



**HAL**  
open science

## Tailored List Decoding of Polar Codes

Malek Ellouze, Camille Leroux, Romain Tajan, Charly Poulliat, Christophe Jégo

► **To cite this version:**

Malek Ellouze, Camille Leroux, Romain Tajan, Charly Poulliat, Christophe Jégo. Tailored List Decoding of Polar Codes. 11th International Symposium on Topics in Coding (ISTC 2021), Aug 2021, Montréal, Canada. pp.141–146, 10.1109/ISTC49272.2021.9594266 . hal-04289467

**HAL Id: hal-04289467**

**<https://hal.science/hal-04289467>**

Submitted on 16 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Tailored List Decoding of Polar Codes

Malek Ellouze\*, Camille Leroux\*, Romain Tajan\*, Charly Poulliat<sup>†</sup> and Christophe Jégo\*

\*University of Bordeaux, Bordeaux INP, IMS Lab, UMR CNRS 5218, France

<sup>†</sup>University of Toulouse, INPT-ENSEEIH, IRIT Lab, UMR CNRS 5505, France

**Abstract**—In this paper, we provide some experimental evidence that a significant number of metric updates and sorting can be avoided during list decoding of polar codes. Starting from the observation that a non-negligible part of the reliable bit channels are located at the tail of the decoding sequence, it is possible to decode the end of the sequence without updating the metrics of list decoding. This tailored CRC-Aided successive cancellation list (T-CA-SCL) decoder can be seen as a simplification of SCL decoding. Moreover, a heuristic is proposed to determine the parameters required by the tailored-CA-SCL decoder. Experimental results show that the proposed tailored list decoding approach reduces the number of metric updates by 45% without significantly altering the error performance. Finally some leads are provided for further research to improve and extend the proposed approach.

**Index Terms**—Polar codes, successive cancellation list decoding.

## I. INTRODUCTION

Introduced by Arikan in [1], polar codes are the first code family to provably achieve the symmetric capacity of binary input discrete memory-less channels using the successive cancellation (SC) decoder. However, for moderate code lengths, the equivalent channels are only partially polarized and the SC decoder has poor performance. To address this problem, the successive cancellation list (SCL) decoder was introduced in [2]. This method consists in keeping  $L$  candidates paths during the whole decoding process simultaneously and choosing the most probable ones. For a sufficiently large value  $L$ , the performance of SCL decoder is similar to those of the maximum likelihood (ML) decoding for high signal-to-noise ratios (SNR). It was also remarked in [2] that even when SCL decoding fails, the transmitted codeword path is within the final list with very high probability. This led [2] and [3] to propose concatenating a Cyclic Redundancy Check (CRC) with a polar code. In addition to increasing the minimum distance of the code, the CRC helps the SCL decoder finding the transmitted codeword path. Indeed, under very similar constraints, CA-SCL outperforms most other decoders and comes close to optimum performance at finite length. The good decoding performance of SCL-based algorithms is reached at the detriment of a computational complexity that is at least  $L$  times larger than SC decoding complexity. This is in part due to the need of the dynamic sorting of the metrics list when it comes to considering new paths. There are mainly two ways to reduce the computational complexity of a CA-SCL decoder. The first option is to reduce the computational complexity of the component SC decoders composing the SCL [4][5]. The main idea is to identify particular sub-codes in the polar code

in such a way that specific parallel decoding algorithm can be locally applied. These SC-reduction methods are static in the sense that the complexity reduction does not depend on the decoded codeword or on the SNR. They are thus beneficial for both hardware [6] and software implementations [7].

A second way is to reduce the size of the list whenever it is relevant during the decoding process [8][9]. Some of these approaches can significantly reduce the average list size but, as any dynamic method, it suffers from a worst case complexity that is actually equivalent (if not worse) to the complexity of the CA-SCL. These list-reduction techniques are more suitable to software implementation in which the worst case latency is not critical. In this work, a list-reduction method that "tailors" the path splitting profile is investigated. Counter to similar list-reduction methods [8][9], the proposed algorithm performs very close to CA-SCL decoding. Moreover, the list tailoring is defined statically so that the complexity reduction does not depend on the SNR and is constant for any decoded codeword. The proposed tailored-CA-SCL (T-CA-SCL) decoder reduces the number of path splitting by  $\sim 35\%$ - $45\%$  for different polar codes. Moreover, it keeps the decoding performance very close to CA-SCL.

