Efficient Sphere Decoding of Polar Codes

Jing Guo
University of Cambridge
jg582@cam.ac.uk

Albert Guillén i Fàbregas
ICREA & Universitat Pompeu Fabra
University of Cambridge
guillen@ieee.org

***Abstract*—The performance of the original successive cancellation decoder of short-length polar codes is inferior to that of the maximum-likelihood decoder. Existing sphere decoding algorithms of polar codes have a high computational complexity even for short lengths. This is because, when exploring the tree defined by the generator matrix of the code, existing algorithms employ loose branching conditions and end up visiting many more nodes than needed. We propose improved branching conditions that significantly reduce the search complexity. A simple example reports an improvement of two orders of magnitude at $\frac{E_b}{N_0} = 4$ dB compared to the standard sphere decoders.**

***Index Terms*—Polar codes, Polar codes decoding, Sphere decoding**

I. Introduction

Polar codes were proposed in [1] as a coding technique that provably achieves the capacity of symmetric binary-input discrete memoryless channels (B-DMCs) with low encoding and decoding complexity. The analysis and construction of polar codes are summarized as follows: (1) Given a B-DMC, virtual channels between the bits at the input of a linear encoder and the channel output sequence are created, such that the mutual information in each of these channels converges to either zero or one as the block length tends to infinity; the proportion of virtual channels with mutual information close to one converges to the original channel's capacity; (2) By transmitting bits through the noiseless virtual channels, under successive cancellation decoding, polar codes achieve the capacity.

The performance of polar codes with successive cancellation decoders is inferior to that of the maximum-likelihood (ML) decoder. Decoders with better finite-length performance have been proposed in the literature. Soft-output decoders such as belief propagation decoders of polar codes were proposed in [2]–[4]. In [5], a list successive cancellation decoder was proposed and the performance was comparable to that of low-density parity-check (LDPC) codes. ML decoding of polar codes implemented by means of sphere decoding (SD) [6], [7] was studied in [8], [9]. These decoders have high computational complexity at block length as short as $N = 64$. Later, [10] proposed SD of binary polar codes via a non-binary tree search, which can decode binary polar codes with length up to 256.

This work has been funded in part by the European Research Council under ERC grant agreement 259663 and the Spanish Ministry of Economy and Competitiveness under grant TEC2012-38800-C03-03.

SD decoding was originally proposed to decode multi-dimensional modulations for fading channels based on lattices in the Euclidean space [11]. In essence, SD decoding performs a depth-first tree search, pruning the search tree according to certain branching conditions, e.g. [12]. SD has been extensively studied in the context of multiple-input multiple-output channels (e.g. [13]–[16]).

In this paper, we improve standard SD branching conditions by computing lower bounds on the optimal decoding metric. We first derive fixed lower bounds that only depend on the received signals. Instead of obtaining fixed lower bounds by enlarging the search space to the real field (as in [12]), we propose fixed lower bounds that keep the finite field constraints. We then propose dynamic lower bounds that are updated during the tree search. Dynamic lower bounds depend both on the received signals and the current tree path.

II. Notation and Preliminaries

Let N be the block length. Row vectors are assumed. Let $\boldsymbol{u}_{\mathcal{I}}$ denote the sub-vector of $\boldsymbol{u}$ with indices $i \in \mathcal{I}$. We use $\boldsymbol{u}_a^b$ to denote the sub-vector $(u_a, \ldots, u_b)$.

