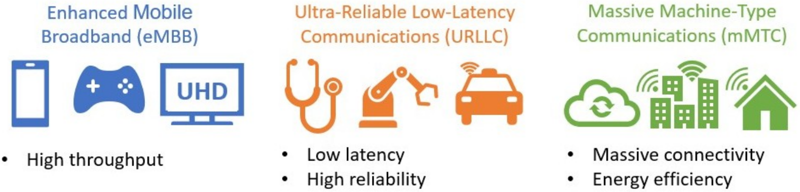
Fast Thresholded SC-Flip Decoding of Polar Codes

Furkan Ercan* and Warren J. Gross
Integrated Systems for Information Processing (ISIP) Lab
McGill University
Montréal, Québec, Canada

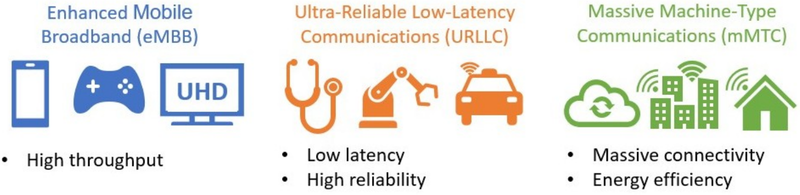
June 8-11, 2020

5G Use Cases



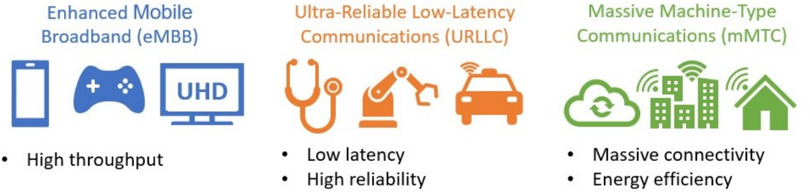
- 5G prioritizes various targets based on the use case.
-
-
-

5G Use Cases



- 5G prioritizes various targets based on the use case.
- Polar codes provably achieve channel capacity.
- They are involved in 5G eMBB control channel.
-

5G Use Cases



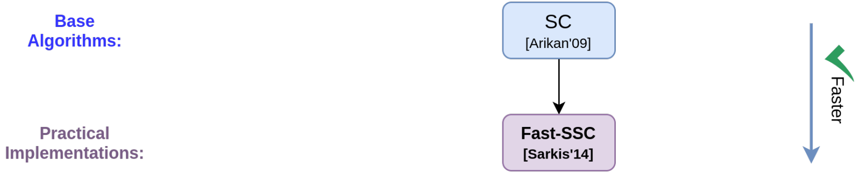
- 5G prioritizes various targets based on the use case.
- Polar codes provably achieve channel capacity.
- They are involved in 5G eMBB control channel.
- Currently, polar codes are being evaluated for other use cases.

An Overview of SC Algorithms

Successive Cancellation (SC) Decoding

- ✓ Simple encoding/decoding
- ✗ Mediocre performance at practical lengths
- ✗ Sequential, long latency

An Overview of SC Algorithms

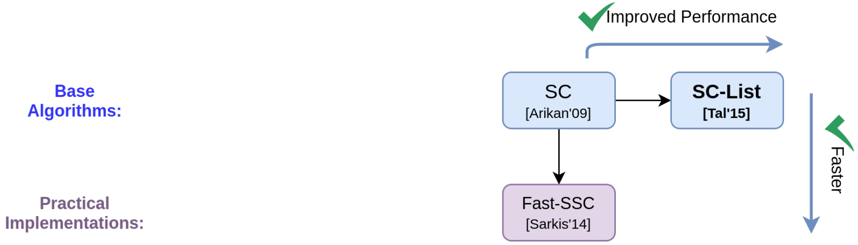


Fast-SSC Decoding

✓ $\approx 10\times$ less latency

- No error correction performance degradation

An Overview of SC Algorithms



SC-List (SCL) Decoding

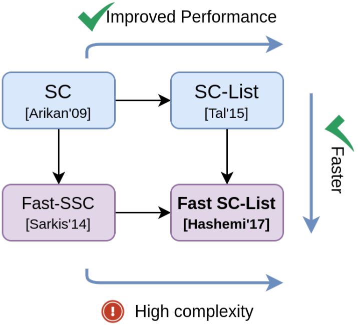
✓ Improved performance

✗ Increased complexity

An Overview of SC Algorithms

Base Algorithms:

Practical Implementations:

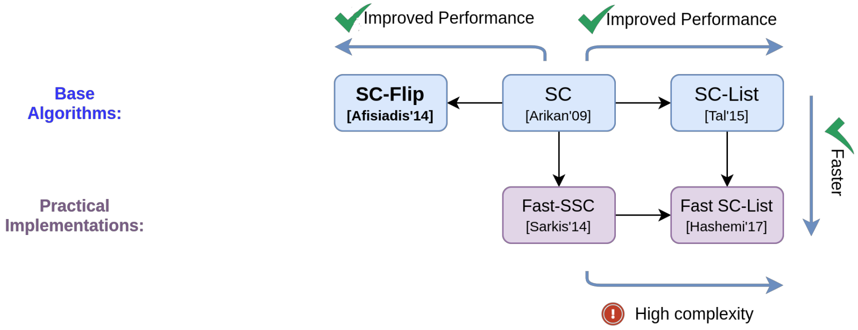


SC-List (SCL) Decoding

✓ Improved performance

✗ Increased complexity

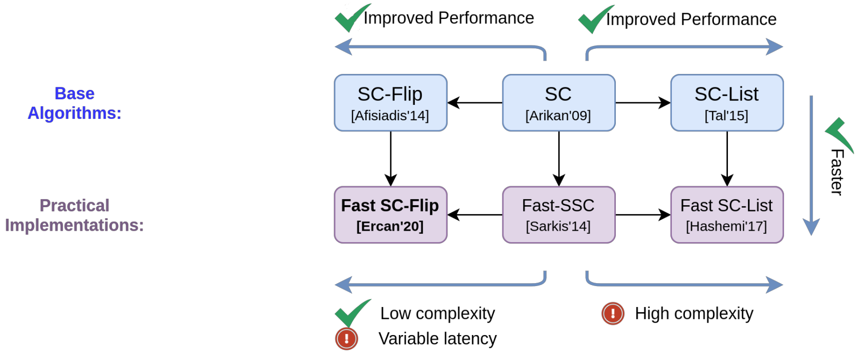
An Overview of SC Algorithms



SC-Flip (SCF) Decoding

- ✓ Some improved performance
- ✓ Low complexity
- ✗ Variable latency

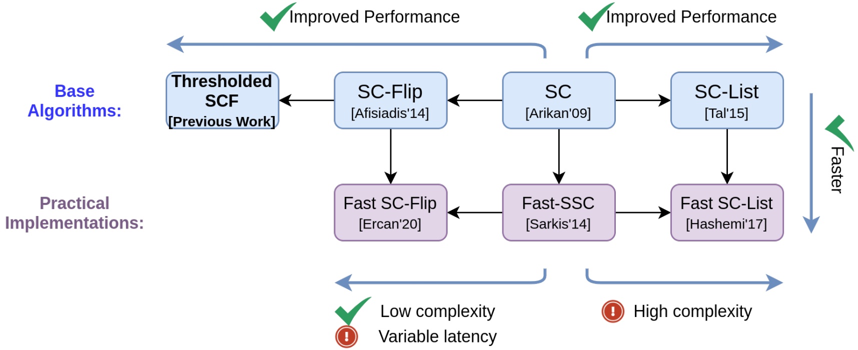
An Overview of SC Algorithms



SC-Flip (SCF) Decoding

- ✓ Some improved performance
- ✓ Low complexity
- ✗ Variable latency

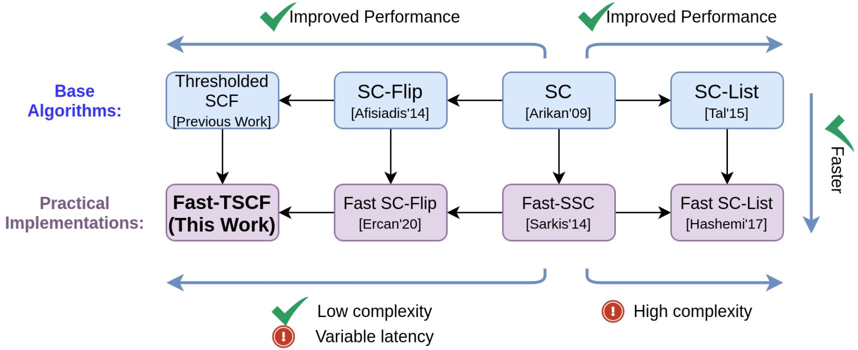
An Overview of SC Algorithms



Thresholded SCF (TSCF) Decoding

- ✓ Better improved performance
- ✓ Lower complexity
- ✗ A lot of precomputations

