# Improved Successive Cancellation Decoding of Polar Codes

Authors: Kai Chen, Kai Niu, Jiaru Lin

Publication: IEEE T. Comm, Aug 2013
Speaker: Jeong-Min Ryu

**Short summary:** Combining the principles of the successive cancellation list (SCL) and the successive cancellation stack (SCS), they proposed a new decoding method for polar codes in this paper**.** The decoding algorithm is called the successive cancellation hybrid (SCH). This proposed algorithm can provide a flexible configuration when the time and space complexities are limited. Performance and complexity analysis based on simulations shows that under proper configurations, all the three improved successive cancellation (ISC) decoding algorithms can approach the performance of the maximum likelihood (ML) decoding but with acceptable complexity.

## I. Introduction

- **Polar codes with successive cancellation (SC) decoding**

Although polar codes are asymptotically capacity achieving, the performance under the SC decoding is unsatisfying in the practical cases with finite-length blocks.

→Alternative decoding schemes:

1) Belief propagation (BP) decoder
 - It can be applied to only BEC.
2) Linear programming decoder
 - It can be applied to only BEC.
3) Maximum likelihood (ML) and maximum a posteriori (MAP) decoders
 - It can work on very short code blocks because of their high complexity.

- **Improved versions of SC decoder**

  1) Successive cancellation list (SCL) decoding algorithm

  - It is introduced to approach the performance of ML decoding with acceptable complexity.

  2) Successive cancellation stack (SCS) decoding algorithm

  - Compared with SCL, it has a much lower time complexity but has a much larger space complexity.

Recently, the polar codes under SCL decoding are found to be capable of achieving the same or even better performance than turbo codes or low- density parity-check codes with the help of cyclic redundancy check (CRC) codes.

An author shows that the CRC-aided SCS decoder can also achieve the same performance. Therefore, polar codes under these CRC-aided decoding schemes can be competitive candidates in future communication systems.

- **In this paper,**

1) SC decoding, SCL decoding and SCS decoding are described.

2) By combining the principles of SCL and SCS, a new decoding algorithm called the successive cancellation hybrid (SCH) decoding is proposed.

3) By reducing unnecessary path searching operations, a pruning technique is proposed.

## II. Background

### *A. Notation*

We denote

- An N-dimensional vector $v_1^N$, $v_1^N = (v_1, v_2, ..., v_N)$.
- A subvector $v_i^j$, $v_i^j = (v_i, v_{i+1}, ..., v_{j-1}, v_j)$ of $v_1^N$, $1 \le i \le j \le N$.
- A subvector $v_{1,o}^j$ with odd indices $(v_k : 1 \le k \le j; k \text{ odd})$.
- A subvector $v_{1,e}^j$ with even indices $(v_k : 1 \le k \le j; k \text{ even})$.

  e.g) $v_1^5 = (5,4,6,2,1)$ , we have $v_2^4 = (4,6,2)$, $v_{1,e}^5 = (4,2), v_{1,o}^5 = (5,6,1)$.

*B. Polar Coding and SC decoding*

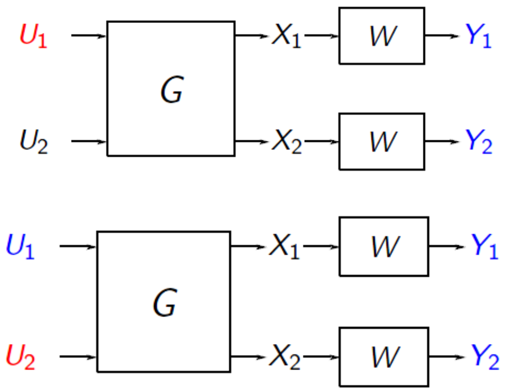
Consider a binary input output symmetric memoryless channel with output probability density function $W(y\,|\,x)$, $y \in Y$, $x \in \mathbb{F}_2$. It can be transformed into a vector channel given by $W_n\left(y_1^n\,|\,u_1^n\right) = W^n\left(y_1^n\,|\,u_1^n G_n\right)$, where $W^n\left(y_1^n\,|\,x_1^n\right) = \prod_{i=1}^{n} W\left(y_i\,|\,x_i\right)$, $G_n = B_s F^{\otimes s}$, $n = 2^s$, $F = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, $\otimes s$ denotes $s$-times Kronecker product of a matrix with itself, and $B_s$ is a bit reversal permutation matrix.