The paper is organized as follows. In Section II, polar coding and SC-based decoders are presented. Then, in Section III, the proposed tailored-CA-SCL algorithm is detailed. In particular, some insights are empirically given to motivate our proposal, showing that it is possible to exploit the fact that the most reliable channels are located at the end of the decoded sequence. In Section IV, some experimental results are provided and future research directions are discussed.

## II. POLAR CODING AND DECODING

### A. Polar coding under SC decoding

Polar codes are defined by the code length  $N = 2^n$ , the code dimension  $K$  and the code rate  $R = K/N$ . The polar code generator matrix is defined as a sub-matrix consisting in  $K$  lines in the following transformation  $G_N = G_2^{\otimes n}$  where  $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$  and  $(\cdot)^{\otimes n}$  denotes the  $n^{\text{th}}$  Kronecker power. In the following,  $a_i^j$  represents the sequence  $(a_i, a_{i+1}, \dots, a_j)$  and  $\hat{a}_i^j$  denotes the estimated value of  $a$ , i.e  $\hat{a}_i^j = (\hat{a}_i, \hat{a}_{i+1}, \dots, \hat{a}_j)$ . We denote by  $\mathbf{u} = u_1^N$  the binary information vector including the frozen bits and by  $\mathbf{x} = x_1^N$  the transmitted codeword such that  $\mathbf{x} = \mathbf{u}G_N$ . Bits of  $\mathbf{x}$  are transmitted over a Binary Memoryless (BMS) channel whose codeword's likelihood is  $W_N(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^N W(y_i|x_i)$ .

The SC decoder estimates  $u_i$  from the received word  $y_1^N$  and the past decisions  $\hat{u}_1^{i-1}$  as:

$$\hat{u}_i = \begin{cases} 0 & \text{if } L(u_i) \geq 0 \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

where  $L(u_i)$  is defined as:

$$L(u_i) = \log \left( \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 1)} \right) \quad (2)$$

and where  $W_N^{(i)}$  is introduced by Arikan as:

$$W_N^{(i)}(\mathbf{y}, u_1^{i-1} | u_i) = \sum_{u_{i+1}^N \in \{0,1\}^{N-i}} \frac{1}{2^{N-i}} W_N(\mathbf{y} | G_N \mathbf{u}). \quad (3)$$

Defining a polar code therefore boils down to pinpointing the  $K$  positions where the information bits are sent, whereas the remaining ones are *frozen* (e.g. they are set to 0). The set of frozen bits is denoted by  $\mathcal{F}$ .

### B. SCL decoding

SC List (SCL) decoding algorithm is an improvement of the SC decoding algorithm by delaying hard-decisions. Instead of solely making the decision based on the LLR value  $s$  in (1), the two decoding possibilities are considered and kept in a list of size  $L$ . Hence each path kept by the SCL algorithm is updated according to an SC decoding process.

Therefore, each decoding path splits into two children paths. A metric is introduced to penalize one decoding path at the expense of the other. It works as follows: at decoding step  $i$ , the  $j^{\text{th}}$  SC decoder computes the LLR  $L_j^{(i)} = L_j(u_i)$ , the path metric  $m_j^{(i)}$  is updated as follows:

$$m_j^{(i)} = \begin{cases} m_j^{(i-1)} & \text{if } L_j^{(i)} \geq 0 \text{ and } \hat{u}_j^i = 0 \\ m_j^{(i-1)} + |L_j^{(i)}| & \text{if } L_j^{(i)} < 0 \text{ and } \hat{u}_j^i = 0 \\ m_j^{(i-1)} + |L_j^{(i)}| & \text{if } L_j^{(i)} \geq 0 \text{ and } \hat{u}_j^i = 1 \\ m_j^{(i-1)} & \text{if } L_j^{(i)} < 0 \text{ and } \hat{u}_j^i = 1 \end{cases} \quad (4)$$

Where  $|\cdot|$  denotes the absolute value operator. The algorithm then selects the best  $L$  candidates having the lowest path metrics. When all the bits are processed, the SCL algorithm selects the path with the smallest metric.