A. Code Construction

Consider the matrix $\boldsymbol{G}_2 = \left[\begin{smallmatrix}1 & 0\\ 1 & 1\end{smallmatrix}\right]$, and let $\boldsymbol{G}_N = \boldsymbol{G}_2^{\otimes n}$ be the $N \times N$ matrix obtained by the Kronecker product of $\boldsymbol{G}_2$ with itself $n = \log_2 N$ times. Let $g_{i,j}$ denote the (i, j)-th entry of $\boldsymbol{G}_N$. To construct codes of rate $R = \frac{K}{N}$, $N - K$ rows are discarded, equivalently, $N - K$ information bits corresponding to those rows are *frozen* to zero. The set of frozen indices is called the frozen set $\mathcal{F}$; its complement is denoted by $\mathcal{F}^c$. Different methods for choosing $\mathcal{F}$ yield different codes. For the sake of simplicity, we only consider two selection rules in this paper. Reed Muller (RM) codes discard rows with lowest Hamming weights, while polar codes discard rows with lowest mutual information of the induced virtual channels. Information bits are denoted by $\boldsymbol{u}$, with $u_i \in \{0, 1\}$ for $i \in \mathcal{F}^c$ and $u_i = 0$ for $i \in \mathcal{F}$. Codewords $\boldsymbol{x}$ are generated by $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}_N$, where $\boldsymbol{u}\boldsymbol{G}_N$ is carried out in GF(2). Let $\mathcal{U}, \mathcal{C}$ denote the set of all information sequences $\boldsymbol{u}$ and the set of all codewords $\boldsymbol{x}$, respectively. Since some bits in $\boldsymbol{u}$ are frozen to be 0, we have $\mathcal{U}, \mathcal{C} \subset \{0, 1\}^N$.

B. Modulation and Channel Model

Throughout this paper, we assume the channel is a binary-input additive white Gaussian noise (AWGN) channel. The coded bit $x_i \in \{0,1\}$ is mapped into the transmitted signal $s_i \in \{+1,-1\}$ via $s_i = 1-2x_i$ for $i = 1,\ldots,N$. The received signal is $y_i = s_i + z_i$, where z_i is i.i.d., additive white Gaussian noise with zero mean and variance σ^2.

C. Sphere Decoding Algorithms

ML decoding over the binary-input AWGN channel is equivalent to solving the following minimization problem,

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}\in\mathcal{U}} \|\bar{\boldsymbol{y}} - \boldsymbol{u}\boldsymbol{G}_N\|^2 \tag{1}$$

where $\bar{\boldsymbol{y}} \triangleq \frac{\mathbf{1}-\boldsymbol{y}}{2}$ and $\mathbf{1}$ is the all-one vector of length N. Here the calculation $\boldsymbol{u}\boldsymbol{G}_N$ is carried out in GF(2) and the result is treated as a real number vector in the rest of the calculation. SD algorithms can be used to solve the above minimization problem. The average complexity of these algorithms is $O(N^3)$ in many scenarios [14]. SD algorithms enumerate all points $\boldsymbol{u} \in \mathcal{U}$ that satisfy the constraint

$$\|\bar{\boldsymbol{y}} - \boldsymbol{u}\boldsymbol{G}_N\|^2 \leqslant r_0^2 \tag{2}$$

Here r_0 is a carefully chosen initial radius for the search. Finding an appropriate initial radius is beyond the scope of this paper. We refer readers to [14], [17] and references therein.

By the construction of the code $\boldsymbol{G}_N$ is a lower triangular matrix, and thus

$$\|\bar{\boldsymbol{y}} - \boldsymbol{u}\boldsymbol{G}_N\|^2 = \sum_{i=1}^{N}\Big(\bar{y}_i - \bigoplus_{j=i}^{N}(g_{j,i}u_j)\Big)^2. \tag{3}$$