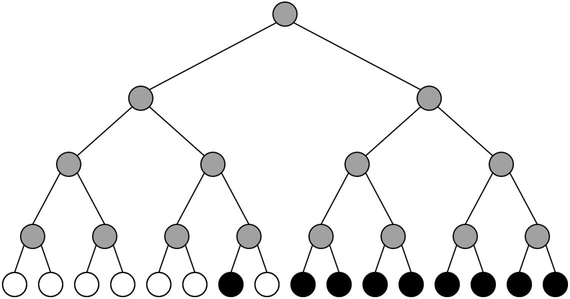
An Overview of SC Algorithms

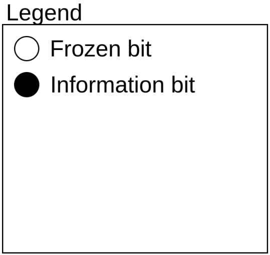


This Work

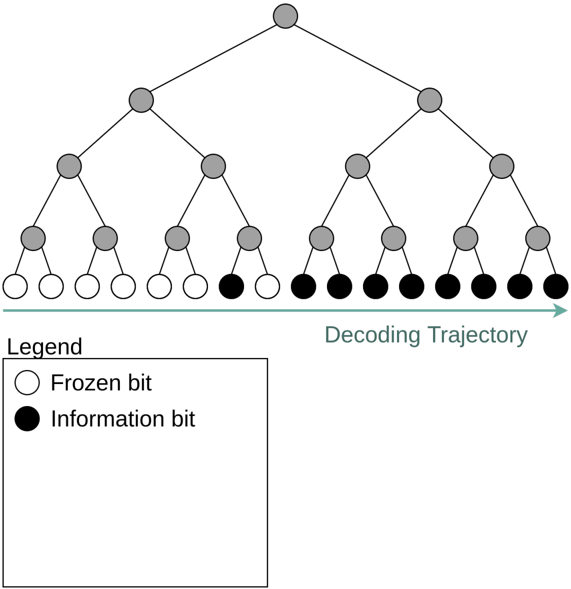
- ✓ No precomputations
- ✓ Introduce fast decoding techniques
- ✓ Hardware implementation

SC-Flip (SCF) Decoding

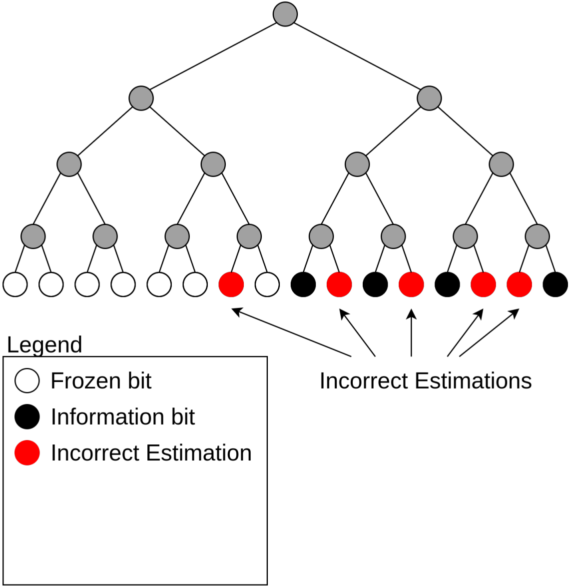




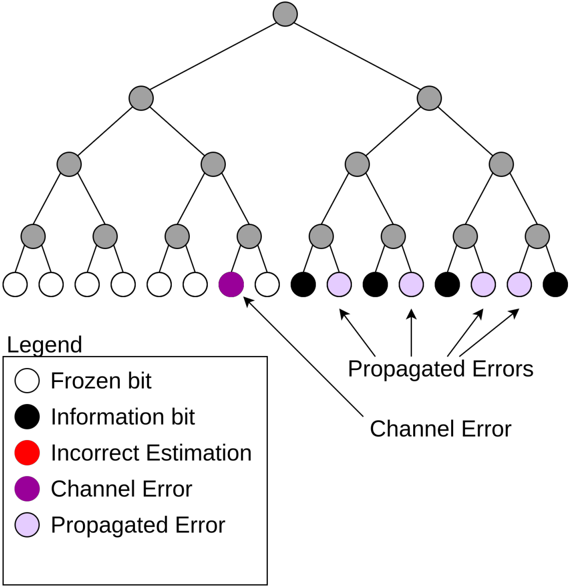
SC-Flip (SCF) Decoding



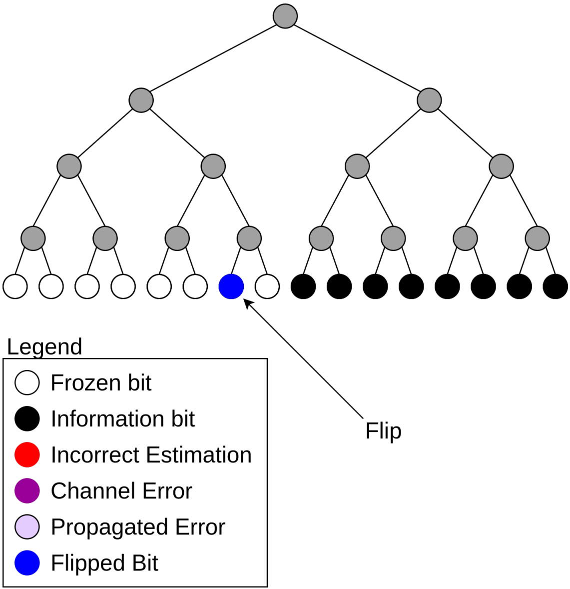
SC-Flip (SCF) Decoding



SC-Flip (SCF) Decoding



SC-Flip (SCF) Decoding



Problems with the SCF Algorithm

- Metric for SCF for node index i: $|L_i|$ where L is LLR.

Problems with the SCF Algorithm

- Metric for SCF for node index i: $|L_i|$ where L is LLR.
- Performance improvement of SCF is limited:
  - Metric cannot distinguish channel errors from propagated errors.

Thresholded SC-Flip (TSCF) Decoding

Thresholded SC-Flip (TSCF) algorithm is an impvovement over SCF decoding:

- The search for bit-flipping is simplified by introducing a **critical set**.
  - Constructed empirically (precomputations)
  - Reduced search effort → reduced complexity

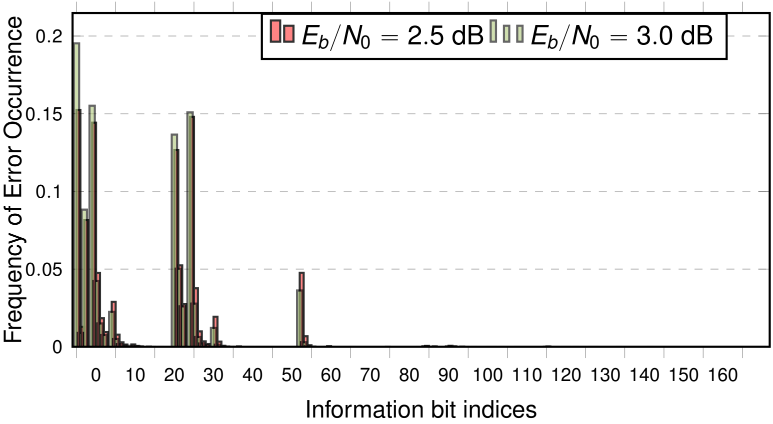
Thresholded SC-Flip (TSCF) Decoding

Thresholded SC-Flip (TSCF) algorithm is an impvovement over SCF decoding:

- The search for bit-flipping is simplified by introducing a **critical set**.
  - Constructed empirically (precomputations)
  - Reduced search effort → reduced complexity
- An **LLR threshold** can filter erroneous indices efficiently.
  - Constructed empirically (precomputations)
  - Efficient index identification → improved performance