Cyclic Redundancy Code (CRC) Aided SCL decoder, denoted by CA-SCL, is an improvement of SCL that consists of introducing a CRC within the information vector  $\mathbf{u}$  in order to check if a path is "valid" or not. This idea has been proposed in [2] where it has been noticed that when SCL fails, the good path often appears in the final list of paths. CA-SCL algorithm chooses the candidate with a valid CRC first and if no or several candidates have a valid CRC, the one with the lowest SCL metric is chosen.

Note that when considering a CA-SCL algorithm, the code rate will be defined as  $R = (K - L_{CRC})/N$  where  $L_{CRC}$  denotes the number of CRC bits.

SCL and CA-SCL decoding algorithms are therefore improvements to the basic SC decoding algorithm. They do guarantee

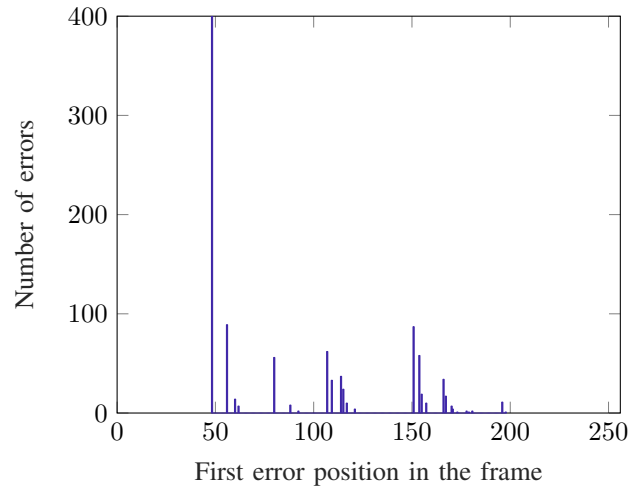


Fig. 1: First error position histogram for a (256,128) polar code decoded with an SC decoding algorithm at  $\frac{E_b}{N_0} = 3$  dB

better performances at the cost of a higher computational complexity.

### III. TAILORED SUCCESSIVE CANCELLATION LIST DECODING

In this section, a new decoding algorithm is introduced to guarantee a computational complexity reduction compared to the CA-SCL algorithm with a small loss in performance. It takes advantage from the fact that the majority of the most reliable bits are located at the end of the encoded sequence. Based on this observation, a new process enables to combine both SC and SCL algorithms with no significant performance loss.

#### A. Decoding tail in SC-based algorithms

It has been observed in [2] that a genie-assisted SC algorithm drastically improves the decoding performance. In such an approach, a "genie" is allowed to correct up to two errors during the SC decoding process. In other words, in SC-based decoding, the very first errors have a major impact on the decoding performance. In Figure 1, the distribution of the first error positions of the SC decoder is provided for 1000 sequences that an SC failed to decode from a (256, 128) polar code. This Monte Carlo simulation shows that the end of the decoding sequence, here denoted as "tail", is almost error-free. This means that applying list-decoding at the end of the sequence is superfluous. In fact, a simple SC decoder would provide very similar performance.

Based on this observation, we propose a simplification of the CA-SCL decoding algorithm in which the "tail" of the sequence is decoded with SC decoding. More precisely, starting from a certain index  $\omega = N - T$ , the path metrics are no longer updated nor is the metric sorting process. Instead, the  $L$  paths are kept and an SC decoding is performed independently on each one of these paths. More details of this decoding algorithm are given in section III-B. When considering the