The vector channel can be further decomposed into equivalent subchannels

$$W_N^{(i)}\left(y_1^N, u_1^{i-1}\,|\,u_i\right) = \sum_{u_{i+1}^N} \frac{1}{2^{N-1}} W_N\left(y_1^N\,|\,u_1^N\right).$$

**For example)**

According to this, we can write $(W,W) \mapsto \left(W_2^{(1)}, W_2^{(2)}\right)$ for any given B-DMC $W$.




$$W_2^{(1)}\left(y_1^2\,|\,u_1\right) \triangleq \sum_{u_2} \frac{1}{2} W_2\left(y_1^2\,|\,u_1^2\right)$$
$$= \sum_{u_2} \frac{1}{2} W\left(y_1\,|\,u_1 \oplus u_2\right) W\left(y_2\,|\,u_2\right)$$
$$W_2^{(2)}\left(y_1^2, u_1\,|\,u_2\right) \triangleq \frac{1}{2} W_2\left(y_1^2\,|\,u_1^2\right)$$
$$= \frac{1}{2} W\left(y_1\,|\,u_1 \oplus u_2\right) W\left(y_2\,|\,u_2\right)$$

In transmitting a binary message block of $K$ bits, the $K$ most reliable polarize channels $\left\{W_N^{(i)}\right\}$ with indices $i \in \mathbb{I}$ are selected to carry these information bits; the others are used to transmit a fixed binary sequence called frozen bits. The index set $\mathbb{I} \subseteq \{1, 2, \dots, N\}$ is called the information set, and $|\mathbb{I}| = K$. The complement set of $\mathbb{I}$ is called the frozen set and is denoted by $\mathbb{F}$.

Given $y_1^n$ and estimates $\hat{u}_1^{i-1}$ of $u_1^{i-1}$, the SC decoding algorithm attempts to estimate $u_i$. This can be implemented by computing the following way:

$$\hat{u}_i = \begin{cases} h_i\left(y_1^N, \hat{u}_1^{i-1}\right), & i \in \mathbb{I} \\ u_i, & i \in \mathbb{F} \end{cases}$$

where

$$h_i\left(y_1^n, u_1^{i-1}\right)=\begin{cases}0, \text{ if } \dfrac{W_n^{(i)}\left(y_1^n, u_1^{i-1} \mid u_i=0\right)}{W_n^{(i)}\left(y_1^n, u_1^{i-1} \mid u_i=1\right)} \geq 1 \\ 1, \text{ otherwise}\end{cases}.$$

## III. Improved Successive Cancellation Decoding

### *A. Unified Description using Code Tree Representation*

The SC decoding can be represented as a path search procedure on a code tree. For a polar code with code length *N*, the corresponding code tree $\mathbb{T}$ is a perfect binary tree. Specifically, $\mathbb{T}$ can be represented as a 2-tuple $(\mathbb{V}, \mathbb{E})$, where $\mathbb{V}$ and $\mathbb{E}$ denote the set of nodes and the set of edges, respectively, and $|\mathbb{V}|=2^{N+1}-1, |\mathbb{E}|=2^{N+2}-2$.