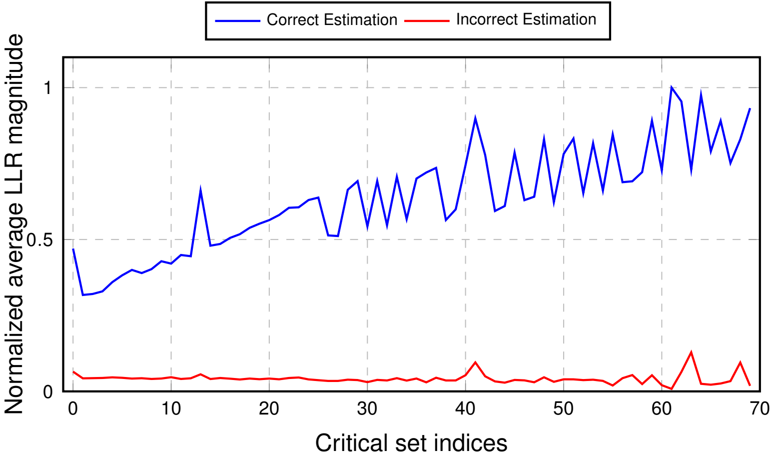
Demonstration: Critical Set

- Example: $PC(N, K) = PC(1024, 170)$



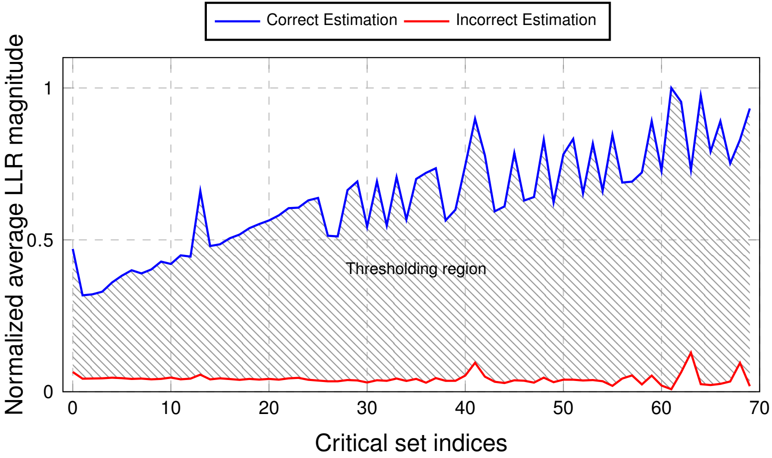
Demonstration: LLR Threshold

- Example: $PC(N, K) = PC(1024, 170)$

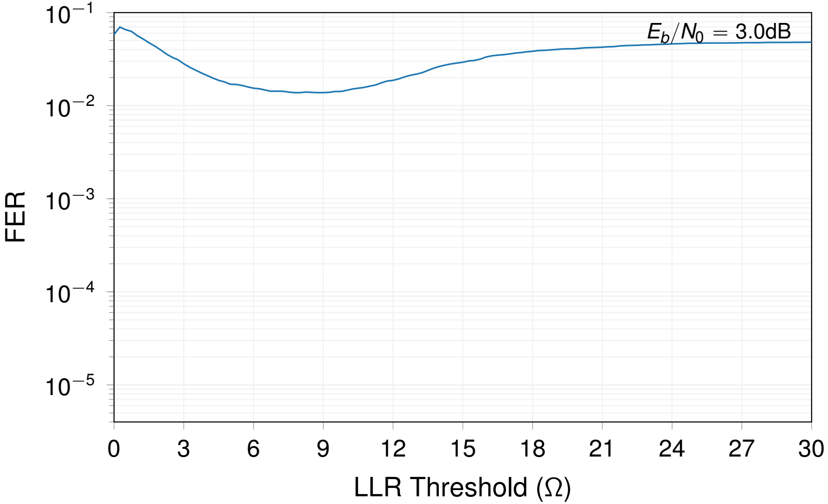


Demonstration: LLR Threshold

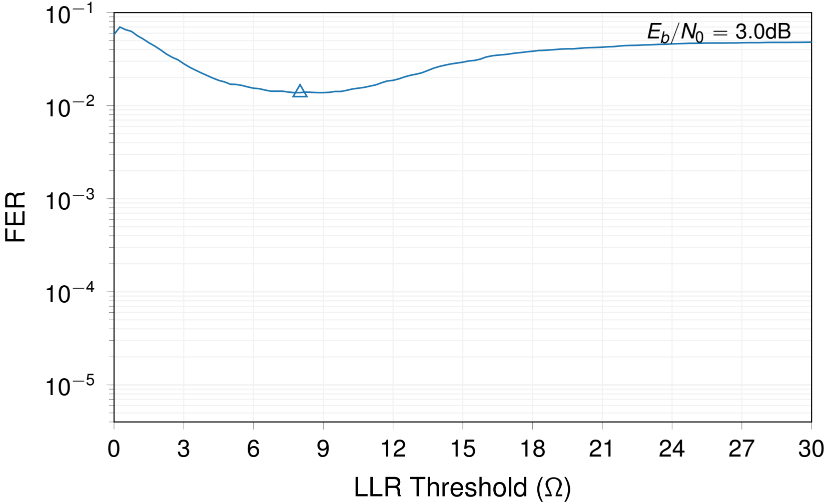
- Example: $PC(N, K) = PC(1024, 170)$



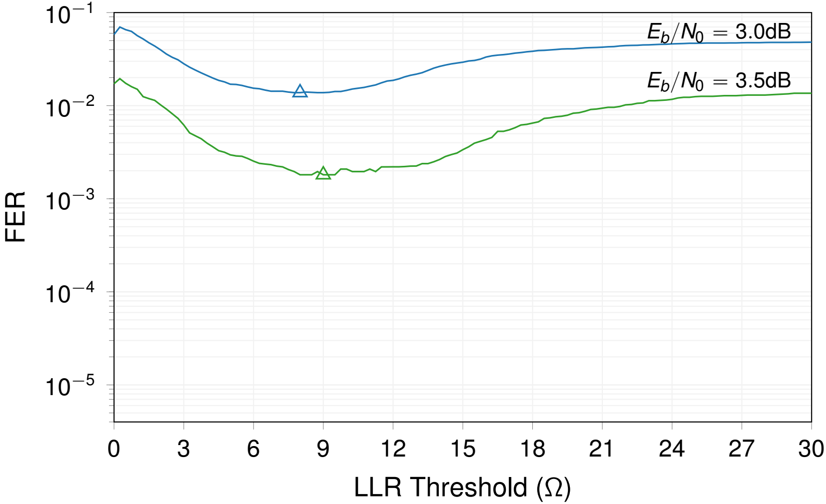
Threshold Sweep for Best Performance



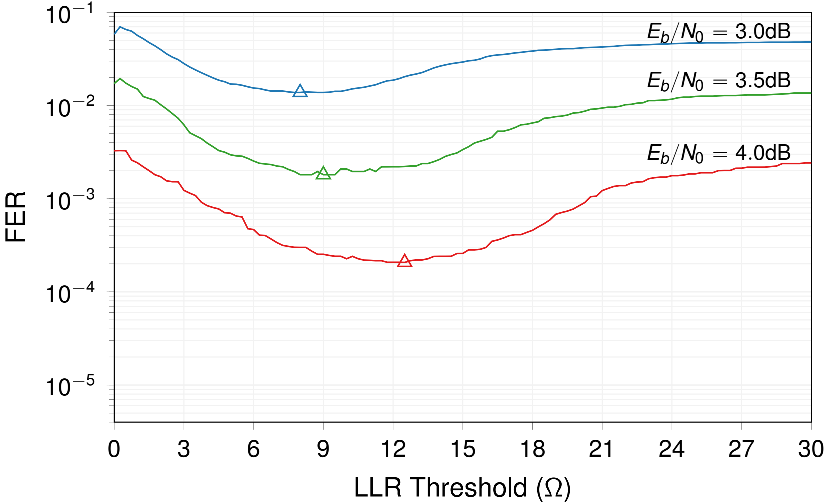
Threshold Sweep for Best Performance



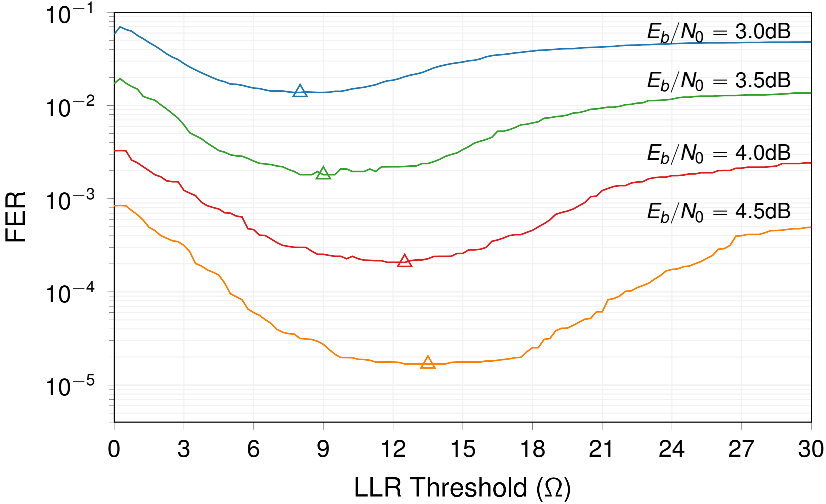
Threshold Sweep for Best Performance



Threshold Sweep for Best Performance

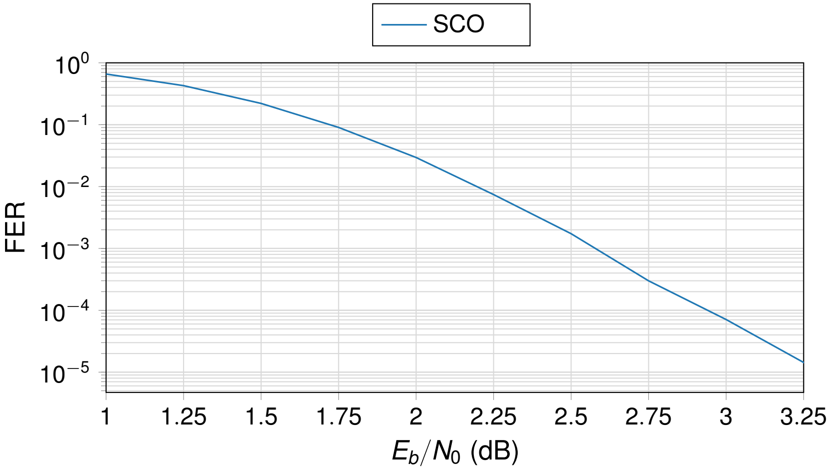