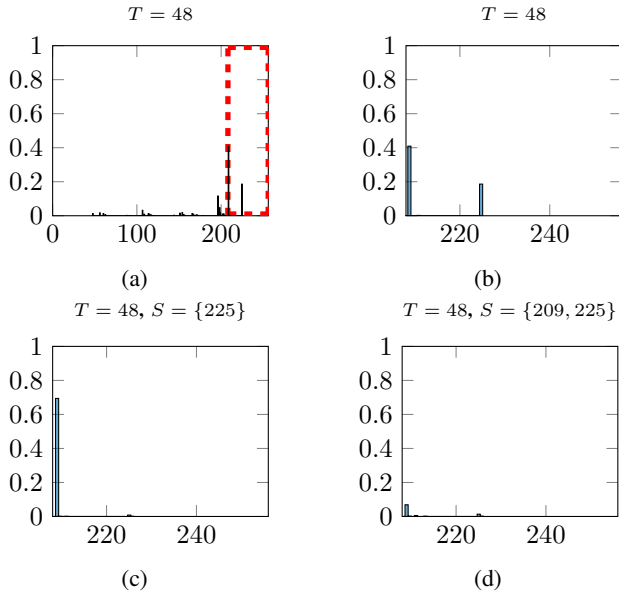


Fig. 2: Distribution of the first errors occurring on the tail

---

**Algorithm 1:** T-CA-SCL decoding

---

**Input:** Polar code  $\mathcal{P}(N, K, \mathcal{F}), T, S, L$

```

1 for  $i \in \llbracket 1; N - T \rrbracket$  do
2   | Perform stage  $i$  of SCL
3 end
4  $\mathcal{L} \leftarrow$  list of the SCL paths;
5 for  $i \in \llbracket N - T + 1; N \rrbracket$  do
6   | if  $i \notin \mathcal{F}$  and  $i \notin S$  then
7     |   foreach  $l \in \mathcal{L}$  do
8       |   | SC decoding of  $u_i$  on path  $l$ 
9     |   end
10  | else
11  |   perform stage  $i$  of SCL
12 end
13  $valid \leftarrow$  The path with the least metric that verifies
   the CRC ;
14 if  $valid = false$  then
15   | Report a decoding failure
16 else
17   | Return  $\mathbf{u}[valid]$ 
18 end
```

---

$N = 256$  polar code in figure 1, for  $T = 256$ , the decoding process is equivalent to perform SC decoding on  $L$  identical paths. When  $T = 0$ , the decoding process is equivalent to a CA-SCL and the number of errors is then significantly lower than for  $T = 256$ . For intermediate  $T$  values, some error peaks appear in the tail zone. Indeed, the SC algorithm is sensitive to the least reliable channels. However, if  $T$  is sufficiently low ( $T < 32$ ), no significant error peak is visible. The error performance should then be close to CA-SCL while the list decoding is replaced by SC decoding in the tail.

In Figure 2a, the percent of the distribution of the first errors

that occur on the tail are provided for  $T = 48$ . One should notice that the error rate is dominated by an error peak located at index  $i = 225$ . To reduce the complexity, we propose to increase  $T$  (Figure 2b) and to reduce the error peak by allowing the list decoder to be updated at index  $i = 225$ . This extra split on the decoding paths enables for a longer tail while avoiding significant decoding performance loss (Figure 2c). We introduce  $S$  as the set of the indices where splitting is considered on the tail. In this case,  $|S|$  defines the number of splits that are performed on the tail. In order to reduce further the complexity, other extra splits can be added. In Figure 2d, the histogram is shown when  $|S| = 2$ :  $S = \{209, 225\}$ . One observes that after two splits, there are no peaks dominating the histogram anymore.

This observation is confirmed in Figure 3 where the evolution of the frame error rate is shown according to the tail length  $T$ . For  $T < 31$ , the frame error rate (FER) of T-CA-SCL is equivalent to CA-SCL. For  $T > 30$ , when no extra split is applied, a performance loss occurs. However, with one extra split at position  $i = 225$ , the FER remains equivalent to CA-SCL until  $T = 47$ .

*B. Tailored SCL Decoding algorithm*

Algorithm 1 gives the details of the T-CA-SCL decoding algorithm. The decoding process is performed into two major steps:

- 1) The first step consists in decoding  $u_1^{N-T}$ . During this decoding step, T-CA-SCL is identical to an SCL decoder. We denote  $\mathcal{L}$  the list of the candidate paths at the end of this step. This phase is represented for  $L = 4$  in Figure 4.
- 2) The second step starts with the decoding of the  $N - T + 1^{th}$  bit. During this step, two different treatments are considered depending on the decoding stage:
  - $i \notin S$ : Typically, the bits at the end of the sequence lead to one path being much more favorable than the other. This means that in most cases, it is very unlikely to make a false decision by choosing the favorable path. At this point, an SC decoder is performed on each one of the previously kept paths separately. This means that no path sorting nor metric updating are needed.
  - $i \in S$ : Sometimes, the bits at the end of sequence can be unreliable. Hard decision making at this point may lead to non negligible errors. In this particular case, an SCL decoding is performed on those specific bits. As an example, figure 4 shows a splitting on the tail for  $u_{S_1}$ .