We use $\bigoplus_{j=a}^{b}(\cdot)$ to denote the summation over $GF(2)$. Define $m_i(\boldsymbol{u}_i^N) \triangleq \Big(\bar{y}_i - \bigoplus_{j=i}^{N}(g_{j,i}u_j)\Big)^2$, where the summation $\bigoplus_{j=i}^{N}(g_{j,i}u_j)$ is carried out in GF(2) and the result is treated as a real number in the rest of the calculation. Eq. (2) can be written as

$$\sum_{i=1}^{N} m_i(\boldsymbol{u}_i^N) \leqslant r_0^2. \tag{4}$$

Since $m_i(\boldsymbol{u}_i^N)$ only depends on u_i^N, Eq. (4) can be solved in a back-substitution manner. Starting from level $\ell = N$, it finds all u_N such that

$$m_N(u_N) \leqslant r_0^2, \tag{5}$$

and then for every level $\ell = N-1,\ldots,1$, it finds all u_ℓ such that

$$\sum_{i=\ell}^{N} m_i(\boldsymbol{u}_i^N) \leqslant r_0^2. \tag{6}$$

When we reach level $\ell = 1$, all points that satisfy Eq. (4) are found. We need to point out that if $\ell \in \mathcal{F}$, the set of feasible solutions for Eq. (6) is $\{0\}$ since u_ℓ is frozen to 0; if $\ell \in \mathcal{F}^c$, the feasible set for Eq. (6) is $\{0,1\}$.

This procedure can be interpreted as a depth-first tree search algorithm. We call $m_\ell(\boldsymbol{u}_\ell^N)$ the branch metric at level ℓ and Eq. (6) the branching condition at level ℓ. All surviving leafs in the tree are points that satisfy Eq. (4). In order to find the ML point, [11] adaptively update the radius r_0. Once a valid solution is found, the radius r_0 is updated and the tree search is restarted with the new radius r_0. The authors in [13] proposed an implementation that does not require restarting the tree search after the radius is updated. We use this implementation for all simulations in this paper.

III. Sphere Decoding with Fixed Lower Bounds

In this section, we discuss a method to speed up SD by using stricter branching conditions. The key idea is to find lower bounds to the quantity

$$\min_{\boldsymbol{u}\in\mathcal{U}} \sum_{i=1}^{\ell-1} m_i(\boldsymbol{u}_i^N) \tag{7}$$

for each $\ell = N-1,\ldots,1$ based on $\boldsymbol{y}_1^N, \boldsymbol{u}_\ell^N$. Let Λ_ℓ be one such bound, then we have

$$\sum_{i=\ell}^{N} m_i(\boldsymbol{u}_i^N) + \Lambda_\ell \leqslant r_0^2. \tag{8}$$

When the lower bound is non-trivial, i.e. $\Lambda_\ell > 0$, Eq. (8) gives a stricter branching condition than Eq. (6).

Lemma 1. *For every $i = 1,\ldots,N, \boldsymbol{u}\in\mathcal{U}$, we have that*

$$m_i(\boldsymbol{u}_i^N) \geqslant \min_{x_i\in\{0,1\}} (\bar{y}_i - x_i)^2. \tag{9}$$

Proof: We have that

$$m_i(\boldsymbol{u}_i^N) \geqslant \min_{\boldsymbol{u}\in\{0,1\}^N} m_i(\boldsymbol{u}_i^N) \tag{10}$$

$$= \min_{\boldsymbol{u}\in\{0,1\}^N} \Big(\bar{y}_i - \bigoplus_{j=i}^{N}(g_{j,i}u_j)\Big)^2 \tag{11}$$

$$= \min_{x_i\in\{0,1\}} (\bar{y}_i - x_i)^2. \tag{12}$$

Eq. (10) comes from the fact that $\mathcal{U} \subset \{0,1\}^N$. We have Eq. (12) by letting $x_i = \bigoplus_{j=i}^{N}(g_{j,i}u_j)$ with $x_i \in \{0,1\}$ since the summation is carried out in GF(2). □

Let

$$\mathrm{LB}(i) = \min_{x_i\in\{0,1\}} (\bar{y}_i - x_i)^2 \tag{13}$$

we now use the branching condition

$$\sum_{i=\ell}^{N} m_i(\boldsymbol{u}_i^N) + \sum_{i=1}^{\ell-1}\mathrm{LB}(i) \leqslant r_0^2 \tag{14}$$

at level ℓ to decide whether the path $\boldsymbol{u}_\ell^N$ should be kept in the tree search or not. We call SD algorithms with branching conditions Eq. (14) *SD algorithms with fixed lower bounds*. Compared to standard SD algorithms, SD algorithms with fixed lower bounds have stricter conditions which enable us to remove branches that would eventually be outside the sphere at an earlier stage. Since $\mathrm{LB}(i), i = 1, \ldots, N$ only depends on $\boldsymbol{y}$, the extra computational complexity caused by calculating fixed lower bounds is $O(N)$.