Threshold Sweep for Best Performance



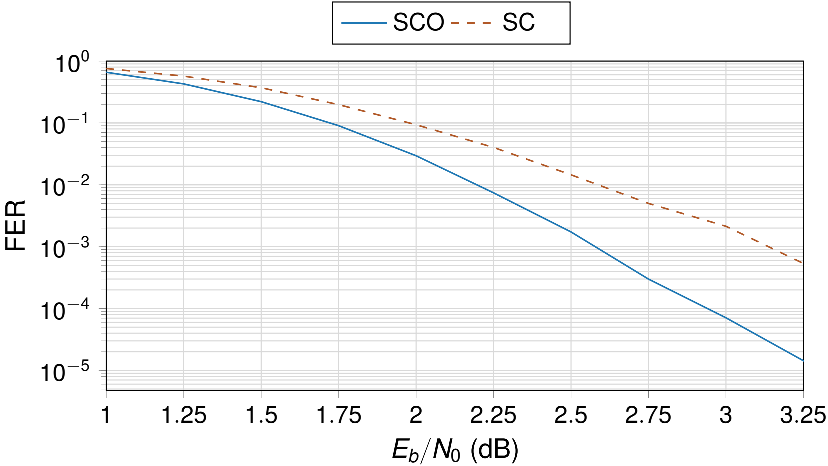
TSCF Performance

- Example: $PC(1024, 512)$, 16 bit CRC, $T_{\max} = 10$.
- Ω for TSCF is optimized for $E_b/N_0 = 2.5$ dB.



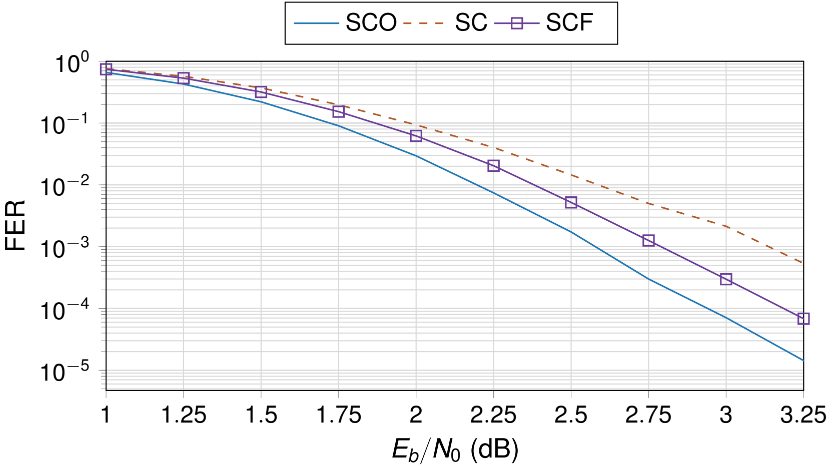
TSCF Performance

- Example: $PC(1024, 512)$, 16 bit CRC, $T_{\max} = 10$.
- Ω for TSCF is optimized for $E_b/N_0 = 2.5$ dB.



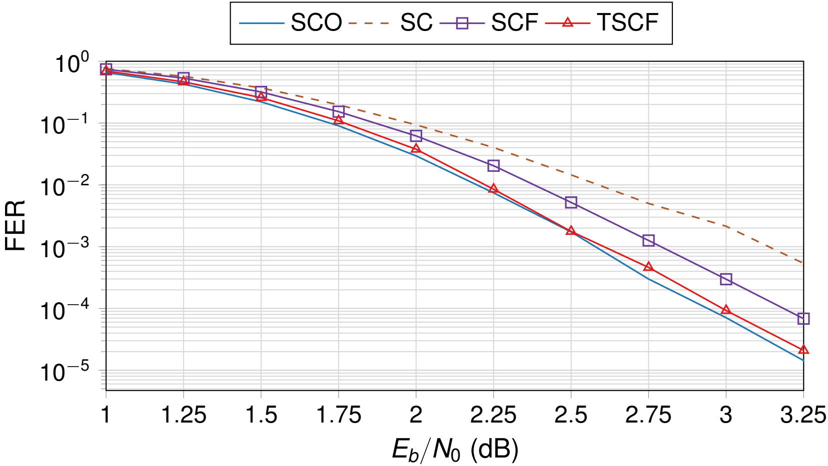
TSCF Performance

- Example: $PC(1024, 512)$, 16 bit CRC, $T_{\max} = 10$.
- Ω for TSCF is optimized for $E_b/N_0 = 2.5$ dB.



TSCF Performance

- Example: $PC(1024, 512)$, 16 bit CRC, $T_{\max} = 10$.
- Ω for TSCF is optimized for $E_b/N_0 = 2.5$ dB.