Note that frozen bits are set to zero and are omitted in Algorithm 1. A CRC check is then applied to return the decoded sequence. By this way, the T-CA-SCL algorithm can achieve lower computational complexity by removing both metrics sorting and updating for a significant number of decoding stages.

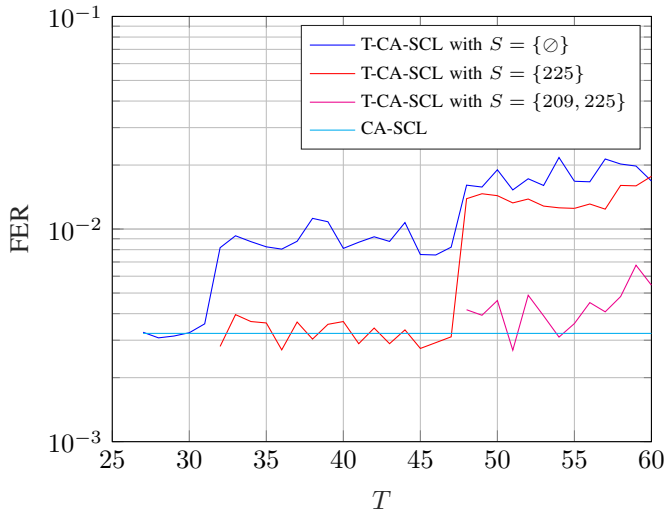


Fig. 3: FER of CA-SCL and T-CA-SCL for different  $T$  value. (256,128) polar code,  $L = 8$ ,  $L_{CRC} = 24$  and  $\frac{E_b}{N_0} = 3$  dB.

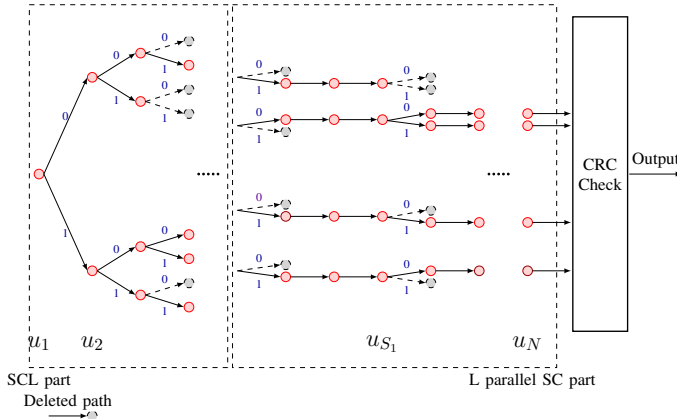


Fig. 4: T-CA-SCL decoding process illustration.

## IV. EXPERIMENTAL RESULTS

### A. Generation of T-CA-SCL parameters

Before the evaluation of FER performance, for each investigated polar code, the parameters  $T$  and  $S$  have to be fixed. The heuristic described in Algorithm (2) performs Monte Carlo simulation on 100000 frames with a T-CA-SCL decoder and gradually increases the tail size  $T$  until the FER performance decreases. As explained in Section III-A, some unreliable channels are located at the end of the sequence. In order to further increase  $T$  without significantly degrading the performance, the decoder can perform list decoding on the unreliable bits with indices  $S$ .

Algorithm 2 was applied to 6 different polar codes ( $R = \{1/2, 3/4\}$  and  $N = \{128, 256, 512\}$ ) selected from the 5G standard [10]. Table I summarizes  $T$  and  $S$  values obtained for these codes. For some configurations, the most unreliable bits are located one after another. This means that splits have to be performed on all these bits and this drastically lowers the performance. The ratio  $\rho = (T - |S|)/K$