A. Simulations

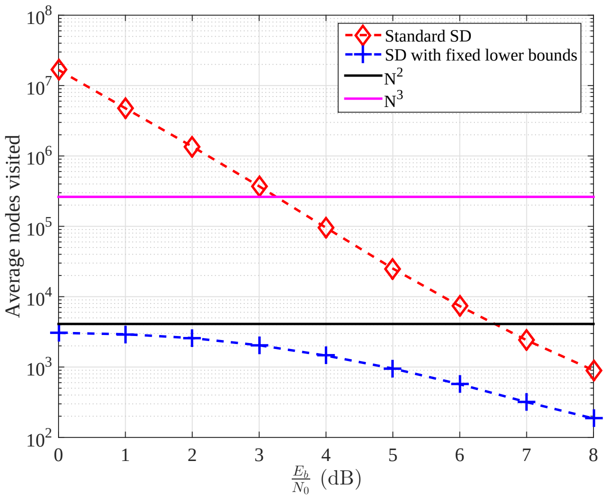

Fig. 1. Average number of nodes visited for RM $(64, 57)$.

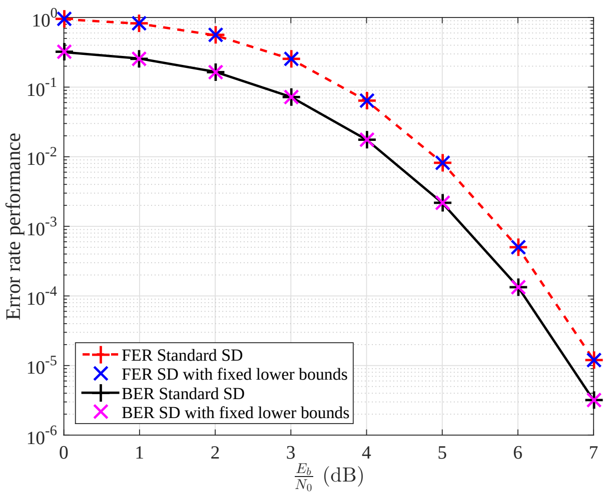

Fig. 2. Error rate performance for RM $(64, 57)$.

Fig. 1 shows the complexity of SD algorithms with fixed lower bounds for a Reed-Muller (RM) $(64, 57)$ code. The complexity is measured by the number of nodes visited during the SD. We use the Schnorr-Euchner Enumeration [18] as the enumerating order at each level. The initial square radius r_0^2 is set to be $+\infty$.[1] We observe that the complexity of SD algorithms with fixed lower bounds is significantly lower than that of standard SD algorithms over the whole SNR range. We then compare our results with decoding algorithms that achieve ML performance proposed in [8] and [9]. Note that the simulation setup is the same for Fig 1, [8, Fig. 4] and [9, Fig. 3]. We can see that our algorithms have advantages in terms of average number of nodes visited over [8, Fig. 4] and [9, Fig. 3] for all SNR range. For example, at $\frac{E_b}{N_0} = 4$ dB, the number of average nodes visited reported in [8, Fig. 4] is of the order of 10^5 and 10^4 in [9, Fig. 3], while for SD algorithms with fixed lower bounds, the number of average nodes visited is at the order of 10^3 (Fig. 1).

Fig. 2 confirms that error rate performance of standard SD algorithms and that of SD algorithms with fixed lower bounds is the same.