TSCF Algorithm - Pros and Cons

Pros:

- ✓ Reduced search complexity
- ✓ Improved decoding performance

TSCF Algorithm - Pros and Cons

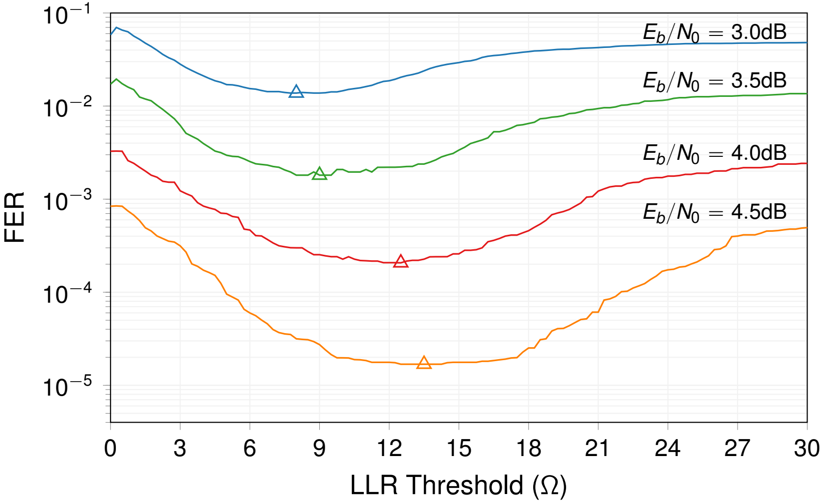
Pros:

- ✓ Reduced search complexity
- ✓ Improved decoding performance

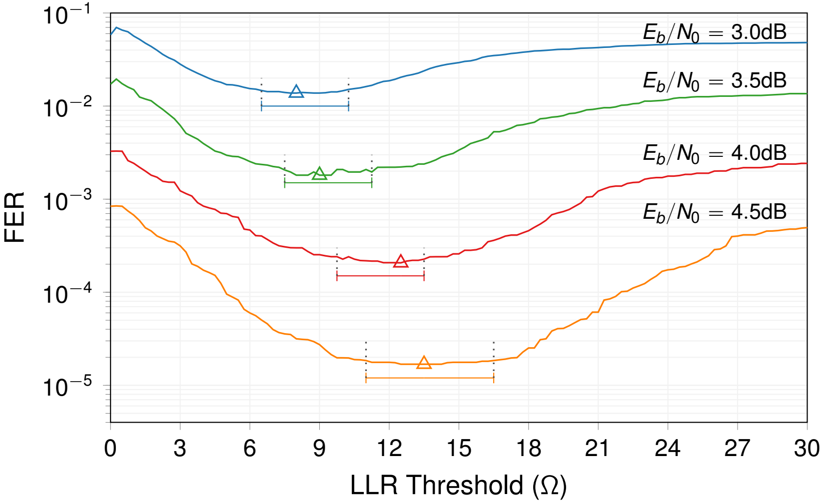
Cons:

- ✗ Precomputations for LLR threshold
- ✗ Precomputations for critical set
- ✗ No fast decoding techniques
- ✗ No practical implementation

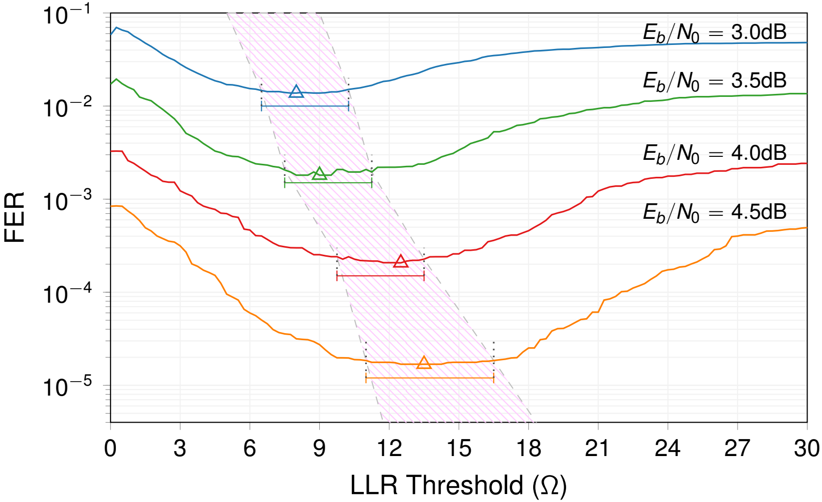
A New Approach to LLR Thresholding



A New Approach to LLR Thresholding

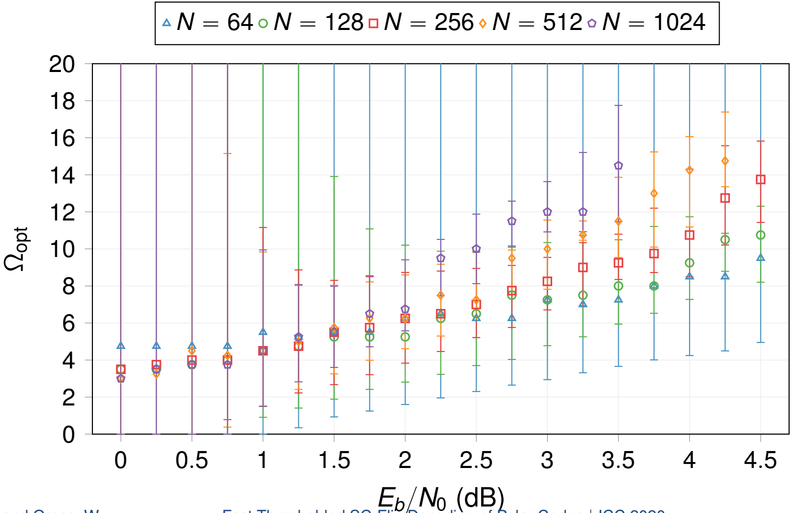


A New Approach to LLR Thresholding



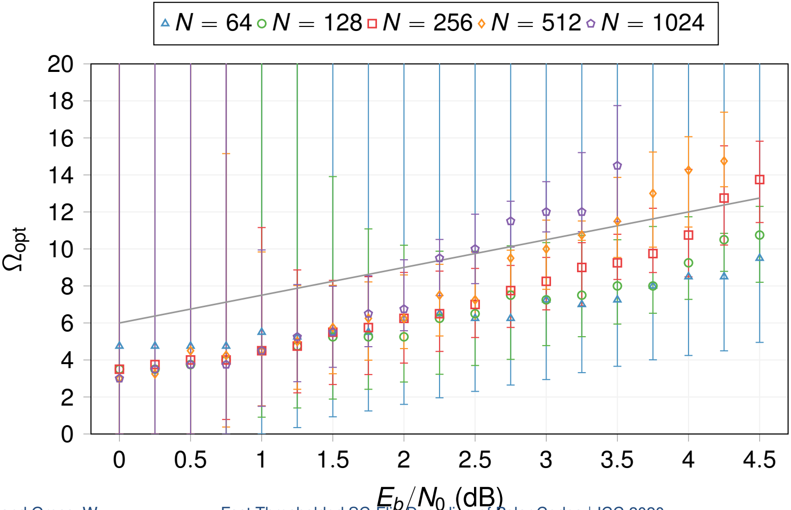
LLR Threshold Regression

- 5G polar codes

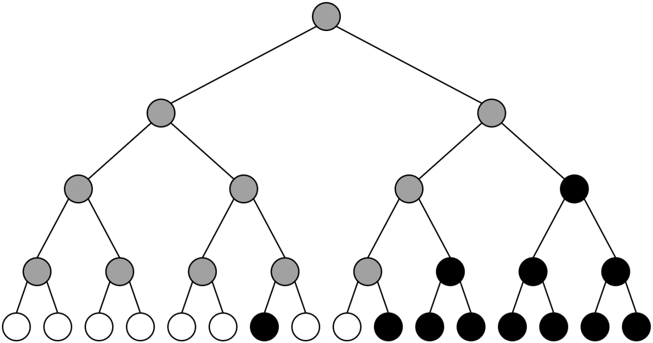


LLR Threshold Regression

- $\Omega_{(\text{approx})} = 2 \times E_b/N_0(dB) + 6$

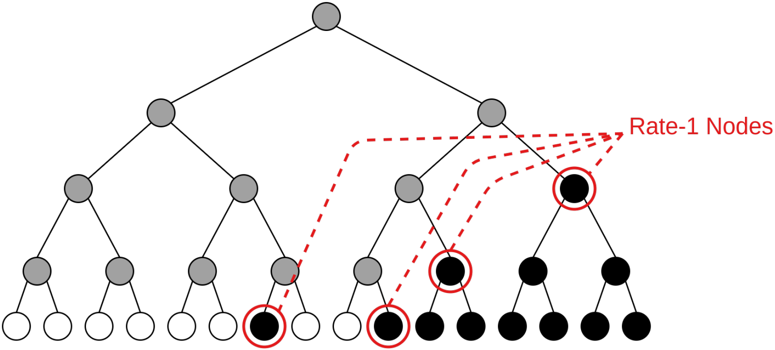


Correlation of Critical Sets



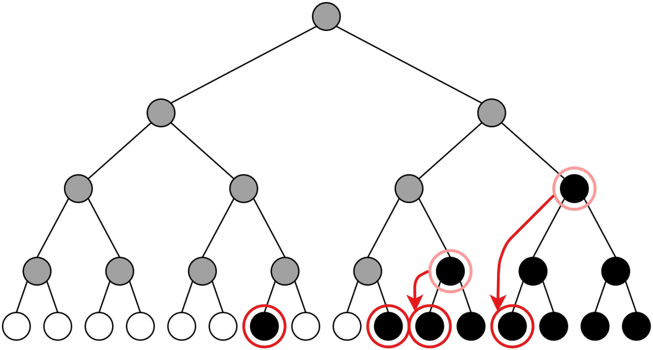
[1] Z. Zhang, K. Qin, L. Zhang and G. T. Chen, "Progressive Bit-Flipping Decoding of Polar Codes: A Critical-Set Based Tree Search Approach," in IEEE Access, vol. 6, pp. 57738-57750, 2018.