### Algorithm 2: T-CA-SCL algorithm parameter generation

**Input:** Polar code  $\mathcal{P}(N, K, \mathcal{F})$

- 1  $S = \{\emptyset\}$ ,  $T = 0$ ,  $j = 0$ ;
- 2  $\mathbf{Y}$  : set of 10000 noisy codewords;
- 3  $FER \leftarrow T\text{-CA-SCL}(\mathcal{P}, T, S, \mathbf{Y})$ ;
- 4 **while**  $FER \sim FER_{CA-SCL}$  **do**
- 5      $FER \leftarrow T\text{-CA-SCL}(\mathcal{P}, T, S, \mathbf{Y})$ ;
- 6     **if**  $FER \sim FER_{CA-SCL}$  **then**
- 7          $T \leftarrow T + 1$ ;
- 8     **else**
- 9          $s_j = T$ ;
- 10          $j \leftarrow j + 1$ ;
- 11     **end**
- 12 **end**
- 13 **return**  $(T, S)$

represents the proportion of bits decoded without splitting. Depending on the considered code, this ratio can reach as much as  $\sim 45\%$ . In other words, almost half of the information bits are decoded without metric update nor metric sorting without degrading the decoding performance.

### B. Decoding performance

Error rate Monte Carlo simulations were performed on the six selected polar codes. The decoding performance curves in Figures 5 and 6 confirm that the  $T$  and  $S$  parameter selection enables the T-CA-SCL algorithm to perform very close to CA-SCL algorithm. The same trends were observed for larger list sizes.

Even if the proposed approach is different from previous works, the idea of limiting the list size during SCL decoding was previously investigated in [8]. More precisely, during a decoding step  $i$ , for each path  $l$ , the computed LLR for bit  $i$  is compared to a value derived from density evolution. If this bit is more reliable than the expected value, the associated path does not split. Depending on the considered SNR value, it enables an average reduction in the number of kept paths. This heuristic was later improved in [9], where the frequently splitting paths are eliminated and a CRC is applied.

Unlike these two approaches, T-CA-SCL algorithm does not significantly degrade the decoding performance compared to CA-SCL algorithm. Moreover, the path splitting reduction does not depend on the received word  $\mathbf{y}$ , nor the SNR value since the non-splitting paths are determined statically. Actually, they are the same for all simulated SNRs. Therefore, the latency reduction is constant and beneficial for both hardware and software implementations. Furthermore, in [8] and [9], the LLR is compared with a value that has to be stored for each bit and for each SNR value while our approach only requires to memorize the position of the non splitting paths.