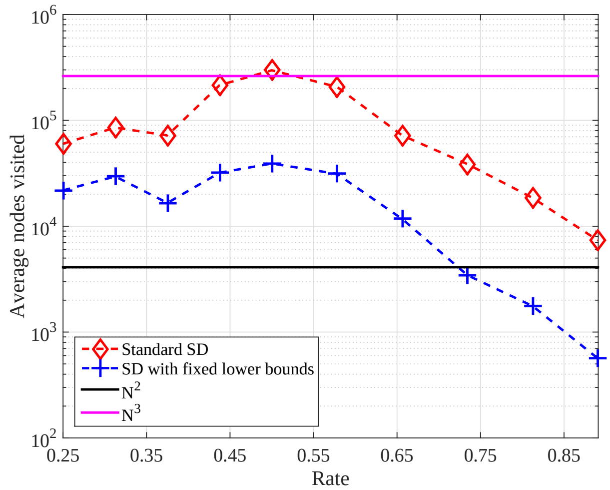

Fig. 3. Average number of nodes visited for RM $(64, K)$ with $R = \frac{K}{64}$, $\frac{E_b}{N_0} = 6$ dB.

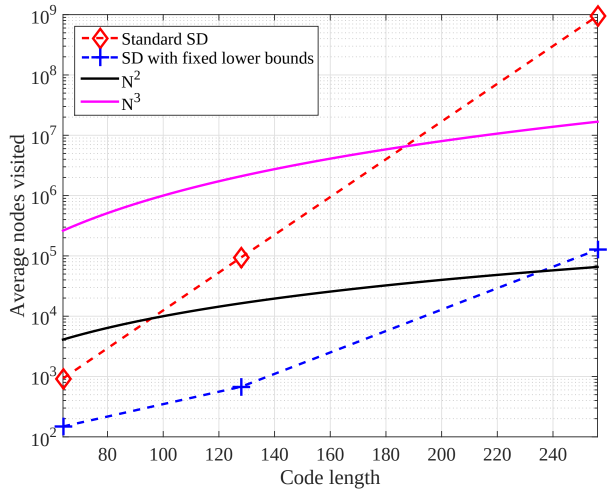

Fig. 4. Average number of nodes visited for polar code $(N, \lfloor RN \rfloor)$ with $R = 0.89$, $\frac{E_b}{N_0} = 8$ dB.

We now evaluate the tradeoff between the complexity of the algorithms and the code rate. Fig. 3 shows that the complexity decreases as the rate increases. SD algorithms with fixed lower bounds show lower complexity over standard SD algorithms over all rate regions.

Fig. 4 shows how the complexity increases with the length for a fixed rate. We can see that the complexity of SD

[1] The initial square radius is set to be sufficient large to guarantee that the ML point falls within the sphere. Moreover, the first point found is always the Babai point, thus the complexity of SD algorithms is not sensitive to increasing r_0^2 beyond the squared Babai distance [13].

algorithms with fixed lower bounds grows slower with the block length than the standard SD algorithm.

IV. Sphere Decoding with dynamic lower bounds

From Fig. 3, as the code rate decreases, the complexity of SD with fixed lower bounds grows. This is due to the fact that the fixed lower bounds are obtained as if there were no frozen bits (i.e. uncoded). Thus, since fixed lower bounds do not take advantage of the presence of frozen bits, fixed lower bounds for low rates could be improved upon. A possible improvement would be to prune the searching tree by using dynamic lower bounds that also depend on the current path $\boldsymbol{u}_\ell^N$ at level ℓ. In order to introduce our dynamic lower bounds, we first need to introduce further notation and preprocessing.

