Rate Compatible LDPC codes over GF(q)

Demijan Klinc

Georgia Institute Of Technology

Advisors: Steven W. McLaughlin and Jeongseok Ha

Motivation

- in wireless communication systems operating over time-varying channels code rate should be adaptively changed depending on the channel quality to achieve maximum throughput

  → **rate-compatible codes**

- Nonbinary LDPC codes can considerably outperform binary LDPC codes at short block lengths
- The problem of puncturing nonbinary LDPC codes over binary input channels has not been considered so far

Overview of the problem

- We consider binary-input channels
- Rate-compatible puncturing of binary LDPC codes is already well understood
- With nonbinary codes, multiple channel uses are required to transmit a single code symbol
- The puncturing problem becomes one of not only determining which symbols to puncture but also how many bits per symbol to puncture

Bit spreading (1)

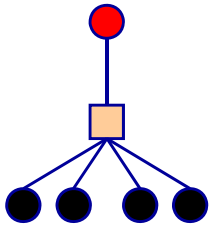
- Initial uncertainty about a code symbol in the decoder rises exponentially with the number of bits that were punctured in that symbol
- Example:

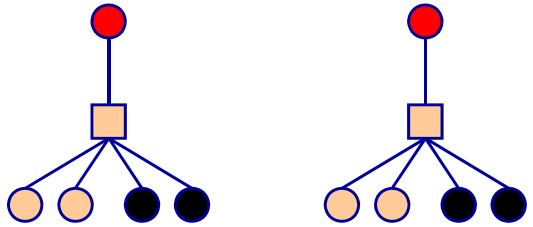
# punct.bits/symb	1	2	3	4	5	6
Zero LLR comp.	1	3	7	15	31	63

- **Key Idea:** spread the punctured bits over a higher fraction of symbols to decrease the number of punctured bits/symbol

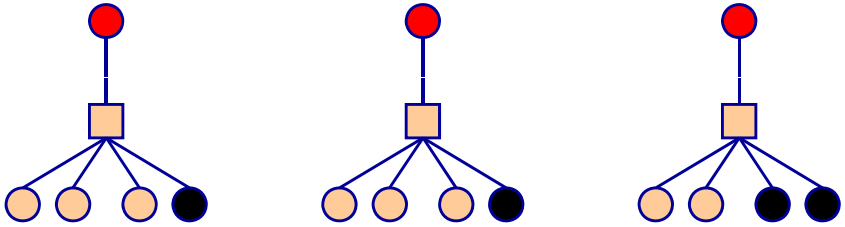
Bit spreading (2)

Instead of puncturing symbolwise





Spread the punctured bits over symbols to decrease puncturing depth



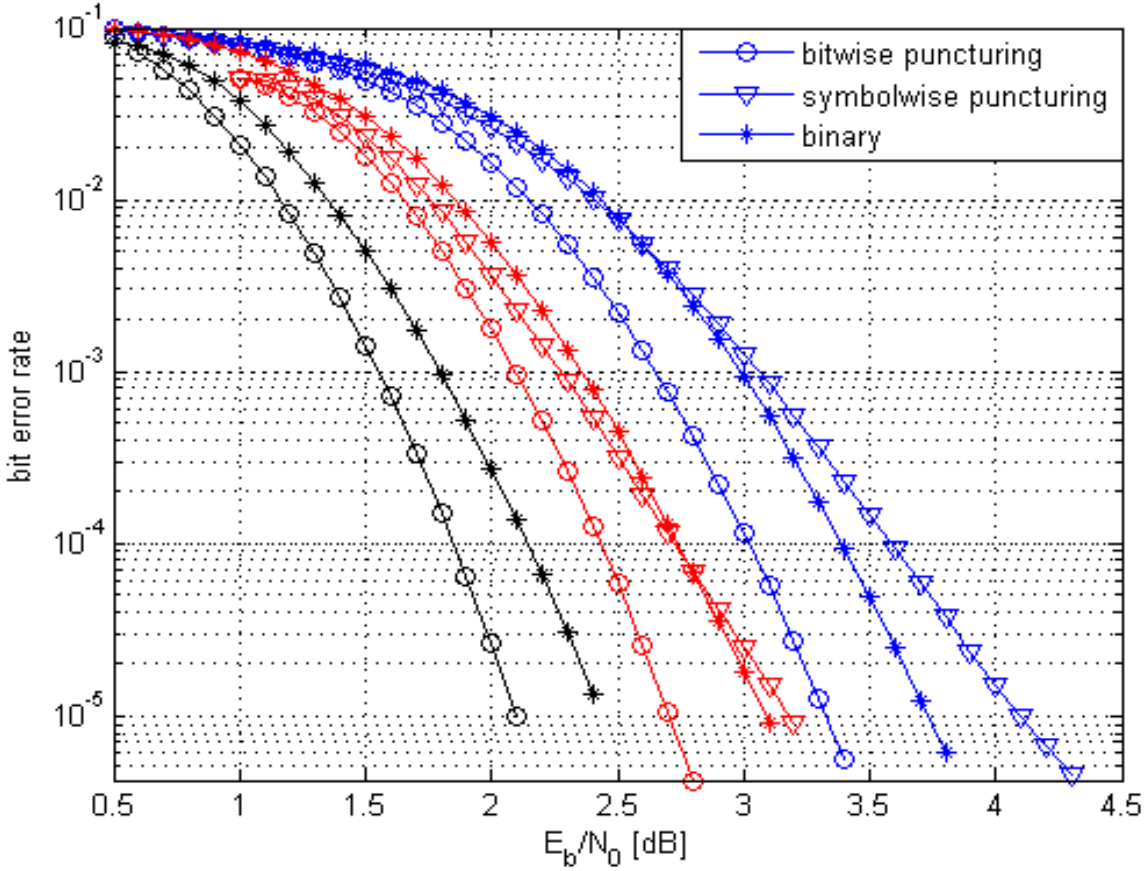
- Symbol node
- Unpunct. bit node
- Punct. bit node