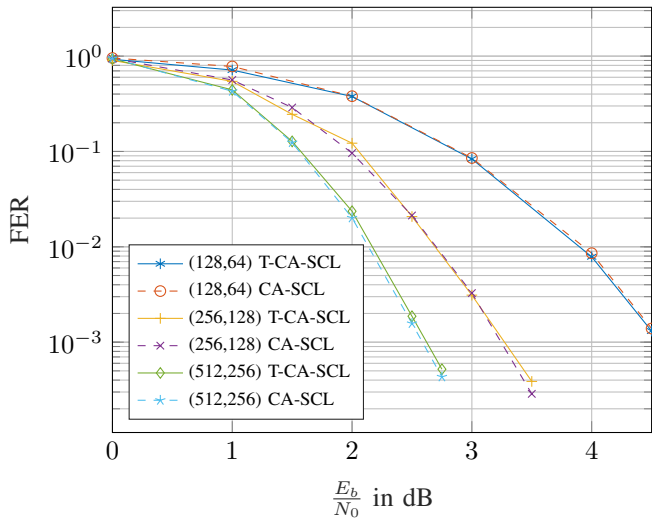


Fig. 5: CA-SCL versus T-CA-SCL.  $R = 1/2$  Polar codes,  $L = 8$

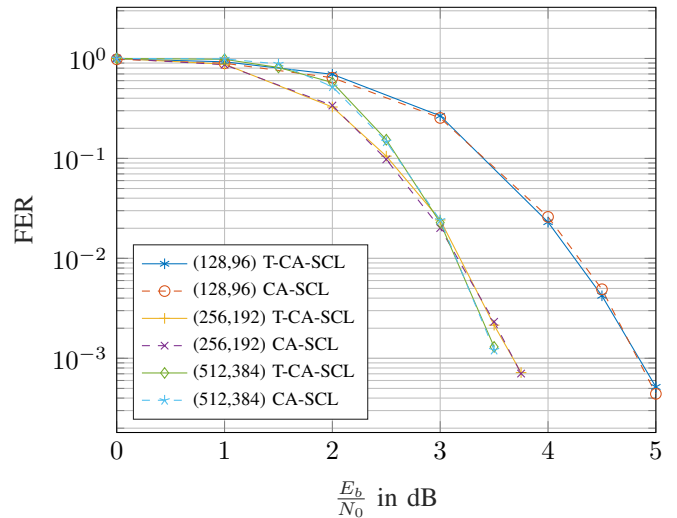


Fig. 6: CA-SCL versus T-CA-SCL.  $R = 3/4$  Polar codes,  $L = 8$

### C. Extension to other FEC codes and decoding algorithms

This work shall be seen as a first step towards further improvements. First, the heuristic to determine  $T$  and  $S$  values is probably sub-optimal. A more analytical approach should be investigated to find a better set of non-splitting paths. Similarly to [8][9], the proposed T-CA-SCL was applied to a plain CA-SCL decoding algorithm. But a simplified-SCL (CA-SSCL) would also benefit from the described tailoring. Since the numbers of metric updates and sorting are lower in CA-SSCL than in CA-SCL, one should however expect a lower gain in terms of computational complexity. Finally, in the case of polarization-adjusted convolutional (PAC) codes [11] or more generally polar codes with dynamic frozen bits [12], the frozen bits are not constant during the decoding process. Therefore, the pruning strategy applied in SSCL algorithms is not directly applicable. In this context, the T-CA-SCL appears as an effective way to reduce the complexity of list decoding when tree pruning is not possible. The analysis of the distribution of errors in PAC codes and the relevance of T-CA-SCL remain to be explored and analyzed in future works.

TABLE I: Polar code and T-CA-SCL decoding parameters

$N$	128	256	512	128	256	512
$K$	64	128	256	96	192	384
$R$	1/2	1/2	1/2	3/4	3/4	3/4
$S$	113	209,225	289	97	∅	∅
$ S $	1	2	1	1	0	0
$T$	23	54	93	45	87	126
$\rho(\%)$	34	36	36	46	45	32

## V. CONCLUSION

A new rule that consists of avoiding splitting on some bits has been introduced in this paper. Actually, it is possible to

remove splitting at the end of each decoded sequence. This approach enables to avoid the computationally heavy sorting for a significant number of iterations during the decoding process. This new algorithm is labelled as T-CA-SCL. Thus, starting from a particular position,  $L$  number of SC decoding can be executed in parallel instead of complex SCL decoding. The experimental results show that the computational complexity can be highly reduced while keeping almost similar error decoding performance.

## REFERENCES

- [1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. on Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [2] I. Tal and A. Vardy, "List decoding of polar codes," in *2011 IEEE Int. Symposium on Inf. Theory Proceedings*, 2011, pp. 1–5.
- [3] K. Niu and K. Chen, "Crc-aided decoding of polar codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668–1671, 2012.
- [4] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1378–1380, 2011.
- [5] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 946–957, 2014.
- [6] P. Giard, G. Sarkis, A. Balatsoukas-Stimming, Y. Fan, C.-y. Tsui, A. Burg, C. Thibeault, and W. J. Gross, "Hardware decoders for polar codes: An overview," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 149–152.
- [7] M. Léonardon, A. Cassagne, C. Leroux, C. Jégo, L. Hamelin, and Y. Savaria, "Fast and flexible software polar list decoders," *CoRR*, vol. abs/1710.08314, 2017.
- [8] Z. Zhang, L. Zhang, X. Wang, C. Zhong, and H. V. Poor, "A split-reduced successive cancellation list decoder for polar codes," *IEEE J. Sel. Areas in Commun.*, vol. 34, no. 2, pp. 292–302, 2016.
- [9] L. Zhang, A. Cao, J. Qiao, and Y. He, "A crc-aided sr-scl decoding algorithm for polar codes," in *2019 IEEE MTT-S International Wireless Symposium (IWS)*, 2019, pp. 1–3.
- [10] 3GPP, "Nr; multiplexing and channel coding."
- [11] M. Moradi, A. Mozammel, K. Qin, and E. Arikan, "Performance and complexity of sequential decoding of PAC codes," *CoRR*, vol. abs/2012.04990, 2020.
- [12] P. Trifonov and V. Miloslavskaya, "Polar codes with dynamic frozen symbols and their decoding by directed search," in *2013 IEEE Information Theory Workshop (ITW)*, 2013, pp. 1–5.