Given a frozen set $\mathcal{F}$, generate an $N \times N$ matrix $\hat{\boldsymbol{G}}$ by substituting the ith row in $\boldsymbol{G}_N$ with the all-zero vector, for all $i \in \mathcal{F}$. Let $\hat{g}_{i,j}$ denote the (i,j)th entry of $\hat{\boldsymbol{G}}$. Let $\hat{\boldsymbol{G}}_\ell$ denote the $\ell \times \ell$ submatrix of $\hat{\boldsymbol{G}}$ which contains the first ℓ rows and columns of $\hat{\boldsymbol{G}}$.

For every $\ell \in \mathcal{F}^c$, find all non-zero identical columns in $\hat{\boldsymbol{G}}_{\ell-1}$. Let d_ℓ denote the number of such columns. Define a set of sets $\mathcal{I}_\ell$ such that each element in $\mathcal{I}_\ell$ is a set that contains the indices of non-zero identical columns in $\hat{\boldsymbol{G}}_{\ell-1}$.

A. SD with Dynamic Lower Bounds

We first illustrate the idea of SD with dynamic lower bounds by an example. Then we will give general algorithms. For an $(8,4)$ polar code with the frozen set $\mathcal{F} = \{1,2,3,5\}$. After preprocessing, we have

$$\hat{\boldsymbol{G}} = \begin{bmatrix} 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0\\ 0&0&0&0&0&0&0&0\\ 1&1&0&0&1&1&0&0\\ 1&0&1&0&1&0&1&0\\ 1&1&1&1&1&1&1&1 \end{bmatrix}, \tag{15}$$

and

$$d_8 = 0,\ \mathcal{I}_8 = \emptyset; \tag{16}$$
$$d_7 = 3,\ \mathcal{I}_7 = \{\{6,5\},\{4,3\},\{2,1\}\}; \tag{17}$$
$$d_6 = 1,\ \mathcal{I}_6 = \{\{4,3,2,1\}\}; \tag{18}$$
$$d_4 = 0,\ \mathcal{I}_0 = \emptyset. \tag{19}$$

Assume we are at level $\ell = 7$. Therefore, (u_7, u_8) are fixed to $(0,0)$ for simplicity. Thus, we now would like to bound $\min_{\boldsymbol{u}\in\mathcal{U}} \sum_{i=1}^{6} m_i(\boldsymbol{u}_i^N)$. A convenient bound is

$$\begin{aligned}\min_{\boldsymbol{u}\in\mathcal{U}} \sum_{i=1}^{6} m_i(\boldsymbol{u}_i^N) &\geqslant \min_{\boldsymbol{u}\in\mathcal{U}} \sum_{i=1}^{2} m_i(\boldsymbol{u}_i^N) \\ &+ \min_{\boldsymbol{u}\in\mathcal{U}} \sum_{i=3}^{4} m_i(\boldsymbol{u}_i^N) \\ &+ \min_{\boldsymbol{u}\in\mathcal{U}} \sum_{i=5}^{6} m_i(\boldsymbol{u}_i^N) \qquad (20)\\ &= \min_{v_1\in\{0,1\}} \sum_{i=1}^{2} (\bar{y}_i - v_1)^2 \\ &+ \min_{v_2\in\{0,1\}} \sum_{i=3}^{4} (\bar{y}_i - v_2)^2 \\ &+ \min_{v_3\in\{0,1\}} \sum_{i=5}^{6} (\bar{y}_i - v_3)^2. \qquad (21)\end{aligned}$$

We have Eq. (21) since

$$v_1 = \bigoplus_{j=1}^{8} (\hat{g}_{j,1} u_j) = \bigoplus_{j=2}^{8} (\hat{g}_{j,2} u_j), \tag{22}$$
$$v_2 = \bigoplus_{j=3}^{8} (\hat{g}_{j,3} u_j) = \bigoplus_{j=4}^{8} (\hat{g}_{j,4} u_j), \tag{23}$$
$$v_3 = \bigoplus_{j=5}^{8} (\hat{g}_{j,5} u_j) = \bigoplus_{j=6}^{8} (\hat{g}_{j,6} u_j). \tag{24}$$

