistema SCMA é normalmente combinado com códigos corretores de erros, tipicamente códigos LDPC. E neste caso, um excelente desempenho é obtido quando o desempenho do SCMA não codificado é bom o suficiente para garantir a convergência do decodificador do LDPC. Apesar de não termos realizado simulações do esquema proposto combinado com códigos corretores de erros — que fica como trabalhos futuros —, espera-se que um resultado satisfatório possa ser obtido no caso codificado utilizando-se valores pequenos de r . Desse modo, acredita-se que uma grande redução de complexidade possa ser obtida com uma pequena perda de desempenho.

VI. CONCLUSÕES

Neste trabalho, uma modificação do decodificador Log-MPA para SCMA, que apresenta bom desempenho, com uma redução significativa de complexidade de detecção, foi proposta. O detector é uma versão do Log-MPA que se baseia numa estrutura de árvore que, quando podada de maneira adequada, proporciona uma redução no número de palavras-código processadas pelo detector iterativo. Resultados de simulação para a taxa de erro de bit foram apresentados. Observou-se que o detector proposto possibilita um compromisso arbitrário entre desempenho e complexidade de detecção. Sendo assim, a contribuição deste trabalho é de grande interesse em cenários práticos.

AGRADECIMENTOS

Este trabalho foi parcialmente financiado pelo CNPq (148251/2018-8, 306780/2017-8) e apoiado pela CAPES através do Programa PrInt, #698503P.

REFERÊNCIAS

- [1] H. Nikopour and H. Baligh. Sparse code multiple access. In *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 332–336, Sep. 2013.
- [2] R. Alizadeh, N. Bélanger, Y. Savaria, and F. Boyer. Performance characterization of an scma decoder. In *2016 14th IEEE International New Circuits and Systems Conference (NEWCAS)*, pages 1–4, June 2016.
- [3] R. Hoshyar, F. P. Wathan, and R. Tafazolli. Novel low-density signature for synchronous cdma systems over awgn channel. *IEEE Transactions on Signal Processing*, 56(4):1616–1626, April 2008.
- [4] M. Taherzadeh, H. Nikopour, A. Bayesteh, and H. Baligh. Scma codebook design. In *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pages 1–5, Sep. 2014.
- [5] L. Li, J. Wen, X. Tang, and C. Tellambura. Modified sphere decoding for sparse code multiple access. *IEEE Communications Letters*, 22(8):1544–1547, Aug 2018.
- [6] Mojtaba Vaezi, Zhiguo Ding, and H Vincent Poor. *Multiple access techniques for 5G wireless networks and beyond*. Springer, 2019.
- [7] L. Lu, Y. Chen, W. Guo, H. Yang, Y. Wu, and S. Xing. Prototype for 5g new air interface technology scma and performance evaluation. *China Communications*, 12(Supplement):38–48, December 2015.
- [8] P. Robertson, E. Vilebrun, and P. Hoeher. A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain. In *Proceedings IEEE International Conference on Communications ICC '95*, volume 2, pages 1009–1013 vol.2, June 1995.
- [9] P. Gao, Y. Du, B. Dong, W. Zhu, Z. Chen, and X. Wang. Low-complexity cs-aided mpa detector for scma systems. *IEEE Communications Letters*, 22(4):784–787, April 2018.
- [10] J. Bao, Z. Ma, M. Xiao, T. A. Tsiftsis, and Z. Zhu. Bit-interleaved coded scma with iterative multiuser detection: Multidimensional constellations design. *IEEE Transactions on Communications*, 66(11):5292–5304, Nov 2018.