An *i*-length decoding path $\{e_1, e_2, \ldots, e_i\}$ consists of *i* edges, with $e_j \in \mathbb{E}_j$, $j \in \{1, 2, \ldots, i\}$. A vector $v_1^i$ is used to depict the above decoding path, where $v_i$ corresponds to the binary label of edge $e_i$. The reliability of a decoding path $v_1^i$ can be measured using a posteriori probabilities (APPs)

$$P_N^{(i)}\left(v_1^i \middle| y_1^N\right)=\frac{W_N^{(i)}\left(y_1^N, v_1^{i-1} \middle| v_i\right)}{2P\left(y_1^N\right)}.$$

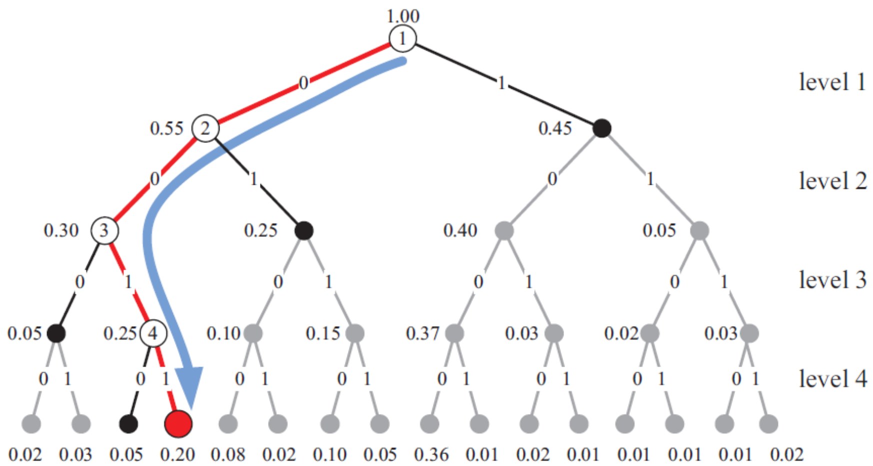



Figure 1. An example of code tree for *N*=4 and *K*=4. The bold branches show the decoding path of SC with $u_1^4=0011$.

There are recursive expressions for the APPs as follows,

$$P_N^{(2i-1)}\left(v_1^{2i-1}\middle|y_1^N\right)=\sum_{v_{2i}\in\{0,1\}}P_{N/2}^{(i)}\left(v_{1,o}^{2i}\oplus v_{1,e}^{2i}\middle|y_1^{N/2}\right)P_{N/2}^{(i)}\left(v_{1,e}^{2i}\middle|y_{N/2+1}^{N}\right),$$
$$P_N^{(2i)}\left(v_1^{2i}\middle|y_1^N\right)=P_{N/2}^{(i)}\left(v_{1,o}^{2i}\oplus v_{1,e}^{2i}\middle|y_1^{N/2}\right)P_{N/2}^{(i)}\left(v_{1,e}^{2i}\middle|y_{N/2+1}^{N}\right).$$

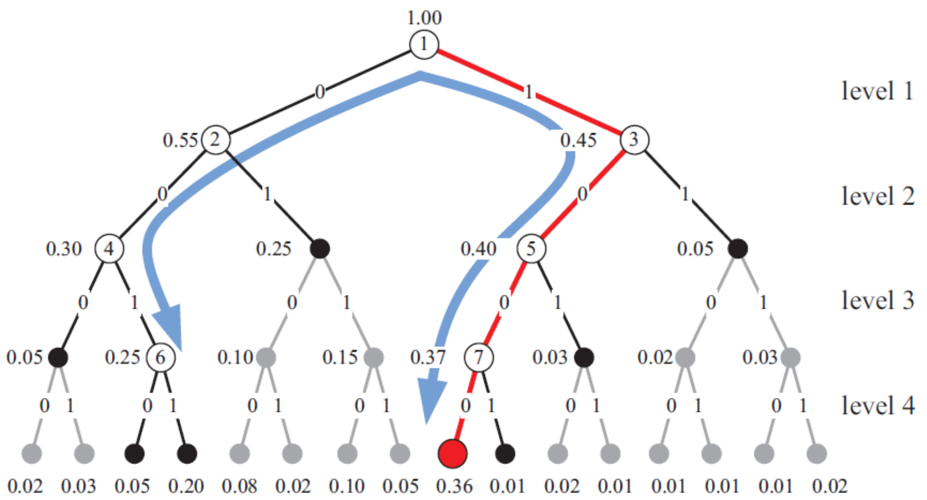
The SC decoding of polar codes can be regarded as a greedy search algorithm on the code tree. Among the two branches in each level, only the one with the larger probability is selected for further processing.