Eqs. (22)-(24) follow since $(u_7, u_8) = (0,0)$ and columns $\{6,5\},\{4,3\},\{2,1\}$ of $\hat{\boldsymbol{G}}_6$ are identical, respectively. Eq. (21) gives a tighter bound than the fixed lower bounds discussed in the previous section. Furthermore, it is particularly convenient since, thanks to the fact that columns $\{6,5\},\{4,3\},\{2,1\}$ of $\hat{\boldsymbol{G}}_6$ are identical, we only need to minimize over 1 variable instead of 2 in each term of Eq. (21). This idea is generalized in the following.

Lemma 2. *At level $\ell \in \mathcal{F}^c$, let $t_{\ell,i} = \bigoplus_{j=\ell}^{N} (g_{j,i} u_j)$. If $d_\ell > 0$, then for all $\mathcal{I} \in \mathcal{I}_\ell$, we have*

$$\sum_{i\in\mathcal{I}} m_i(\boldsymbol{u}_i^N) \geqslant \min_{u\in\{0,1\}} \sum_{i\in\mathcal{I}} (\bar{y}_i - u \oplus t_{\ell,i})^2 \tag{25}$$
$$\geqslant \sum_{i\in\mathcal{I}} LB(i) \tag{26}$$

where $LB(i)$ is defined in (13).

Proof:

$$\sum_{i\in\mathcal{I}} m_i(\boldsymbol{u}_i^N) = \sum_{i\in\mathcal{I}}\Big(\bar{y}_i - \bigoplus_{j=i}^{N}(g_{j,i}u_j)\Big)^2 \quad (27)$$

$$= \sum_{i\in\mathcal{I}}\Big(\bar{y}_i - \Big(\bigoplus_{j=i}^{\ell-1}(g_{j,i}u_j)\Big)\oplus t_{\ell,i}\Big)^2 \quad (28)$$

$$\geqslant \min_{u\in\{0,1\}}\sum_{i\in\mathcal{I}}(\bar{y}_i - u\oplus t_{\ell,i})^2 \quad (29)$$

$$\geqslant \sum_{i\in\mathcal{I}}\min_{u\in\{0,1\}}(\bar{y}_i - u\oplus t_{\ell,i})^2 \quad (30)$$

$$= \sum_{i\in\mathcal{I}}\mathrm{LB}(i). \quad (31)$$

Eq. (29) comes from the fact that, $\bigoplus_{j=i}^{\ell-1}(g_{j,i}u_j)$ gives the same value for all $i\in\mathcal{I}$. □

Note that the calculation of $t_{\ell,i}$ forms part of the calculation $\bigoplus_{j=i}^{N}(g_{j,i}u_j)$. Thus $t_{\ell,i}$ can be stored and re-used during the decoding process. The extra complexity of dynamically updating lower bounds comes from the minimization step Eq. (29).

Now we summarize the steps for SD with dynamic lower bounds:

- **Preprocessing:** Given an (N,K) polar code with a frozen set $\mathcal{F}$, calculate $d_\ell,\mathcal{I}_\ell$ for all $\ell\in\mathcal{F}^c$.
- **Dynamically update lower bounds:** At level ℓ. If $d_\ell>0$, for all $\mathcal{I}\in\mathcal{I}_\ell$:
  - Calculate $t_{\ell,i}=\bigoplus_{j=\ell}^{N}(g_{j,i}u_j)$, for all $i\in\mathcal{I}$.
  - Calculate

$$\min_{u\in\{0,1\}}\sum_{i\in\mathcal{I}}(\bar{y}_i - u\oplus t_{\ell,i})^2. \quad (32)$$

  - Update lower bounds on $\sum_{i\in\mathcal{I}} m_i(u_i^N)$ using Eq. (32).