Correlation of Critical Sets



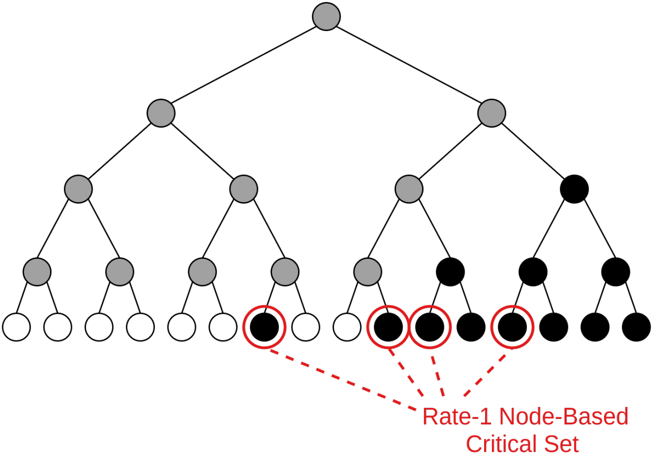
[1] Z. Zhang, K. Qin, L. Zhang and G. T. Chen, ”Progressive Bit-Flipping Decoding of Polar Codes: A Critical-Set Based Tree Search Approach,” in IEEE Access, vol. 6, pp. 57738-57750, 2018.

Correlation of Critical Sets



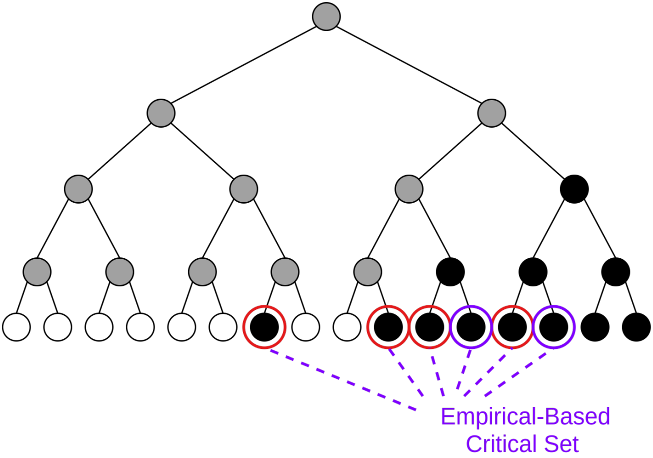
[1] Z. Zhang, K. Qin, L. Zhang and G. T. Chen, ”Progressive Bit-Flipping Decoding of Polar Codes: A Critical-Set Based Tree Search Approach,” in IEEE Access, vol. 6, pp. 57738-57750, 2018.

Correlation of Critical Sets



[1] Z. Zhang, K. Qin, L. Zhang and G. T. Chen, "Progressive Bit-Flipping Decoding of Polar Codes: A Critical-Set Based Tree Search Approach," in IEEE Access, vol. 6, pp. 57738-57750, 2018.

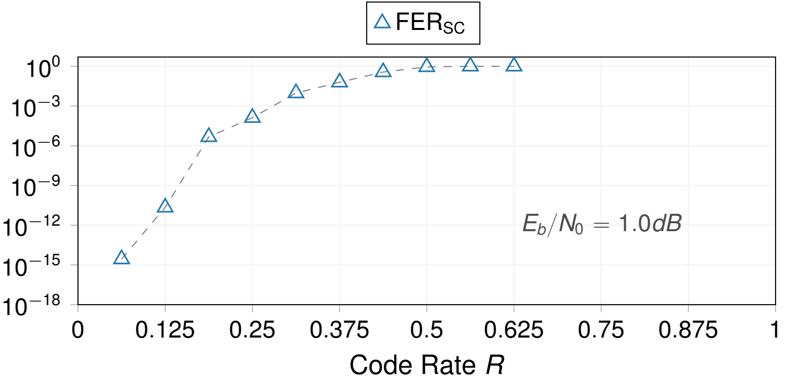
Correlation of Critical Sets



[1] Z. Zhang, K. Qin, L. Zhang and G. T. Chen, "Progressive Bit-Flipping Decoding of Polar Codes: A Critical-Set Based Tree Search Approach," in IEEE Access, vol. 6, pp. 57738-57750, 2018.

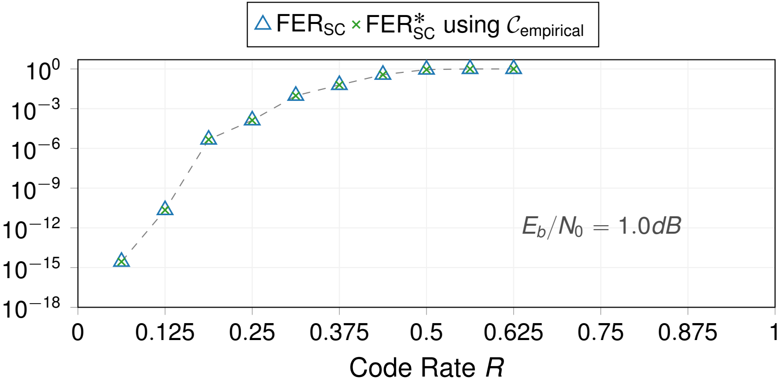
Correlation of Critical Sets

- $\text{FER}_{\text{SC}} = 1 - \left[\prod_{i \in \mathcal{X}} \left(1 - \Pr(\text{error}_i)\right) \right].$
- $\mathcal{X}$ can be information bits, or a critical set $\mathcal{C}$.
- 5G polar code, N=1024.



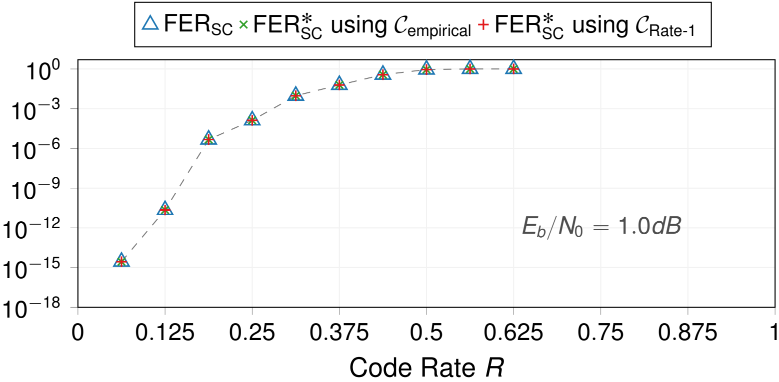
Correlation of Critical Sets

- $\text{FER}_{\text{SC}} = 1 - \left[\prod_{i \in \mathcal{X}} \left(1 - \Pr(\text{error}_i)\right) \right].$
- $\mathcal{X}$ can be information bits, or a critical set $\mathcal{C}$.
- 5G polar code, N=1024.