In the example, four path extensions are required, that is, one for each level. However, the decoding path obtained by SC is not guaranteed to be the most probable one. As shown in the example, the one labeled "1000" has the largest probability of all the $N$-length paths, but it fails in the competition at the first level.

The performance of SC is limited by the bit by bit decoding strategy. Whenever a bit is wrongly determined, correcting it in the future decoding procedure becomes impossible.

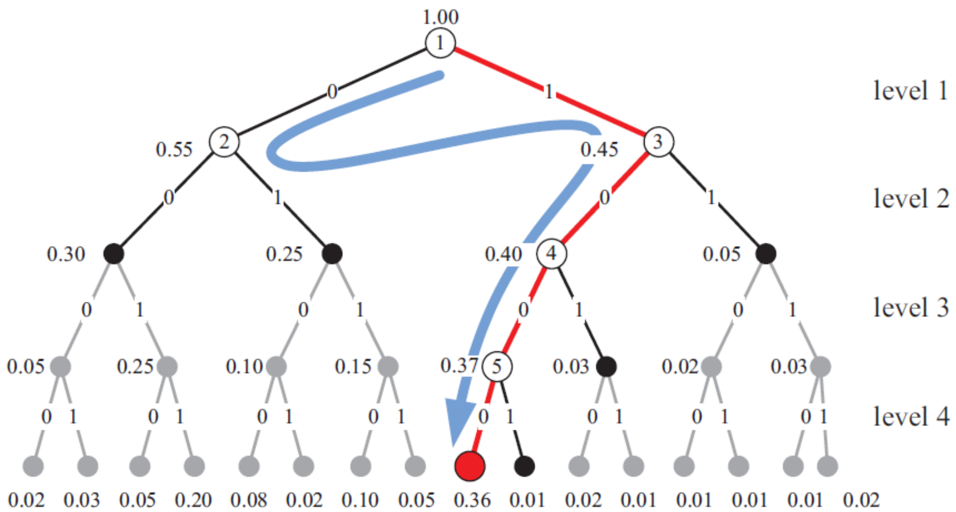
*B. SC List and SC Stack Decoding Algorithms*

*1) SCL Decoding*




Unlike SC where only one path is reserved after processing at each level, SCL allows a maximum of $L$ candidate paths to be further explored at the next level. SCL can be regarded as a breadth-first search algorithm on the code tree $\mathbb{T}$ with a search width $L$. At each level, SCL doubles the number of candidates by appending a bit 0 or a bit 1 to each of the candidate paths. It then selects a maximum of $L$ ones with the largest metrics and stores them in a list for further processing at the next level.

*2) SCS Decoding*




The SCS decoder uses an ordered stack S to store the candidate paths and tries to find the optimal estimation by searching along the best candidate in the stack. Whenever the top path in the stack that has the largest path metric reaches length $N$, the decoding process stops and outputs this path. Unlike the candidate paths in the list of SCL, which always have the same length, the candidates in the stack of SCS have difference lengths.

*3) Complexity*

As for the implementation aspect, a space-efficient structure is suggested in [15] to implement the SC decoder, and the time and space complexities are $O(NlogN)$ and $O(N)$, respectively.

The SCL decoding algorithm maintains $L$ decoding paths simultaneously, and each path consumes an $O(N)$ space; thus, its space complexity of SCL is $O(LN^2)$ and time complexity is $O(LN \log N)$.

In the worst cases, the time and space complexities of SCS are $O(LN \log N)$ and $O(DN)$, respectively.