Puncturing Algorithm

- Two-Step Algorithm:
  - 1st step: determine which symbols that are allowed to be punctured under the constraint that recovery is guaranteed for each node
  - 2nd step: spread punctured bits over the bits chosen in the previous step to minimize the puncturing depth of each punctured symbol

Simulation Results (1)

- Left to right: unpunctured (r = 0.5); r = 0.6; r = 0.7;

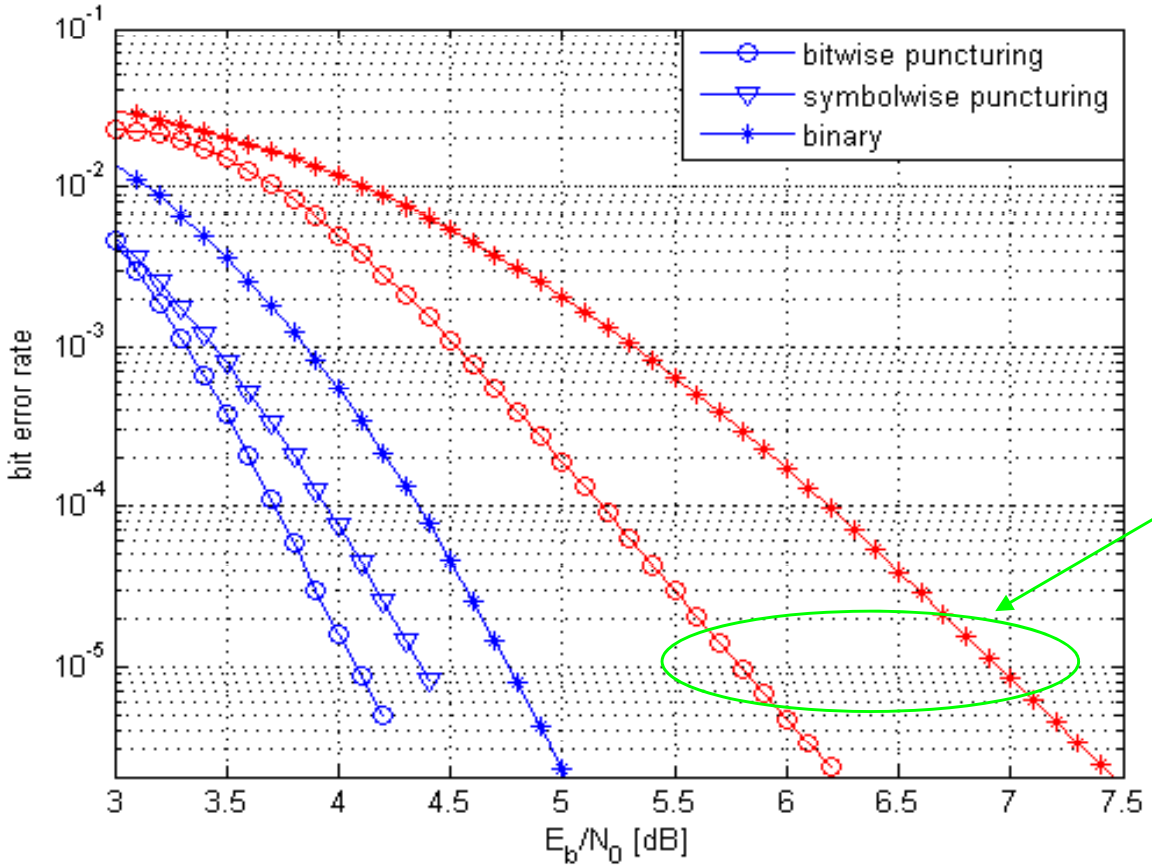


Proposed puncturing scheme consistently outperforms symbolswise puncturing and binary codes

The nonbinary code is over GF(64) with block length 142 symbols.

Simulation Results (2)

- Left to right: r = 0.8; r = 0.9;



Gap over binary codes exceeds 1 dB at r = 0.9

Future Work

- Develop theoretical framework for analysis of puncturing patterns for nonbinary LDPC codes
- Density evolution analysis over the BEC channel (in progress)
- Application of rate-compatible punctured LDPC codes for distributed source coding