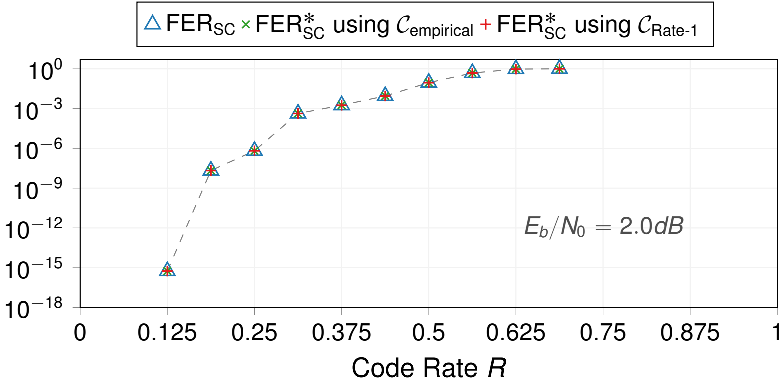
Correlation of Critical Sets

- $\text{FER}_{\text{SC}} = 1 - \left[\prod_{i \in \mathcal{X}} \left(1 - \Pr(\text{error}_i)\right) \right].$
- $\mathcal{X}$ can be information bits, or a critical set $\mathcal{C}$.
- 5G polar code, N=1024.



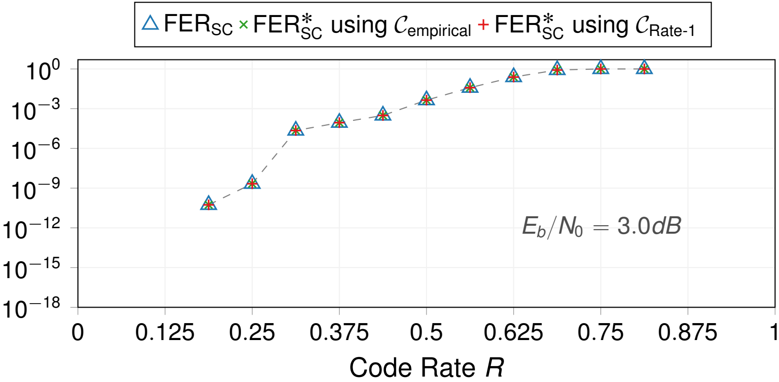
Correlation of Critical Sets

- $\text{FER}_{\text{SC}} = 1 - \left[\prod_{i \in \mathcal{X}} \left(1 - \Pr(\text{error}_i)\right) \right].$
- $\mathcal{X}$ can be information bits, or a critical set $\mathcal{C}$.
- 5G polar code, N=1024.



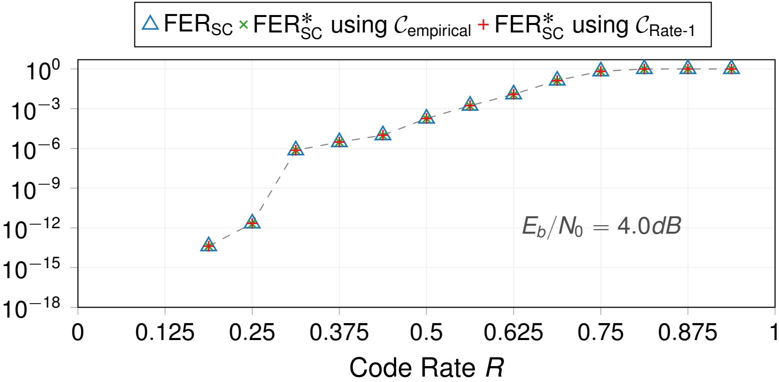
Correlation of Critical Sets

- $\text{FER}_{\text{SC}} = 1 - \left[\prod_{i \in \mathcal{X}} (1 - \Pr(\text{error}_i))\right].$
- $\mathcal{X}$ can be information bits, or a critical set $\mathcal{C}$.
- 5G polar code, N=1024.



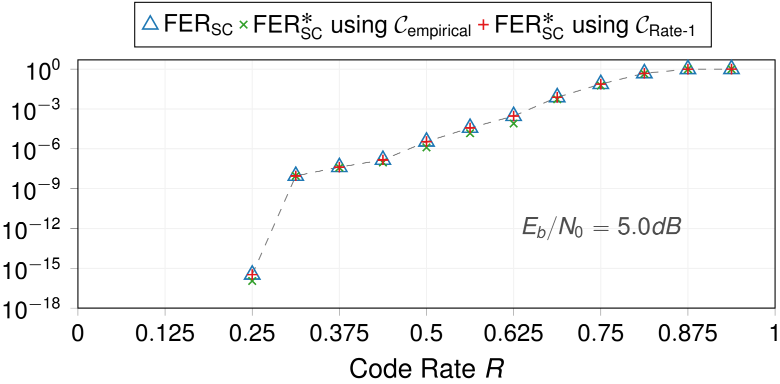
Correlation of Critical Sets

- $\text{FER}_{\text{SC}} = 1 - \left[\prod_{i \in \mathcal{X}} \left(1 - \Pr(\text{error}_i)\right) \right].$
- $\mathcal{X}$ can be information bits, or a critical set $\mathcal{C}$.
- 5G polar code, N=1024.



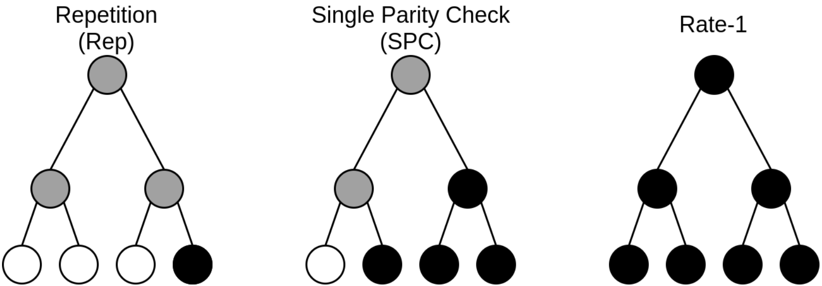
Correlation of Critical Sets

- $\text{FER}_{\text{SC}} = 1 - \left[\prod_{i \in \mathcal{X}} \left(1 - \Pr(\text{error}_i)\right) \right].$
- $\mathcal{X}$ can be information bits, or a critical set $\mathcal{C}$.
- 5G polar code, N=1024.



Fast-TSCF Decoding

- New critical set approach allows for fast decoding.
- Special nodes: Repetition, single parity check (SPC), Rate-1
- Use LLR thresholding at the top of the special nodes.



Decoding of Special Nodes

- Decoding of special nodes for SCF algorithm was implemented previously (Fast-SCF decoding) [1].
- Idea: Use thresholding at the top-node calculations.
- Example: Rate-1 nodes

$$\eta_{\text{Rate-1}} = \begin{cases} \arg\ \min |\alpha^S_{0:N_v-1}|, & \text{if } \min |\alpha^S_{0:N_v-1}| \leq \Omega \\ \varnothing, & \text{otherwise.} \end{cases}$$

[1] F. Ercan, T. Tonnellier, and W. J. Gross, Energy-efficient hardware architectures for fast polar decoders, IEEE Transactions on Circuits and Systems I: Regular Papers, pp. 114, 2019.

Decoding of Special Nodes

- Decoding of special nodes for SCF algorithm was implemented previously (Fast-SCF decoding) [1].
- Idea: Use thresholding at the top-node calculations.
- Example: Rate-1 nodes

Bit-flipping index

$$\eta_{\text{Rate-1}} = \begin{cases} \boxed{\arg\min |\alpha^S_{0:N_v-1}|,} & \text{if } \min |\alpha^S_{0:N_v-1}| \leq \Omega \\ \varnothing, & \text{otherwise.} \end{cases}$$

---

[1] F. Ercan, T. Tonnellier, and W. J. Gross, Energy-efficient hardware architectures for fast polar decoders, IEEE Transactions on Circuits and Systems I: Regular Papers, pp. 114, 2019.