### *C. Hybrid SCL and SCS*

As the name suggests, SCH is a hybrid of SCL and SCS. SCH has two working modes called on-going and waiting. Initially, the SCH decoder works in the on-going mode by searching along the best candidate path using an ordered stack, which is similar to SCS. However, when the number of candidate paths in the stack is about to achieve the maximum depth, SCH stops searching forward and switches to the waiting mode. In the waiting mode, SCH turns to extend the shortest path in the stack until all the candidate paths in the stack have the same length. The processing in the waiting mode is somewhat similar to that in that in SCL, and the number of paths in the stack is decreased to the search width L. Then, SCH switches back to the on-going mode. Fig. 5 shows a graphic illustration of this process. This decoding procedure goes on until an $N$ length path appears at the top of the stack.

The time and space complexities of the SCH decoding in the worst cases are $O(LN \log N)$ and $O(DN)$, respectively. the time complexity of SCH is usually less than that of SCL but is slightly greater than that of SCS.

### *D. Tree Pruning Technique*

When searching the path on the code tree, the candidate paths with extremely small metrics and their descendants will hardly have the chance to be reserved for the future process. Obviously, computation and storage overhead can be reduced by dropping these paths.

Given a search width $L$, let $L^{(i)}$ denote the set of all the possible $i$-length candidate paths, and $\left|L^{(i)}\right| = L$ for all $i \in \{1, 2, ..., N\}$. An additional vector $a_1^N$ is used to record the pruning reference, where $a_i$ is the largest metric of all the visited paths in $L^{(i)}$, i.e.,

$$a_i = \max_{v_1^i \in L^{(i)}} M_N^{(i)}\left(v_1^i \middle| y_1^N\right).$$

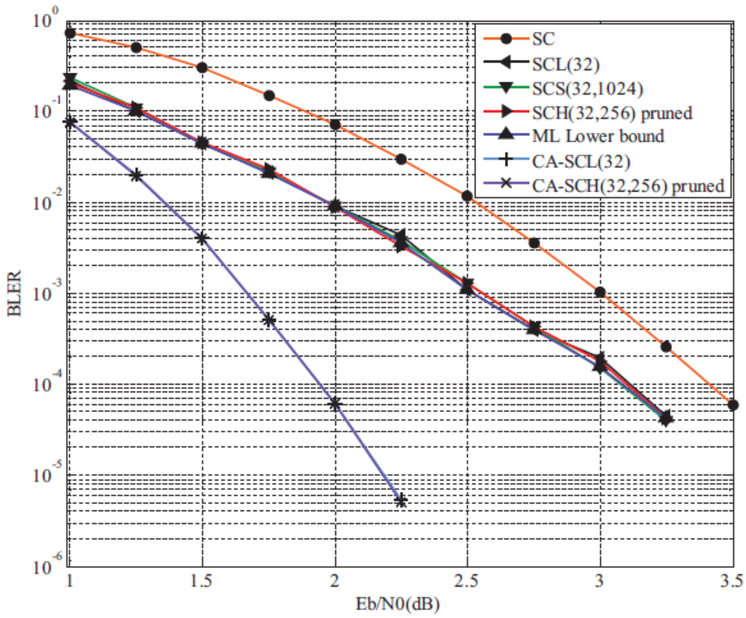
We introduce a new parameter called the pruning threshold $\tau$, and $\tau \geq 1$. During the processing at level-$i$ in the code tree, an $i$-length path with a metric smaller than $a_i - \ln(\tau)$ is dropped directly, i.e., these deleted paths have the following metrics:

$$M_N^{(i)}\left(v_1^i \middle| y_1^N\right) < a_i - \ln(\tau).$$

## IV. NUMERICAL RESULTS

The block error rate (BLER) performance and complexity are analyzed via numerical results over binary-input AWGN channels.

For convenience, we use SCL($L$) to denote the SCL decoder with search width $L$. Let SCS($L$,$D$) and SCH($L$,$D$) denote the SCS and SCH decoders with search width $L$ and maximum stack depth $D$, respectively. All the codes used in this section have code length $N = 1024$ and code rate $R = 1/2$. For all the pruned ISC decoding schemes, $\tau = 1.6 \times 10^9$.