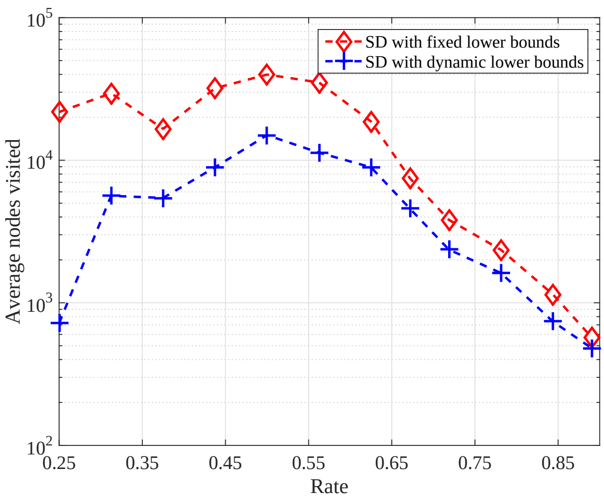

Fig. 5. Average number of nodes visited for RM $(64,K)$ for $\frac{E_b}{N_0}=6$ dB.

Fig. 5 compares the complexity of SD with fixed and dynamic lower bounds. We observe that dynamic lower bounds, further reduce the complexity in the low rate region. We should note that the average computational complexity for each node of dynamic lower bound is higher because we need to update the lower bounds during the tree search.

V. Conclusions

We have proposed several techniques to lower the complexity of ML decoding of polar and RM codes. This is achieved by means of improving the standard branching conditions of the sphere decoder by introducing lower bounds on the remaining metric. The proposed techniques improve the search complexity of the algorithm by several orders of magnitude.

References

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[2] E. Arıkan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447 – 449, June 2008.

[3] U. U. Fayyaz and J. R. Barry, "Polar codes for partial response channels," in *IEEE Int. Conf. Commun.*, Budapest, Hungary, June 2013.

[4] J. Guo, M. Qin, A. Guillén i Fàbregas, and P. H. Siegel, "Enhanced belief propagation decoding of polar codes through concatenation," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, June 2014.

[5] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, July - Aug. 2011.

[6] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications," in *Proc. ACM SIGSAM*, 1981, vol. 15.

[7] U. Fincke and H. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Computat.*, vol. 44, no. 170, pp. 463–471, Apr. 1985.

[8] S. Kahraman and M. E. Çelebi, "Code based efficient maximum-likelihood decoding of short polar codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, July 2012.

[9] K. Niu, K. Chen, and J. Lin, "Low-complexity sphere decoding of polar codes based on optimum path metric," *IEEE Commun. Lett.*, vol. 18, no. 2, pp. 332–335, 2014.

[10] S. Kahraman, E. Viterbo, and M. E. Çelebi, "Folded tree maximum-likelihood decoder for kronecker product-based codes," in *Proc. 51-st Annu. Allerton Conf. Commun., Control and Computing*, Monticello, IL, Oct. 2013.

[11] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1639–1642, July 1999.

[12] M. Stojnic, H. Vikalo, and B.Hassibi, "A branch and bound approach to speed up the sphere decoder," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, Mar. 2005, vol. 3, pp. 429–432.

[13] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[14] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Process.*, vol. 53, pp. 2806–2818, Aug. 2005.

[15] A. D. Murugan, H. El Gamal, M. O. Damen, and G. Caire, "A unified framework for tree search decoding: rediscovering sequential decoding," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 933–953, Aug. 2005.

[16] M. El-Khamy, H. Vikalo, B. Hassibi, and R. J. McEliece, "Performance of sphere decoding of block codes," *IEEE Trans. Commun.*, vol. 57, no. 10, pp. 2940–2950, Oct. 2009.

[17] W. Zhao and G. B. Giannakis, "Sphere decoding algorithms with improved radius search," *IEEE Trans. Commun.*, vol. 53, no. 7, pp. 1104–1109, July 2005.

[18] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programm.*, vol. 66, no. 1-3, pp. 181–191, 1993.