Decoding of Special Nodes

- Decoding of special nodes for SCF algorithm was implemented previously (Fast-SCF decoding) [1].
- Idea: Use thresholding at the top-node calculations.
- Example: Rate-1 nodes

Bit-flipping index

$$\eta_{\text{Rate-1}} = \begin{cases} \arg\min |\alpha^{S}_{0:N_v-1}|, & \text{if } \min |\alpha^{S}_{0:N_v-1}| \leq \Omega \\ \varnothing, & \text{otherwise.} \end{cases}$$

Threshold condition

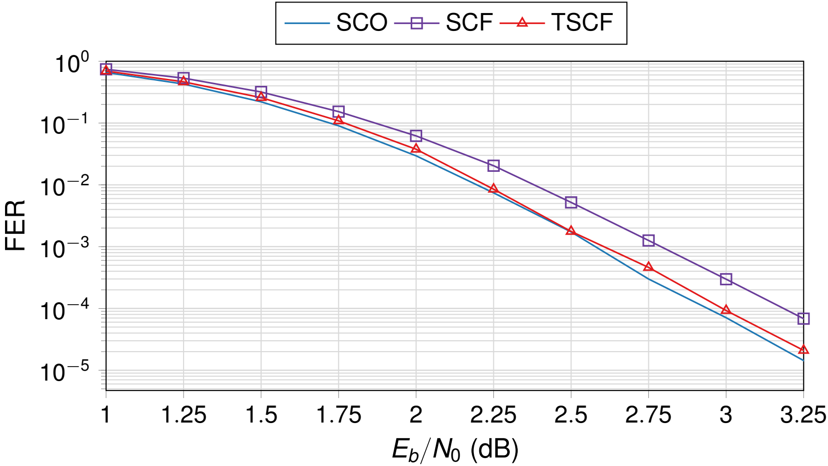
[1] F. Ercan, T. Tonnellier, and W. J. Gross, Energy-efficient hardware architectures for fast polar decoders, IEEE Transactions on Circuits and Systems I: Regular Papers, pp. 114, 2019.

Hardware Implementation

- Fast-SCF decoder is modified to implement the Fast-TSCF decoder:
  - Sorter is not required in TSCF algorithm.
  - Channel estimation is introduced as an input for Ω.
  - All fast decoding techniques are modified with Ω.
- Implemented in VHDL, validated with test benches.

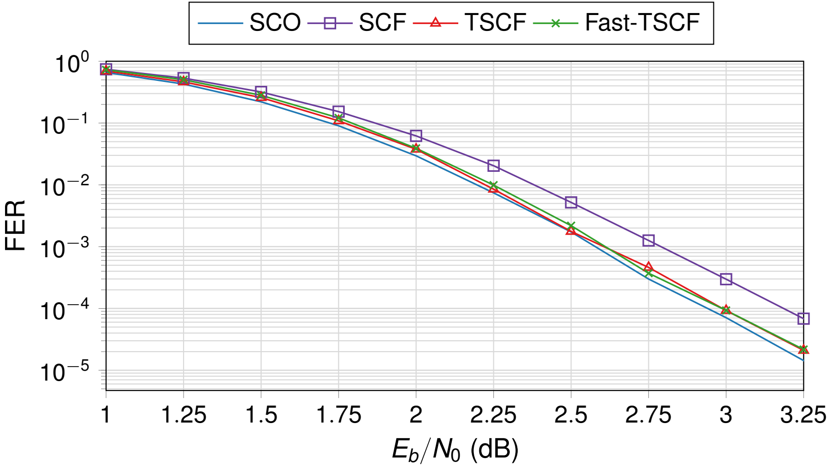
Results: Performance

- $PC(1024, 512)$, 16 bit CRC, $T_{\max} = 10$.



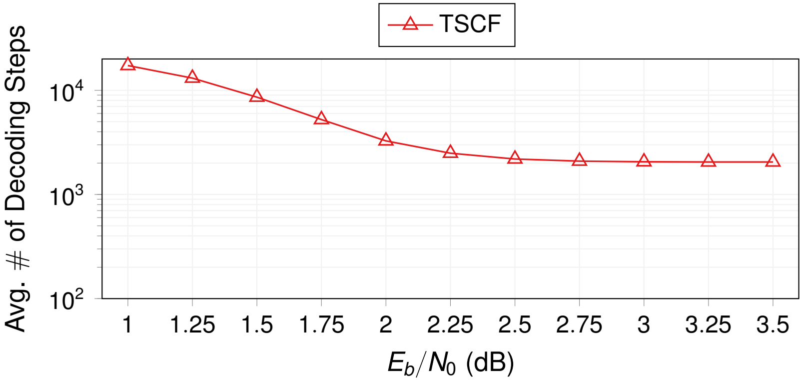
Results: Performance

- $PC(1024, 512)$, 16 bit CRC, $T_{\max} = 10$.



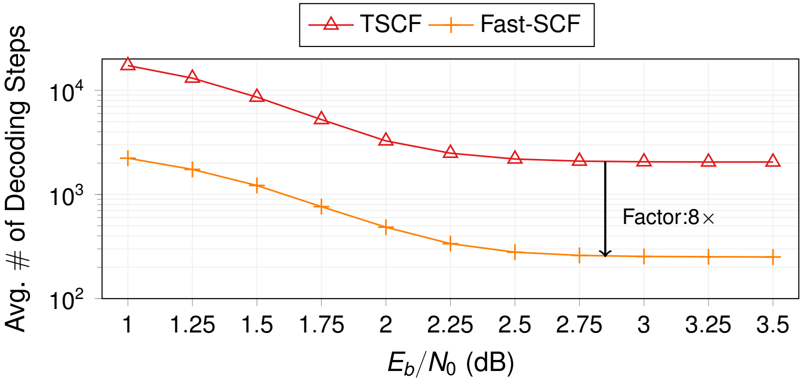
Results: Latency

- $PC(1024, 512)$, 16 bit CRC, $T_{\text{max}} = 10$.



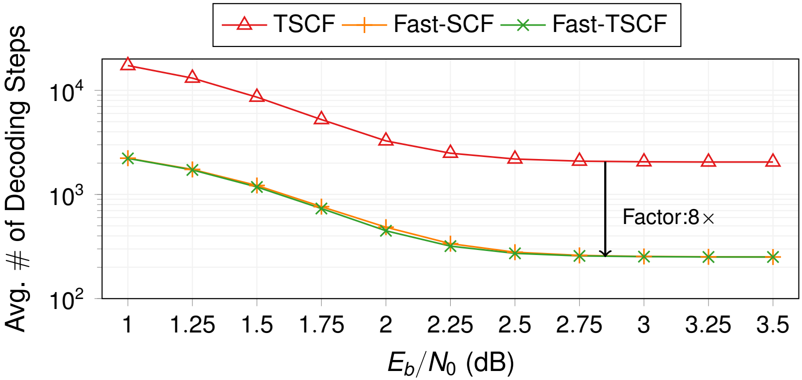
Results: Latency

- *PC*(1024, 512), 16 bit CRC, $T_{\text{max}} = 10$.



Results: Latency

- $PC(1024, 512)$, 16 bit CRC, $T_{\max} = 10$.



Results: ASIC Synthesis

Table: TSMC 65 nm CMOS synthesis results comparison for Fast-TSCF decoding against state-of-the-art, using $PC(1024, 512)$.

	Fast-TSCF	Fast-SCF[1]	Fast-SSCL[2]
Technology (nm)	65	65	65
Supply(V)	1.0	1.0	N/A
Frequency (MHz)	480	455	885
Avg. Coded T/P (Mbps)	1595(a)	1511(a)	**1861**
Area (mm^2)	**0.49**	0.56	1.05
Area Efficiency (Gbps/mm^2)	**3.2**	2.71	1.78

(a) Average value at target FER$= 10^{-4}$.

(b) List size for Fast-SSCL is $L = 2$.

[1] F. Ercan, T. Tonnellier, and W. J. Gross, Energy-efficient hardware architectures for fast polar decoders, IEEE Transactions on Circuits and Systems I: Regular Papers, pp. 114, 2019.

[2] S. A. Hashemi, C. Condo, and W. J. Gross, Fast and flexible successive cancellation list decoders for polar codes, IEEE Transactions on Signal Processing, vol. 65, no. 21, pp. 57565769, Nov 2017.

Conclusion

- Answering how to make TSCF algorithm practical and fast

Conclusion

- Answering how to make TSCF algorithm practical and fast
- We showed how to:
  - Replace empirical threshold with a function of E_b/N_0.
  - Correlate empirical critical set with an analytical one.
  - Introduce fast decoding techniques for TSCF.
  - Hardware implementation.

Conclusion

- Answering how to make TSCF algorithm practical and fast
- We showed how to:
  - Replace empirical threshold with a function of E_b/N_0.
  - Correlate empirical critical set with an analytical one.
  - Introduce fast decoding techniques for TSCF.
  - Hardware implementation.
- Compared to
  - Fast-SCF: 0.24 dB performance improvement.
  - Fast-SSCL: 82% better area efficiency.
  - TSCF: 88% fewer decoding steps & no precomputational dependencies.

Thank you for your attention!