As the figure shows, the curves of SCL, SCS, and the pruned SCH are overlapped. Similar to SCL and SCS, the pruned SCH under this configuration can also achieve a performance that is very close to that of ML. For the CRC-aided decoding schemes, the performance of the CRC-aided SCL (CA-SCL) decoding scheme and the CRC-aided SCH (CA-SCH) decoding scheme with pruning are almost the same.

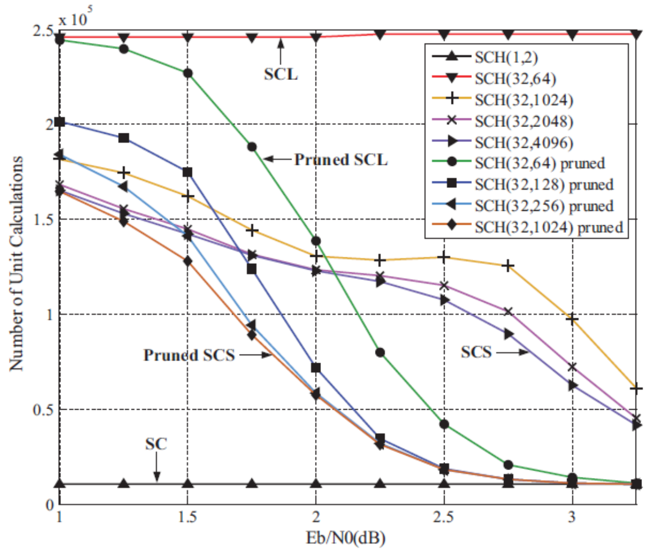



Figure 2. Average time complexities of SCH with different configurations.

The SCH decoding without and with pruning are equivalent to the conventional SCL and pruned SCL, respectively. SCH with D = 4096 and pruned SCH with D = 1024 are equivalent to the conventional SCS and pruned SCS, respectively, because all computations are taken in the on-going mode.

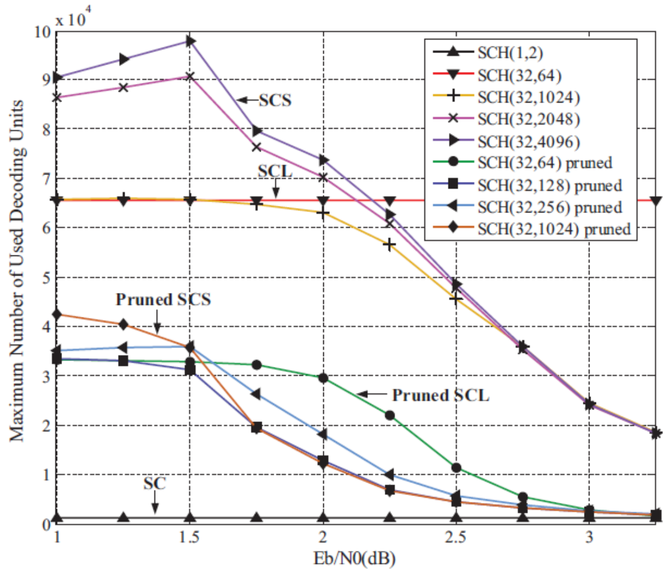



Figure 3. Average space complexities of SCH with different configurations.

The space complexity is significantly reduced by the pruning technique in the moderate and high SNR regimes.

## V. Conclusion

The SC decoding algorithm of polar codes and its improved versions, namely, SCL and SCS, are restated as path search procedures on the code tree of polar codes. By combining the principles of SCL and SCS, a generic ISC decoding scheme called the SCH decoding is proposed. This proposed scheme can provide a flexible configuration when the time and space complexities are limited. To avoid unnecessary path searching, a pruning technique suitable for all three ISC decoding schemes is proposed. In the moderate and high SNR regime, the pruned ISC decoders can approach the performance of ML decoding with the time and space complexities very close to those of SC.